UNIVERSIDAD DE VALLADOLID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

TRABAJO FIN DE MASTER

INGENIERO DE TELECOMUNICACIÓN

802.11p standard and V2X applications on commercial Wi-Fi cards

AUTOR: Javier Fernández Pastrana

TUTOR: Juan Carlos Aguado Manzano

ENERO 2017

TITULO: **802.11p standard and V2X applications on commercial Wi-Fi cards**

AUTOR: Javier Fernández Pastrana

TUTOR: Juan Carlos Aguado Manzano

DEPARTAMENTO: Teoría de la Señal y Comunicaciones e Ingeniería Telemática

<u>**Miembros del Tribunal**</u>

PRESIDENTE: Patricia Fernández del Reguero

SECRETARIO: Ramón J. Durán Barroso

VOCAL: Ignacio de Miguel Jiménez

SUPLENTE: Mª Jesús Verdú Pérez

FECHA DE LECTURA:

CALIFICACIÓN:

RESUMEN DEL PROYECTO

En este proyecto se implementa la capa de acceso correspondiente al estándar 802.11p en un entorno Linux con tarjetas Wi-Fi comerciales y se testea para la incorporación en un producto comercial de la compañía [redacted]. El estándar 802.11p está englobado dentro del marco del vehículo conectado y de la siguiente generación de Sistemas de Transporte Inteligentes (ITS). La mayor aportación que hace este proyecto frente a otros realizados es que en lugar de utilizar tarjetas exclusivamente dedicadas al mundo ITS, aquí se ha buscado tarjetas Wi-Fi comerciales convencionales de bajo precio que fueran compatibles con el estándar, se ha utilizado código libre para la implementación del estándar en una distribución Linux, se han fijado los requisitos que se debería cumplir para poder ser utilizado en entornos V2X reales y se han realizado los test necesarios para asegurar su comercialización. Para algunos de estos test se ha utilizado un dispositivo ITS implementado por el [redacted]. Finalmente, este proyecto ha sido implementado en un producto comercial fabricado por [redacted], de tal forma que se ha actualizado el dispositivo añadiendo esta nueva característica que le permitirá ser utilizado en el futuro como vehículo conectado.

ABSTRACT

During this project has been implemented the 802.11p standard (access layer). The project has been developed for a Linux environment using conventional Wi-Fi wireless cards. Moreover, it was tested so that it can be used in a professional product sold by the company [redacted]. The 802.11p standard belongs to the next generation of the connected vehicle as well as the future Intelligent Transportation Systems. The main difference between this project and other commercial projects is that we intend to implement de access layer using a conventional and cheap Wi-Fi wireless card, whereas in most of the commercial projects adopts a proprietary approach. Besides, it has been used open source projects to program the implementation in a Linux distribution and we have set the requirements that the device should meet so that it could be use in VX2 real environments. Finally, we have carried out the necessary tests in order to ensure that it could be commercialize. Some of these tests were carried out with a real ITS device provided by the [redacted] in order to check the interoperability of this project with a real infrastructure. As a consequence, this solution was included in a professional product used for automotive solutions and sold [redacted]. Hence, the product has been upgraded and its technical specifications improved by using cheaper hardware.

PALABRAS CLAVE

802.11p, V2x, OCB, Intelligent Transportation System, Cooperative ITS, Vehicular communications, Outside the Context of a BSSID, connected vehicle, WAVE, ITS, CA
DENM, BTP, GeoNetworking, DCC, Linux, IW, 802.11, Wireshark, datalogger.

Contents

List of Figures

Table of contents

Appendix Figures

Abbreviations

AP	Access Point
ASTM	American Society for Testing and Materials
BIM	Backend Integration Manage
BRAN	Broadband Radio Access Networks
BTP	Basic Transport Protocol
CAM	Cooperative Awareness Messages
CCA	Clear Channel Assessments
CRDA	Central Regulatory Domain Agent
DCC	Decentralized Congestion Control
DENM	Decentralized Environmental Notification Messages
ECC	Electronics Communications Committee
EDCA	Enhanced Distributed Channel Access
EIRP	Effective Isotropic Radiated Power
FCC	Federal Communications Commissions
GAC	Geographically-Scoped Anycast
GBC	Geographically-Scoped Broadcast
GUC	Geographically-Scoped Unicast
HMI	Human Machine Interface
I2V	Infrastructure to Vehicle
IBSS	Independent BSS
ITS	Intelligent Transport Systems
LDM	Local Dynamic Map
LPV	Local Position Vector
LS	Location Service
MAP	Topological definition of lanes for a road-segment
MLME	MAC Layer Management Entity
OCB	Outside the Context of a BSS
OFDM	Orthogonal Frequency-Division Multiplexing
PLCP	Physical Layer Convergence Procedure
PMD	Physical Medium Dependent
POTI	Positioning and Timing management
PSDU	PLCP Service Data Unit
RLAN	Radio Local Area Network
SC	Service Consumer
SHB	Single Hop Broadcast
SP	Service Provider
SPAT	Signal Phase And Timing
STA	Station
TA	Timing Advertisement
TCP	Transmission Control Protocol
TOPOM	Road Topology Message
TSB	Topologically Scoped Broadcast
UDP	User Datagram Protocol
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle

V2X	Vehicle to any component
WAVE	Wireless Access in Vehicular Environments
WINIC	Wireless Network Interface Controller
WLAN	Wireless Local Area Network
WM	Wireless Medium

1 Introduction

Intelligent Transport Systems (ITS) were defined in EU Directive 2010/40/EU [1] as *"'(...) systems in which information and communication technologies are applied in the field of road transport, including infrastructure, vehicles and users, and in traffic management and mobility management, as well as for interfaces with other modes of transport"*

The deployment of this new technology poses an enormous challenge, since ITS comprises several kind of fields of application, which implies that it will be needed different kind of technology communications to make the ITS network possible. Moreover, we have to take into account the vast transportation infrastructure needed in highways, streets, bridges, etc, as well as the growing number of different transport vehicles. Besides, it is important to remember that it does not only apply to mechanical transport, but these systems will also include pedestrians, cyclists, etc.

It is believed that all these changes will give rise to the next age of transport system generation based on information and communications technologies. This new age of transportation will provide tools to enable authorities, operators and travellers to take more informed and "intelligent" decisions.

Transport infrastructure involves a great deal of services, communications and protocols that are currently under development. Particularly there has been made a huge effort in defining and developing new wireless communications, which are mainly focused on the communication of the main elements of the system:

- From Vehicle to Vehicle (V2V), via ad-hoc (or cellular) communication or based on Infrastructure involvement.
- From Vehicle to Infrastructure (V2I).
- From Infrastructure to Vehicle (I2V).

All these communications are usually referred as V2x communications. This concept has been settled by the community to refer, in an easy way, any possibility of communication between vehicles and infrastructure. In Figure 1 is given an example of a Cooperative ITS environment, where is represented several types of communication technologies, WLAN, satellite communications, mobile, ITS-G5, etc. Besides, it is showed some examples of user applications such as navigation, safety systems, crash avoidance, etc.

During this project we are mainly focusing at physical and MAC layers of the OSI stack, these layers correspond to the access layer in ITS stack. Therefore, it will be given some examples of upper layers and user applications, but they will always be geared towards the access layer.

The main actors of this technology are called stations and depending on where they are located (vehicles, infrastructure, pedestrians, etc.), they will have a specific purpose and definition:

- Road Side Unit (RSU) ITS: A RSU provides ITS applications from the roadside. It can provide ITS applications independently or cooperatively with Service Centre or other RSUs.

- On Board Unit (OBU) ITS: An OBU provides ITS applications to vehicle drivers and/or passengers. It may require an interface for accessing in-vehicle data from the in-vehicle network or in-vehicle system.
- Personal ITS: It provides ITS applications to personal and nomadic devices.

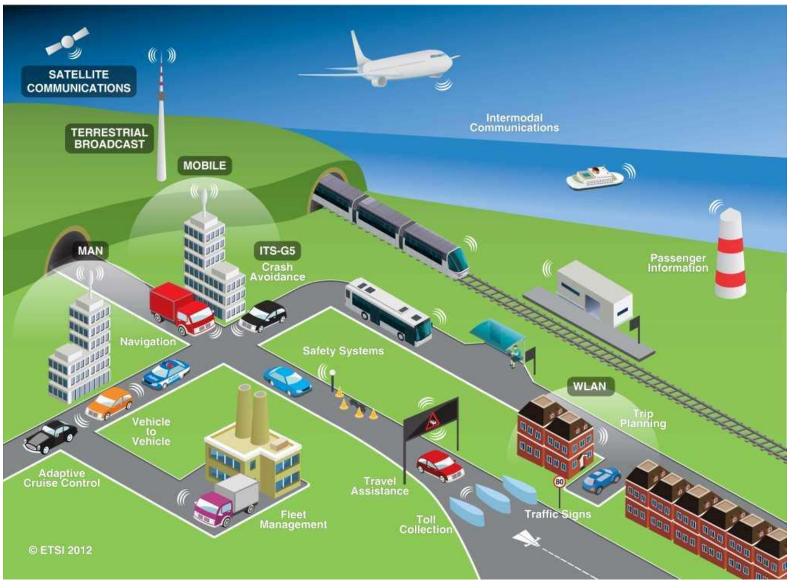


Figure 1. Example of Cooperative ITS environment [2]

Apart from the previous stations, the infrastructure requires a central device, which controls every situation as well as connects the whole infrastructure with the external network. This is the function of the Service Centre:

- ITS Service Centre: It is the responsible for providing all the centralized ITS applications. It plays the role of traffic operator, road operator, services provider or content provider. Furthermore, a Service Centre may require further connection with backend systems such as Internet.

A great deal of all these technologies, devices and applications are currently under development and test. Particularly, in Europe a complete set of European projects have been carried out in order to test the possibilities. Just to name a few, SCOOP [3] and Eco-AT [4] which are responsible of the development of the ITS European corridors (section 0), GCDC project which is an US project, however it was tested on the 28th of May 2016 [5] in European roads where ten European teams take part in. Other project is developed by [redacted], the importance of this project is that they have built an ITS environment where can be tested each ITS device as a real ITS infrastructure.

Currently, there are three different standards for ITS environments. They are defined for the three main markets in the world: US (IEEE 1609 [7]), UE (ETSI ITS G5 [2]) and Japan (ARIB STD-T109 [8]). However, all these standards keep the same idea of what implies an ITS environment and the goals that are wanted to be achieved, although the network implementation of each one is different. For example, in the physical layer frequency allocation is different for all of them, which means that it will be difficult design compatible hardware. Secondly, if we compare the rest of the upper protocols of the stack, they are different too. This situation leads to the development of different pieces of software for each standard, which makes even more difficult for the industry to find a single suitable hardware solution for all standards at the same time. There are two relevant exceptions to this rule.

First at all, the most important, the Access layer is common at least for European and American standards. This layer is defined in the standard 802.11p, commonly named WLANp. However, it is necessary to point out that in the physical layer the range frequency and allocation is slightly different. Secondly, presentation and application layers could be similar, although this is not completely decided.

In the EU, the current status of the Cooperative Intelligent Transport Systems can be followed through the European Commission webpage [9]. The last piece of news about these cooperative systems is dated on the 30th of November 2016 [10], when it was defined the strategy that will make possible to deploy vehicles that can communicate to each other and to the transport infrastructure on EU roads as of 2019 [11]. As a consequence, as it is said in the work programme of C-ITS platform [12], nowadays it is already possible the deployment of Cooperative ITS but more progress in higher levels of automation are still necessary.

Currently, the main solution to interact with the ITS network is using commercial products which are specifically designed for the ITS Access layer. It implies that these components usually are expensive, because the most of times they have a custom-made hardware along with the fact that it is not a wide range of possible solutions. However, Atheros has released a family [13] of commercial cheap Wi-Fi cards that are able to work in the range of frequencies and meeting the main requirements of 802.11p standard. At the same time, the company released also the driver, so that anyone could modify it in order to adapt it to their necessities. This has provoked several open projects (section 2.2) to be carried out in order to adapt the driver to 802.11p standard and even the whole protocol stack for ITS (section 3.10), as much for American as for European one. As a result, it becomes an excellent opportunity to develop ITS devices and it will be the first aim of this project to check the possibilities of these cards and the open projects to be used for ITS applications.

Nevertheless, every new technology introduced in the competitive car industry must be rigorously, extensively and exhaustively tested, specifically if devices not initially designed for this purpose are going to be used. [redacted]

[redacted]

Figure 2. [redacted]

Given the importance that these devices will have in the validation of any new ITS device that is included in the car, [redacted]

If a commercial device is to be upgraded, external validation will be necessary. For this purpose, we count with the collaboration of [redacted] is a validation centre for ITS and connected mobility. It has built a real environment which enables to simulate urban and interurban mobility solutions. It offers a 3km of highway and 2km of rural road and the next generation networks are implemented and tested.

One of the main contributions of this project is the Road Side Unit simulator which is able to emulate 20 vehicles in a real environment, implementing this way a complete ITS environment. As it is shown in the Figure 3, the ITS environment can be completely configured by adding the demanded ITS stations and communication protocols.

[redacted]

Figure 3. ITS simulation map [15]

The ITS environment is configurable using an Android APP installed in an extern device. Through this application, it is possible to place the ITS network in any point of the world map and designing your own architecture. Afterwards, executing the simulation is enough to start to send the preconfigured packets.

During this project, the Road Side Unit designed by [redacted] has been used to test and validate that the modifications included in our project meet ITS specifications. But, the flexibility of the [redacted] RSU allows us to address a collateral target that is to get a general idea of the ITS G5 standard stack and what protocols are in each layer of the stack. This point is important in order to take a well-understanding of the behaviours of the Cooperative ITS and of the Access layer (802.11p) that is wanted to implement.

In conclusion, the next objectives have been established:

1. To look for suitable cheap Wi-Fi cards.
2. To modify the Linux Kernel to allow these cards to work with 802.11p standard, by using mainly any suitable open source project.
3. To modify the user level software to work with 802.11p standard, by using mainly any suitable open source project.
4. To upgraded a commercial device to use these cards.
5. To test by all means that the new application is working as expected.

The rest of the memory is organized as follows:

- **State of Art (chapter 2):** It will be provided a general overview of the current ITS European status. Also, giving the current situation about the commercial wireless cards that can be used with 802.11p standard, the deployment of ITS infrastructure in the European Union giving the current open source projects to implement ITS protocol stack as well as it will be provided some commercial solutions for ITS environments. Finally, it will be referenced some network simulators that are useful to get an idea of how the ITS networks work.
- **Specifications and requirements (chapter 0):** The main goal of this chapter is to explain what implies at physical layer the 802.11p standard for a conventional Wi-Fi wireless card. Explaining what are the 802.11p particularities in order to find a suitable wireless card. Besides, it will be crucial to give and explanation of how is implemented the physical layer on 802.11p standard in Linux distributions. Once it is found appropriate wireless cards, it will be explained the new added features and how can be installed in a Linux device.
  In order to know what areas we have to test, it will be given a short view of how the whole ITS stack works, and why there are cross-layer functionalities, such as multi-hop channel or the transmission power, that are controlled by upper layers although they are linked to the physical layer. Therefore, it is essential to check if our wireless card supports these working conditions.
- **User manual (chapter 4):** It will be given a simple guide of how to configure and launch the 802.11p features to interact with ITS communications using a Linux device. Moreover, it will be included a manual of how to configure Wireshark to decode ITS packets as well as a demonstration of how could be dissected some packets that were captured during the project implementation.
- **Performance evaluation (chapter 5)**: The main target of this section is to check if the commercial Wi-Fi wireless cards selected for 802.11p are suitable and meet all the specifications imposed by the ITS standards. It means, we want to know if these wireless cards could work in a real ITS environment as good as a professional solution or if they meet at least the enough requirements to implement a prototype.
- **Conclusions (chapter 6):** At last, it is reflected if we have achieved all the agreed targets and extract a firm conclusion about the usage of conventional Wi-Fi wireless cards for 802.11p standard.

- **Future work (chapter 7):** It will be provided some ideas of how this project can be continued as well as some projects that can be attached to this one in order to provide a full solution.

2 State of Art

Nowadays several companies are developing commercial products for ITS networks. During this chapter we are going to make a short review of some of these ITS devices. Furthermore, it will be given an overview of the existing ITS projects, ITS network simulators, which is the current status of Cooperative ITS (C-ITS) in the European Union and what we want to achieve during this project taking into account all the mentioned points.

Since the decision to use 802.11p standard for vehicular communications as a firm solution (on the 14th of March 2008 ECC [16]), multitude of companies and organisations have helped to develop and deploy the Cooperative ITS. Nowadays it is still in progress but, as it was said, it was decided to roll out vehicles that use ITS infrastructure on EU roads as of 2019.

One of the first open source projects for ITS solutions was the Grand Cooperative Driving Challenge [17] (GCDC), which finished in October of 2016. It was born as a competition in which private and state companies took part to develop the vehicular communications, especially the 802.11p standard on European stack. It was given documentation and software support to adapt the ATH5K driver [18] to this standard. However, after a time it was refused the public access to this information. Anyway, it was enough to start to modify the drivers of ATH5K and later the ATH9K [19]. The Atheros software community started with the ATH5K, so the first OCB (Outside of a Context of a BSSID) adapted wireless cards were these ones. Sometime later, the ATH9K appeared, and Atheros released the driver software. The software community was able to implement the OCB mode among other updates.

Currently, it only exists one Open Source method to implement the 802.11p standard, and it is using the driver modifications made for ATH5K or ATH9K [20]. It has not been found other Wi-Fi wireless cards that are able to support this standard, which implies both things, the physical specifications and a suitable driver. Commercial products for WLANp also exist (the most popular ones are manufactured by NXP and Qualcomm-Atheros), but these devices are only for professional solutions, it means that they cannot be purchased by a conventional user, because they have not released these products to any client.

These are the reasons why, at the moment, the cards which are controlled by the ATH5K or ATH9K driver are the only ones suitable for the majority of research and development projects. During this project it will be used the ATH9K driver, because it is newer than ATH5K and it is easier to find suitable Wi-Fi wireless cards with this chipset in the market.

2.1 C-ITS Infrastructure.

Meanwhile, the European Union has supported several projects during the last years in order to develop and deploy several infrastructure demonstrations to test this technology. For example, a demonstration between Helmond and Eindhoven was carried out by GCDC project. A more complete view of the different European initiatives can be seen in Figure 4.a, whereas in Figure 4.b it is shown that the final idea is to create a strong network transportation structure based on nine European Corridors. The corridors in a more advanced ITS infrastructure deployment are those related with the next European projects:

- SCOOP project which involves Austria, Spain, France, Portugal [3].
- European Corridor which involves Austria, Germany, and the Netherlands [4].

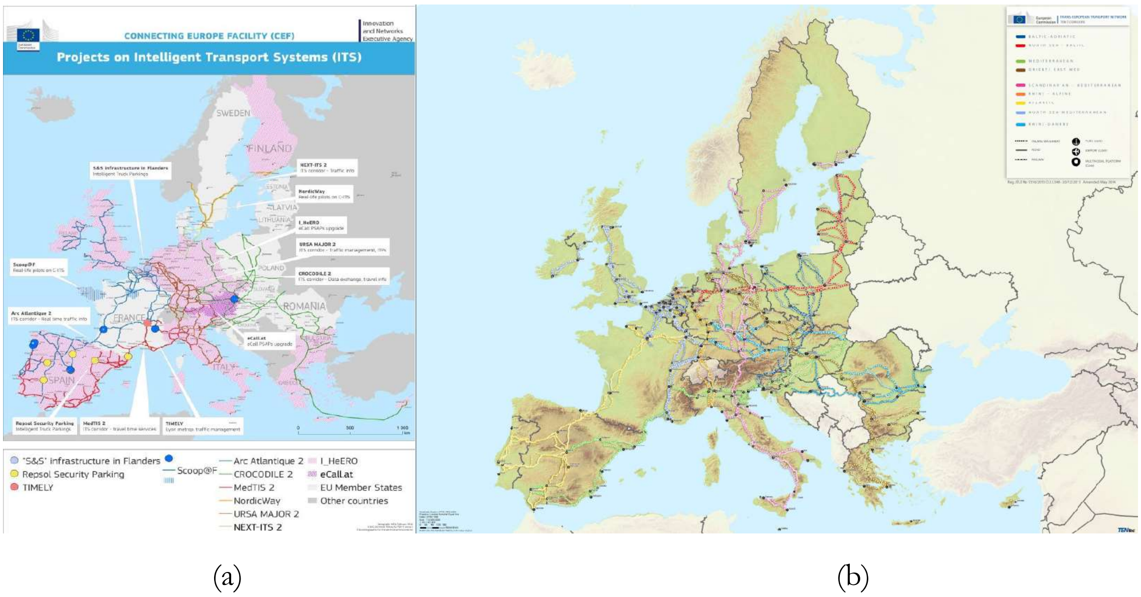

CONNECTING EUROPE FACILITY (CEF)
Innovation and Networks Executive Agency
Projects on Intelligent Transport Systems (ITS)
SWEDEN
FINLAND
ESTONIA
LATVIA
LITHUANIA
POLAND
ROMANIA
FRANCE
SPAIN
ITALY
S&S infrastructure in Flanders
Intelligent Truck Parkings
NEXT-ITS 2
ITS corridor - Traffic info
NordicWay
Real-life pilots on C-ITS
I_HeERO
eCall PSAPs upgrade
URSA MAJOR 2
ITS corridor - Traffic management, ITPs
CROCODILE 2
ITS corridor - Data exchange, travel info
Scoop@F
Real-life pilots on C-ITS
Arc Atlantique 2
ITS corridor - Real time traffic info
eCall.at
eCall PSAPs upgrade
Repsol Security Parking
Intelligent Truck Parkings
MedTIS 2
ITS corridor - travel time services
TIMELY
Lyon metrop. traffic management
0
500
1 000
km
'S&S' infrastructure in Flanders
Repsol Security Parking
TIMELY
Scoop@F
Arc Atlantique 2
CROCODILE 2
MedTIS 2
NordicWay
URSA MAJOR 2
NEXT-ITS 2
I_HeERO
eCall.at
EU Member States
Other countries
(a)
(b)

Figure 4. (a) European ITS projects [21]. (b) The overall TEN-T corridor map

Finally, it is worth mentioning the [redacted] (section 1) consortium that is involved in all project phases relating to design and testing of intelligent transport systems and services. They perform the relevant validation and demonstration activities necessary for successful development [redacted]. This consortium has kindly provided us with a side road unit to test the 802.11p device developed during this project.

2.2 Open software projects for ITS stack

Open software projects have become a great opportunity to test the possibilities of many new technologies. This is also true for the protocols implementing the upper layers of the Cooperative ITS. Although it is not the final objective of this project to implement the complete protocol stack, any Cooperative ITS device that we want to develop should use this kind of project as a start point. Nowadays, a few of open source projects have been developed in order to support the next generation of vehicular networking. The major barrier to the WLANp open source development is the difficulty in finding open source wireless card drivers along with finding a wireless card that meets the technical specifications.

- Vanetza Github project [22]: it has implemented the GeoNetworking protocol (section 3.10.5) and the Basic Transport Protocol (section 3.10.4) as well as the Decentralized Congestion Control (section 3.8) mechanism and aspects of the Facilities layer (section 3.10)
- GeoNetworking Github project [23]: it was born as a proposal for the GCDC competition. They have implemented more than 50% of the project which includes Cooperative Awareness Message (section 3.10.1), Decentralized Environmental Notification Message (section 3.10.2), ASN.1 PER (Packed Encoding Rules are ASN.1 encoding rules for producing a compact transfer syntax for data structures described in ASN.1), BTP and GeoNetworking protocols.
- 802.11p on Linux [24]: It is already finished, although it is being merged with the official Linux kernel release [25]. It implements the Access layer of the stack (802.11p). It provides the first steps and provides the driver modifications for ATH9K.

The third project has been the basis of this research.

2.3 Commercial solutions for Cooperative ITS

On the other side, if we focus on the private development of Cooperative ITS, we can find multitude of companies that offers every kind of products. The devices developed by Cohda-Wireless are very popular. Among other things, they build up Road Side Units (RSU) and On Board Units (OBU). Besides, they also collaborate with the company NXP, which is a provider of 802.11p wireless radio cards. It means that they do not use a conventional Wi-Fi wireless card, but they use specific designed wireless devices to work with 802.11p standard.

The company Arada-Systems also develops some popular devices for ITS networks. Currently, they offer products for each part for a Cooperative ITS. The information given for these products has attracted our attention, because the technical datasheet for each device is provided, what will be useful to compare our results with their specifications. It will be explained with more detail in the section 5.3.1.

2.4 Software simulators for ITS networks

Due to difficulty in deploying real ITS infrastructure, ITS simulators are crucial for the development of this technology, as they enable to understand and predict the behaviours of the configured network. It has been developed several simulators for Cooperative ITS. One of the most used simulators is OMNet++ [26], which provides a multitude of possible configurations for almost any kind of network. "Veins" [27] is a framework developed for this program in order to design ITS environments. It is able to simulate how individual ITS devices behave in different scenarios, as for example intersections controlled by traffic lights, crashes, etc.

There are other popular ITS simulators such as iTetris [28] which is a modular simulation platform for large scale evaluation of cooperative ITS applications. Other software product for ITS is Marben [29] which offers a user-friendly C++ API and pre-defined safety applications that speeds up the development and integration of ITS applications into On Board Units or Road Side Units.

To sum up, it has been show the Cooperative ITS has a solid based. And it will be deployed in the future. However, as we will explain during this memory there are several radio technologies that can be implemented at Access layer. For the time being, the 802.11p standard is a firm solution for this access layer and it has suitable properties for ITS communications. However, with the arrival of new mobile technologies such as 5G the old implementations may be replaced.

3 Specifications and requirements

As it was discussed in previous sections, there are several projects that implement WLANp. However, the most frequently used in this project in order to achieve the objectives proposed has been “802.11p on Linux”, which was developed at the Faculty on Electrical Engineering of Czech Technical University in Prague [30]. This project provides the first steps and gives solid foundation to implement WLANp (the first two layer of the stack) in any Linux device. This project provides a modification of the Atheros driver ATH9K. These modifications were made for 3.18 Linux kernel. However, as I shall show during this memory, they can be imported in upper Linux kernel versions.

In order to understand the modifications that the standard WLANp includes in physical layer, it will be explained some noticeable features that have to be taking into account to better understand the behaviours of 802.11p. The explanation of these characteristics will help to interpret the results obtained from the performed tests and explained in the next sections as well as to understand the changes that were made in the ATH9K driver source code.

3.1 Technical specifications of Wi-Fi wireless card

The 802.11p standard is a “branch” of the 802.11a. As it is explained in the section 3.4, it was changed some of its characteristics in order to define a protocol which was able to work in a high mobility network. For this reason, the first technical specification that we need is that our Wi-Fi wireless card deploys the 802.11a.

Secondly, as it will be explained in the section 3.4, we are going to include a new operation working mode called Outside the Context of a BSSID (OCB). This new function is a branch of the well-known operation mode Independent BSS (IBSS) that it is used to create Ad-hoc networks that contains no access points. Typically, the most of Wi-Fi wireless cards support this mode, but it is crucial check it before implementing this project.

Thirdly, in the previous chapter we defined that the ATH9K is the driver that it is going to be used during this project. Therefore, we have to check which are the Atheros chipsets that supports this driver, this information is available in the Linux wireless Wiki [31]. Also, it is included which are the commercial devices that use this driver. The ATH9K is a driver for PCI-e or miniPCI-e wireless cards, it means that the future ITS device must include this port to connect the card.

Taking into account all the previous technical requirements and if they are available or not in European markets, the wireless cards that have been used during the project are:

- Atheros HB92 AR9280 Dell U608F (miniPCI-e half size).
- Atheros AR9280 Ubiquiti Networks SR71-E (miniPCI-e full size).
- Atheros AR9382 Sparklan WPEA-121N (miniPCI-e half size).
- Qualcomm Atheros AR9462 Sparklan WPEA-251N (BT) (miniPCI-e half size).

There is also an ATH9K driver for USB devices, it is called ATH9K_htc. Both drivers share some libraries but it is not the same driver. Knowing the limitations of using a PCI port, during this project it was tested if a device which uses the ATH9K_htc driver was able to support 802.11p. In order to test this driver, it was used the Netgear WNDA 3200, which supports the 802.11a/b/g/n standards and uses the AR7010 AR9280 Atheros chipsets.

3.2 Description of the physical layer on WLAN standard (IEEE 802.11-2012 [32])

The standard WLAN is able to work at bitrates of 6, 9, 12, 18, 24, 36, 48 and 54 Mbit/s. In order to provide these bitrates and, at the same time, be able to cope with the difficulties of wireless transmission, WLAN make use of the well-known OFDM technique. The basic idea of OFDM is to divide the available frequency spectrum into multiple narrow subchannels, so that the effects of any interference can be avoided or dismissed. Specifically, standard WLAN uses 52 subcarriers in a bandwidth of 20 MHz, where 48 are used for data and the other 4 are pilot carriers. Each bitrate is modulated using binary, quadrature phase shift keying (BPSK or QPSK) or 16 – 64 quadrature amplitude modulation (16QAM – 64QAM). Besides there are 3 different coding rates to be used depending on the modulation and the bitrate: 1/2, 2/3 and 3/4 (this ratio gives the proportion of sent useful bits to total sent bits).

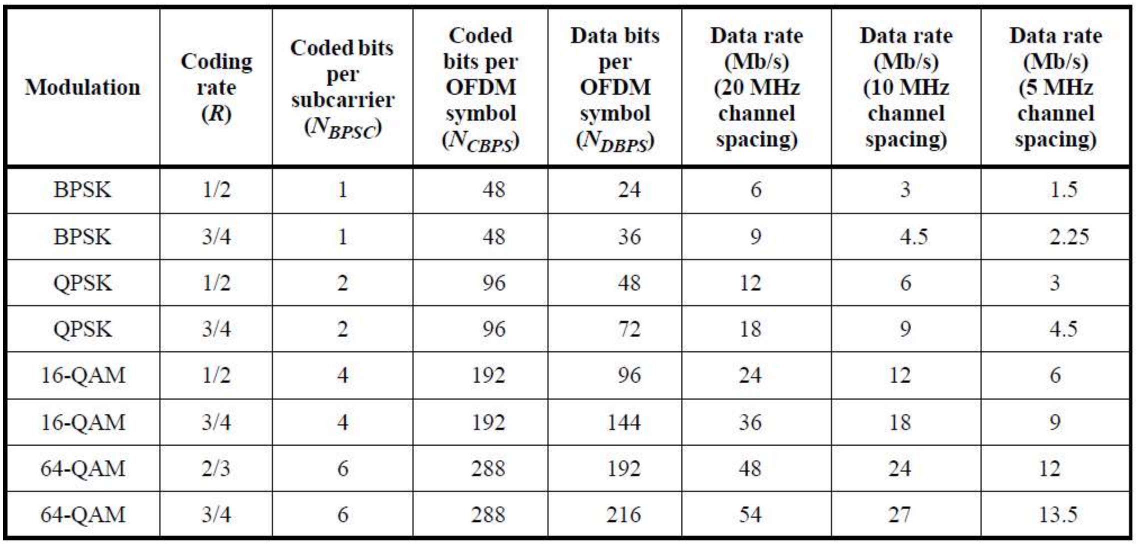
It is also provided by using the OFDM technique a “half-clocked” mode that uses 10MHz channel bandwidths with communication capabilities of 3, 4.5, 6, 9, 12, 18, 24 and 27 Mbit/s. The bitrates 3, 6 and 12 Mbit/s are mandatory in this operation mode. This operation mode doubles the symbol time as well as the clear channel assessments (CCA) times. The purpose of CCA mechanism is to detect if the medium is in busy condition or not. In order to do this two kinds of carriers are used, the carrier sense and the clear channel assessment.

Furthermore, there is a third mode named “quarter-clocked” that uses 5MHz channel spaces. In this case, the supported bitrates are the half of the bitrates of the previous mode (the supported and mandatories bitrates). Besides, the symbol time and the CCA times will be four times longer.

In Figure 6 is explained how the half-clocked and quarter-clocked modes work. As the number of subcarriers is always the same, but the total bandwidth used is smaller, the period of the symbol must be doubled or quadrupled, what have a straightforward effect in the bit rate.

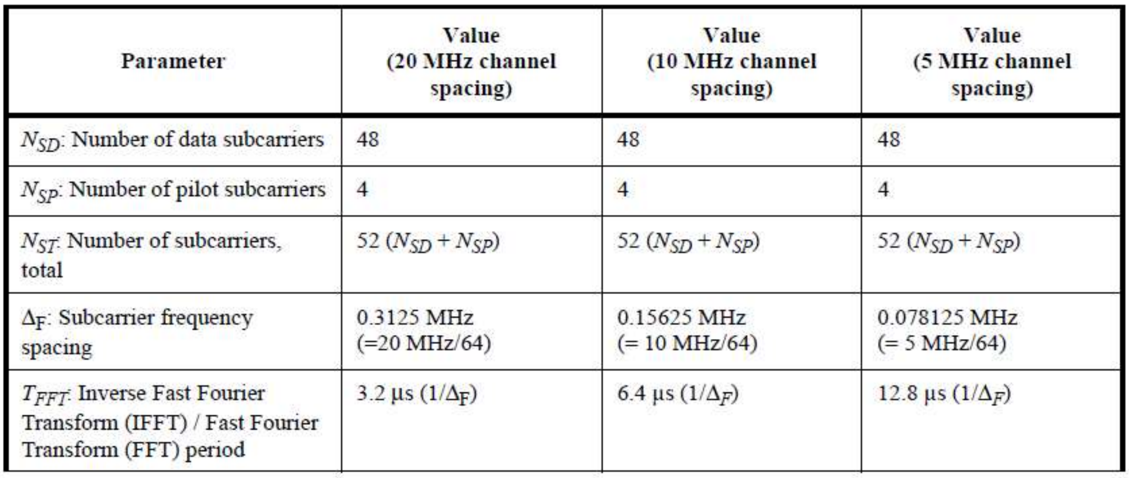
These two last operation modes have a huge importance in 802.11p, as they are the ones that are intended to be used in vehicular environments at 5.9GHz.

A summary of all this information and the different combinations that are allowed in the IEEE 802.11-2012 standard is shown in Figure 5 [32].



Modulation	Coding rate (R)	Coded bits per subcarrier (N_{BPSC})	Coded bits per OFDM symbol (N_{CBPS})	Data bits per OFDM symbol (N_{DBPS})	Data rate (Mb/s) (20 MHz channel spacing)	Data rate (Mb/s) (10 MHz channel spacing)	Data rate (Mb/s) (5 MHz channel spacing)
BPSK	1/2	1	48	24	6	3	1.5
BPSK	3/4	1	48	36	9	4.5	2.25
QPSK	1/2	2	96	48	12	6	3
QPSK	3/4	2	96	72	18	9	4.5
16-QAM	1/2	4	192	96	24	12	6
16-QAM	3/4	4	192	144	36	18	9
64-QAM	2/3	6	288	192	48	24	12
64-QAM	3/4	6	288	216	54	27	13.5

Figure 5. Modulation-dependent parameter. (Table 18-4 [32])



Parameter	Value (20 MHz channel spacing)	Value (10 MHz channel spacing)	Value (5 MHz channel spacing)
N_{SD}: Number of data subcarriers	48	48	48
N_{SP}: Number of pilot subcarriers	4	4	4
N_{ST}: Number of subcarriers, total	52 ($N_{SD} + N_{SP}$)	52 ($N_{SD} + N_{SP}$)	52 ($N_{SD} + N_{SP}$)
Δ_F: Subcarrier frequency spacing	0.3125 MHz (=20 MHz/64)	0.15625 MHz (= 10 MHz/64)	0.078125 MHz (= 5 MHz/64)
T_{FFT}: Inverse Fast Fourier Transform (IFFT) / Fast Fourier Transform (FFT) period	3.2 µs ($1/\Delta_F$)	6.4 µs ($1/\Delta_F$)	12.8 µs ($1/\Delta_F$)

Figure 6. Timing related parameters (Table 18-5 [32])

Using OFDM in wireless environments has many advantages but one of the main disadvantages is that the receptor, apart from being by far more complicated than in other transmission techniques, must devote part of the reception time to "calibrate" the transmission medium. This way it is important to study the physical frames, where information is sent to calibrate and synchronize the receptor, in order to know the exact payload that must be included to calculate the real bitrate that our system will support.

OFDM PHY provides all these services of the physical layer on WLAN, this is, it is the layer in charge for choosing the right parameters combination for a specific bit rate, and the encapsulation of all the information in order to allow a correct reception of the data. OFDM PHY contains three functional entities: The PMD (Physical Medium Dependent) function, the PHY convergence function and the layer management function. In order to understand how the user data is encapsulated into the physical frame and the cost that must be paid in order to avoid the interference problem, a more detail explanation of this layer is given.

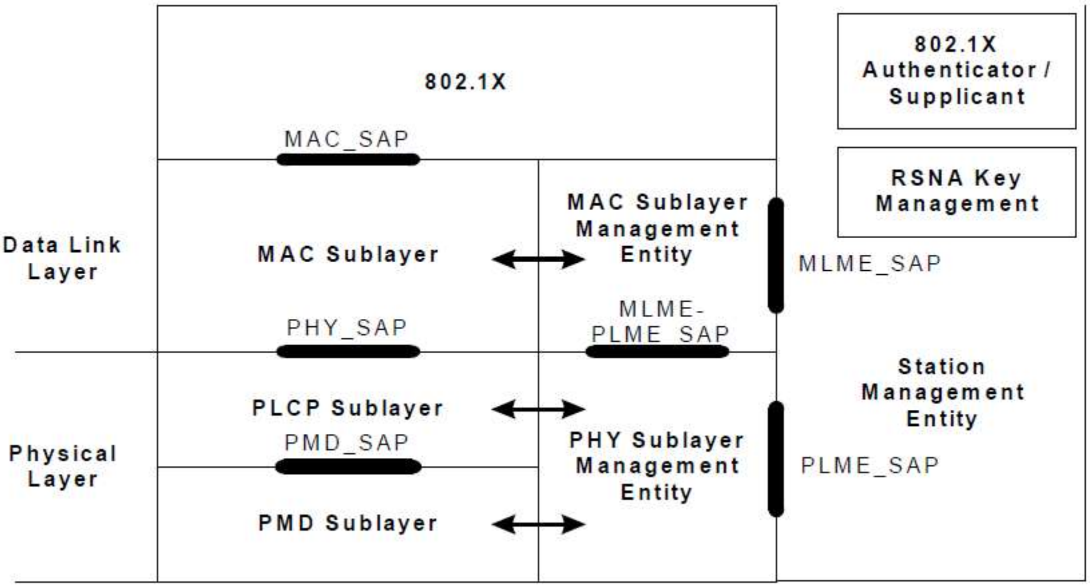


Figure 7. Portion of the ISO/IEC basic reference model covered in the standard 802.11-8012 [32]

The architecture of OFDM PHY is showed in Figure 7. We can see that the physical layer has two sublayers, the PLCP (Physical Layer Convergence Procedure) sublayer and PMD sublayer. The PMD sublayer defines the characteristics and method for transmitting and receiving data through a Wireless Medium (WM) between two or more STAs (Stations). On the other hand, PLCP sublayer is responsible for the convergence of the PMD system and the PHY service. It defines a method for mapping the IEEE 802.11 PSDUs (PLCP Service Data Unit) into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD services.

Therefore, the PSDU (MAC layer) is encapsulated in the final frame which is mapped in the PLCP sublayer. Figure 8 shows the resulting PHY packet. As it is shown, a Preamble field and a signal field are added. The Preamble field is used by the receptor to calibrate the physical conditions of the transmission medium, whereas the signal field contains information about the packet length and the data rate at which the data field is transmitted. The preamble and the signal field always have a fixed duration. For example, the signal field is always transmitted at the lowest bitrate, which is 3 Mbit/s, and its length is 24 bits, therefore it takes 8μs. Through a simple calculation, it is easy to reach the conclusion that to send the Preamble takes exactly 32 μs (As it is shown in Table 1). As a consequence, it should be taken into account all these headers to calculate the real bitrate, which should be close to the theoretical, from the bitrate observed by the user. Moreover, since the supported WLANp bitrates are 3, 6 and 12 Mbps, but the headers always have exactly the same duration (40 μs Preamble + Signal), this is equivalent to say that at 3 Mbit/s we are sending 120 bits, at 6 Mbps 240 bits and at 12 Mbps 480 bits in these headers. Hence, the higher is the bitrate, the higher is the cost that the headers will have. As a result, the bit rate observed by the user should be further to the real bit rate for the higher ones.

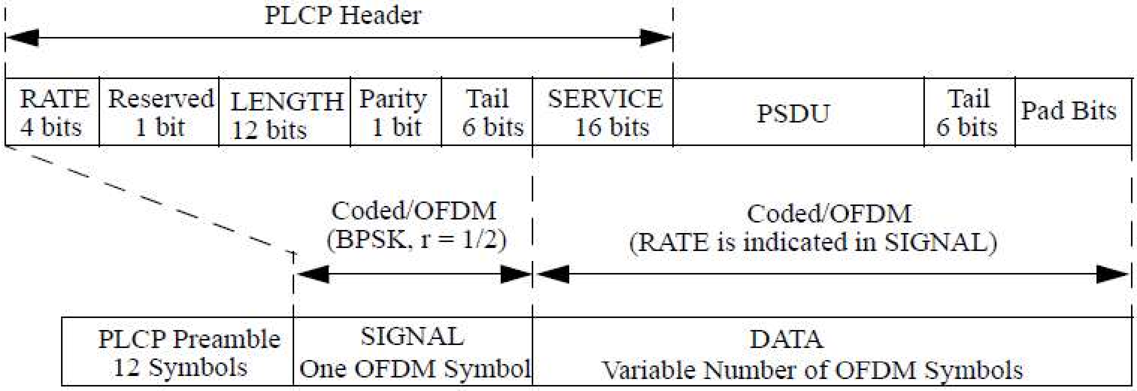


Figure 8. Resulting PHY packet

Field	Subfield	Description	Duration (µs)
Preamble	N/A	Synchronizing the receiver.	32
Signal	Rate	Specifies the transfer rate at which the data field in the PPDU will be transmitted.	8
	Reserved	For future use.	
	Length	The length of the packet.	
	Parity	Parity bit.	
	Tail	Used for facilitate decoding and calculation of rate and length subfields.	
Data	Service	Used for synchronizing the descrambler at receiver.	Depending on selected transfer rate and packet length.
	PSDU	The data from the MAC layer including header and trailer, i.e. MPDU.	
	Tail	Used for putting the convolutional encoder to zero state.	
	Pad bits	Bits added to reach a multiple of coded bits per OFDM symbol (i.e. 48, 96, 192, 288, see Figure 5).	

Table 1. Description of the different fields of the PPDU (Figure 8) (ref. [33] Table B.2)

3.3 Description of 5.9 GHz frequency band

In 2008 the Electronics Communications Committee (ECC) and European Commission took the decision [16] of addressing frequency designation within the band 5875-5925 MHz for the harmonised implementation of Intelligent Transport Systems (ITS). They reserved three 10 MHz channels for safety-related communications and the ECC recommended two more channels to be reserved for non-safety applications. On the other hand, the US Federal Communications Commissions (FCC) reserved the frequency band 5850-5925 MHz for exclusive use by ITS applications in 1999. Later (2007) it was specified to divide into seven 10 MHz channels with the option of combining two channels (20MHz) [34].

Finally, in EU, it was defined all the frequency ranges that are shown in Table 2, meanwhile in U.S it was defined the frequency range 5850MHz-5925MHz. It means that both frequency ranges are shifted 5MHz, therefore they are not compatible at physical layer.

Type	Channel type	Central frequency	IEE 802.11[32] channel number	Channel spacing	Default data rate	TX power limit	TX power density limit
ITS-G5A	G5-CCH	5900 MHz	180	10 MHz	6Mbit/s	33 dBm EIRP[1]	23 dBm/MHz
	G5-SCH2	5890 MHz	178	10 MHz	12Mbit/s	23 dBm EIRP	13 dBm/MHz
	G5-SCH1	5880 MHz	176	10 MHz	6Mbit/s	33 dBm EIRP	23 dBm/MHz
ITS-G5B	G5-SCH3	5870 MHz	174	10 MHz	6Mbit/s	23 dBm EIRP	13 dBm/MHz
	G5-SCH4	5860 MHz	172	10 MHz	6Mbit/s	0 dBm EIRP	-10 dBm/MHz
ITS-G5D	G5-SCH5	5910 MHz	182	10 MHz	6Mbit/s	0 dBm EIRP	-10 dBm/MHz
	G5-SCH6	5920 MHz	184	10 MHz	6Mbit/s	0 dBm EIRP	-10 dBm/MHz
ITS-G5C	G5-SCH7 (not used)	5470 to 5725 MHz	94 to 145	Several	Dependent on channel spacing	30 dBm EIRP (DFS master)	17 dBm /MHz
						23 dBm EIRP (DFS Slave)	10dBm/MHz

Table 2. European channel allocation for ITS services

During this project, we have focused on European standard as well as the European protocol stack, due to the upper layers in European Standard are different from American.

The European standard, ITS-G5, defines four different bands; depending on the final application the frequency range will be different:

- ITS-G5A frequency band is exclusively for ITS road traffic safety applications.
- ITS-G5B frequency band is set for ITS non-safety road traffic applications.
- ITS-G5C is referred to broadband radio access networks (BRAN), radio local area network (RLAN) and wireless local area network (WLAN).
- ITS-G5D frequency band is set for future usage of ITS road traffic applications.

The standard also defines the maximum transmission power limits and bitrates. Every channel has its own bitrate associated, which is mandatory to use. On the other hand, it is given the maximum transmission power density limit, because depending on the environmental conditions can be more stringent requirements. As an example, in Figure 9 is shown how the safety applications are allowed to use a higher transmit power, whereas the non-safety applications transmit less power in order to not interfere in the other channels.

Decentralized Congestion Control (DCC) mechanism is responsible for controlling the transmission power in ITS-G5 stations. It is only applicable to G5A, G5B and G5D bands.

[1] EIRP: Effective Isotropic Radiated Power

It is defined in the standard ITS 102-687 [35] and the main targets of DCC mechanism are to maintain network stability, throughput efficiency and fair resource allocation.

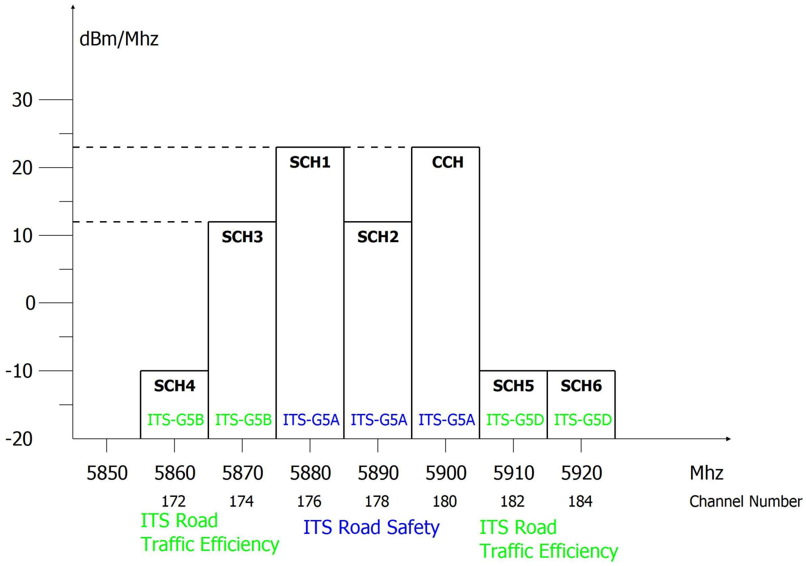


Figure 9. Maximum limit of mean spectral power density for each channel type in ITS-G5A, ITS-G5B, and ITS-G5D

3.4 Description of Outside the Context of a BSS (OCB) mode

Once the spectrum was decided by the FCC in 1999, it was necessary to take the next step and develop a standard that would make appropriate use of the available frequency band. Few years later, the American Society for Testing and Materials published the ASTM E2212 standard specification [36]. This standard was the first that defined the ITS technology and it was based on a modified version of 802.11a OFM physical layer. After it was published, the ASTM took care of allowing the easy use of hardware designed for 802.11a, since the ASTM was concerned that the new standard would be hard to implement and maintain as an independent add-on to both 802.11 and 802.11a.

Furthermore, the hardware was not the only problem to be caused by basing the new protocol on 802.11a. Protocol 802.11a defines a fix association between stations and APs in the mode BSS. This association is extremely inappropriate for a situation of high mobility as the kind of networks defined in ITS. Moreover, the stations can only join to one BSS domain, which is also against the idea of an environment with multiple devices communicating at the same time.

One of the solutions to adapt 802.11a to 802.11p was to introduce in the MAC layer a new operation mode, called OCB mode. This mode allows operating without BSS joining, authentication, association procedures, beacons, encryption, etc.

The OCB mode allows every station to work as an AP. There is not an association between stations that would require a negotiation of the transmission conditions. Therefore, it is mandatory the definition of well-known channels to transmit and receive information. Besides, other modification was the frequency bandwidth changed to 5MHz and 10MHz, making the communications more robust to the multi-path delay and inter-symbol interference because this new specification doubled all the symbols times as it was shown in the previous section.

The final result was published in 2010 and it was named 802.11p standard. Finally it was integrated in 802.11 standard in 2012 [37].

3.5 Description of Linux wireless networking

Since 3.19 Linux kernel version (release notes [38] and official kernel commit [39]), the OCB mode is available in kernel configuration menu. It means that the system enables the Wi-Fi devices to use this operation mode, similarly to the other operation modes such as managed, monitor, IBSS, AP, etc. This is the first determining and necessary change to use WLANp technology on Linux devices. However, it is not enough to implement the whole stack of the E.T.S.I. ITS G5 standard. The manufacturer Atheros released the source code of ATH9K series. Therefore, it was possible to modify the Wi-Fi driver to implement the first two layers of the stack (physical and MAC), as well as it offered an access point to user level.

Nowadays, those driver code changes are being included in the latest releases of Linux kernel. However, these modifications are being gradual because they should be tested and merged to the Master branch of Linux kernel of Linus Torvalds[25]. Hence, currently all these changes have to be done manually. In the following sections it will be explained how to include them in order to be capable of operating in OCB mode and working with Atheros cards.

The most common framework used by Wi-Fi drivers is mac80211/cfg80211, as it is illustrated in Figure 10, which describes the Linux kernel implementation architecture. Taking into account this stack, it is straightforward to know which layers must be modified in order to support WLANp.

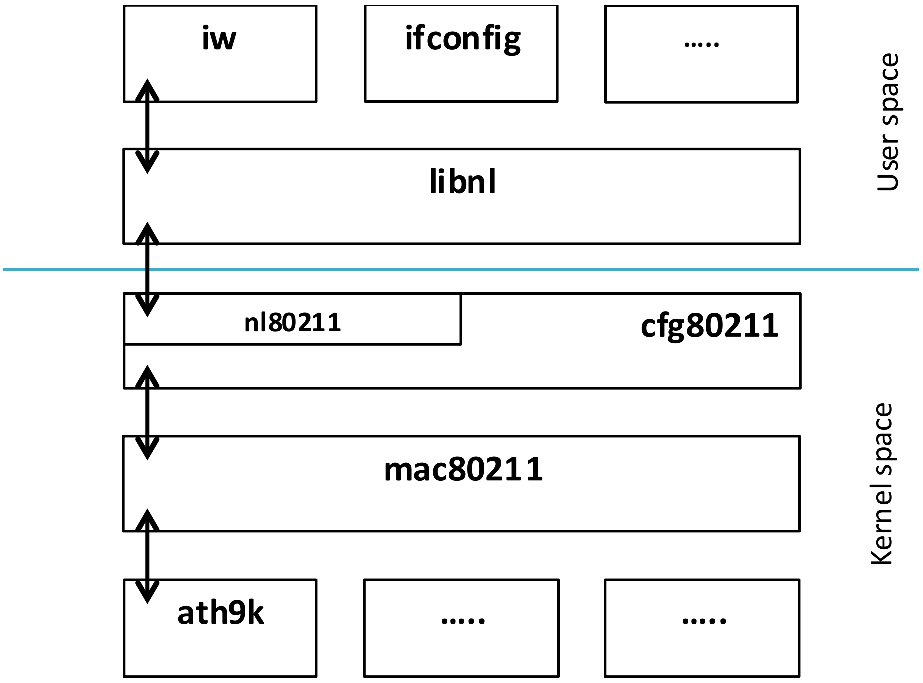


Figure 10. IEEE 802.11 Linux kernel implementation architecture [30]

3.5.1 mac80211 layer

There are two kinds of Wi-Fi devices, depending on where MLME (MAC Layer Management Entity), whose target is to implement the Physical layer state machine, is implemented. It can be implemented either by firmware or hardware, or by host-based software that is launched in the main CPU. When the MLME is implemented by hardware the Wi-Fi device is named "full MAC" and when it is by software then is known as Soft MAC, which is the most common type nowadays because it is able to offer an accurate control of the hardware.

3.5.2 cfg80211

The cfg80211 is a layer between user space and mac80211 that provides the management services in the kernel side. Therefore, it allows the user space to control and configure wireless devices.

3.5.3 nl80211

It is in the same layer that cfg80211. However, it is controlled only by the user space. It provides a flexible way to enable the communication between user space and kernel through Netlink socket. Netlink is a datagram oriented service which is the responsible for the communication between the user space processes and the kernel modules [40].

3.6 Software dependencies

In this section the software dependencies that were necessary to install before starting the installation of the kernel as well as user programs are described. Moreover, the software versions will be given. However, it does not mean that this project cannot work in other software releases, but that these other software versions have just not been tested.

Dependency	Program which needs it
build-essentials	Common libraries for compiling
libncurses-dev	*menuconfig* – If you want to edit the kernel configuration though a graphical interface
libnl1-1.1.4-3.6	IW program
libnl-1_1-devel-1.1.4-3.6	IW program
python-M2Crypto-0.22.5-5.1	Wireless regulatory database
libgcrypt-devel-1.6.1-29.1	Central Regulatory Domain Agent
libgpg-error-devel-1.13-3.6	Central Regulatory Domain Agent

Table 3. Software dependencies

3.7 Installation of modified kernel and user programs

Before compiling the kernel, it had been modified the source code of the ATH9K driver. The files that must be changed are shown in Figure 11 and the right modifications are described in

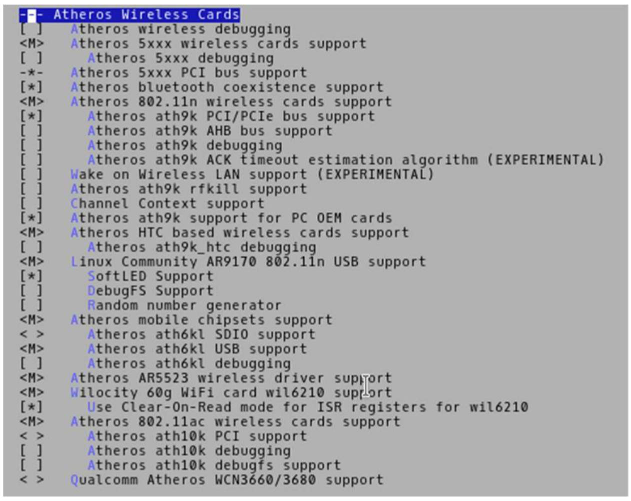
Appendix A, as well as in the original commit of "802.11p on Linux" [20]. Those files can be manually modified in any 3.19 upper Linux kernel version or they can be replaced by one of the patches that were developed during the project, patch kernel 4.2.8 [41] or kernel 4.7.rc3[42] (Taking into account the kernel version). Furthermore, if the last stable release is used, it is advisable to check the source code before changing it, as for example in 4.7 Linux kernel release, some of those code lines have been included. Consequently, in the future the ATH9K driver will be updated to support 8011.p in all the software releases and no manually changes should be done.

- `drivers/net/wireless/ath/ath9k/ani.c`
- `drivers/net/wireless/ath/ath9k/ath9k.h`
- `drivers/net/wireless/ath/ath9k/common-init.c`
- `drivers/net/wireless/ath/ath9k/debug.c`
- `drivers/net/wireless/ath/ath9k/htc_drv_init.c`
- `drivers/net/wireless/ath/ath9k/hw.c`
- `drivers/net/wireless/ath/ath9k/hw.h`
- `drivers/net/wireless/ath/ath9k/init.c`
- `drivers/net/wireless/ath/ath9k/main.c`
- `drivers/net/wireless/ath/ath9k/recv.c`
- `drivers/net/wireless/ath/regd.c`

Figure 11. Driver files that have been changed

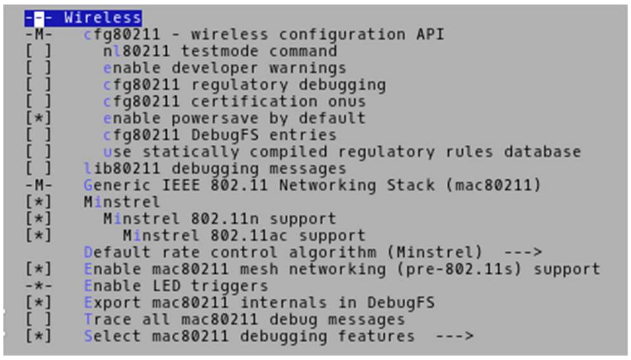
The only way to apply the changes that had been made is compiling the kernel. Moreover, in the previous steps to the compilation, the proper configuration flags of the modules must be activated in order to support OCB mode. This flag is CONFIG_MAC80211_OCB_DEBUG. On the other way, it must be kept the default configuration for the ath9k.

Figure 12 shows the default configuration for the drivers of Atheros wireless cards. Figure 13 shows the default configuration of the 802.11 stack that was explained in Figure 10; in this menu it is not necessary to enable more flags. However, in the next menu is crucial changing the debugging features as it is shown in Figure 14. It means activating the "verbose OCB debugging". Afterwards, the configuration can be saved and the configuration menu closed.



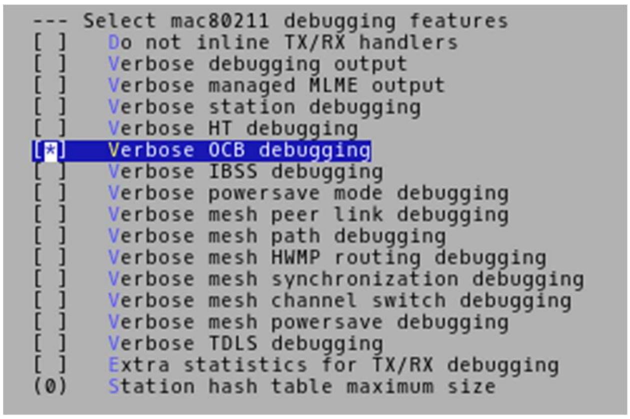
```
--- Atheros Wireless Cards
[ ]   Atheros wireless debugging
<M>   Atheros 5xxx wireless cards support
[ ]     Atheros 5xxx debugging
-*-   Atheros 5xxx PCI bus support
[*]   Atheros bluetooth coexistence support
<M>   Atheros 802.11n wireless cards support
[*]     Atheros ath9k PCI/PCIe bus support
[ ]     Atheros ath9k AHB bus support
[ ]     Atheros ath9k debugging
[ ]     Atheros ath9k ACK timeout estimation algorithm (EXPERIMENTAL)
[ ]   Wake on Wireless LAN support (EXPERIMENTAL)
[ ]   Atheros ath9k rfkill support
[ ]   Channel Context support
[*]   Atheros ath9k support for PC OEM cards
<M>   Atheros HTC based wireless cards support
[ ]     Atheros ath9k_htc debugging
<M>   Linux Community AR9170 802.11n USB support
[*]     SoftLED Support
[ ]     DebugFS Support
[ ]     Random number generator
<M>   Atheros mobile chipsets support
< >     Atheros ath6kl SDIO support
<M>     Atheros ath6kl USB support
[ ]     Atheros ath6kl debugging
<M>   Atheros AR5523 wireless driver support
<M>   Wilocity 60g WiFi card wil6210 support
[*]     Use Clear-On-Read mode for ISR registers for wil6210
<M>   Atheros 802.11ac wireless cards support
< >     Atheros ath10k PCI support
[ ]     Atheros ath10k debugging
[ ]     Atheros ath10k debugfs support
< >   Qualcomm Atheros WCN3660/3680 support
```

Figure 12. Default configuration for Atheros drivers before kernel compiling



```
--- Wireless
-M-   cfg80211 - wireless configuration API
[ ]     nl80211 testmode command
[ ]     enable developer warnings
[ ]     cfg80211 regulatory debugging
[ ]     cfg80211 certification onus
[*]     enable powersave by default
[ ]     cfg80211 DebugFS entries
[ ]     use statically compiled regulatory rules database
[ ]   lib80211 debugging messages
-M-   Generic IEEE 802.11 Networking Stack (mac80211)
[*]   Minstrel
[*]     Minstrel 802.11n support
[*]       Minstrel 802.11ac support
      Default rate control algorithm (Minstrel)  --->
[*]   Enable mac80211 mesh networking (pre-802.11s) support
-*-   Enable LED triggers
[*]   Export mac80211 internals in DebugFS
[ ]   Trace all mac80211 debug messages
[*]   Select mac80211 debugging features  --->
```

Figure 13. IEEE 80211 kernel architecture default configuration



```
--- Select mac80211 debugging features
[ ]   Do not inline TX/RX handlers
[ ]   Verbose debugging output
[ ]   Verbose managed MLME output
[ ]   Verbose station debugging
[ ]   Verbose HT debugging
[*]   Verbose OCB debugging
[ ]   Verbose IBSS debugging
[ ]   Verbose powersave mode debugging
[ ]   Verbose mesh peer link debugging
[ ]   Verbose mesh path debugging
[ ]   Verbose mesh HWMP routing debugging
[ ]   Verbose mesh synchronization debugging
[ ]   Verbose mesh channel switch debugging
[ ]   Verbose mesh powersave debugging
[ ]   Verbose TDLS debugging
[ ]   Extra statistics for TX/RX debugging
(0)   Station hash table maximum size
```

Figure 14. mac80211 debugging features configuration

The paths of the previous configuration menus are:

- Figure 12: *Device drivers -> Network device support -> Wireless LAN -> Atheros Wireless Cards.*
- Figure 13: *Networking support - > Wireless.*
- Figure 14: *Networking support - > Wireless -> Select mac80211 debugging features.*

After compiling the kernel, all the necessary changes in the kernel space will have already been done. Therefore, the next steps will be to install all the necessary resources for the user space. Besides, this is the only way to check that every change was successfully implemented.

3.7.1 IW

IW Linux program allows users to use and configure the wireless devices. This program works directly with nl80211 layer. It supports the newest drivers that have been added to the kernel. Furthermore, the last release includes the OCB mode in order to work without an IBSS. However, during this project it was used the initial modified IW program [43].

In order to check if your IW program supports OCB, it could be checked as is shown in Figure 15. If there is no answer to this command, it means that your IW version does not support OCB. Therefore, the IW modified program must be installed [43].

```
/sbin/iw | grep -i ocb
        dev <devname> ocb leave
        dev <devname> ocb join <freq in MHz> <5MHZ|10MHZ> [fixed-freq]
```

Figure 15. IW output – OCB support

Finally, it has been shown that sometimes an installation error occurred. It has not been bared the root of the problem. However, it had been found a solution for this error which is described in Figure 16. The reason of this error could be a compatibility issue between the IW program and the Operating System, as it has only been found in some Linux distributions.

```
#"/sbin/iw: error while loading shared libraries: libnl.so.1: cannot open shared
object file: No such #file or directory"
#libnl-1
        nano /usr/include/netlink/object.h
#libnl-3
        nano /usr/include/libnl3/netlink/object.h
#Coment the following declarations:
        extern int nl_object_get_refcnt(struct nl_object *);
        extern struct nl_cache * nl_object_get_cache(struct nl_object *);
        static inline void * nl_object_priv(struct nl_object *obj);
```

Figure 16. Error during the IW program installation

3.7.2 Wireless regulatory database for CRDA

We know that 802.11 standard defines several frequency ranges, although the 2.4GHz and 5GHz bands are the most used for WLAN. Each frequency range is divided into multiple sub-channels. However, each country can apply their own regulations; it means that the channel allocation can be different between countries as well as the maximum transmission power which could be limited in some regions. These local regulations are defined in the wireless regulatory domain which is a regulatory database defined in the user space. It allows distributions to provide updates without kernel upgrades.

In order to update the regulatory domain to support WLANp channel allocation, it is only necessary to change the database from the regulatory domain. The database is defined in "db.txt" file. As it could be seen inside, it is defined the frequency ranges and the allowed power that it can be used in each country. Furthermore, it can be used the official release of wireless regdb [44], and then the database can be modified.

In the CTU-IIG project [30] it was modified the information about Germany (DE) country, as it is shown in Figure 17. However, it was added a new country, defined as country AA, in order to raise the power transmission for each frequency range. It is shown in the Figure 18.

Afterwards it is only necessary to compile and install the wireless regulatory domain.

```
Country DE: DFS-ETSI
  # entries 279004 and 280006
  (2400 - 2483.5 @ 40), (100 mW)
  # entry 303005, 304002 and 305002
  (5150 - 5350 @ 80), (100 mW), NO-OUTDOOR
  # entries 308002, 309001 and 310003
  (5470 - 5725 @ 80), (500 mW), DFS
  # For ITS-G5 evaluation
  (5850 - 5925 @ 20), (100 mW), NO-CCK, OCB-ONLY
  # 60 gHz band channels 1-4, ref: Etsi En 302 567
  (57240 - 65880 @ 2160), (40), NO-OUTDOOR
```

Figure 17. Germany frequency ranges definition

```
country AA:
        (2402 - 2482 @ 40), (30)
        (5170 - 5250 @ 80), (30)
        (5250 - 5330 @ 80), (30)
        (5490 - 5730 @ 80), (30)
        (5735 - 5835 @ 80), (30)
        # For ITS-G5 evaluation
        (5840 - 5935 @ 10), (30)
```

Figure 18. Added country for regulatory domain definition

3.7.3 Central Regulatory Domain Agent

Finally, the previous step to take effect, it must be installed a CRDA (Central Regulatory Domain Agent). During the wireless regulatory database installation several files were created (/usr/lib/crda/pubkeys), "regulatory.bin" and the keys "root.key.pub.pem" and "linville.key.pub.pem". These files are used by CRDA to complete the installation. It can be used the CRDA version provided by CTU-IIG [45] or the version included in the official release [46]. In both cases, it is essential to copy the pubkeys (located in /usr/lib/crda/pubkeys.), which were generated by wireless regdb installation, into CRDA/pubkeys.

Finally, to check that the CRDA and wireless regdb were successfully installed, it can be executed the command that is shown in Figure 19. In the same figure it is shown the right answer when everything is correct. Moreover, IW program can be used to check if we can change the country of the regulatory domain. Executing the command "iw reg get" we see the current country used and if we execute "iw reg set AA", the system changes the country to AA.

```
# Test CRDA + generated regulatory.bin
/sbin/regdbdump ../802.11p-wireless-regdb/regulatory.bin | grep -i ocb
       country 00: invalid
       (5850.000 - 5925.000 @ 20.000), (20.00), NO-CCK, OCB-ONLY
```

Figure 19. Command executed and expected result when CRDA and wireless regdb are successfully installed.

3.8 ITS-G5 transceiver channel configuration

The Road Side Units (RSU) and On Board Units (OBU) are an indispensable part for any ITS network. Depending on the number of available channels and the number of transceivers, there are three kinds of RSU and OBU: single-radio single-channel, single-radio multi-channel, multi-radio multi-channel. This difference derives from the three kinds of standards that are being developed, Japan, EU and U.S. The EU family of standards assumes that multi-radio multi-channel system.

The three standards have huge differences because they are implemented with different protocol stack. For example, there are differences in the ways the standards access the channel, in the case of U.S. standard, which is called Wireless Access in Vehicular Environments (WAVE), uses an alternating access scheme with Enhanced Distributed Channel Access (EDCA) systems for each respective channel type. On the other hand, the European standard ITS G5, the model is based on a state machine. Moreover, depending on the final application it will use one channel or other.

This state machine it is defined in the standard TS 102-687 [35] which is defined by the Decentralized Congestion Control Mechanisms for ITS. One of the components of DCC is the Access Control Loop, which implements this state machine. Every channel is controlled by one instance of this states machine. It implements three different states, depending on the observed busy time of the medium.

Relaxed	15% - 40 %
Active	<15%
Restrictive	>40%

For example, it is recommended to use a threshold between 15% and 40% for CCH. It means that, if the busy time is less than 15%, the state will be active. If it is between 15% - 40%, it will be relaxed and for more than 40% the state will be restrictive.

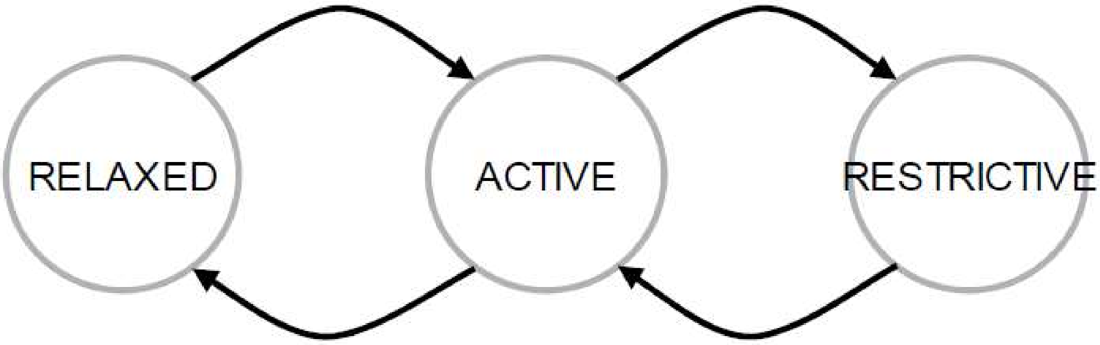


Figure 20. DCC access state machine

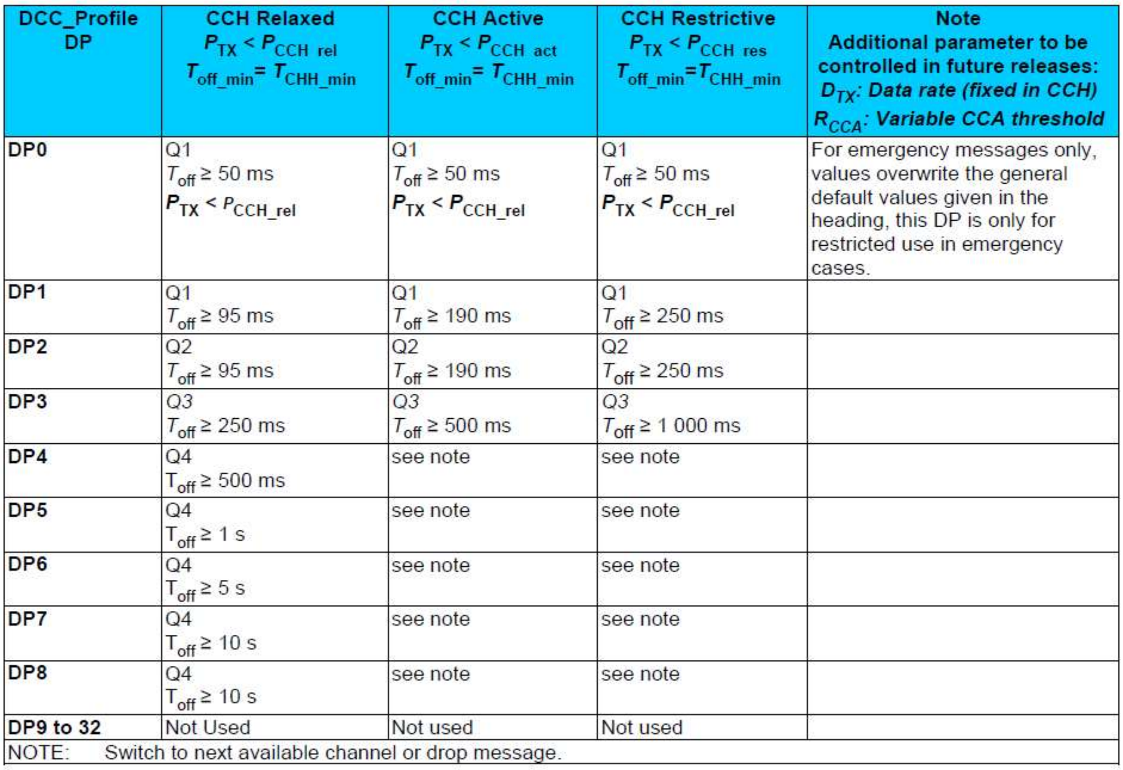
One important consequence of being in one state or other is that the minimum time between two consecutive messages will change. DCC mechanism also defines different profiles depending on the importance of the messages. For example, messages for road safety have higher frequency than for traffic efficiency. An example is given in Figure 21 in which, depending on the profile and the state machine, the time between messages is different.

3.8.1 General considerations of DCC

ITS stations equipped with ITS G5A and G5B transceiver shall adopt channel access requirements for all transmitted messages based on cross layer DCC. The DCC is supposed to dynamically adapt to channel conditions by changing certain parameters of the MAC and PHY, as it was explained in the section 3.3.

3.8.2 Access layer requirements based on DCC profiles

DCC defined 33 different profiles (DP0 to DP32), each profile is given for every channel. ITS station operating on an ITS G5A channel shall comply with the parameters given for the different channels and DCC states, as it is shown in Figure 21. Please note that in this figure there are defined the parameters Qx. These are the queues where the messages are sent. This way, profile DP0 and profile DP1 serve the same queue. The rest of timing requirements can be found in ITS 102-724 [47].



DCC_Profile DP	CCH Relaxed $P_{TX} < P_{CCH\ rel}$ $T_{off\_min} = T_{CHH\_min}$	CCH Active $P_{TX} < P_{CCH\ act}$ $T_{off\_min} = T_{CHH\_min}$	CCH Restrictive $P_{TX} < P_{CCH\ res}$ $T_{off\_min} = T_{CHH\_min}$	Note Additional parameter to be controlled in future releases: *D_{TX}: Data rate (fixed in CCH)* *R_{CCA}: Variable CCA threshold*
DP0	Q1 $T_{off} \geq 50$ ms $P_{TX} < P_{CCH\_rel}$	Q1 $T_{off} \geq 50$ ms $P_{TX} < P_{CCH\_rel}$	Q1 $T_{off} \geq 50$ ms $P_{TX} < P_{CCH\_rel}$	For emergency messages only, values overwrite the general default values given in the heading, this DP is only for restricted use in emergency cases.
DP1	Q1 $T_{off} \geq 95$ ms	Q1 $T_{off} \geq 190$ ms	Q1 $T_{off} \geq 250$ ms	
DP2	Q2 $T_{off} \geq 95$ ms	Q2 $T_{off} \geq 190$ ms	Q2 $T_{off} \geq 250$ ms	
DP3	Q3 $T_{off} \geq 250$ ms	Q3 $T_{off} \geq 500$ ms	Q3 $T_{off} \geq 1\ 000$ ms	
DP4	Q4 $T_{off} \geq 500$ ms	see note	see note	
DP5	Q4 $T_{off} \geq 1$ s	see note	see note	
DP6	Q4 $T_{off} \geq 5$ s	see note	see note	
DP7	Q4 $T_{off} \geq 10$ s	see note	see note	
DP8	Q4 $T_{off} \geq 10$ s	see note	see note	
DP9 to 32	Not Used	Not used	Not used	
NOTE: Switch to next available channel or drop message.				

Figure 21. CCH: Access layer requirement table based on DCC profiles and congestion state. (Table 1 [47])

Every channel that is active has to take into account its own table with its times. If it is a multichannel station, it will implement one state machine per active channel. Finally, the message received by the access layer will be transmitted using the first possible channel.

3.8.3 Access layer Channel monitoring

The Access Layer should monitor the following status indicators of ITS G5 and G5B channels [47]:

- Channel Busy Ratio: Relative time in % the channel is busy.
- Receiver Signal Strength Indicator statistic.
- Frames transmission indication (if a message has been successfully transmitted or dropped)

Optionally:

- Notification of Tx power reduction on a per message base in case the message could not be transmitted using the asked Tx power level.

3.8.4 ITS G5A

As it was explained the frequency band of ITS G5A is allocated between 5.875 GHz-5.905 GHz and it contains the channels CCH, SCH1 and SCH2 whose purpose is the road safety service.

These channels are under control of DCC mechanism. The Control Channel (CCH) will be the default channel for the transmission of DP1 and DP2 type messages (see Figure 21). While the DP3 to DP8 type messages are only allow if the state of the machine is “Relaxed”. The Service Chanel 1 (SCH1) is the default channel for announcing and offering ITS services for safety and road efficiency under the DCC state "Active" and "Restrictive" of the CCH. Finally, the Service channel 2 (SCH2) is the second service channel for ITS G5A and it is used as an alternative channel for traffic safety-related services. Nevertheless, his frequency allocation is between CCH and SCH1 since it could cause interference issues.

3.8.5 ITS G5B

ITS G5B frequency band is from 5.855 GHz to 5.9875 GHz and it includes the SCH3 and SCH4. Besides, it is considered for general purpose ITS services.

As well as the example of Figure 21, ITS G5B band has its own requirements for the access layer. They are shown in Table 4 and 5 of TS 102 724 standard [47].

3.8.6 Multi-hop Channel Operations

Multi-hopping of any type of message is only allowed using CCH and if its DCC state is "Relaxed". It means that a single transceiver in safety critical conditions cannot hop to another channel while its state is "Active" or "Restrictive". Furthermore, SCH1 allows multi-hop for any type of messages, provided that its DCC state is "Relaxed". If any message has to be forwarded (only in Active or Restrictive), it can be used the SCH2 in order to avoid increasing channel congestion on SCH1. Therefore, if all ITS G5A channels are congested and their status is "Active" or "Restrictive", it is not allowed multi-hopping.

On the other hand, the channels that are available for non-safety services (G5B) can use multi-hop capabilities in order to avoid congestion situations. Besides, as ITS G5A, multi-hop operations are only allowed in DCC state "Relaxed".

3.9 ITS G5 functional transceiver configuration

3.9.1 Example 1 – Switching channel

The following example has been abstracted from Annex D of TS 102-724 standard [47].

ITS G5 communications should be capable of operating on single channels or multiple channels according to the requirements of ITS applications.

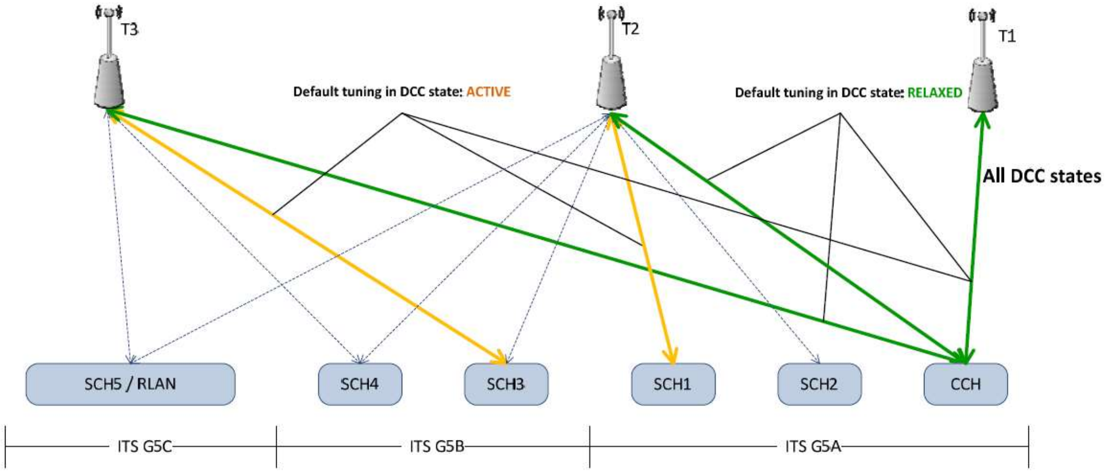

T3
T2
T1
Default tuning in DCC state: ACTIVE
Default tuning in DCC state: RELAXED
All DCC states
SCH5 / RLAN
SCH4
SCH3
SCH1
SCH2
CCH
ITS G5C
ITS G5B
ITS G5A

Figure 22. Multi-Transceiver and Multi-Channel Architecture Figure D.1 [47]

Every ITS stations which can use more than one transceiver should operate the next configurations:

- Transceiver Configuration 1 (T1): The transceiver works exclusively in CCH.
- Transceiver Configuration 2 (T2): The transceiver can be tuned on demand to an arbitrary ITS channel (G5A or G5B). If the state is "Active" or "Restrictive", the Services announcements should be taking place on SCH1. If the state is "Relaxed", the announcement will be on CCH.
- Transceiver Configuration 3 (T3): The transceiver can be tuned on demand to an arbitrary ITS channel (G5A or G5B). If the state is "Active" or "Restrictive", the Services announcements should be taking place on SCH3. If the state is "Relaxed", the announcement will be on CCH.

In non-safety applications, the transceiver can be operated in T2 and T3 configurations. Moreover, in this case G5C channels are allowed, but these channels are not under control of DCC mechanism.

3.9.2 Example 2 - Transmission power tuning function

In section 3.8 was explained the DCC mechanism. It was shown that this mechanism is able to adjust the power transmission depending on the observed busy time of the medium. Some reasons why DCC implements this feature are because of channel interference. When an ITS station transmits a message it introduces spurious emissions in adjacent channels or even channels beyond.

DCC strategies define some methods in order to reduce the effects of this problem. Furthermore, it should be taken into account, that every station is continually monitoring its active channels. They monitor:

- Channel Busy Ratio.
- Receiver Signal Strength Indicator statistic.

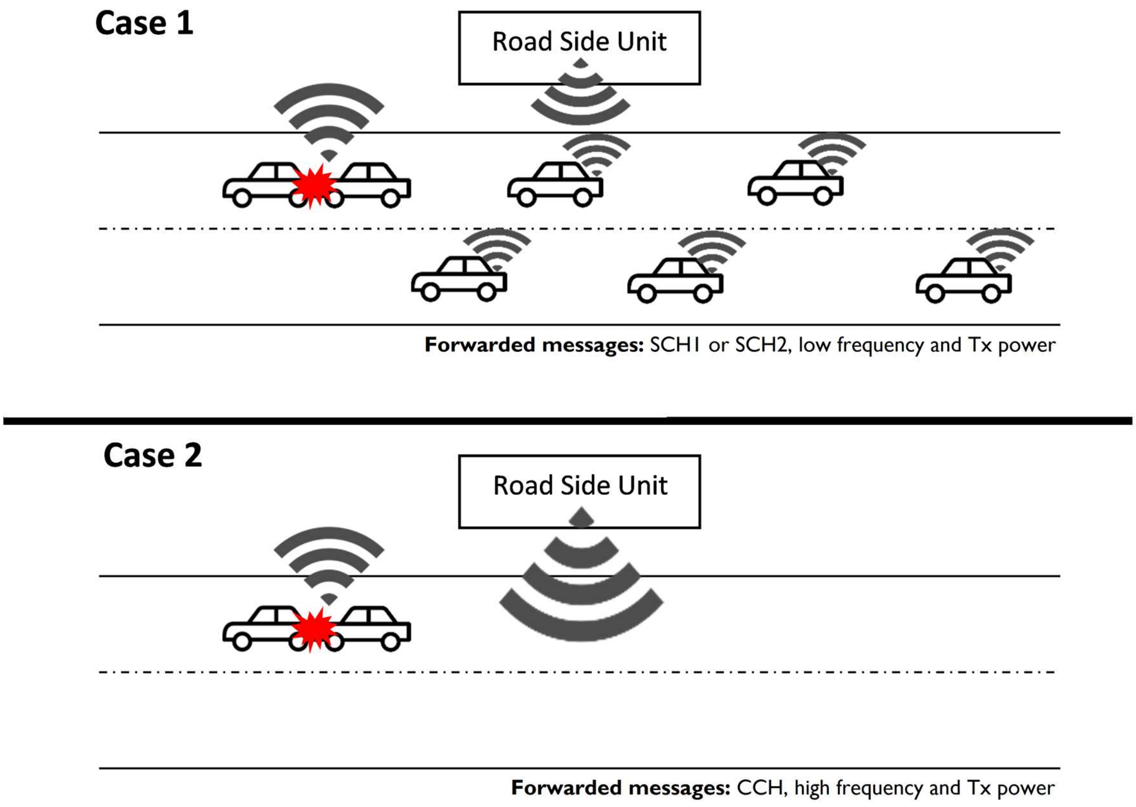

Case 1
Road Side Unit
Forwarded messages: SCH1 or SCH2, low frequency and Tx power
Case 2
Road Side Unit
Forwarded messages: CCH, high frequency and Tx power

Figure 23. Example ITS environment

An example is given below Figure 23. In both cases, an accident has occurred. In the first case, the environment is crowded. However, in the second one it is isolated. In both scenarios, the On Board Unit of the damaged car sends a broadcast message through the CCH. The nearest Road Side Unit realises that an accident took place. Afterwards, it sends the message warning the rest of the drivers. Then, this message is forwarded by each OBU that has received it.

- First case: The OBU receives the message and checks if the state of the CCH is “Relaxed”. In congested situation should be “Active” or “Restrictive”. Therefore, the message is forwarded through the SCH1 (if it is available, otherwise SCH2) with lower frequency and Tx power, in order to not saturate the spectrum.
- Second case: The OBU receives the message and checks if the state of the CCH is “Relaxed”. It is the only OBU in this place. Therefore, the message is forwarded using de CCH channel with the higher frequency and power, in order to be received for other distant stations.

3.10 Higher-layer protocols

We have seen how the first two layers of ITS networking (Access layer) work, as well as some differences with the American standard (IEEE WAVE) in these layers. During this project we have focused on 802.11p because ITS G5 makes heavy use of this standard and radio technology. However, there are more access protocols and technologies. Therefore, it is essential to show the other access protocols as well as to give a short overview of the upper layers of European standard.

It was explained that ETSI ITS G5 is based on multi-transceiver to cope with the multi-channels operation. It means that every vehicle that wants to exchange safety information should be equipped with one transceiver that it is always tuned to the control channel. And optionally, it can be equipped with more transceivers to exchange information using the service channels.

As it can be shown in Figure 24, there are cross-layer services that are provided by vertical Management and Security layers (more information in chapter 7 of [37]). The Management layer is responsible for configuration of ITS station, it means exchanging information among the different layers. On the other hand, the Security layer provides security and privacy services at different layers of the communication stack.

As it is defined in the OSI stack, the highest layer is the Application layer, which focuses on the final user or client and mostly it has commercial targets as traffic efficiency, infotainment and business but it also implements some non-commercial applications such as safety applications.

Just below, it is the Facilities layer, which provides common support for all levels of ITS stack. It is subdivided into Application, Information and Communication support. Examples of protocols used in this layer are Cooperative Awareness Messages (CAM), Decentralized Environmental Notification Messages (DENM) or the Local Dynamic Map (LDM) which is a database that stores information about the local neighbourhood.

Straightaway, it is defined the Transport and Network layer, to provide not only standard TCP/UDP over IP services, but also the Basic Transport Protocol (BTP) over GeoNetworking service. The aim of GeoNetworking is to employ the access layer for multi-hop data dissemination to nodes that are identified solely by their geographic position. It means that instead of addressing messages to named nodes, they need to be delivered to nodes identified by their distance to the sender, the current road or similar attributes.

At the bottom, we can find the Access layer, where 802.11p is placed, besides ITS G5, Wi-Fi or other radio technologies such as 3G support. Whole ITS stack it is defined in TS 102 636-3 [48] standard.

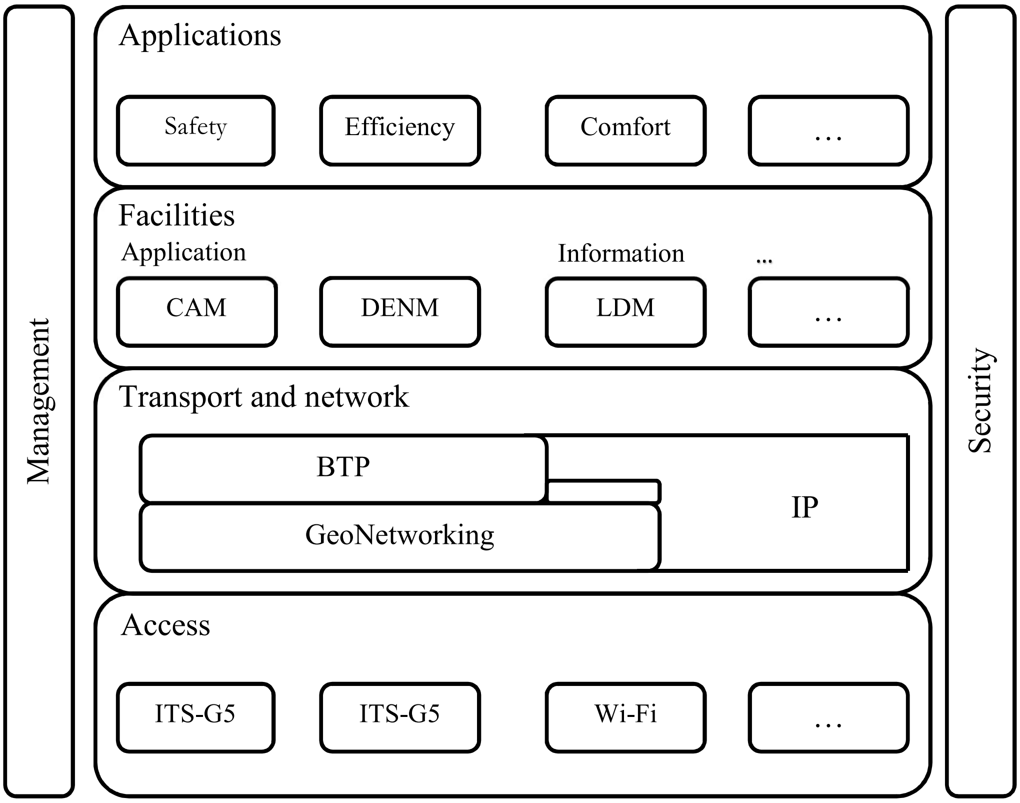


Figure 24. An excerpt from the ETSI ITS protocol stack (page 128 [37])

3.10.1 Cooperative Awareness Messages (CAM)

This protocol is defined in EN 302 637-2 [49] and it defines the Cooperative Awareness Messages as messages that are exchanged in ITS network between stations to create and maintain awareness of each other and to support the cooperative performance of vehicles using the road network. Depending on the type of station, a CAM message may contain different information. For example, a vehicle station includes in CAM messages status information as time, position, motion state, activated systems, etc. Besides, it is included information about the vehicle such as the dimensions, type and role in the road traffic, etc.

All this information is used by the receiver stations for example to estimate the collision risk with the originating station and if it is necessary, the OBU may inform the driver of the vehicle via the Human Machine Interface (HMI).

The generation and the transmission of CA messages are managed by the Cooperative Awareness (CA) basic service. The CA is a service from Facilities layer and it provides to CAM the sending and receiving services. By this way, multiple ITS applications can rely on CA basic service (An example of how CA basic service works is given in page 37 of [50]).

3.10.2 Decentralized Environmental Notification Messages (DENM)

This protocol is defined in EN 302 637-3 [51]. DENM is used to describe the different situations or events that can be detected by a station. Therefore, it is used by the ITS Application to alert road users of a detected event. The transmission of DENM should persist while the event is present and the receiver station may decide if this event involves an alert and warning to the user or not.

The following DENM are defined:

- New DENM: A new message is generated by the DEN basic service. It is given a new identifier, as well as is included information about the position, event type, detection time, etc. (more attributes are defined in chapter 7 of [51]).
- Update DENM: It is a message transmitted by the same origin station, with the same identifier. However, it is updated the information about the event.
- Cancellation DENM: Also it is send by the same station with the same message ID, and whose aim is to inform about the termination of an event.
- Negation DENM: This type is used to announce the termination of an event if the originating station has the capacity to detect the termination of an event which has been previously announced by other station.

An example of how works the DEN basic service is given in page 25 of [50].

3.10.3 Local Dynamic Map (LDM)

This protocol is defined in EN 302 895 [52] standard. The LDM is a conceptual data store located in every ITS station, which contains information that could be relevant to the operation of ITS Applications and related road safety and traffic efficiency. All this information can be provided for any ITS source independently of ITS transmission protocol.

Thus, the basic functionality of LDM is to provide a repository of information for Facilities and Applications. Facilities such as the CA and DEN basic services can store information into the LDM. Moreover, the Applications can retrieve information from the LDM. An example for the last three protocols explained is given in Figure 25 and how CA and DEN service can access to LDM facility to read or write information.

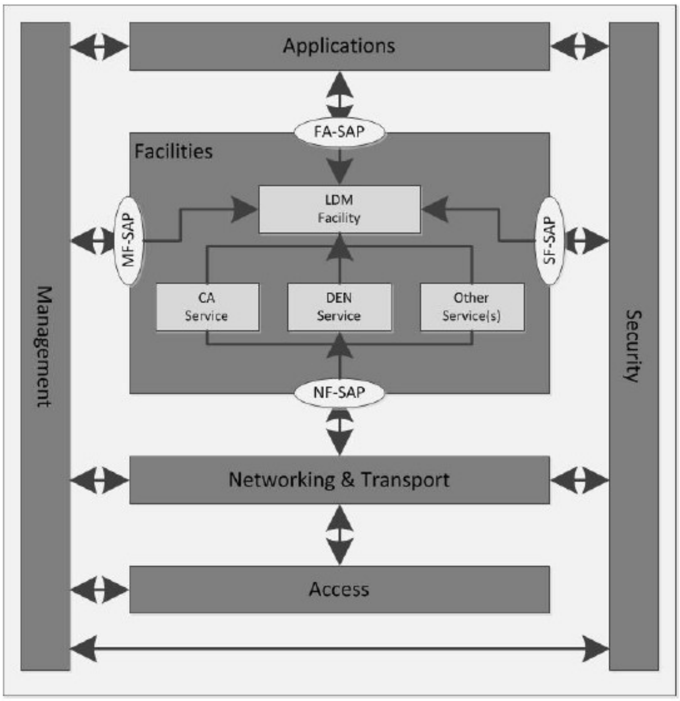


Figure 25. CAM, DENM and LDM interaction (Figure 1 of [52]).

An example of how the LDM works and the information support facilities functional requirements are is given in page 47 of [50].

3.10.4 Basic Transport Protocol (BPT)

This protocol is defined in EN 102 636-5-1 [53] standard. The basic Transport Protocol provides an end-to-end service without connection in ITS ad hoc network. Similar to UDP (User Datagram Protocol), it offers a minimal transport service. Besides, it also provides to protocols of Facilities layer to have directly access to the services provided by GeoNetworking. It means that this protocol enables a communication between the immediate upper layer and the Transport and Network layer.

3.10.5 GeoNetworking

This protocol is defined in TS 102 636-4-1 [54] and TS 102 636-3 [48] standards. GeoNetworking is a protocol that provides the transport of the packets in ad hoc network. The ITS networks are based on geographical addressing and routing between ITS stations.

GeoNetworking is a protocol designed for highly mobile networks nodes and frequent changes in the network topology. The protocol supports a distributed and self-organizing ad-hoc networking concept. It is able to send messages to one or multiple vehicles, depending on the final application. Attending on geographic routing, GeoNetworking provides to functions: geographic addressing and geographic forwarding.

Each time a station receives a packet, the station evaluates the destination address. If this node is not the destination, it forwards the message according to its knowledge of the network topology. For the forwarding process, a location service is used to identify the GeoAdhoc routers and the location of a destination node. Finally, it also supports routing messages to a geographic area. It is useful if it is required that every node receives the same message, for example in safety conditions.

All these concepts are defined in GeoNetworking as GeoUnicast, GeoBroadCast and topologically scoped broadcast. All of them are defined in the standard referenced at the beginning of this subsection.

3.10.6 Service Announcement Message (SAM)

During the previous sections, it has been explained the respective protocols of Networking and Transport layer and other two protocols of Facilities layer. However, there is a multitude of protocols in this layer.

As it is shown in Figure 26, the Facilities layer can be implemented with a wide range of protocols. These protocols have a well-defined purpose. The goal is implement one protocol per service, in such a way that, combining all these functionalities we have a polyvalent and flexible ITS network. Furthermore, this implementation approach enables a scalable platform for future required services. In this section, it will be explained the protocol responsible of the multi switching channel at Facilities layer.

Unfortunately, at the time of writing this chapter, some of the standards that are going to be explained were still being finalized. We provide here the current trends, but some details might evolve in the future. Also, very few studies could be found, mainly this section is based on the book "Vehicular ad hoc Networks: Standards, Solutions, and Research" [55] and the research carried out by De Martini Laura [56].

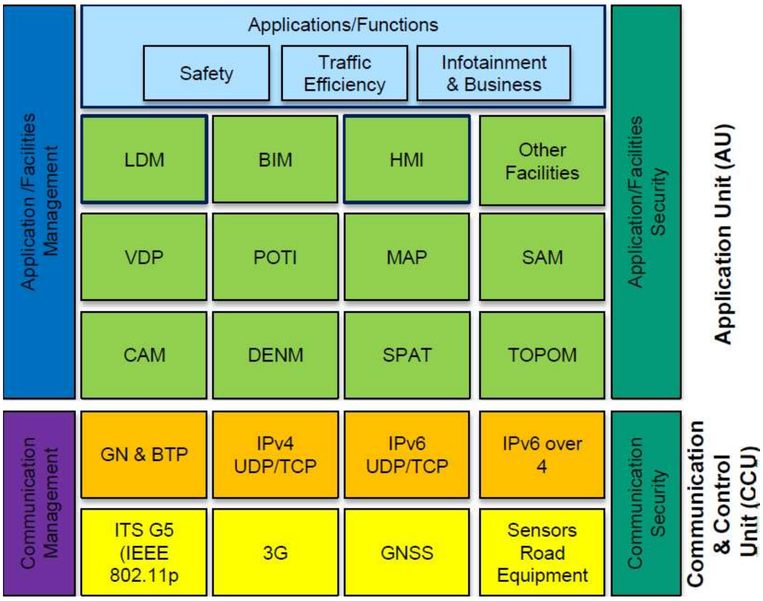


Figure 26. ETSI ITS G5 stack - Example protocols [57]

Among all the protocols that can be implemented in the Facilities layer, we have to spotlight the Service Announcement Messages (SAM). These messages are controlled by the Service Advertisement which is a management functionally providing a mechanism to announce availability of an ITS service in a single-hop communication link. These advertisements are not application messages themselves, though they may contain information allowing the user application to decide whether to connect. This concept is based on the WAVE Service Announcements (WSAs) of the American standard which is described in IEEE 1609.3 [58]. Meanwhile, for European Standard it will be published in TS 102 890-2 standard because at the time of writing this section, it is not available for downloading (to check the current status: Status of DTS/ITS-0020044 [9]).

The Service Announcement Message must only be sent through a Service Channel. This service cannot use the Control Channel. As it is defined in TS 102 724 [47], the Control Channel is the reference channel of all ITS stations operating in safety-related context, where CAM, DENM, TOPO (Road Topology) and MAP (Map) are transmitted. Other types of messages may be sent on Service Channels which should be announced using SAM.

This idea is represented in the Figure 27, where it is shown the different between the Transceiver A and B. The Transceiver A is working in Control Channel. Depending on the channel load, it could be possible to change the allocation of a service to a Service Channel because the CCH has exceeded the limit of channel load. It can continue sending the data in Service Channel 1 or 3, depending on the Transceiver Configuration (Section 3.9). In order to know, if a transceiver must commute the channel, it is implemented a mechanism in which every station knows when it have to hop the channel by himself. It will not be advertised to the other stations as it happens in the rest of situations. This mechanism is described with a flow chart included in Appendix figure 14.

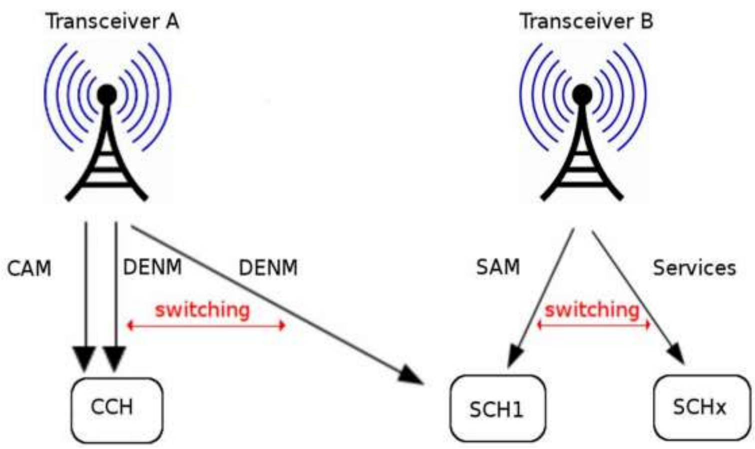


Figure 27. ETSI ITS G5 Channel and Message Specification [56]

On the other hand, the rest of protocols can be only used in the Service channel and it is possible to switch the channel under crowded conditions provided that a previous announcement is made by SAM. In these situations, we can identify two kinds of stations, the Service Provider (SP), which is the station responsible for sending the SAM on SCH1 and offer services on SCHx and the Service Consumer (SC), which after the reception on SCH1 of the announcement switches to SCHx in order to get the related services. Any station can play both roles.

This explanation was extracted from the following standards:

- ETSI TS 102 894-1 V1.1.1 clause 6.2.5 [50].
- ETSI TS 102 636-4-2 V1.1.1 Annex A [59].
- ETSI TS 102 940 V1.1.1 clause 4.1.1.5 [60].
- Status of ITS Communication Standards HTG3-1 2012-11-12 clause ME-01 [61].

Once more time, we have to emphasise that multi-channel operation implementation is still in progress.

In fact, in the first case of Figure 27, the mode operation can be disabled including the 1-bit field Channel Offloading Bit in the common header of GeoNetworking (page 208 [55]). In the case that a station has not the Decentralized Congestion Control (DCC) implemented, it will not be able to calculate the channel load. As a result, it should disable the multi-channel operation using this bit field.

About the second case of Figure 27, we suggest interested readers to read the chapters 7.4.2.1, 7.4.2.2 [55] and the Annex A of TS 102 636-4-2 [59]. Because, in these documents a first approach to how could be SAM protocol (TS 102 890-2) is given.

4 User manual

In this section, it is explained how to run WLANp in a Linux device, as well as how we can make our first tests in order to know if the installation was successful. It is given an example of an easy communication between stations.

Besides, it is explained how the different messages that can be sent in ITS environment can be decoded.

4.1 First steps

Once the installation of all user programs was successful, it can be checked with a wireless card if OCB mode is supported and consequently if WLANp at physical and MAC layers are supported as well.

First of all, it is necessary to check if the WLANp channel is available for the user. We must be certain that we have selected the right wireless regulatory domain by executing the command “iw reg get”. Afterwards, we can list the physical properties of our wireless card “iw list”, and we have to see something similar to Figure 28. Depending on what wireless card is being used and the domain selected, we will have enabled some channels or others.

```
* 5850 MHz [170] (17.0 dBm) (no IR)
* 5855 MHz [171] (17.0 dBm) (no IR)
* 5860 MHz [172] (17.0 dBm) (no IR)
* 5865 MHz [173] (17.0 dBm) (no IR)
* 5870 MHz [174] (17.0 dBm) (no IR)
* 5875 MHz [175] (17.0 dBm) (no IR)
* 5880 MHz [176] (17.0 dBm) (no IR)
* 5885 MHz [177] (17.0 dBm) (no IR)
* 5890 MHz [178] (17.0 dBm) (no IR)
* 5895 MHz [179] (17.0 dBm) (no IR)
* 5900 MHz [180] (17.0 dBm) (no IR)
* 5905 MHz [181] (17.0 dBm) (no IR)
* 5910 MHz [182] (17.0 dBm) (no IR)
* 5915 MHz [183] (17.0 dBm) (no IR)
* 5920 MHz [184] (17.0 dBm) (no IR)
* 5925 MHz [185] (17.0 dBm) (no IR)
```

Figure 28. Available channels (Country AA)

In the case that everything is correct and the WLANp channels are available as it is shown in Figure 28, the next step is to check if it is possible to create an OCB interface and to join to one of the channels. An example of how this could be done is given in Appendix figure 12. If the wireless cards support the OCB mode and the OCB interface is able to join to some WLANp channel, it will mean that everything is ready to work at the application layer. Thus, every change related with the two first layers of OSI stack was successful and now we have a system that supports WLANp. Therefore, now it can be plausible to implement or to use the upper protocols. The easiest one is IP, because it is only required to give an IP address and a default gateway to use it.

An example is given in Appendix figure 13 of how to make an easy connectivity test. If it is transmitted any data through one interface of the station and the receiver is listening in the same channel, this data will be capture using tcpdump program in the selected interface. In this way, the OCB interfaces could be tested.

It is important to point, that these modifications can be used for channels defined for US standard. The only change that must be done, is to select the right regulatory domain ("reg set AA"). The database developed for this project (Figure 18) enables to work in these channels. However, if it use the database provided by the original project [24], it will have problems with the channels 170 and 171 because they are not defined in the regulatory domain definition.

4.2 Suitable Wi-Fi wireless cards

Once it has been explained the steps that we have to follow in order to implement the 802.11p standard we can check if our wireless cards are compatible or not with 802.11p standard. During this project it was made a complete study of potential compatible wireless cards analysing a higher number of devices of different brands (Atheros, Realtek, Ralink, Intel, Broadcom, etc.). One of the mainly difficulties was find a potential wireless card that we can be bought in the European Union, because most of them are sold in the United States and China. The characteristics of more than 200 cards were analysed, taking into account the requirements defined in the section 3.1, and as we advanced in that section, only a few of these cards meet the technical specifications. Finally, we check if they are able to work in OCB mode and which channels they support. The results are shown in the Table 4.

Wireless Card	Type	Work in OCB mode	ITS channels available
Atheros AR9280 Dell	miniPCI-e	Yes	Yes
Atheros AR9280 Ubiquiti	miniPCI-e	Yes	No
Atheros AR9382 Sparklan	miniPCI-e	Yes	Yes
Qualcomm Atheros AR9462 Sparklan	miniPCI-e	Yes	Yes
Atheros AR7010 AR9280 Netgear	USB	No	Yes

Table 4. Compatible Wi-Fi wireless card with 802.11p

As we can see in the Table 4, there are wireless cards that cannot support the 802.11p standard. In the case of Atheros AR9280 manufactured by Ubiquity, it has the same wireless chip as the Atheros AR9280 manufactured by Dell. However, it is not available the new channels defined for ITS environments.

In the case of the USB wireless card, it is available the new channels. However, it cannot support the OCB mode, because it uses a different driver, the ATH9K_htc which has not been modified to support this working mode.

4.3 [redacted] – Road Side Unit

As it was introduced during the first chapter, during this project it has been used a real Road Side Unit (RSU [redacted]) for some of our test. This RSU is able to simulate a whole ITS environment, it means that it is able to create virtual stations (RSUs and OBUs) and to stablish communications among devices. The usefulness of this ITS device is that this instrument can simulate any kind of ITS unit at the same time, also it supports several ITS protocols and it implements the 802.11p standard at access layer.

This device is used by the [redacted] consortium in order to be placed in its facilities. In this way, other companies that want to test their own technology can do it in a pseudo real ITS environment. Because, as it was said, this instrument is able to simulate, intelligent traffic lights, any kind of motorised transport, as well as it can be configured in any kind of situations such as defining the traffic direction, the conditions of the road and how many ITS devices we want. However, every packet it is sent using the same wireless card (it is shared the transmission time), it means that all the packets are sent from the same place, therefore it cannot be tested the high mobility of the ITS network.

In the next section this device is configured and used in order to perform a test. The next protocols are used:

- Basic Transport Protocol.
- Decentralized Environmental Notification Message.
- Infrastructure to Vehicle Information.
- GeoNetworking.

As the device can simulate several ITS devices at the same time, it is possible to configure which protocol is used by each ITS device, defining the frequency of the packets as well as the properties of the road and the vehicles.

4.4 Wireshark and ITS plugin

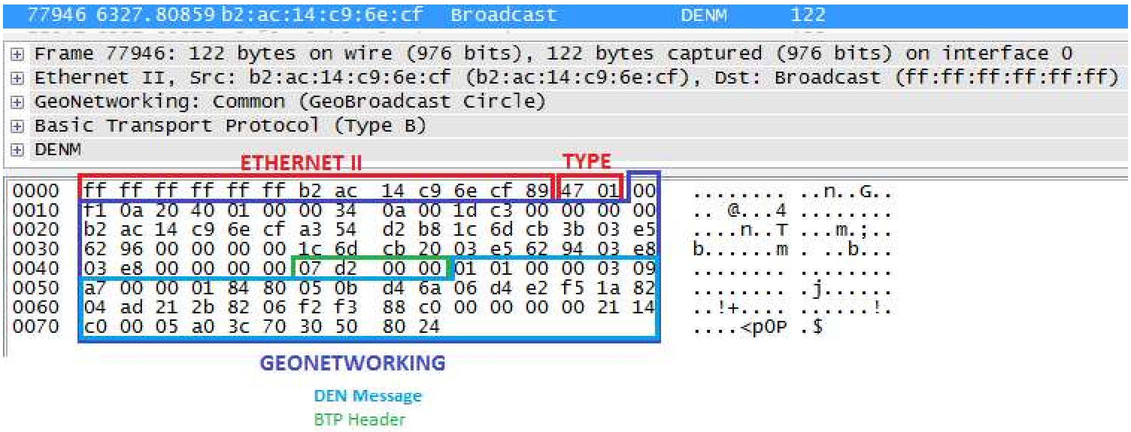
During the project we focused on the Access layer and we managed to capture all the exchanged traffic in ITS environments. One of the most popular free tools for networks analysis is Wireshark software. In addition, ETSI has published an ITS plugin for Wireshark [62]. In such a way that we are able to dissect and decode every packet to analyse the information that it contains.

An example of how the messages can be decoded is given below. We took as samples the messages sent by the Road Side Unit developed by [redacted] (Section 1).

4.4.1 Decoding a Decentralized Environmental Notification Message

In order to understand how is built a DEN Message, it can be useful to review Figure 24. As we can see in this figure, a DENM belongs to Facilities layer. Afterwards it is added the BTP packet in Transport and Network Layer and the resulting packet is encapsulated in a GeoNetworking packet.

At last, depending on the technology of the Access layer, it is added the destination and the source MAC address (in the case of WLANp). In the Figure 29 we can see a real DEN Message captured using Wireshark as well as it can be seen the structure explained before.



```
77946 6327.80859 b2:ac:14:c9:6e:cf   Broadcast            DENM      122
⊞ Frame 77946: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0
⊞ Ethernet II, Src: b2:ac:14:c9:6e:cf (b2:ac:14:c9:6e:cf), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ GeoNetworking: Common (GeoBroadcast Circle)
⊞ Basic Transport Protocol (Type B)
⊞ DENM
0000  ff ff ff ff ff ff b2 ac  14 c9 6e cf 89 47 01 00   ........ ..n..G..
0010  f1 0a 20 40 01 00 00 34  0a 00 1d c3 00 00 00 00   .. @...4 ........
0020  b2 ac 14 c9 6e cf a3 54  d2 b8 1c 6d cb 3b 03 e5   ....n..T ...m.;..
0030  62 96 00 00 00 00 1c 6d  cb 20 03 e5 62 94 03 e8   b......m . ..b...
0040  03 e8 00 00 00 00 07 d2  00 00 01 01 00 00 03 09   ........ ........
0050  a7 00 00 01 84 80 05 0b  d4 6a 06 d4 e2 f5 1a 82   ........ .j......
0060  04 ad 21 2b 82 06 f2 f3  88 c0 00 00 00 00 21 14   ..!+.... ......!.
0070  c0 00 05 a0 3c 70 30 50  80 24                     ....<pOP .$
```

Figure 29. DEN Message

As we saw during the previous chapter, the Basic Transport Protocol is a protocol that provides an end-to-end connectionless transport service. As it is defined in the standard EN 102 636-5-1 [53], it consists of 4-byte protocol header. There are two kinds of headers, BTP-A and BTP-B.

- BTP-A: This header carries the source and the destination port (Figure 30). The destination port identifies the protocol entity at Facility layer in the destination station. On the other hand, the source port identifies the port that the ITS facilities layer protocol entity in the source has used to send this packet, as well as in case of a reply, this is the port that should be addressed in the absence of the other information.
- BPT-B: This header carries the destination port but no the source port (Figure 31). It has the same function than the same field in the BTP-A. However, it should be filled with the well-known values defined in annex B, table B1 of EN 102 636-5-1 [53]. In the case that one of these predetermined ports is being used, this header provides additional info in Destination port field.

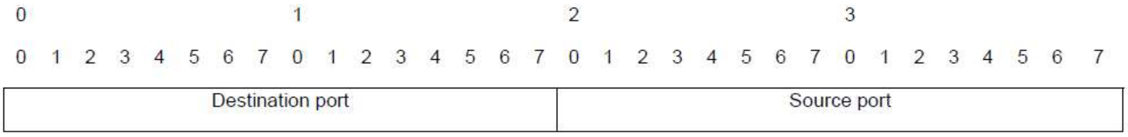


Figure 30. BTP-A Header format

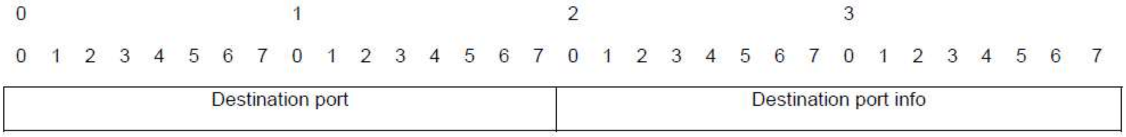


Figure 31. BTP-B Header format

The goal of BTP is to multiplex the messages such as CAM and DENM in the ITS Facilities layer. In this case, we can see in the Figure 29 that the BTP frame is: 07 D2 00 00. Therefore, it is a BTP-B header, because it no contains source information. Moreover, if we convert the destination port 07 D2 into a decimal value, it results 2002. If we check this value in the table B1 of the EN 102 636-5-1 [53] standard, we will see that this code corresponds to a DENM ITS Facility layer.

Following with the dissection, the rest of the bytes belong to the DEN Message. The general structure of a DENM is illustrated in Figure 32. And we are going to compare this structure with the packet that we have captured.

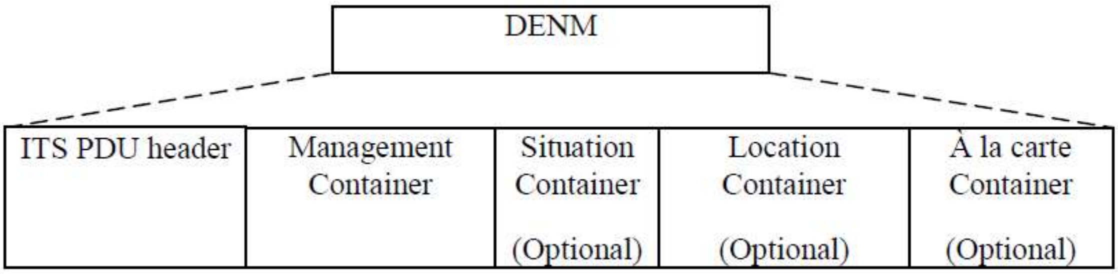

DENM
ITS PDU header
Management Container
Situation Container (Optional)
Location Container (Optional)
À la carte Container (Optional)

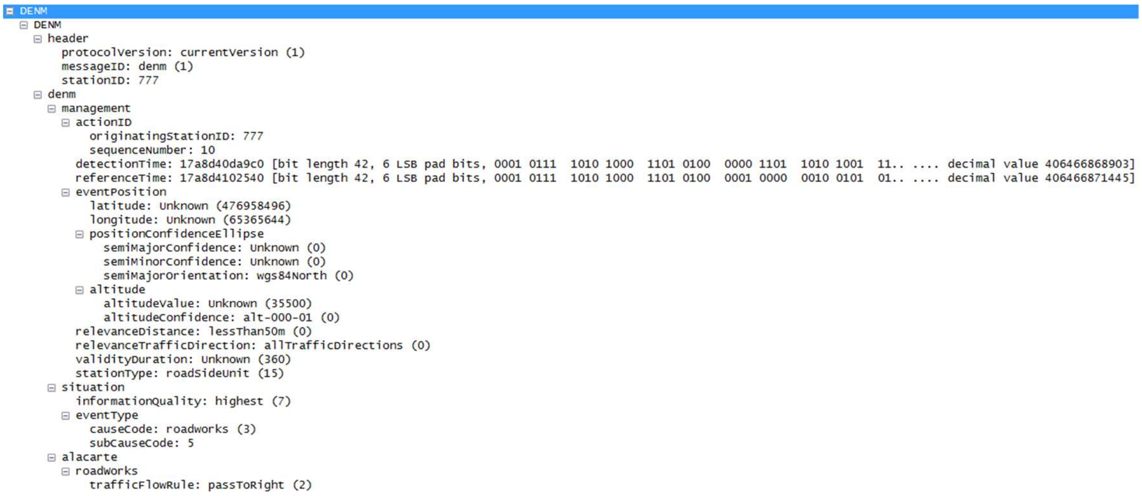
Figure 32. General structure of a DENM

The DEN Message is divided into five main fields, the header, and the rest of the fields containing the real information that is wanted to be sent. Into the header is sent the protocol version that is being using, the ID of the message and the source station. Afterwards it follows:

- The management container: It contains information related to the DENM management and the DENM protocol.
- The situation container: It contains information related to the type of the detected event.
- The location container: It contains information of the event location, and the location referencing.
- The *à la carte* container: It contains information specific to the case of use which requires the transmission of additional information that is not included in the three previous containers.

All the subfields of DENM are defined in the chapter 7 of EN 302 637-3 standard [51], as well as the dictionaries to decode the information, as it can be seen in the table 10 of the same document.

Using the Wireshark plugin for ITS communications, we can decode the DEN Message in easy way, as it is illustrated in Figure 33. And as it is shown, it matches with the description given before.



```
⊟ DENM
  ⊟ DENM
    ⊟ header
        protocolVersion: currentVersion (1)
        messageID: denm (1)
        stationID: 777
    ⊟ denm
      ⊟ management
        ⊟ actionID
            originatingStationID: 777
            sequenceNumber: 10
          detectionTime: 17a8d40da9c0 [bit length 42, 6 LSB pad bits, 0001 0111  1010 1000  1101 0100  0000 1101  1010 1001  11.. .... decimal value 406466868903]
          referenceTime: 17a8d4102540 [bit length 42, 6 LSB pad bits, 0001 0111  1010 1000  1101 0100  0001 0000  0010 0101  01.. .... decimal value 406466871445]
        ⊟ eventPosition
            latitude: Unknown (476958496)
            longitude: Unknown (65365644)
          ⊟ positionConfidenceEllipse
              semiMajorConfidence: Unknown (0)
              semiMinorConfidence: Unknown (0)
              semiMajorOrientation: wgs84North (0)
          ⊟ altitude
              altitudeValue: Unknown (35500)
              altitudeConfidence: alt-000-01 (0)
          relevanceDistance: lessThan50m (0)
          relevanceTrafficDirection: allTrafficDirections (0)
          validityDuration: Unknown (360)
          stationType: roadSideUnit (15)
      ⊟ situation
          informationQuality: highest (7)
        ⊟ eventType
            causeCode: roadworks (3)
            subCauseCode: 5
      ⊟ alacarte
        ⊟ roadworks
            trafficFlowRule: passToRight (2)
```

Figure 33. DENM captured

4.4.2 Infrastructure to Vehicle Information (IVI) Messages

The IVI messages support mandatory and advisory road signage such as contextual speeds and roadworks warnings. IVIM either provides information of physical road signs such as static or variable road signs, virtual signs or roadworks.

The IVIM is defined in CEN ISO/TS 19321 [63] standard. Its structure is defined in the Figure 34. It was defined as an extensible data structure which can be split into several containers. Also this message is encapsulated in a message with the ITS Common Header, as it was explained for DEN Messages.

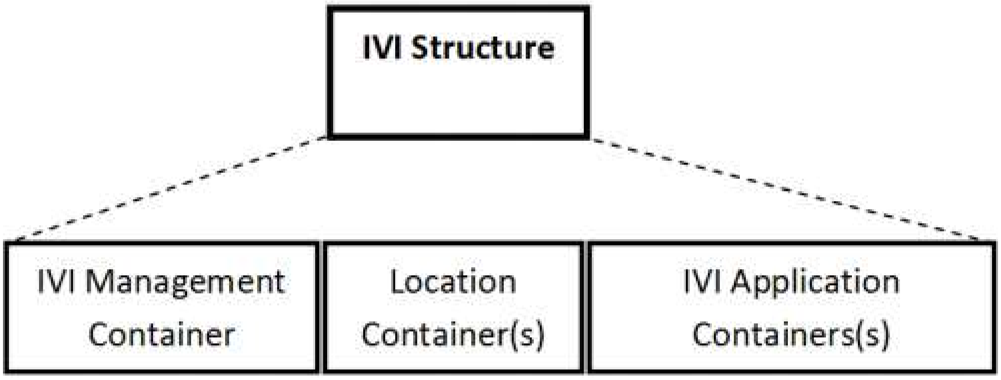
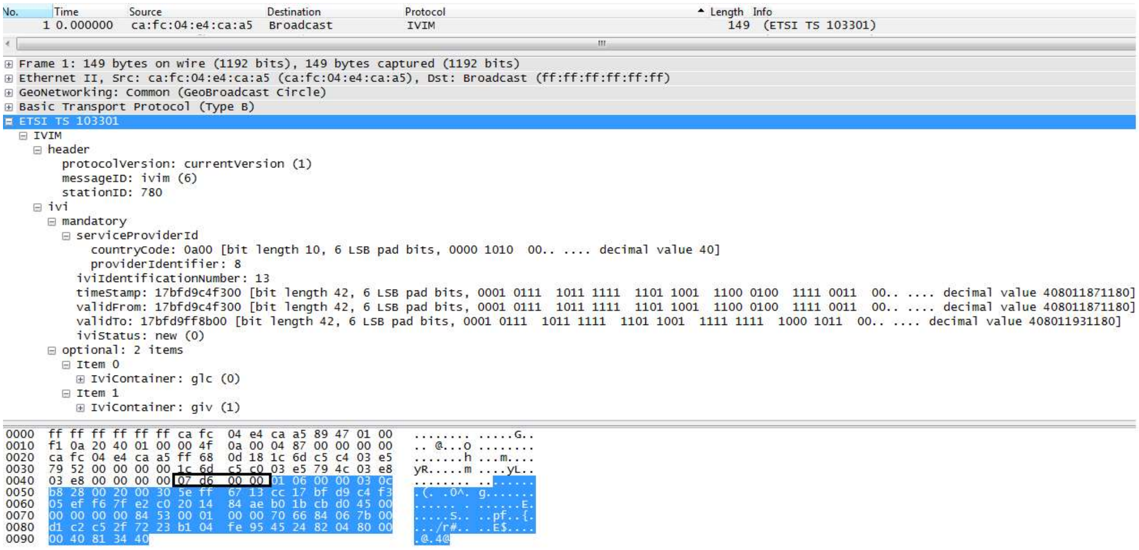


Figure 34. IVI Message structure

The structure of the previous figure is reflected in the Figure 35, which is a captured packet from the Road Side Unit. The BTP Header (07 D6 00 00) indicates that is an IVI Message which the 2006 code. Thereupon, we can find the IVI Message where it can be find the next fields:

- IVI Management Container (Header): It contains the protocol version and the ID of the message and the station.
- Location Container: It is a compulsory field which includes the metadata of the IVI Message, such as the timestamp and what is his expiration time for this packet.
- Finally, it is included the containers. These fields embrace the road signage information. This information is represented by codes defined in ISO/TS

14823:2008 [64]. By this way the stations can decode the messages and interpret the ITS situation.



```
No.   Time        Source             Destination    Protocol                              Length  Info
    1 0.000000    ca:fc:04:e4:ca:a5  Broadcast      IVIM                                     149  (ETSI TS 103301)

⊞ Frame 1: 149 bytes on wire (1192 bits), 149 bytes captured (1192 bits)
⊞ Ethernet II, Src: ca:fc:04:e4:ca:a5 (ca:fc:04:e4:ca:a5), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ GeoNetworking: Common (GeoBroadcast Circle)
⊞ Basic Transport Protocol (Type B)
⊟ ETSI TS 103301
  ⊟ IVIM
    ⊟ header
        protocolVersion: currentVersion (1)
        messageID: ivim (6)
        stationID: 780
    ⊟ ivi
      ⊟ mandatory
        ⊟ serviceProviderId
            countryCode: 0a00 [bit length 10, 6 LSB pad bits, 0000 1010  00.. .... decimal value 40]
            providerIdentifier: 8
          iviIdentificationNumber: 13
          timeStamp: 17bfd9c4f300 [bit length 42, 6 LSB pad bits, 0001 0111  1011 1111  1101 1001  1100 0100  1111 0011  00.. .... decimal value 408011871180]
          validFrom: 17bfd9c4f300 [bit length 42, 6 LSB pad bits, 0001 0111  1011 1111  1101 1001  1100 0100  1111 0011  00.. .... decimal value 408011871180]
          validTo: 17bfd9ff8b00 [bit length 42, 6 LSB pad bits, 0001 0111  1011 1111  1101 1001  1111 1111  1000 1011  00.. .... decimal value 408011931180]
          iviStatus: new (0)
      ⊟ optional: 2 items
        ⊟ Item 0
          ⊞ IviContainer: glc (0)
        ⊟ Item 1
          ⊞ IviContainer: giv (1)

0000  ff ff ff ff ff ff ca fc  04 e4 ca a5 89 47 01 00   ........ .....G..
0010  f1 0a 20 40 01 00 00 4f  0a 00 04 87 00 00 00 00   .. @...O ........
0020  ca fc 04 e4 ca a5 ff 68  0d 18 1c 6d c5 c4 03 e5   .......h ...m....
0030  79 52 00 00 00 00 1c 6d  c5 c0 03 e5 79 4c 03 e8   yR.....m ....yL..
0040  03 e8 00 00 00 00 07 d6  00 00 01 06 00 00 03 0c   ........ ........
0050  b8 28 00 20 00 30 5e ff  67 13 cc 17 bf d9 c4 f3   .(. .0^. g.......
0060  05 ef f6 7f e2 c0 20 14  84 ae b0 1b cb d0 45 00   ...... . ......E.
0070  00 00 00 00 84 53 00 01  00 00 70 66 84 06 7b 00   .....S.. ..pf..{.
0080  d1 c2 c5 2f 72 23 b1 04  fe 95 45 24 82 04 80 00   .../r#.. ..E$....
0090  00 40 81 34 40                                     .@.4@
```

Figure 35. IVI Message

4.4.3 Decoding GeoNetworking messages

The other kinds of packets that have been captured from the RSU are the GeoNetworking beacons. This protocol defines different kind of headers message, depending on the final application purpose. Every kind is defined in the EN 302 636-4-1 [65] standard:

- GUC: Geographically-Scoped Unicast packet header.
- TSB: Topologically Scoped Broadcast packet header.
- SHB: Single Hop Broadcast packet header.
- GBC and GAC: Geographically-Scoped Broadcast and Geographically-Scoped Anycast packet headers.
- BEACON packet header.
- LS Request and LS Reply: Location Service packet headers.

The packets captured during the simulation were of the type Beacon packet header. These packets are used for beaconing, which is a technique used to advertise periodically a GeoAdhoc router's with a position vector to ITS neighbours. A Beacon packet shall be sent periodically unless the GeoAdhoc router sends another GeoNetworking packet that carries the GeoAdhoc router's Local Position Vector (LPV). When a GeoAdhoc starts up, it shall send an initial beacon to announce its presence to other GeoAdhoc routers.

The Beacon packet consists of a Basic Header, a Common Header, and a Logic Position Vector as shown in Figure 36.

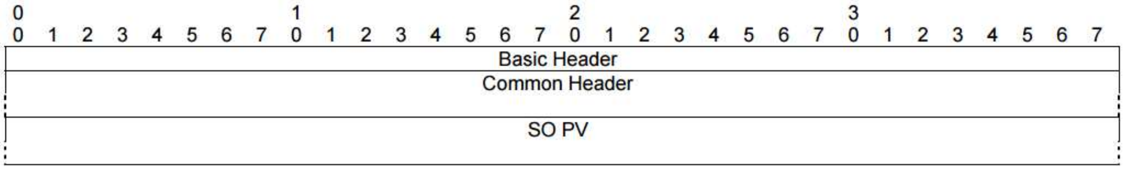
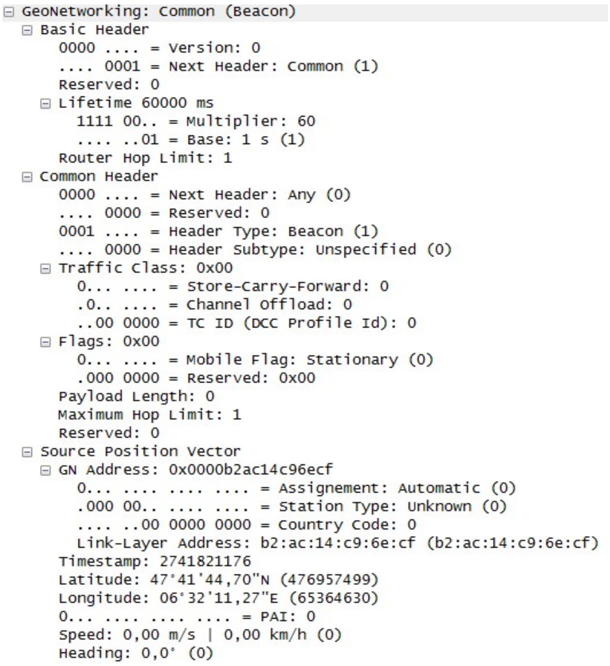


Figure 36. GeoNetworking Beacon packet.

- Basic header: The size is 4 bytes and its main target is to identify the GeoNetworking that is been using.
- Common header: The size is 8 bytes and it carries metadata information about the following data, as well as, it carries information about status of the ITS environment, like the traffic class.
- SO PV: It is the source position vector and it carries geographical information, such as longitude, latitude of and speed of the ITS station which send the GeoNetworking packet.

Using the Wireshark ITS plugin, we check if our captured packets match with this structure. As it is illustrated in the Figure 37, we can see the 3 main fields which are defined in this packet.



```
⊟ GeoNetworking: Common (Beacon)
  ⊟ Basic Header
      0000 .... = Version: 0
      .... 0001 = Next Header: Common (1)
      Reserved: 0
    ⊟ Lifetime 60000 ms
        1111 00.. = Multiplier: 60
        .... ..01 = Base: 1 s (1)
      Router Hop Limit: 1
  ⊟ Common Header
      0000 .... = Next Header: Any (0)
      .... 0000 = Reserved: 0
      0001 .... = Header Type: Beacon (1)
      .... 0000 = Header Subtype: Unspecified (0)
    ⊟ Traffic Class: 0x00
        0... .... = Store-Carry-Forward: 0
        .0.. .... = Channel Offload: 0
        ..00 0000 = TC ID (DCC Profile Id): 0
    ⊟ Flags: 0x00
        0... .... = Mobile Flag: Stationary (0)
        .000 0000 = Reserved: 0x00
      Payload Length: 0
      Maximum Hop Limit: 1
      Reserved: 0
  ⊟ Source Position Vector
    ⊟ GN Address: 0x0000b2ac14c96ecf
        0... .... .... .... = Assignement: Automatic (0)
        .000 00.. .... .... = Station Type: Unknown (0)
        .... ..00 0000 0000 = Country Code: 0
        Link-Layer Address: b2:ac:14:c9:6e:cf (b2:ac:14:c9:6e:cf)
      Timestamp: 2741821176
      Latitude: 47°41'44,70"N (476957499)
      Longitude: 06°32'11,27"E (65364630)
      0... .... .... .... = PAI: 0
      Speed: 0,00 m/s | 0,00 km/h (0)
      Heading: 0,0° (0)
```

Figure 37. GeoNetworking Beacon captured

5 Performance evaluation

As it was explained during the section 3.7, the ATH9K driver was modified in order to support the OCB mode. Therefore, some of the tests aim at checking the OCB behaviours using the Atheros wireless cards. It means, checking if each and every one of the changes were successful and comparing the results with the 802.11p specifications explained during the chapter 0. Moreover, it is wanted to analyse if these modifications are enough in order to work in a real ITS environment.

This part of the project is necessary to know the basis of ITS working mode as well as it helps us understand the WLANp behaviours and spot possible malfunctioning or physical limitations. In this chapter the reasons to perform the different tests and the results of them are presented. Some conclusions leading to the appropriateness of the devices chosen in this project to implement ITS devices are drawn from the obtained results. The rest of the chapter is organized as follows, in the first section, performed measurements, we discuss general WLANp behaviour and present the questions are intended to solve. Besides, we explain the possible limitations and features that should be checked. In the next section covers the software tools or programs that have been used during every test and where they can be found. Finally, we present several scenarios, the importance of every one and the results in them.

5.1 Performed measurements

During the modification of the Wi-Fi card driver, and comparing the working modes IBSS and OCB, some doubts arise. IBSS allows a negotiation between the receptor and the transmitter so that can be an explicit adaptation to the transmission conditions by modifying the parameters of the communications such as the bitrate, the modulation format, etc. WLANp also defines a specific bit rate for every channel spacing (Figure 5), but as the communication is a broadcast one, there were doubts about how the adaptation to the different transmission conditions was made. In fact, we did not found any serious study relating the changes in the driver with a test in a real ITS environment. This way, the main problems to solve were:

1. Check if the wireless card supports every working OCB mode, including the different bit rates and modulations (BPSK, QPSK and 16-QAM). As we did not have a constellation analyser, we only check the bitrates in different environments, so that observing adaptation of the bit rate we could suppose that all the necessary changes were taking place. For this reason, the performance of the device was analysed in non-ideal situations.
2. Another important parameter is the change of the power transmission. The power transmission is mainly controlled by the DCC mechanism. As we have not implemented this mechanism in our device, the test was made to infer indirectly the change of the power transmission when we force it in the driver.
3. It was explained during the examples showed in the previous chapters (Section 3.10.6), that in a real environment, the RSU can transmit packets using five different channels. Thus, it is logic to verify if the wireless cards are able to work in all these channels and if they have the same transmission and reception qualities.

4. Finally, it is crucial to check if the system is faster enough to switch between channels. ITS real environments can be working in several channels at the same time. It could be interesting to listen to several channels alternatively, but one question arises, when a card switch between two channels are it going to lose much information? This is crucial when form example a RSU warns that the next packet is going to be transmitted using one of the service channels. It this message is only transmitted once, then when the OBU switch to listen to the new service channel, then it should be able to get directly this message.

5.2 Software tools

In this section, it is explained the software used to understand the particular aspects of WLANp taking into account the limitations in test devices we had. Particularly, a complete test of our system would have supposed to have had a constellation analyser, a spectrum analyser and other test devices. However, facing the fact that we had only one ITS device along with our Wi-Fi cards, several software tests were design so that indirect conclusions could be drawn. For example, in order to measure the throughput traffic, it was sent a specific amount of bytes.

There are plenty of network monitoring and analysis programs. We can structure the software tools differentiating the software that is launched in the receiver and sender. The main target of receiver stations is data logging all the packets that they receive. Meanwhile, the sender station sends a data stream. We want to capture the traffic at second layer (MAC sublayer Figure 7), therefore it is possible to use user programs as "tcpdump", "Wireshark" or even a handmade program using "pcap" libraries. On the other side, since all our conclusions are drawn directly from the observed bit rate, we want to send our data without interference of upper protocols. Therefore, it was used a socket to send the data stream directly to physical layer. In this manner, by including only the PHY headers we have more control about the communication delays.

5.3 Scenarios

Every scenario is composed of two kinds of V2X stations, during the test we are going to refer as "Road Side Unit" the device which will be the responsible for sending the data and as "On Board Unit" the device which will be the receiver station. It is worth noting that we are not using the RSU of [redacted] (section 4.3), but we are using this notation to define the test environment. The reason why both are defined as sender and receiver is because the OBU never answers the RSU and vice versa.

This scenario fits particularly well with our purpose because, at the moment, our main target in [redacted] devices is to capture every V2X communications and to understand the first two layers of the stack, PHY and MAC.

The scenarios have been designed taking into account the wireless card physical limitations, because these components were not designed to support WLANp, and this way some doubts arise about if they are able to fulfil all the requirements of the standard. Thus, the scenarios that were simulated were:

- Analysis of channel spacing versus supported data rate.
- Analysis of reliable working in adverse conditions.

- Analysis of the variation of the supported transmission power.
- Analysis of operability of every ITS channel.
- Analysis of channel switching limitations.

Moreover, it has been considered appropriate to show a summary of the results of other researches and the technical specifications of Cooperative ITS devices. It was not found many researches about the throughput, power transmission and delay in switching channel for 802.11p, but we found various documents that were useful to compare.

One study that has been found is “Throughput and Delay Limits of 802.11p and Its Influence on Highway Capacity” [66]. The results that they achieved can be seen in the Figure 38 which shows the relationship between payload and throughput. For our purposes of comparison, we are going to take as a reference the maximum payload because it is the most similar case to ours. These results will be used for the sections 5.3.1, 5.3.2 and 5.3.4.

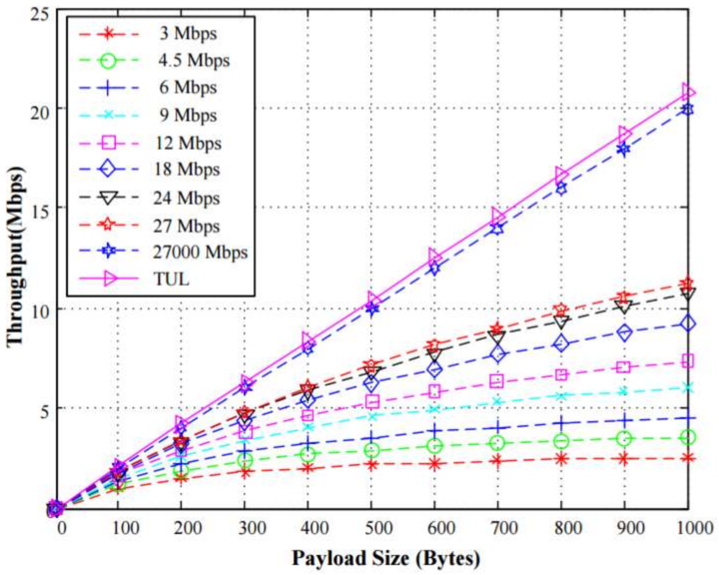
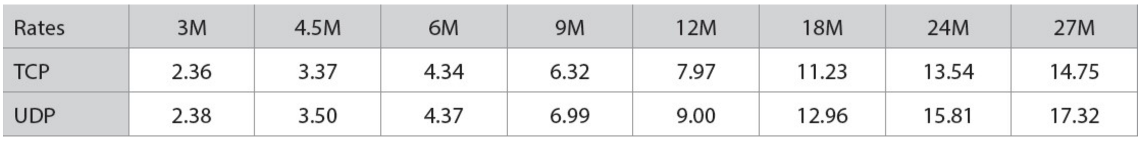


Figure 38. Maximum throughputs and throughput upper limit of 802.11p [66]

Other source of information is the provided datasheets of the commercial products designed for Cooperative ITS. This is the case of Arada-Systems. One of the targets of this company is to develop, license and provide solutions for the next generation of Wi-Fi usage including Automotive. They have a partnership with Qualcomm-Atheros, therefore it seems highly likely that their devices use Qualcomm-Atheros Wi-Fi wireless chipsets the same as in our case.

Currently, they develop a wide range of ITS products, including RSUs and OBUs. We are going to take as a reference one of the datasheets of them. It is called “LocoMate Classic On Board Unit” [67]. Of special interest for our case is the throughput traffic specifications on one of the ITS channels, which is shown in Figure 39.



Rates	3M	4.5M	6M	9M	12M	18M	24M	27M
TCP	2.36	3.37	4.34	6.32	7.97	11.23	13.54	14.75
UDP	2.38	3.50	4.37	6.99	9.00	12.96	15.81	17.32

Figure 39. Throughput Traffic Test Results Half-Rates on Channel 172 (Mbps) without Channel Switch [67].

TCP and UDP are the common protocols used to measure the throughput traffic in this kind of transmission systems. We can take these results as a reference, but never as tight results for us because they are using different upper protocols that several delays that we cannot quantify. These results will be used for the sections 5.3.1, 5.3.2 and 5.3.4.

5.3.1 Comparison between real and theoretical data rates depending on the channel spacing

The standard EN 302 663 [33] defines the physical channel allocation. By allocation must be understood not only the frequency where the channel is located, but also the bitrate, transmission power limit, transmission power density limit and channel spacing. Taking into account those specifications, it is advisable to check if our wireless cards support those bitrates.

Moreover, as it is defined in clause 18 of IEEE 802.11-2012 [32], depending on the channel spacing, a different bitrate for each modulation is used. In order to see the difference between 20MHz, 10MHz and 5MHz it was made a bitrate test for each speed that is used for WLANp.

The aim of this test is analyse if the driver modifications for WLANp [34] follow the standard recommendations, as we have no equipment to check directly if the physical modulation (BPSK, QPSK or QAM) and bitrate is the right one. Thus, for the one modulation we should obtain different data rates depending on the channel spacing, as it is illustrated in Figure 5.

Every test was run enough times to obtain reliable measurements. It is illustrated, through the confidence interval. Every communication used the CCH to send the data, because it is the most use channel in ITS environments.

Test conditions:

• Type of software:	RAW_ETHERNET Socket
• PSDU size (Figure 8):	1514 Bytes
• Distance between stations:	~40 centimetres
• Transmission power:	17dBm
• Frequency:	5900 MHz – Control Channel
• Confidence margin:	99.975

Previous operations

In order to obtain accurate measurements, we should take into account all the bits transmitted. As it was shown in Figure 8, it is added more headers in physical layer. And depending on the selected bitrate, it will send more or less bits with the same PSDU. The length of the final frame is calculated for each modulation, BPSK, QPSK and 16-QAM. An example of how to do this is given in 802.11-2012 standard section “18.3.2 PLCP frame format” [32].

The PSDU size is 1514 Bytes including headers of upper layers. The user data field was filled with 1496Bytes. The headers are the source and destination MAC address and it is also included a 4 Bytes CRC field.

BPSK frame

Data bits per OFDM symbol: 24 bits

Coded bits per OFDM symbol: 48bits

Coding rate: 1/2

PREAMBLE = 32μs = 96bits

SIGNAL = 4+1+12+1+6 = 24bits

DATA = 16 + PSDU*8 + 6 + PAD-BITS

16 + 1514*8 + 6 = 12134; 12134/48 = 252.79 ≈ 253.

253*48 = 12144; PAD-BITS = 12144-12134 = 10bits

FINAL-FRAME = 12144 + 24 + 96 = 12264 bits

PSDU user data = 11968bits

The efficiency of BPSK is 97.58%

QPSK frame

The preamble and the signal field are always sent using BPSK modulation. Therefore, we have to differentiate between the bits sent using BPSK and QPSK.

Data bits per OFDM symbol: 48bits

Coded bits per OFDM symbol: 96bits

Coding rate: 1/2

PREAMBLE = 32μs = 96bits

SIGNAL = 4+1+12+1+6 = 24bits

DATA = 16 + PSDU*8 + 6 + PAD-BITS

16 + 1514*8 + 6 = 12134; 12134/96 = 126.39 ≈ 127.

127*96 = 12192; PAD-BITS = 12192-12134 = 58bits

FINAL-FRAME = 12192 + 24 + 96 = 12312bits

PSDU user data = 11968bits

The efficiency of BPSK is 97.21%

16-QAM frame

The preamble and the signal field are always sent using BPSK modulation. Therefore, we have to differentiate between the bits sent using BPSK and 16-QAM.

Data bits per OFDM symbol: 96bits

Coded bits per OFDM symbol: 192bits

Coding rate: 1/2

PREAMBLE = 32μs = 96bits

SIGNAL = 4+1+12+1+6 = 24bits

DATA = 16 + PSDU*8 + 6 + PAD-BITS

16 + 1514*8 + 6 = 12134; 12134/192 = 63.18 ≈ 64.

64*192 = 12288; PAD-BITS = 12192-12134 = 154bits

FINAL-FRAME = 12288 + 24 + 96 = 12408bits

PSDU user data = 11968bits

The efficiency of BPSK is 96.45%

Results	Modulation	Coded - Data bits per OFDM symbol	Data rate (Mbit/s) (10 MHz channel spacing)	Efficiency (%) - Confidence interval	Data rate (Mbit/s) (5 MHz channel spacing)	Efficiency (%) - Confidence interval
Theoretical	BPSK[2]	48 - 24	3	93.06% [2.7919 - 2.7921]	1.5	93,39% [1.4085 - 1.4087]
Real			**2.7920**		**1.4086**	
Theoretical	QPSK[2]	96 - 48	6	88.06% [5.2834 -5.2847]	3	89.49% [2.6845 - 2.6851]
Real			**5.2840**		**2.6849**	
Theoretical	16-QAM[2]	192 - 96	12	70,10% [8.4112 – 8.4118]	6	70.95% [4.2558 - 4.2583]
Real			**8.4116**		**4.2570**	

Table 5. Comparison between real and theoretical data rates depending on the channel spacing

Each real data rate has been calculated taking into account the previous results obtained for each modulation. Because, depending on the modulation, the size of the frame may change. The resulting frame is in each case:

- BPSK 1533 Bytes
- QPSK 1539 Bytes
- 16-QAM 1551 Bytes

Results show (Table 6) that we can achieve efficiency level nearby to theoretical values.

Modulation	Theoretical efficiency	Real efficiency (10MHz channel spacing)
BPSK	97.58%	93.06%
QPSK	97.21%	88.06%
16-QAM	96.45%	70.10%

Table 6. Comparison between theoretical and real efficiency for each modulation

Considering the available data rates for 802.11p described during the chapter 0, we only have to focus on 3, 6 and 12 Mbits. We are going to compare the given results at the beginning of this section (Figure 38), the results provided by Arada (Figure 39) and our obtained results. We must underline that we do not know how the results of the other articles are achieved, we only know that they use the same access layer (802.11p) and the payload of the packet. Therefore, the Table 7 is indicative because it only shows a behaviour of how similar devices work.

[2] The coding rate for every modulation is the same: 1/2

Source	Type	3 Mbps	6 Mbps	12 Mbps
Test Figure 38	Frame 1000 Bytes	2.5	5	8
Arada Systems (Figure 39)	UDP frame	2.38	4.37	9
Our results, (Atheros AR9382)	Frame 1514 Bytes	2,79	5.28	8.41

Table 7. Throughput Traffic comparison among different devices.

To sum up, the aim of this test was to probe if the Wi-Fi wireless card certainly follows the theoretical pattern and maintains a compatible data rates for each modulation, as well as comparing our results with others researches in order to verify the proper functioning of the wireless card. Hence, we can observe that our results are similar to the other sources. Therefore, we can conclude that our system is working properly. Nevertheless, during the next test we will compare more results.

5.3.2 Analysis of data rates in adverse environmental conditions.

We want to study the performance of the Atheros wireless cards when the stations are separated and other transmission channels than the common channel are used. The range of frequencies allowed in WLANp is between 5855MHz – 5925MHz. The commercial wireless cards that are being used during the project are not WLANp devices. It means that the manufacturer does not guarantee a well performance in WLANp frequencies. Hence, our interest in analysing at least qualitatively the performance of the ITS channels.

The stations were placed around 15 meters between them. In the transmission path there were also two thin walls made of glass and plastic that can interfere in the transmission. Besides, there should not be frequency interferences since no ITS environments are near the stations. In order to have a measurement of the quality of the communications, the number of lost packets are calculated, which is possible because it is known how many packets are sent and received.

Test conditions:

• Type of software:	RAW_ETHERNET Socket
• PSDU size (Figure 8):	1514 Bytes
• Distance between stations:	~15 meters
• Transmission power:	17dBm
• Frequency:	CCH, SCH4 and SCH6.
• Bandwidth	10MHz

Channel	Data rate 3 Mbps	Lost packets %	Data rate 6 Mbps	Lost packets %	Data rate 12 Mbps	Lost packets %
CCH	2.668	0,0012	4.964	0,0012	8.281	0,0396
SCH4	2,662	0,0012	4.951	0,0012	8.258	0,0396
SCH6	2,663	0,0012	4.947	0,0012	8.259	0,0396

Table 8. Comparison among lost packets, data rare and the transmission channel.

The aim of this test is to check if it could be any malfunctioning when the stations are separated and use different frequency channels. Especially, when it is used the channels that are allocated in each end of the ITS G5 frequencies, this is, the lowest frequency is 5860 MHz and the highest 5920 MHz.

Once again we can show that the obtained results follow the theoretical ones, excepting data rates have slight fallen, but this can be due to statistical nature of the tests.

5.3.3 RSSI power

The aim of this test is to check if the wireless cards are able to fix a transmission power. As it was said in ETSI ITS G5 is compulsory to use a limited power (usually express as dB). Besides, the standard requires that depending on the state of the environment, depending if it is crowded or not, all the transmitters must adapt their power signal in order to not interfere with the neighbour's signals. Thus, it is crucial to establish if the driver and the cards are working as it is intended. Please, note that in IBSS mode the power transmission is fixed automatically by the protocols. As we are using a modified driver we must ensure that when a transmission power is fixed this is really happening.

The technical specification of Sparklan WPEA-111N [68] indicates that the output power at 5GHz range is 12dBm ±1,5dBm, whereas it could not be found the datasheet of Dell U608F and it could be only possible to find the datasheet of AR9344 [69] which may have some similarities because belongs to the same family, and [68] indicates that the output power at 5GHz range is 14dBm ±2dBm. Apart from the technical specifications it is also possible to take into account the information provided by the driver about the maximum transmission power allowed. This information does not always match with the technical specifications but it is used by the driver to fix the limits of transmission.

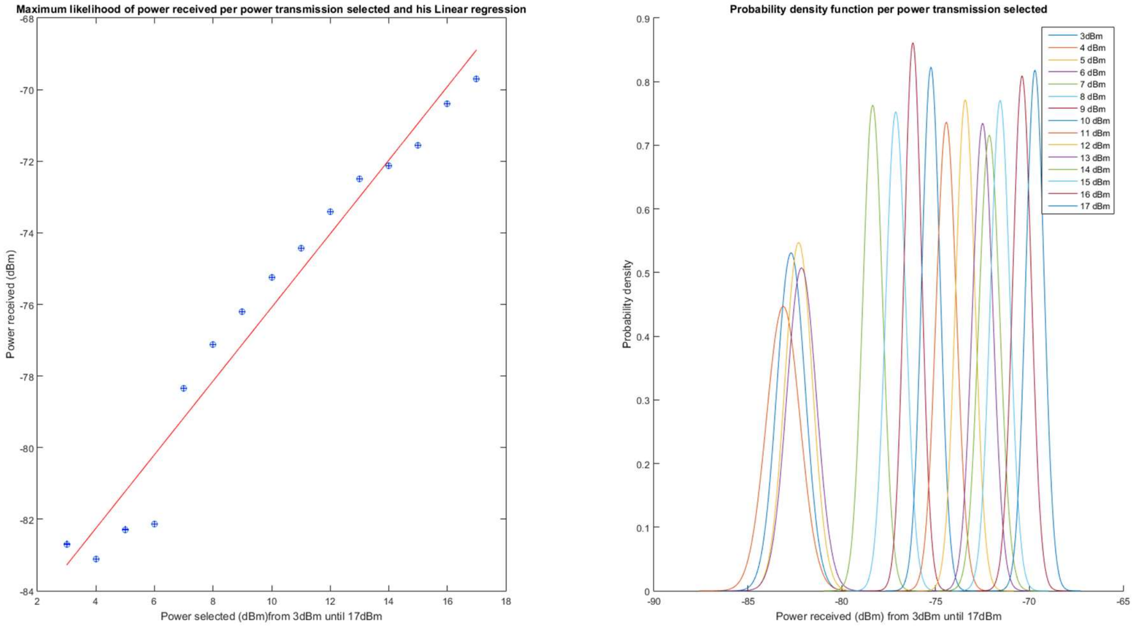
• Atheros HB92 AR9280	17dBm
• Atheros HB116 AR9382	18dBm

The cards are able to measure the signal power at the reception and this value can be consulted with the name of RSSI. It was not found a way to obtain the RSSI parameter using and OCB interface. However, it can be got by using an interface in monitor mode, but this interface only can work with 20MHz bandwidth. Nevertheless, the spectral density should be the same using 20MHz or 10MHz channel spacing. Therefore, the set-up of this experiment consists of one Wi-Fi card configured in OCB mode and 20MHz channel spacing and another Wi-Fi card configure in monitor mode. The card in OCB mode will send sets of packets at different transmission power and the card in monitor mode will gather the RSSI value for each packet.

Test conditions:

• Measure:	RSSI (dBm)
• Distance between stations:	~15 meters
• Transmission power:	3dBm to 17dBm
• Frequency	5900 MHz
• Bandwidth	20MHz

The result of this test is shown in Figure 40.



(a)

(b)

Figure 40. (a) Maximum likelihood of power received per power transmission selected and his Linear regression. (b) Probability density function per power transmission selected

In Figure 40.a. it can be seen how the received power decreases while the transmitted power falls too. In the same figure is has been applied a simple linear regression model. The experimental values match pretty well with the linear model, which demonstrates that the transmission card is modifying the transmission power when it is asked by software. Furthermore, it has been calculated the confidence interval for each point. However, it cannot be appreciated in the figure because the order of magnitude is not comparable. In order to get an idea of the confidence interval, Probability Density Function is shown in Figure 40.b. It is worth noting that we have not demonstrated that the transmission power is the one selected in the card, but only that if we get a transmission power as reference, then the observed increments of power in the receptor matches perfectly with the increments of power in the transmitter. We can see also that for transmission power below a limit, there is no matching between receptor and transmitter, this is due to the reception power is below the sensibility of the card, and thus, only noise is detected.

5.3.4 Reliability of ITS channels

As it was explained, ETSI ITS G5 standard defines 7 different channels. Every transmission starts in Control Channel, and depends on the final application, the communication continues in the same channel (CCH); switch to Service Channel 1 (SCH1) or to Service Channel 3 (SCH3). And if any of these service channels are busy, it is possible to commute to SCH2 or SCH4. Besides, it can be used the ITS G5C band, which provides the SCH5 and SCH6.

To ensure that every channel works under WLANp conditions, it was tested every supported modulation in each ITS channel.

Test conditions:

• Type of software:	RAW_ETHERNET Socket
• Distance between stations:	~40 centimetres
• Transmission power:	17dBm
• Frequency	CCH, SCH1 to SCH6
• Bandwidth	10MHz
• Confidence margin:	99.975
• Wireless card	Atheros AR9382 Sparklan

Channels	3 Mbps (Mbps)	Confidence interval	6 Mbps (Mbps)	Confidence interval	12 Mbps (Mbps)	Confidence interval
CCH	2.7231	2.7229 - 2.7234	5.2059	5.2054 - 5.2066	8.3936	8.3922 - 8.3951
SCH1	2.7222	2.7211 - 2.7233	5.2031	5.1965 - 5.2099	8.3907	8.3871 - 8.3943
SCH2	2.7233	2.7221 - 2.7246	5.2064	5.2058 - 5.2070	8.3893	8.3834 - 8.3953
SCH3	2.7231	2.7230 - 2.7234	5.2067	5.2062 - 5.2074	8.3927	8.3887 - 8.3968
SCH4	2.7230	2.7227 - 2.7234	5.2062	5.2058 - 5.2068	8.3927	8.3907 - 8.3949
SCH5	2.7228	2.7226 - 2.7231	5.2057	5.2051 - 5.2063	8.3875	8.3838 - 8.3912
SCH6	2.7229	2.7226 - 2.7233	5.2050	5.2021 - 5.2081	8.3948	8.3910 - 8.3987

Table 9. Comparison among channels and supported data rates.

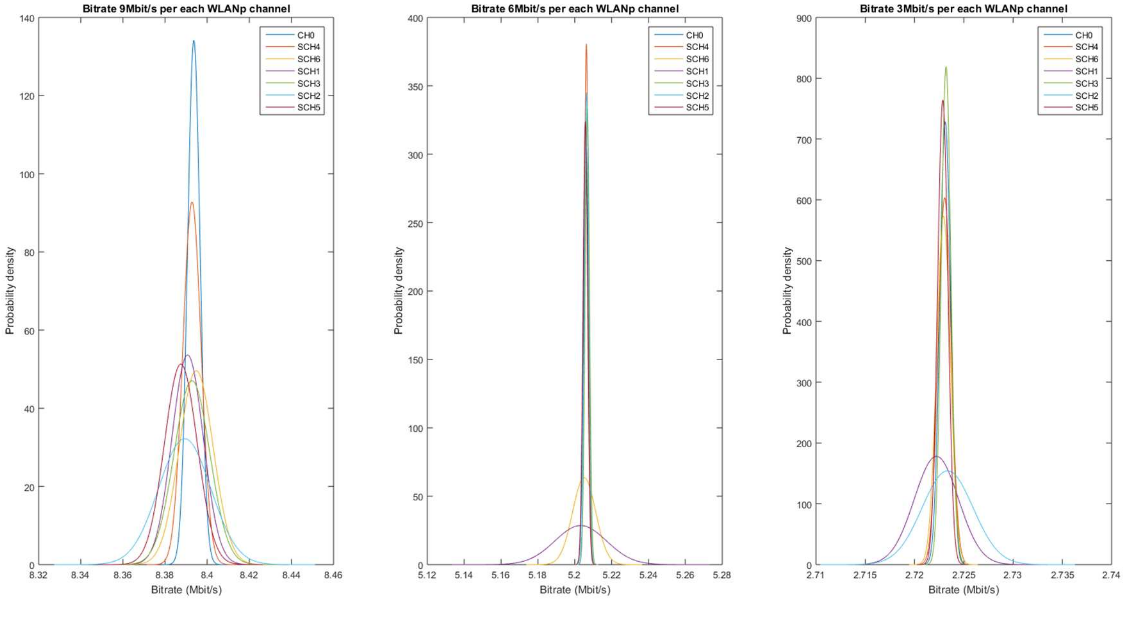


Figure 41. Probability Density Functions for each channel and supported data rate

Once again we can compare the obtained results with other research such as Figure 38 and the datasheet shows in Figure 39. And it is shown that the results preserve the pattern defined at the beginning of the section. Also, it can be set that every channel has similar throughput behaviours. The results may seem better than the obtained results by the Arada product (Figure 39), in the case of 3 Mbps and 6 Mbps. However, we do not know how they made the tests, it is detailed in his datasheet [70] that they used TCP and UDP protocols, but it is not specified the size of the used frames. As we know, it is a huge difference using the minimum and maximum frame size. Therefore, this differences may be associated to this problem.

On the other hand, during this test was detected an unlikely behaviour about the loss of packets. It can be appreciated in Figure 42, where we can find a huge difference between the number of lost packets among ITS channels. But, we have to point out that this behaviour depends on the used Wi-Fi wireless card. This test was made using the Atheros AR9382, however it was observed that using the Atheros AR9280 (older) the results were worst and using the Atheros AR9462 (newer) were better. For reasons of time, it could not be done an exhaustive comparison among wireless cards, but it was observed that the manufactures are improving the 5GHz working mode.

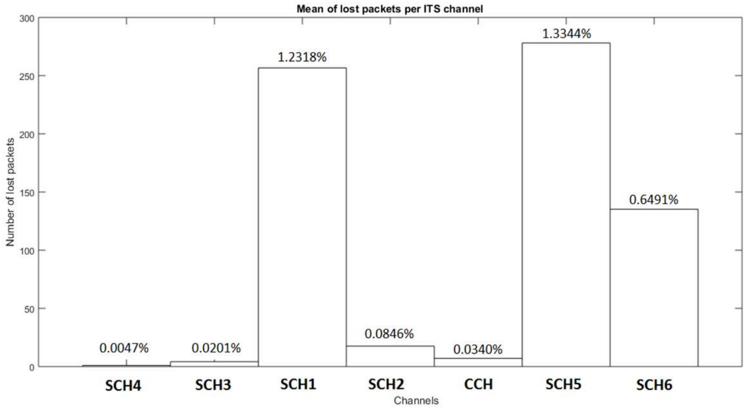


Figure 42. Mean of lost packets per ITS channel

Nevertheless, although in some cases there is a notable difference, it is still a low lost packets percent, because in the worst case involves the 1.3344% of the sent packets.

About the reason of this behaviour, for the moment only it can be supposed that is due to malfunctioning of the physical component. As it was said in previous sections, the wireless cards that we are using were not developed for this purpose. Therefore, it is reasonable that the performance can change from one channel to another.

5.3.5 Switching channel

The Decentralised Congestion Control (DCC) defines the switching channel mechanism. It means that the RSU warns the OBU using the Control Channel which channel will be next to receive the next packet. It is crucial, simulate a scenario where one of the stations receives a packet and, after reading the proper field, switches to other channel.

During this project, it has been found a problem with this mechanism that can be summarised as follows. When one station switches to a channel (leave and join), the first packet that is received in this new channel is not properly read, and only after the second packet, the receptor is able to receive it perfectly.

In order to demonstrate this question, it was performed an experiment. The set-up of the experiment consists of two WLANp stations both using CCH with exactly the same configuration. The sender transmits one packet per two seconds and the receiver is always listening in CCH. When this listening station receives a packet, it will leave and join the Control Channel. The result is shown in Figure 43.

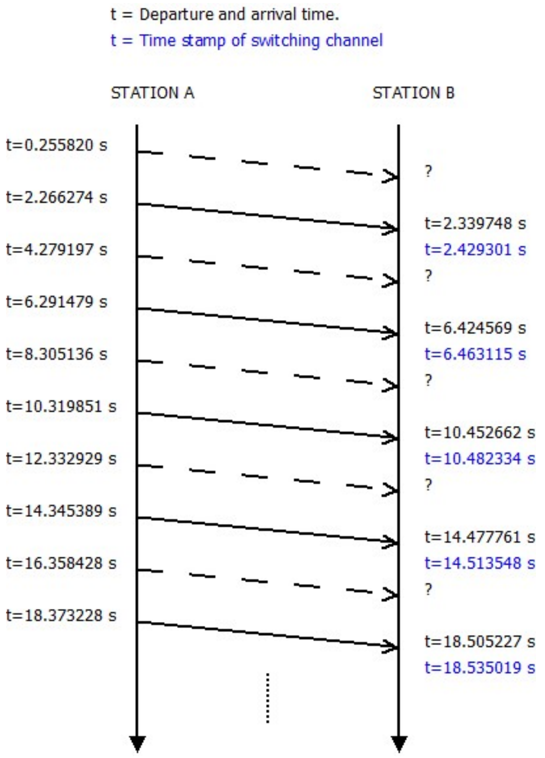


Figure 43. Leaving and joining the channel per packet received (2 seconds)

As it is shown, every packet received after switching channel will be lost. It does not mean that the receiver station does not receive it, but that it could not be read. To embrace this theory, it was made two more tests. In the first one the receiver station leaves and join the Control Channel every 1.5 seconds. In this case it is expected that no packed is read, because the switching channel timer is lower than the frequency receiving packets and that way the card never receives two packets in the same switching period. As it is shown in Figure 44 the result is as expected.

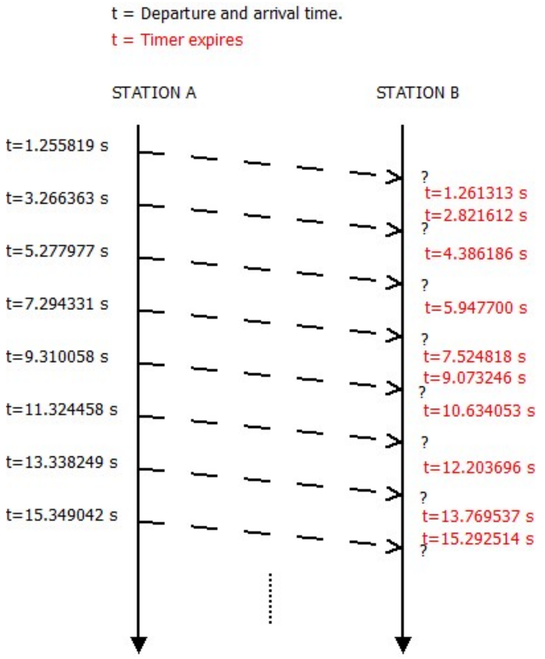


Figure 44. Leaving and joining the CCH when the timer expires

A second test was performed. This second test is designed to discard the possibility that the time needed by the card in order to switch is greater than 2 seconds. In order to demonstrate that this is not the case in this experiment the transmitter station sent the packets with a period of 30 seconds. We know from the first experiment that the switching time must be lower than 4 seconds, as the receiver station always read the second message. If the station is unable to read the first packet after waiting 30 seconds that only can mean that the first packet is never read. As it is shown in Figure 45 the first message is always lost.

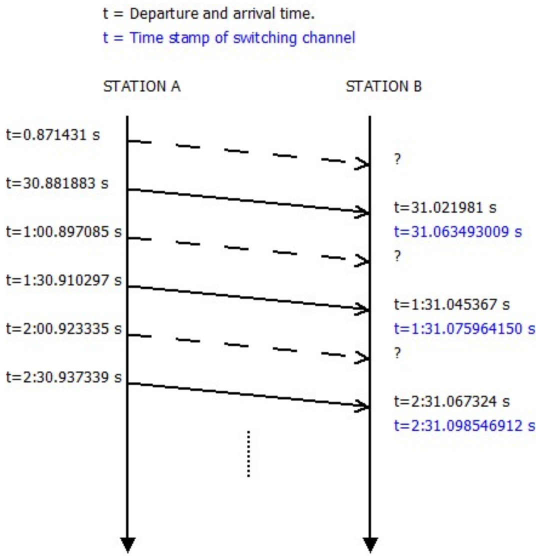


Figure 45. Leaving and joining the channel per packet received (30seconds)

In order to discard any other possibilities, as for example that the programs that we used in the experiment would be doing something wrong, several programs such as “tcpdump”, “tshark” and a handmade program were used. For the sender station most of the times was used a handmade program which was able to send customized frames directly to the wireless card using a socket. Also it was used programs as “ping” and “iperf” which have the same aim, but they have a different working mode as well as send other kind of frames. It was looked for different methods to ensure that the results have full independence of the software program.

It has not been found the source of the problem. The best assumption in our opinion is that this behaviour always happens when it is leaved and joined the channel because the receiver station uses the first packet to analyse the parameters of the channel and the configuration of the communication (bit rate, modulation, etc.). The standard says that in order to configure all these parameters the receptor should use the preamble and the signal field (Figure 8. Nevertheless, it seems that when it is using an OCB interface, it needs more than one packet to synchronize the communication.

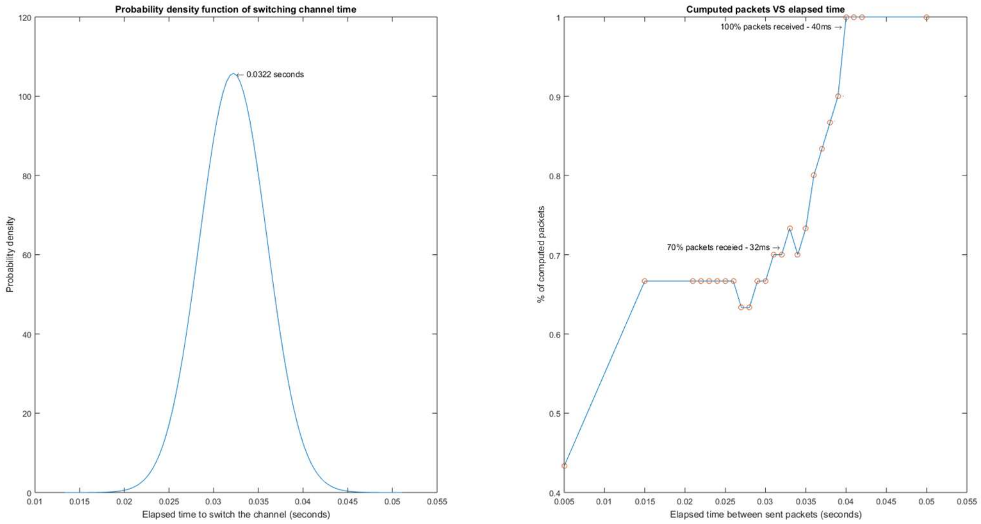
This can be a serious problem, because as it was explained the ETSI ITS G5 defines a multiradio environment, this is, while it is always one wireless card listening the CCH the other one should be commuting to the required Service Channel to listen to the next message. But if this message is not repeated, as the card is always losing the first message received after switching, that means that it will not be able to get the expected message. The only solution is to use as many cards as channels we want to listen to, or the messages to be sent more than once.

5.3.6 Channel switching time

Along this memory it has been shown the importance of frequency hopping in ETSI ITS G5 standard. As a consequence, and taking into account the results of the previous section, it is reasonable to calculate the time to commute between channels, although the first message in the channel will be lost. Moreover, these wireless cards were not specifically designed for WLANp neither the driver software that we are using. In fact, supposedly the card was prepared to work in a selected channel without switching. Therefore, this hardware could have problems switching the working channel.

The standard set that the maximum allowed time to switch is 50ms. This is shown in Figure 21, where DP0 application sends messages every 50ms during emergency situations. As a result, the channel switching time, should be less than 50ms, because in emergency situations this time is consider critical in order to avoid dangers.

In order to avoid the problem described in the previous section (section 5.3.4), two consecutive messages are sent in the same channel. This way, we can calculate the minimum time to switch regardless of this problem.



(a) (b)

Figure 46. Mean time to hop frequency channel

Figure 46.a. shows the Probability Density Function (PDF) of the elapsing time required to hop of channel. It is possible to compute the elapsed time using the "time" Linux command, which computes the time used by the instruction in carrying out the operation. However, this time does not imply that this is the time required by the card to change the frequency, as after executing the instruction the card could need to do other internal operations. Anyway, the results show that it is required 32.2 ms to hop between channels. It means that, not taking into account the lost packet problem, the wireless card should have enough time to switch between channels. However, it was made a second test, whose results are showed in Figure 46.b. In this experiment the number of lost packets are count depending on the elapsed time from the switching event, and it is clear that if we want to guarantee that 100% of the packets are read, then it is necessary to wait up to 40 ms.

Nevertheless, this time is still under the limit of 50ms and it was measured in a practical test. Therefore, we can assume, the wireless card meets the specifications and it will be able to receive all the packets, if the problem losing the first received packet is solved.

5.4 Discussion of the results

Reminding the targets set at the beginning of this project (chapter 1), finally it has been demonstrated that there are Wi-Fi wireless cards suitable for 802.11p. However, it could be only found one driver which supports the OCB mode. Currently, it can be only used the ATH9K wireless cards family, all of them has miniPCI-e or PCI-e connection port. Also, some of them support the ITS channels defined in the section 3.3, because as we saw during the section 4.2, depending on the final assembler manufacturer this characteristic may change.

Regarding the performance of these conventional Wi-Fi wireless cards for 802.11p (sections 5.3.1, 5.3.2, 5.3.3, 5.3.4, 5.3.5 and 5.3.6), it has been demonstrated that these wireless cards are faster enough to switch between channels, it is possible controlling the power transmission as defined by the Decentralized Congestion Control (DCC) and the modified driver supports the data rates defined by 802.11p standard (Table 2). On the other hand, it was observed that every time that it was switched the channel the card always loses the first packet. It was given a possible reason for this malfunctioning. However, if this project must continue in the future, it is advisable keep in mind this behaviour. Moreover, the implementation of the complete protocol stack requires the be able to read the power of every packet received (RSSI value) in order to implement the DCC. However, we have not been able to access to this value, so this is also another problem to be solved.

Due to the results achieved and that in this project the main objective is to record and analyse the transmitted messages from other station, we can conclude that it is possible to implement this project in a commercial solution, such as the [redacted], and it meets the current specifications. In fact, it was demonstrated that it is possible use the [redacted] in ITS environments only mirroring this project in its Linux distribution. Moreover, it was verified with the Road Side Unit developed by [redacted] consortium.

6 Conclusions

Recalling the defined targets of this project, the first aim of this project was finding a suitable Wi-Fi wireless card for our purpose. It has been demonstrated that there are some cards which use the ATH9K driver that can support the 802.11p standard. On the other hand, it also depends on the manufacturer of the wireless card, as it was given the example of the Atheros 9280 assembled by Ubiquiti that cannot support the ITS channels even if it uses the ATH9K driver. Finally, it was checked that USB wireless cards that use the ATH9K_htc driver, are not modified to support 802.11p standard.

On the basis of the software modifications (kernel and user level applications), it has been verified that are suitable for 802.11p standard. Also, it has been proved that the kernel modifications are compatible with upper kernel versions (in the future may be included in the official kernel release). Moreover, as we saw in the section 4.1, the OCB support for the driver ATH9K is being included in official Linux kernel release. Therefore, the upcoming stable versions will include the medicated driver officially.

Focusing on the results of the test, we saw that the throughput traffic, switching channel timing and the power transmission are the expected. However, it was found a malfunctioning behaviour in the switching channel procedure which can be a problem in the future. If this is a problem whose source is to be found or if it just a normal behaviour of 802.11p is something that is still not known. Nevertheless, currently the multi-channel operation is not completely defined by ETSI. Consequently, when the ITS stack will be full defined, this problem may have been solved or other compatible commercial Wi-Fi wireless cards may have been appeared.

To sum up, it has been demonstrated (section 5.4) that this implementation can be used in real ITS environments [redacted]. Furthermore, it can be use in European and American standard if it is followed the instructions given in section 3.7, due to both standards use 802.11p at Access layer. Consequently, this project can be used to capture data from the channels defined by WAVE.

7 Future work

The deployment of a new technology is an open door for developing. The number of applications that can be developed are countless. During this chapter some future paths that can be followed taking as a basis this project will be given.

- To develop the whole ITS stack. Probably using one of the Open Source project explained during this project. "Vanetza" project [22] is an interesting one, it has already implemented GeoNetworking, DCC mechanism, Basic Transport Protocol and aspects of the Facilities layer.
- Implementation of the DCC mechanism. As we have shown during this project, this mechanism takes part into every layer of the stack, not only the Access layer. As a result, this is one of the main parts of the ITS stack, and it should be implemented as soon as possible.
- Switching channel using the DCC mechanism. DCC alerts when the station should commute between channels. However, this mechanism is not responsible of joining or leaving a channel. It should be integrated both mechanisms to complete the process.
- Finding the source or a solution for the problem described in the section 5.3.5. In the future the channel hopping may be used. Therefore, it is crucial to find a solution for this malfunctioning.
- Research about the use of the driver ATH9_htc for 802.11p. This driver belongs to the same family as ATH9K. However, it is use for USB Wi-Fi wireless cards. This interface could be very useful for devices that do not support PCI-e half size.
- Research about the possible solution of programing a bridge between PCI-e and USB interface.
- Finding suitable Open Source projects to implement the US stack, it means the Network and Transport layer and Facilities layer.
- Finding suitable plugins for network protocol analyser (Wireshark, tcpdump, etc.) for US stack. The idea is looking for a plugin for WAVE as it was done for ITS G5 in the section 4.4.

References

[1] Official Journal of the European Union, 'Directive 2010/40/EU'. 07-2010.

[2] ETSI, 'Intelligent Transport Systems - Standards', *ETSI.org*, 15-Dec-2016.

[3] European Commission, 'SCOOP project', Jan-2016.

[4] 'European Corridor - Austrian Testbed for Cooperative Systems'.

[5] GCDC, 'I-GAME event', 28-May-2016.

[6] [redacted].

[7] T. K. IEEE Standards Association, '1609 - Dedicated Short Range Communication'. 15-Dec-2016.

[8] ARIB, 'STD-T109. 700MHz Band Intelligent Transport Systems'. 14-Feb-2012.

[9] ETSI, 'Details of "DTS/ITS-002004" Work item', 03-Feb-2010.

[10] European Commission, 'Cooperative, connected and automated mobility (C-ITS)', Jan. 2016.

[11] European Commission, 'C-ITS platform - Final report', Jan. 2016.

[12] European Commission, 'Work Programme second phase C-ITS Platform', Nov. 2016.

[13] Atheros, 'Backports ATH9K driver', *Linux Backports*, 2007.

[14] [redacted].

[15] [redacted].

[16] ECC, 'ECC Decision (08)01 - "The harmonised use of the 5875-5925 MHz frequency band for Intelligent Transport Systems (ITS)"'. 14-Mar-2008.

[17] GCDC, 'Grand Cooperative Driving Challenge', Oct-2016.

[18] Developer Community, 'Wiki-Debian information about ATH5K drivers', 2016.

[19] Developer Community, 'Wiki-Debian information about ATH9K drivers', 2016.

[20] R. L. CTU-IIG, '802.11p on Linux - Commit "ath9k OCB supported"', 18-Dec-2014.

[21] Isabelle Vandoorne, 'Intelligent (road) transport systems incl. cooperative, connected and automated vehicles: opportunities and developments'. 20-Sep-2016.

[22] Raphael Riebl, '"Vanetza" Githug Project', 06-Sep-2013.

[23] Alex Voronov, '"GeoNetworking" Github Project', 18-Jan-2016.

[24] R. L. CTU-IIG, '802.11p on Linux - Github repository', 18-Dec-2014.

[25] Linus Torvalds, 'Official Linux kernel Github repository', 16-Sep-2001.

[26] *OMNet++. Discrete Event Simulator.* OpenSim Ltd., 2016.

[27] Christoph Sommer, *'Venis' Vehicles in Network Simulation.* 2016.

[28] J. M. Michele Rondinone *et al.*, '"iTetris". A modular simulation platform for the large scale evaluation of cooperative ITS applications'. Elsevier, May-2013.

[29] Marben, *'Marben' Software solution for V2X.* 2015.

[30] R. Lisov´y, M. Sojka, Z. Hanz´alek, 'IEEE 802.11p Linux Kernel Implementation'. Czech Technical University in Prague, 10-Dec-2014.

[31] Wiki, 'ATH9K supported devices', 22-Jun-2015.

[32] IEEE Standards Association, 'IEEE Standard for information technology Telecommunications and information exchange between systems Local and metropolitan area networks'. IEEE Computer Society, 03-2012.

[33] European Standard, 'EN 302 663: "Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band"'. Nov-2012.

[34] Cornell University Law School, 'CFR 90.377. Frequencies available; maximum EIRP and antenna height, and priority communications.' 27-Jun-2007.

[35] ETSI, 'TS 102 687: "Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer part"'. Jul-2011.

[36] American Society for Testing and Materials, 'ASTM E2212 standard: "Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems — 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications"'. 2002.

[37] F. D. Christoph Sommer, *Vehicular Networking*. Cambridge University Press, 2015.

[38] Wiki, 'Release 3.19 kernel Linux', *kernelnewbies.org*, 08702-2015.

[39] Rostislav Lisovy, 'Commit information - Added OCB support in official kernel Linux release.', 11-Apr-2014.

[40] 'Netlink Linux Programmer's Manual', 12-Dec-2016.

[41] Javier Fernandez Pastrana, 'Github ATH9K OCB patch for Linux 4.2.8', 19-Jun-2016.

[42] Javier Fernandez Pastrana, 'Github ATH9K OCB patch for Linux 4.7.rc3', 19-Jun-2016.

[43] CTU-IIG, '"802.11p-iw" project: "Wireless configuration tool"', 23-Sep-2007.

[44] Linux Foundation, 'Official wireless regdb: "Linux Wireless Regulatory Database"', 04-Jun-2015.

[45] CTU-IIG, '"802.11p-crda" Github project. CRDA modified for 802.11p', 18-May-2008.

[46] Linux Foundation, 'Official release CRDA: "Linux Central Regulatory Domain Agent"', 29-Apr-2016.

[47] ETSI, 'TS 102 724: "Intelligent Transport Systems (ITS); Harmonized Channel Specifications for Intelligent Transport Systems operating in the 5 GHz frequency band"'. Oct-2012.

[48] ETSI, 'TS 102 636-3: "Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 3: Network architecture"'. Mar-2010.

[49] ETSI, 'EN 302 637-2: "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service'. Sep-2014.

[50] ETSI, 'TS 102 894-1: "Intelligent Transport Systems (ITS); Users and applications requirements; Part 1: Facility layer structure, functional requirements and specifications"'. Aug-2013.

[51] ETSI, 'EN 302 637-3: "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service"'. Sep-2014.

[52] ETSI, 'EN 302 895: "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM)"'. Jan-2014.

[53] ETSI, 'EN 102 636-5-1: "Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 5: Transport Protocols; Sub-part 1: Basic Transport Protocol"'. Aug-2014.

[54] ETSI, 'TS 102 636-4-1: "Intelligent Transport Systems (ITS); Vehicular communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality"'. Jun-2011.

[55] A. M. Claudia Campolo and Riccardo Scopigno, *Vehicular ad hoc Networks. Standards, Solutions and Research*. Springer, 2015.

[56] De Martini Laura, 'Multi-Channel Congestion Control for ETSI ITS G5 A/B'. Jun-2013.

[57] Lan LIN, 'ETSI G5 technology: The European approach', 13-Jun-2013.

[58] IEEE Standards Association, '1609.3 Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services'. 2016.

[59] ETSI, 'TS 102 636-4-2: "Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 2: Media-dependent functionalities for ITS-G5"'. Oct-2013.

[60] ETSI, 'TS 102 940: "Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management"'. Jun-2012.

[61] European Commission, 'HTG3-1. Status of ITS Communication Standards'. 11-Dec-2012.

[62] E. garciay, *Wireshark ITS Plugin (rev 69)*. Wireshark, 2016.

[63] ISO/TS, '19321:2015: "Intelligent transport systems -- Cooperative ITS -- Dictionary of in-vehicle information (IVI) data structures"'. 15-Apr-2015.

[64] ISO/TS, '14823:2008: "Traffic and travel information -- Messages via media independent stationary dissemination systems -- Graphic data dictionary for pre-trip information dissemination systems"'. 15-Jul-2008.

[65] ETSI, 'EN 302 636-4-1: "Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality'. Oct-2013.

[66] X. D. Yunpeng Wang, G. L. Daxin Tian, and Haiyang YU, 'Throughput and Delay Limits of 802.11p and Its Influence on Highway Capacity'. 2013.

[67] Arada Systems, 'Product datasheet "LocoMate Classic On Board Unit"'. 2016.

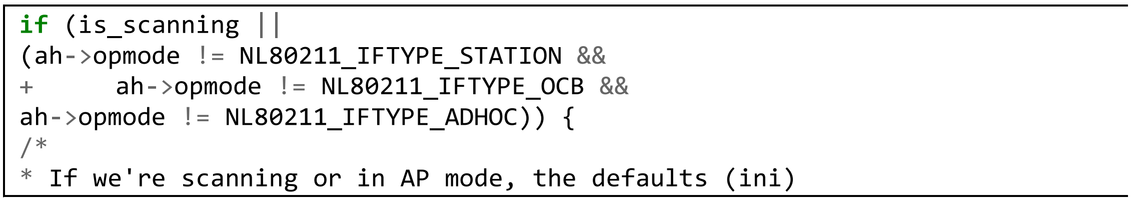
[68] Atheros, 'AR9280 Datasheet'. Oct-2009.

[69] Atheros, 'AR9344 Datasheet'. Dec-2010.

[70] 'Arada_datasheet_obu_2.01_2015.pdf'.

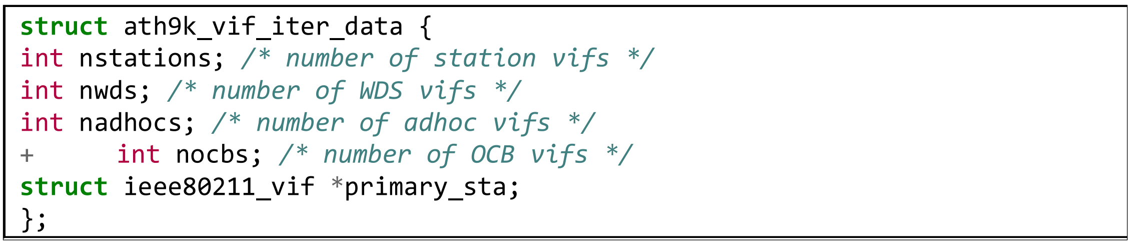
A. Appendix

In this annex it is shown the changes of ath9k driver. The next figures show the difference between the original source and the modified code. Therefore, any developer can write the code in their own Linux kernel version.



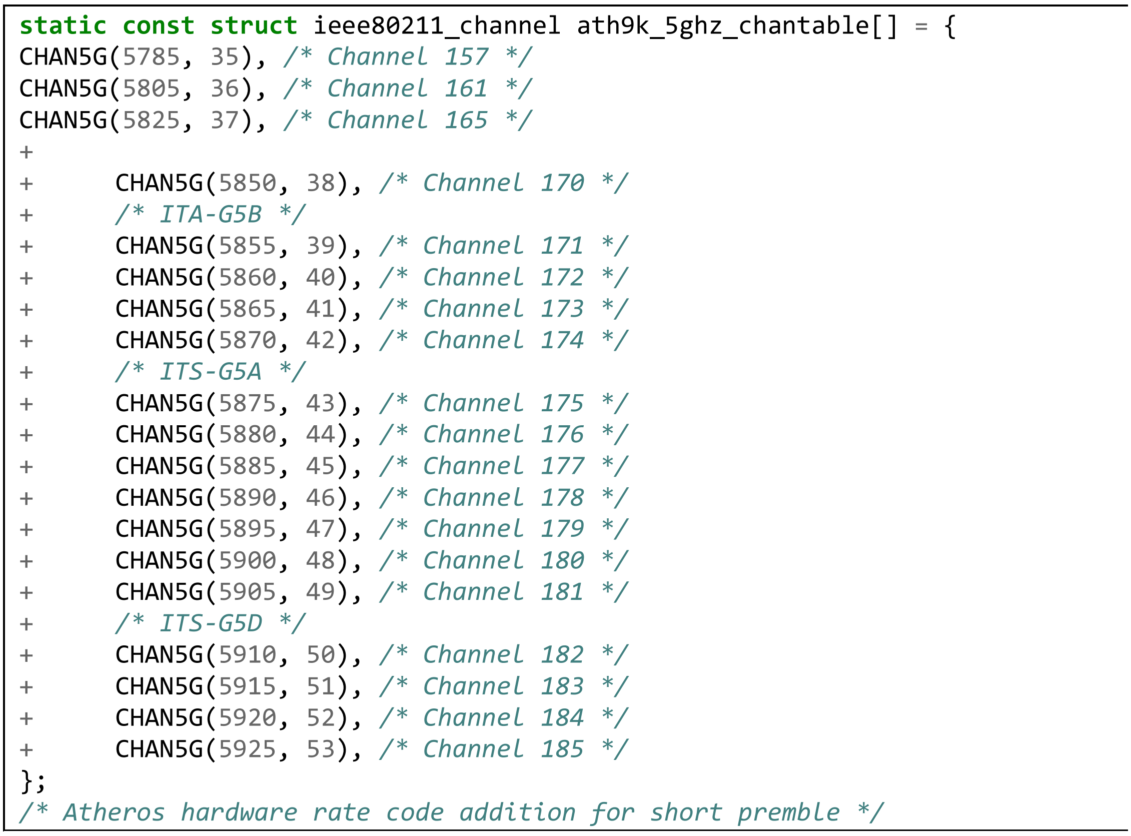
```
if (is_scanning ||
(ah->opmode != NL80211_IFTYPE_STATION &&
+      ah->opmode != NL80211_IFTYPE_OCB &&
ah->opmode != NL80211_IFTYPE_ADHOC)) {
/*
* If we're scanning or in AP mode, the defaults (ini)
```

Appendix figure 1. drivers/net/wireless/ath/ath9k/ani.c



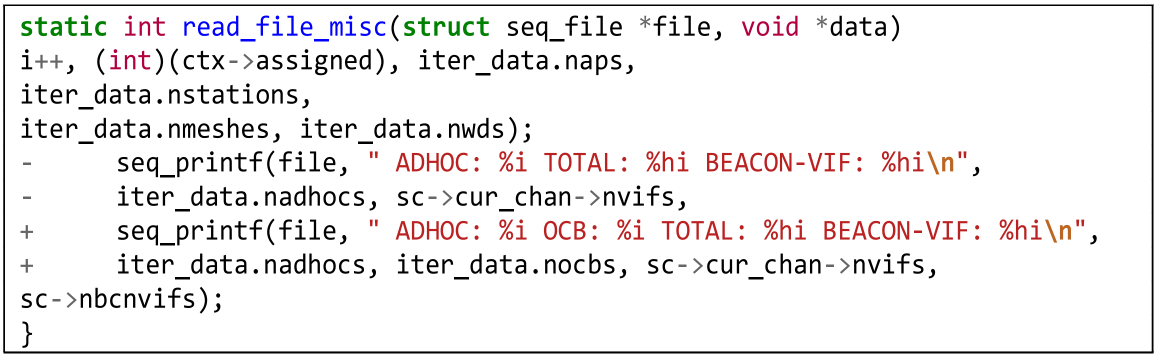
```
struct ath9k_vif_iter_data {
int nstations; /* number of station vifs */
int nwds; /* number of WDS vifs */
int nadhocs; /* number of adhoc vifs */
+      int nocbs; /* number of OCB vifs */
struct ieee80211_vif *primary_sta;
};
```

Appendix figure 2. drivers/net/wireless/ath/ath9k/ath9k.h



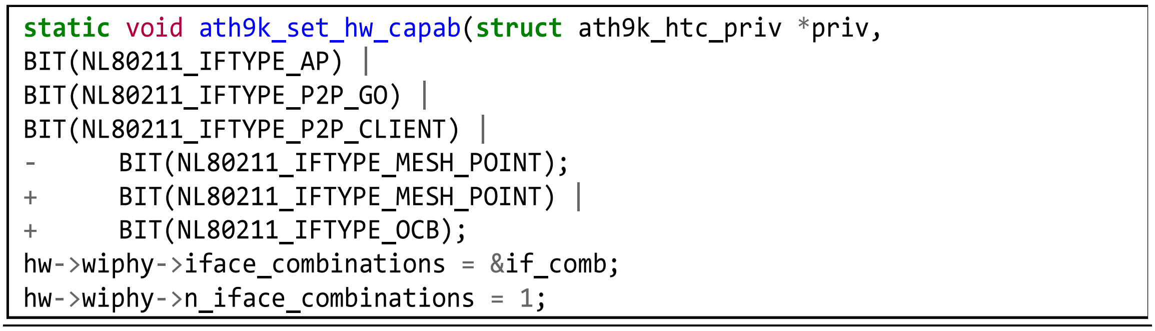
```
static const struct ieee80211_channel ath9k_5ghz_chantable[] = {
CHAN5G(5785, 35), /* Channel 157 */
CHAN5G(5805, 36), /* Channel 161 */
CHAN5G(5825, 37), /* Channel 165 */
+
+      CHAN5G(5850, 38), /* Channel 170 */
+      /* ITA-G5B */
+      CHAN5G(5855, 39), /* Channel 171 */
+      CHAN5G(5860, 40), /* Channel 172 */
+      CHAN5G(5865, 41), /* Channel 173 */
+      CHAN5G(5870, 42), /* Channel 174 */
+      /* ITS-G5A */
+      CHAN5G(5875, 43), /* Channel 175 */
+      CHAN5G(5880, 44), /* Channel 176 */
+      CHAN5G(5885, 45), /* Channel 177 */
+      CHAN5G(5890, 46), /* Channel 178 */
+      CHAN5G(5895, 47), /* Channel 179 */
+      CHAN5G(5900, 48), /* Channel 180 */
+      CHAN5G(5905, 49), /* Channel 181 */
+      /* ITS-G5D */
+      CHAN5G(5910, 50), /* Channel 182 */
+      CHAN5G(5915, 51), /* Channel 183 */
+      CHAN5G(5920, 52), /* Channel 184 */
+      CHAN5G(5925, 53), /* Channel 185 */
};
/* Atheros hardware rate code addition for short premble */
```

Appendix figure 3. drivers/net/wireless/ath/ath9k/common-init.c



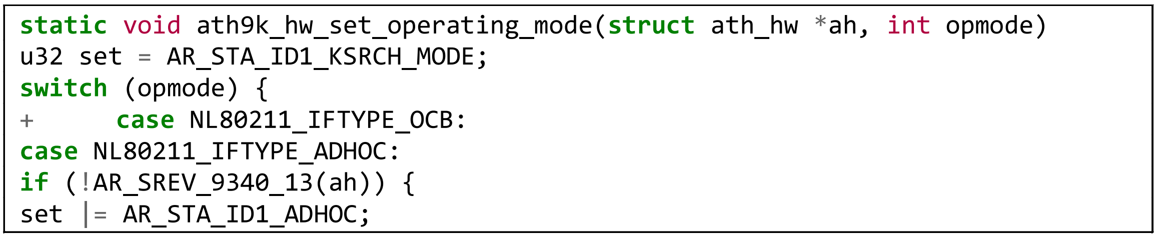
```
static int read_file_misc(struct seq_file *file, void *data)
i++, (int)(ctx->assigned), iter_data.naps,
iter_data.nstations,
iter_data.nmeshes, iter_data.nwds);
-      seq_printf(file, " ADHOC: %i TOTAL: %hi BEACON-VIF: %hi\n",
-      iter_data.nadhocs, sc->cur_chan->nvifs,
+      seq_printf(file, " ADHOC: %i OCB: %i TOTAL: %hi BEACON-VIF: %hi\n",
+      iter_data.nadhocs, iter_data.nocbs, sc->cur_chan->nvifs,
sc->nbcnvifs);
}
```

Appendix figure 4. drivers/net/wireless/ath/ath9k/debug.c



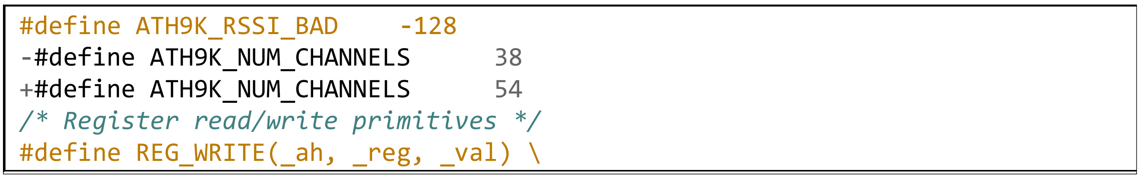
```
static void ath9k_set_hw_capab(struct ath9k_htc_priv *priv,
BIT(NL80211_IFTYPE_AP) |
BIT(NL80211_IFTYPE_P2P_GO) |
BIT(NL80211_IFTYPE_P2P_CLIENT) |
-      BIT(NL80211_IFTYPE_MESH_POINT);
+      BIT(NL80211_IFTYPE_MESH_POINT) |
+      BIT(NL80211_IFTYPE_OCB);
hw->wiphy->iface_combinations = &if_comb;
hw->wiphy->n_iface_combinations = 1;
```

Appendix figure 5. drivers/net/wireless/ath/ath9k/htc_drv_init.c



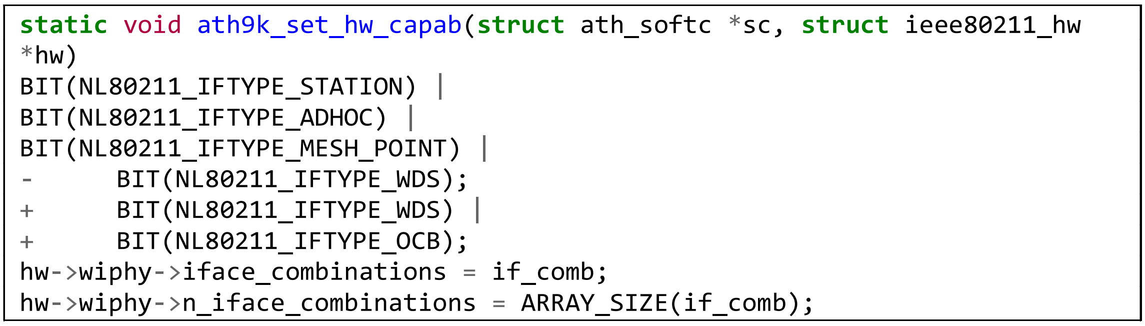
```
static void ath9k_hw_set_operating_mode(struct ath_hw *ah, int opmode)
u32 set = AR_STA_ID1_KSRCH_MODE;
switch (opmode) {
+      case NL80211_IFTYPE_OCB:
case NL80211_IFTYPE_ADHOC:
if (!AR_SREV_9340_13(ah)) {
set |= AR_STA_ID1_ADHOC;
```

Appendix figure 6. drivers/net/wireless/ath/ath9k/hw.c



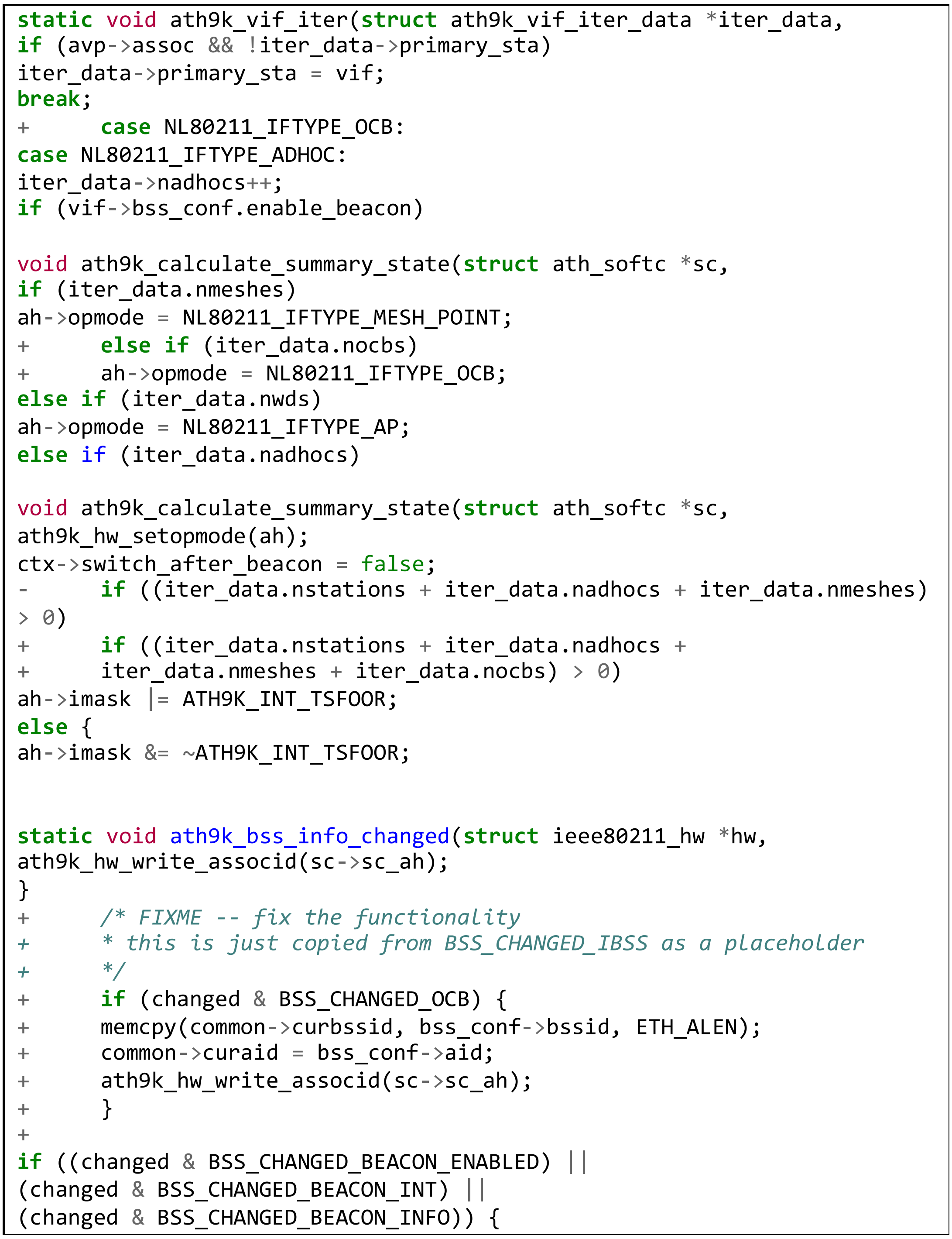
```
#define ATH9K_RSSI_BAD    -128
-#define ATH9K_NUM_CHANNELS      38
+#define ATH9K_NUM_CHANNELS      54
/* Register read/write primitives */
#define REG_WRITE(_ah, _reg, _val) \
```

Appendix figure 7. drivers/net/wireless/ath/ath9k/hw.h



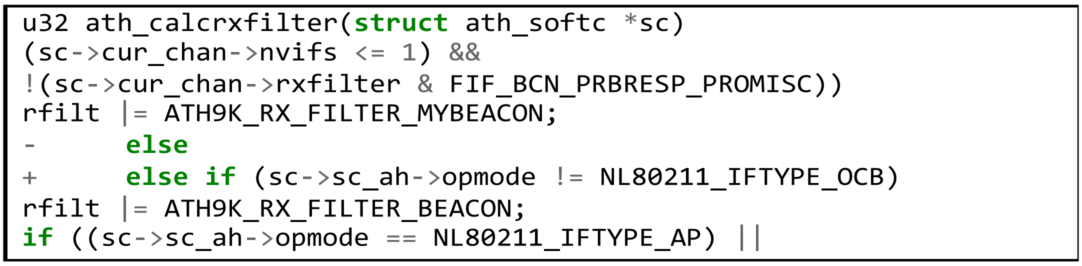
```
static void ath9k_set_hw_capab(struct ath_softc *sc, struct ieee80211_hw
*hw)
BIT(NL80211_IFTYPE_STATION) |
BIT(NL80211_IFTYPE_ADHOC) |
BIT(NL80211_IFTYPE_MESH_POINT) |
-      BIT(NL80211_IFTYPE_WDS);
+      BIT(NL80211_IFTYPE_WDS) |
+      BIT(NL80211_IFTYPE_OCB);
hw->wiphy->iface_combinations = if_comb;
hw->wiphy->n_iface_combinations = ARRAY_SIZE(if_comb);
```

Appendix figure 8. drivers/net/wireless/ath/ath9k/init.c



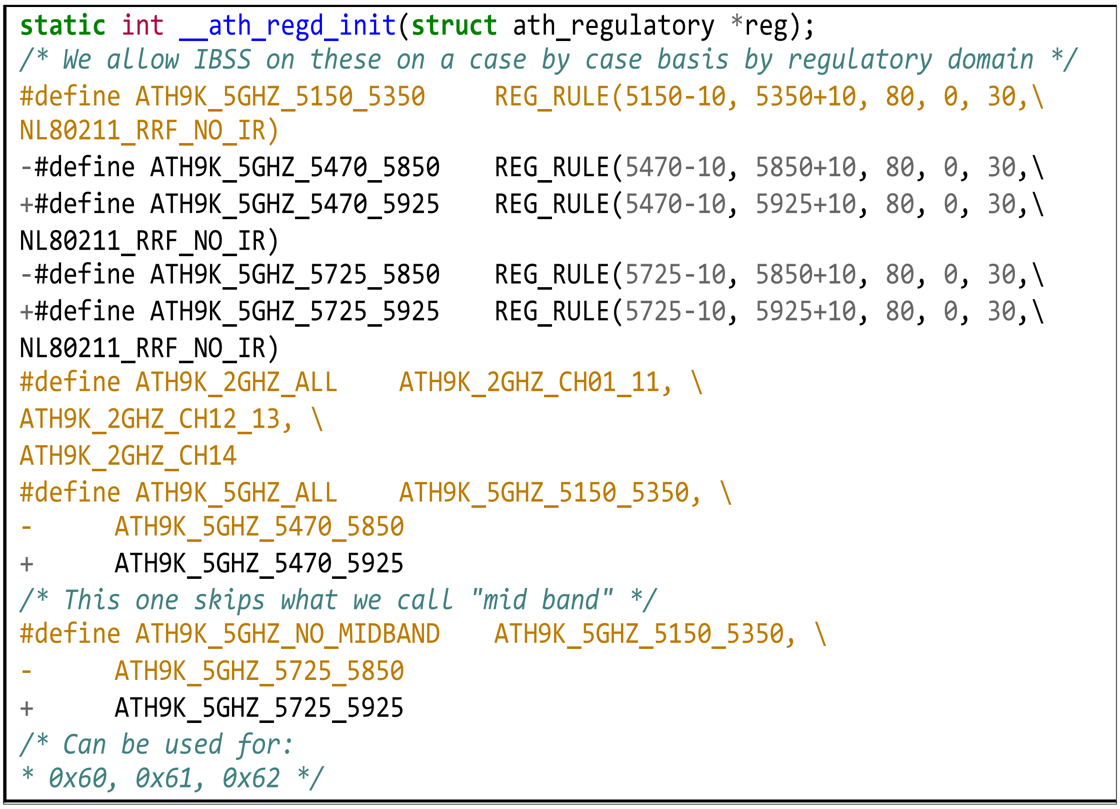
```
static void ath9k_vif_iter(struct ath9k_vif_iter_data *iter_data,
if (avp->assoc && !iter_data->primary_sta)
iter_data->primary_sta = vif;
break;
+      case NL80211_IFTYPE_OCB:
case NL80211_IFTYPE_ADHOC:
iter_data->nadhocs++;
if (vif->bss_conf.enable_beacon)

void ath9k_calculate_summary_state(struct ath_softc *sc,
if (iter_data.nmeshes)
ah->opmode = NL80211_IFTYPE_MESH_POINT;
+      else if (iter_data.nocbs)
+      ah->opmode = NL80211_IFTYPE_OCB;
else if (iter_data.nwds)
ah->opmode = NL80211_IFTYPE_AP;
else if (iter_data.nadhocs)

void ath9k_calculate_summary_state(struct ath_softc *sc,
ath9k_hw_setopmode(ah);
ctx->switch_after_beacon = false;
-      if ((iter_data.nstations + iter_data.nadhocs + iter_data.nmeshes)
> 0)
+      if ((iter_data.nstations + iter_data.nadhocs +
+      iter_data.nmeshes + iter_data.nocbs) > 0)
ah->imask |= ATH9K_INT_TSFOOR;
else {
ah->imask &= ~ATH9K_INT_TSFOOR;


static void ath9k_bss_info_changed(struct ieee80211_hw *hw,
ath9k_hw_write_associd(sc->sc_ah);
}
+      /* FIXME -- fix the functionality
+      * this is just copied from BSS_CHANGED_IBSS as a placeholder
+      */
+      if (changed & BSS_CHANGED_OCB) {
+      memcpy(common->curbssid, bss_conf->bssid, ETH_ALEN);
+      common->curaid = bss_conf->aid;
+      ath9k_hw_write_associd(sc->sc_ah);
+      }
+
if ((changed & BSS_CHANGED_BEACON_ENABLED) ||
(changed & BSS_CHANGED_BEACON_INT) ||
(changed & BSS_CHANGED_BEACON_INFO)) {
```

Appendix figure 9. drivers/net/wireless/ath/ath9k/main.c



```
u32 ath_calcrxfilter(struct ath_softc *sc)
(sc->cur_chan->nvifs <= 1) &&
!(sc->cur_chan->rxfilter & FIF_BCN_PRBRESP_PROMISC))
rfilt |= ATH9K_RX_FILTER_MYBEACON;
-      else
+      else if (sc->sc_ah->opmode != NL80211_IFTYPE_OCB)
rfilt |= ATH9K_RX_FILTER_BEACON;
if ((sc->sc_ah->opmode == NL80211_IFTYPE_AP) ||
```

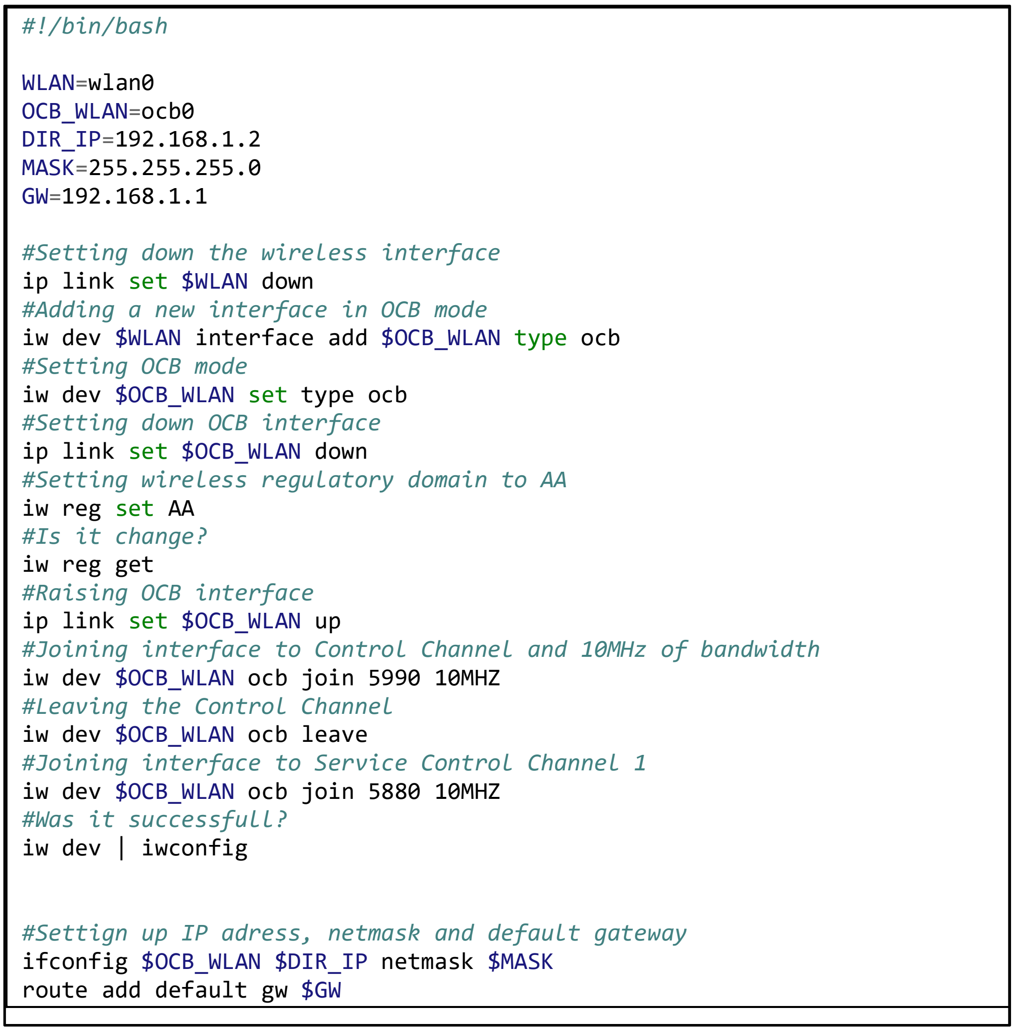
Appendix figure 10. drivers/net/wireless/ath/ath9k/recv.c



```
static int __ath_regd_init(struct ath_regulatory *reg);
/* We allow IBSS on these on a case by case basis by regulatory domain */
#define ATH9K_5GHZ_5150_5350     REG_RULE(5150-10, 5350+10, 80, 0, 30,\
NL80211_RRF_NO_IR)
-#define ATH9K_5GHZ_5470_5850    REG_RULE(5470-10, 5850+10, 80, 0, 30,\
+#define ATH9K_5GHZ_5470_5925    REG_RULE(5470-10, 5925+10, 80, 0, 30,\
NL80211_RRF_NO_IR)
-#define ATH9K_5GHZ_5725_5850    REG_RULE(5725-10, 5850+10, 80, 0, 30,\
+#define ATH9K_5GHZ_5725_5925    REG_RULE(5725-10, 5925+10, 80, 0, 30,\
NL80211_RRF_NO_IR)
#define ATH9K_2GHZ_ALL    ATH9K_2GHZ_CH01_11, \
ATH9K_2GHZ_CH12_13, \
ATH9K_2GHZ_CH14
#define ATH9K_5GHZ_ALL    ATH9K_5GHZ_5150_5350, \
-      ATH9K_5GHZ_5470_5850
+      ATH9K_5GHZ_5470_5925
/* This one skips what we call "mid band" */
#define ATH9K_5GHZ_NO_MIDBAND    ATH9K_5GHZ_5150_5350, \
-      ATH9K_5GHZ_5725_5850
+      ATH9K_5GHZ_5725_5925
/* Can be used for:
* 0x60, 0x61, 0x62 */
```

Appendix figure 11. drivers/net/wireless/ath/regd.c

B. Appendix



```
#!/bin/bash

WLAN=wlan0
OCB_WLAN=ocb0
DIR_IP=192.168.1.2
MASK=255.255.255.0
GW=192.168.1.1

#Setting down the wireless interface
ip link set $WLAN down
#Adding a new interface in OCB mode
iw dev $WLAN interface add $OCB_WLAN type ocb
#Setting OCB mode
iw dev $OCB_WLAN set type ocb
#Setting down OCB interface
ip link set $OCB_WLAN down
#Setting wireless regulatory domain to AA
iw reg set AA
#Is it change?
iw reg get
#Raising OCB interface
ip link set $OCB_WLAN up
#Joining interface to Control Channel and 10MHz of bandwidth
iw dev $OCB_WLAN ocb join 5990 10MHZ
#Leaving the Control Channel
iw dev $OCB_WLAN ocb leave
#Joining interface to Service Control Channel 1
iw dev $OCB_WLAN ocb join 5880 10MHZ
#Was it successfull?
iw dev | iwconfig


#Settign up IP adress, netmask and default gateway
ifconfig $OCB_WLAN $DIR_IP netmask $MASK
route add default gw $GW
```

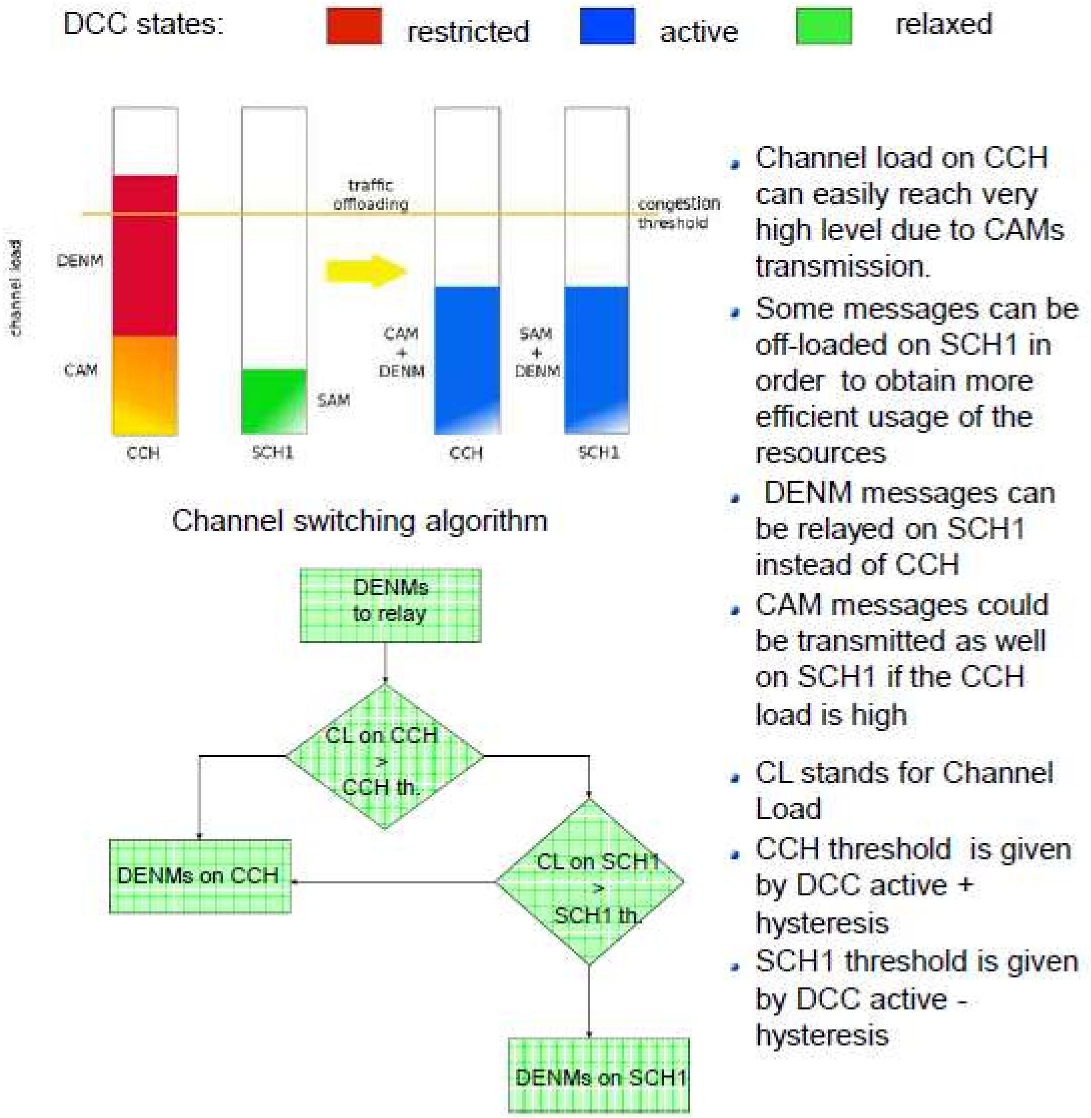
Appendix figure 12. Configuration of OCB interface

```
#/bin/bash
#testing commands for IP layer

WLAN_OCB=ocb0
IP_ADDRESS=192.168.1.2
OUT_FILE=out.pcap

#Ping through a specific interface
ping -I $WLAN_OCB $IP_ADDRESS
#       -I: Interface address
#       -c: Count number of packets
#       -b: Allow pingining a broadcast address
#       -D: Print timestamp before each line

#Capture traffic
tcpdump -i $WLAN_OCB -v -XX -w OUT_FILE
#       -i: Interface address
#       -v: Verbose mode
#       -XX: Print the headers and the data of each packet, in hex and
#            ASCII. The output file *.pcap can be readed with Wireshark
#            program
#       -w: Write the raw packets to file
#       -c: Exit after receiving count packets
```

Appendix figure 13. Example of sending data through IP layer

C. Appendix

The next picture has been taken from the document "Multi-Channel Congestion Control for ETSI ITS G5 A/B".



- Channel load on CCH can easily reach very high level due to CAMs transmission.
- Some messages can be off-loaded on SCH1 in order to obtain more efficient usage of the resources
- DENM messages can be relayed on SCH1 instead of CCH
- CAM messages could be transmitted as well on SCH1 if the CCH load is high
- CL stands for Channel Load
- CCH threshold is given by DCC active + hysteresis
- SCH1 threshold is given by DCC active - hysteresis

Appendix figure 14. Multi-channel flow chart on Control Channel