

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**

### **EVALUACIÓN DEL DESEMPEÑO DE UNA HETNET DE CARAVANAS DE VEHÍCULOS EN UN SECTOR DEL CENTRO HISTÓRICO DE QUITO**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

**JEFFERSON EDUARDO ROBLES CASA**

**DIRECTOR: PhD. MARTHA CECILIA PAREDES PAREDES**

**Quito, marzo 2022**

# **AVAL**

Certifico que el presente trabajo fue desarrollado por Jefferson Eduardo Robles Casa, bajo mi supervisión.

**PhD. Martha Cecilia Paredes Paredes.  
DIRECTOR DEL TRABAJO DE TITULACIÓN**

## **DECLARACIÓN DE AUTORÍA**

Yo, Jefferson Eduardo Robles Casa, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

JEFFERSON EDUARDO ROBLES CASA.

## DEDICATORIA

Dedico este trabajo primero a Dios por haberme permitido llegar tan lejos a pesar de los inconvenientes y duros momentos que he vivido últimamente.

A mi madre, Martha Blanca Robles Casa, que a pesar de ser madre soltera tuvo el coraje y fuerza de velar y sacar adelante a sus dos hijos, esto es especialmente para ella.

También dedico este trabajo al ingeniero Galo Pico Ruiz, quien, a pesar de no ser parte de mi familia, lo siento como un padre y agradezco infinitamente por haberme brindado su apoyo desde mi niñez.

A mi hermano, Marco Giovanni Robles Casa, por sus palabras de aliento cuando pretendía rendirme y gracias a él continúe esforzándome y sin rendirme a pesar de las circunstancias.

A todas esas personas a las que puedo decir que son mis amigos, tanto de la universidad, colegio y a aquellas que conocí hace poco y que me han brindado su apoyo y amistad incluso aunque no se encuentren en el país. También a aquella persona que a pesar de que la conozco desde hace unos pocos meses, me ha brindado la amistad más sincera que he tenido en mi vida.

Finalmente dedico esto a mis dos sobrinos, Jean Marcos y Mateo Nicolas, y quiero decirles que todo es posible con constancia y esfuerzo, espero ser un ejemplo para ellos y que algún día también puedan estar en mi lugar y sentir la satisfacción de estar tan cerca de lograr sus metas.

## **AGRADECIMIENTO**

Agradezco a todos los profesores que he tenido en mi carrera universitaria en la Escuela Politécnica Nacional, gracias a ellos pude adquirir conocimientos científicos y de vida valiosos que me acompañaran a lo largo de mi vida.

Un especial agradecimiento a la PhD. Martha Cecilia Paredes a quien conozco desde tercer semestre cuando recibí la materia de software de simulación y también tuve el honor de recibir cátedra de las materias de comunicaciones inalámbricas y redes celulares en los últimos semestres de la carrera. Muchas gracias doctora por confiar en mí y brindarme su apoyo para realizar este trabajo de titulación.

Gracias también a el PhD. Luis Felipe Urquiza y al MSc. Pablo Barbecho quienes me dieron pautas y me ayudaron a desarrollar este trabajo de titulación.

Finalmente, gracias a la Escuela Politécnica Nacional por brindarme estos años que atesorare en mi mente.

# ÍNDICE DE CONTENIDO

AVAL	1
DECLARACIÓN DE AUTORÍA	2
DEDICATORIA	3
AGRADECIMIENTO	4
ÍNDICE DE CONTENIDO	5
RESUMEN	8
ABSTRACT	9
1 INTRODUCCIÓN	1
1.1 OBJETIVOS	2
1.2 ALCANCE	2
1.3 MARCO TEÓRICO	4
1.3.1 REDES VEHICULARES O VANET	4
1.3.1.1 Arquitectura VANET	6
1.3.1.2 Estandarización de redes vehiculares	7
1.3.1.3 Asignación de espectro DSRC	8
1.3.1.4 IEEE 802.11p	9
1.3.1.5 Caravanas de vehículos ( <i>Platooning</i> )	10
1.3.2 LTE ( <i>LONG TERM EVOLUTION</i> )	11
1.3.2.1 E-UTRAN ( <i>Evolved UTRAN</i> )	12
1.3.2.2 EPC ( <i>Evolver Packet Core</i> )	15
1.3.3 INTERACCION IEEE 802.11p CON LTE	17
1.3.3.1 HETNET Vehicular	17
1.3.3.1.1 Acceso a la red LTE-A para IEEE 802.11p	18
1.3.3.1.2 Backhaul de SC LTE-A	18
1.3.3.1.3 Selección de Car Header móvil para comunicaciones V2I.	19
2 METODOLOGÍA	21
2.1 SIMULADORES	21
2.1.1 OMNET++	21
2.1.2 SUMO	22
2.1.3 OSM	22
2.1.4 VEINS	23
2.2 INSTALACIÓN DE SOFTWARE	23

2.2.1 UBUNTU	23
2.2.2 OMNET++	27
2.2.3 SUMO	28
2.2.4 LIBRERÍAS Y REPOSITORIOS EN OMNET++	29
2.3 DISEÑO	34
2.3.1 SELECCIÓN DE ÁREA PARA IMPLEMENTAR LA HETNET	35
2.3.2 GENERACIÓN DE MAPA	36
2.3.3 INSERCIÓN DE FLUJO VEHICULAR	38
2.3.4 SELECCIÓN DE RUTAS	40
2.3.5 DISEÑO E IMPLEMENTACIÓN DE HETNET	42
2.3.6 CONFIGURACIÓN DE LA HETNET	44
2.3.7 DISEÑO DE ESCENARIOS	47
2.3.8 VINCULACIÓN DE SUMO CON OMNET++	54
3 RESULTADOS Y DISCUSIÓN	57
3.1 PROCESAMIENTO DE ESCENARIOS	58
3.2 ANÁLISIS DE RESULTADOS	68
3.2.1 PRIMER ESCENARIO	68
3.2.1.1 Paquetes WLAN transmitidos vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p	68
3.2.1.2 Paquetes WLAN recibidos vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p	69
3.2.1.3 <i>Packet Delivery Ratio</i> WLAN vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p	70
3.2.1.4 <i>Delay</i> WLAN vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p	71
3.2.1.5 Paquetes LTE transmitidos vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p	72
3.2.1.6 Paquetes LTE recibidos vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p	73
3.2.1.7 <i>Packet Delivery Ratio</i> LTE vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p	74
3.2.1.8 <i>Delay</i> LTE vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p	75
3.2.2 SEGUNDO ESCENARIO	75
3.2.2.1 Paquetes WLAN transmitidos vs Intervalo de envío de mensajes para el estándar IEEE 802.11p	75
3.2.2.2 Paquetes WLAN recibidos vs Intervalo de envío de mensajes para el estándar IEEE 802.11p	76

3.2.2.3 <i>Packet Delivery Ratio</i> WLAN vs Intervalo de envío de mensajes para el estándar IEEE 802.11p	77
3.2.2.4 <i>Delay</i> WLAN vs Intervalo de envío de mensajes para el estándar IEEE 802.11p	77
3.2.2.5 Paquetes LTE transmitidos vs Intervalo de envío de mensajes para el estándar IEEE 802.11p	78
3.2.2.6 Paquetes LTE recibidos vs Intervalo de envío de mensajes para el estándar IEEE 802.11p	79
3.2.2.7 <i>Packet Delivery Ratio</i> LTE vs Intervalo de envío de mensajes para el estándar IEEE 802.11p	79
3.2.2.8 <i>Delay</i> LTE vs Intervalo de envío de mensajes para el estándar IEEE 802.11p	80
4 CONCLUSIONES Y RECOMENDACIONES	81
4.1 CONCLUSIONES	81
4.2 RECOMENDACIONES	82
5 REFERENCIAS BIBLIOGRÁFICAS	83
ANEXOS	87



## RESUMEN

En este trabajo de titulación se evalúa el desempeño de una red HETNET (*Heterogeneous Network*) formada por caravanas de vehículos que se desplazan en un sector del centro histórico de Quito. La red HETNET está formada por una red celular LTE (*Long Term Evolution*) y una red vehicular VANET (*Vehicular Adhoc Network*) de corto alcance.

Para llevar a cabo este proyecto, se revisan conceptos teóricos de forma breve de HETNET, VANET, caravanas de vehículos y algunos otros conceptos asociados con las comunicaciones inalámbricas, los cuales se encuentran descritos en el primer capítulo del presente trabajo.

En el capítulo 2 se revisan las características de los simuladores que ayudarán a implementar la HETNET. También se describe cómo se instalan los simuladores, cómo se implementó la red en un sector del centro histórico de Quito. Y finalmente, se establecen escenarios para evaluar el rendimiento de la HETNET de caravanas de vehículos dentro de un área del centro histórico de Quito. Para los escenarios se consideran características físicas y técnicas distintas. En el primer escenario se consideran dos parámetros variables que son la potencia de transmisión de los vehículos y la posición del CH (*Car Header*) dentro de la caravana de vehículos. El segundo escenario tiene como parámetros variables el intervalo de envío de mensajes IEEE 802.11p y la cantidad de vehículos que pertenecen a cada caravana de vehículos.

Finalmente, se analizan los resultados en donde se distingue que el número de automóviles dentro de la simulación, la potencia de transmisión del estándar IEEE 802.11p y el intervalo de envío de mensajes entre vehículos tienen un impacto considerable en el rendimiento de la red HETNET en función del número de paquetes enviados, número de paquetes recibidos, PDR (*Packet Delivery Ratio*) y el *delay* de los paquetes enviados tanto para el estándar IEEE 802.11p como para la red LTE.

**PALABRAS CLAVE:** HETNET, PLATOONING, IEEE 802.11p, LTE, Caravana de vehículos, Centro histórico de Quito.

## **ABSTRACT**

This degree work evaluates the performance of a HETNET (Heterogeneous Network) formed by caravans of vehicles that move in a sector of the historical center of Quito. The HETNET network is made up of an LTE (Long Term Evolution) cellular network and a short-range VANET (Vehicular Adhoc Network) vehicular network.

To carry out this project, theoretical concepts of HETNET, VANET, vehicle caravans and some other concepts associated with wireless communications are briefly reviewed, which are described in the first chapter of this work.

Chapter 2 reviews the simulator features that will help implement HETNET. It also describes how the simulators are installed, how the network was implemented in a sector of the historic center of Quito. And finally, scenarios are established to evaluate the performance of the HETNET of vehicle caravans within an area of the historic center of Quito. Different physical and technical characteristics are considered for the scenarios. In the first scenario, two variable parameters are considered, which are the transmission power of the vehicles and the position of the CH (Car Header) within the vehicle caravan. The second scenario has as variable parameters the interval for sending IEEE 802.11p messages and the number of vehicles that belong to each vehicle caravan.

Finally, the results are analyzed where it is distinguished that the number of cars within the simulation, the transmission power of the IEEE 802.11p standard, and the interval for sending messages between vehicles have a considerable impact on the performance of the HETNET network depending on of the number of packets sent, number of packets received, PDR (Packet Delivery Ratio) and the delay of the packets sent for both the IEEE 802.11p standard and for the LTE network.

**KEYWORDS:** HETNET, PLATOONING, IEEE 802.11p, LTE, Quito's Historic center.

# 1 INTRODUCCIÓN

La movilidad en el centro histórico de Quito se ha visto mermada cada vez más con el paso de los años, producto del incremento del parque automotor de la ciudad [1], además de la estrechez de las calles de la zona céntrica.

Para mitigar este efecto, se puede considerar como alternativa, utilizar automóviles inteligentes que puedan mantener una conducción controlada y segura para transeúntes y conductores.

Generalmente este tipo de automóviles utilizan redes vehiculares, en donde los vehículos se comunican entre ellos por medio del estándar IEEE 802.11p, pero es necesario desplegar un tipo de infraestructura denominada RSU (*Road-Side Unit*) que utiliza el estándar IEEE 802.11p para interactuar con los vehículos circundantes [2], [3].

Los RSUs básicamente sirven para enviar información de los vehículos que se encuentren cerca de la infraestructura hacia otros RSUs que se encuentren distantes [2] y, con ello también, establecer una comunicación con otros vehículos que se encuentren lejos para que estos, a su vez, por ejemplo, puedan determinar el mejor camino que deberían seguir para evitar aglomeraciones de tráfico.

Desplegar RSUs conlleva un costo extra para aquellas entidades que deseen brindar el servicio con vehículos inteligentes por lo que se propone utilizar, como alternativa, una infraestructura que ya se encuentre instalada en el centro histórico de Quito.

La red celular LTE (*Long Term Evolution*) se puede utilizar en lugar de los RSUs para la comunicación entre vehículos de larga distancia, pero para la comunicación a corta distancia se podría utilizar el estándar IEEE 802.11p [4]. La combinación de los estándares IEEE 802.11p y LTE forma una HETNET (*Heterogeneous Network*) que debe cumplir ciertas condiciones para no saturar la red LTE [4], [5].

Si todos los vehículos envían paquetes hacia los *eNodeBs*, pueden saturar la red causando problemas a los UEs (*User Equipment*) de los usuarios que se encuentren dentro del rango de cobertura en la que operan junto con la red vehicular; para mitigar esto, en la literatura se propone que exista un CH (*Car Header*) que se encargue de recolectar la información de los vehículos circundantes y reenvíe esos datos al *eNodeB* [4] - [6].

También existe un tipo de tecnología para automóviles inteligentes que se denomina tren de carga o caravanas de vehículos (*Platooning*), en donde la velocidad y el frenado de

cada automóvil se encuentra sincronizado dado que comparten datos a través de redes vehiculares [7], [8].

En este trabajo, se propone simular una red HETNET para una caravana de vehículos (*platooning*) y evaluar el desempeño de esta red usando como escenario un sector del centro histórico de Quito. Los resultados para dichas evaluaciones se presentan en términos de paquetes IEEE 802.11p transmitidos, paquetes IEEE 802.11p recibidos, eficiencia o PDR (*Packet Delivery Ratio*) IEEE 802.11p, el retardo o *Delay* IEEE 802.11p, paquetes LTE transmitidos, paquetes LTE recibidos, eficiencia o PDR LTE y el retardo o *Delay* LTE.

## 1.1 OBJETIVOS

El objetivo general de este Proyecto Técnico es evaluar el desempeño de una HETNET de caravanas de vehículos que se desplazan en un sector del centro histórico de Quito.

Los objetivos específicos del Proyecto Técnico son:

- ✓ Describir una red HETNET vehicular, los estándares IEEE 802.11p, y LTE y caravanas de vehículos.
- ✓ Diseñar la HETNET para caravanas de vehículos para un sector del centro histórico de Quito.
- ✓ Implementar una simulación de la HETNET de caravanas de vehículos dentro de un sector del centro histórico de Quito.
- ✓ Evaluar el desempeño de la HETNET de caravanas de vehículos mediante el análisis de gráficas estadísticas que se obtendrán luego de simular dos escenarios.

## 1.2 ALCANCE

En la fase teórica se revisan, de forma concisa, los conceptos asociados a este trabajo de titulación. Se empieza describiendo una HETNET a fin de entender su funcionamiento, además, es oportuno realizar un breve vistazo a cada estándar que integran la HETNET, el IEEE 802.11p [7] y LTE [9]. Adicionalmente, se explica qué es una caravana de vehículos (*Platooning*) o tren de carretera, cómo se conforman y qué tipos de caravanas existen según sus características.

En la fase de diseño e implementación se establece un sector del centro histórico de la ciudad de Quito sobre el cual se evalúa el comportamiento de la HETNET de caravanas de vehículos. Para generar los mapas, sobre el cual se implementa la HETNET de caravanas de vehículos se utilizan las herramientas OSM (*Open Street Maps*) y JOSM (*Java Open Street Maps Editor*).

Por otro lado, se usa SUMO (*Simulation of Urban Mobility*) para simular el tráfico vehicular. La carga vehicular creada por SUMO será aleatoria por lo que se seleccionan cinco rutas distintas que tengan puntos de salida y llegada distintos con el objetivo de que las caravanas traten de no encontrarse dentro del mapa.

Las comunicaciones inalámbricas corren por cuenta de OMNET++, en donde se integran dos (2) librerías esenciales. La primera es INET Framework que se encarga del modelamiento del estándar IEEE 802.11p. La segunda librería es SimuLTE (*Simulator LTE*) que posibilita la creación de redes LTE.

En la fase de diseño e implementación también, se definen dos escenarios con las siguientes características:

- El primer escenario considera dos parámetros variables que serán la potencia de transmisión de los vehículos y la posición del CH dentro de la caravana de vehículos.
- En el segundo escenario será variable el intervalo de envío de mensajes IEEE 802.11p y a la cantidad de vehículos que pertenecen a cada caravana de vehículos.

Estos escenarios son importantes en una HETNET de caravanas de vehículos debido a que definen la distancia máxima para la comunicación de los vehículos, la cantidad de paquetes que circulan por la red, el tiempo de arribo de los paquetes al CH y la cobertura que el CH tendrá dentro de la caravana de vehículos.

En la fase de simulación y presentación de resultados se integra una librería más a OMNET++ que se denomina VEINS (*Vehicles in Network Simulation*), la cual se encarga de vincular la simulación del tráfico vehicular que se genera en SUMO, con la simulación de las redes inalámbricas, que se generan en OMNET++.

Adicional, en la fase de simulación y presentación de resultados se evalúa el desempeño de la HETNET de caravanas de vehículos, en un sector de la ciudad de Quito mediante gráficas estadísticas para cada uno de los escenarios propuestos.

Las gráficas del primer escenario están en función de Paquetes transmitidos vs Potencia de IEEE 802.11p, Paquetes recibidos vs Potencia de IEEE 802.11p, PDR (*Packet Delivery Ratio*) vs Potencia de IEEE 802.11p y *Delay* vs Potencia de IEEE 802.11p.

En el segundo escenario, las gráficas se encuentran en función de Paquetes transmitidos vs Intervalo de envío de mensajes, Paquetes recibidos vs Intervalo de envío de mensajes, PDR vs Intervalo de envío de mensajes y *Delay* vs Intervalo de envío de mensajes.

Una vez se tengan las gráficas se procede a analizar las similitudes y diferencias que se presentan con las simulaciones generadas para cada escenario.

Todo este trabajo de titulación se realiza mediante simulaciones por lo que no tiene producto final demostrable.

## **1.3 MARCO TEÓRICO**

En el marco teórico de este trabajo de titulación se revisan, las redes vehiculares en donde también se detallan características de las caravanas de vehículos (Platooning), luego se resumen las características de las redes celulares (LTE) y finalmente se explica como las redes mencionadas anteriormente se aplican en las caravanas de vehículos para formar una red HETNET vehicular.

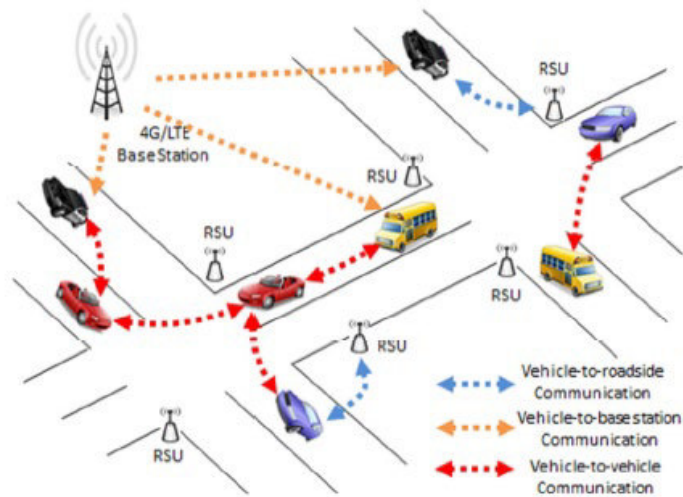
### **1.3.1 REDES VEHICULARES O VANET**

Las redes vehiculares o también conocidas como VANET (*Vehicular Ad-hoc Network*) son redes inalámbricas orientadas a la comunicación entre vehículos. Los automóviles pasan a ser los nodos en una VANET los cuales se comunican entre ellos sin que exista ninguna infraestructura existente [10].

Una VANET se utiliza para informar de cualquier evento que se encuentre en la carretera o las condiciones del clima, pero algunas funcionalidades que se han desarrollado últimamente con respecto a las VANETs son los mensajes acerca de posición, velocidad entre autos para que sea sustentable el manejo asistido, o de plano que el vehículo se maneje por su cuenta sin la ayuda del ser humano [6].

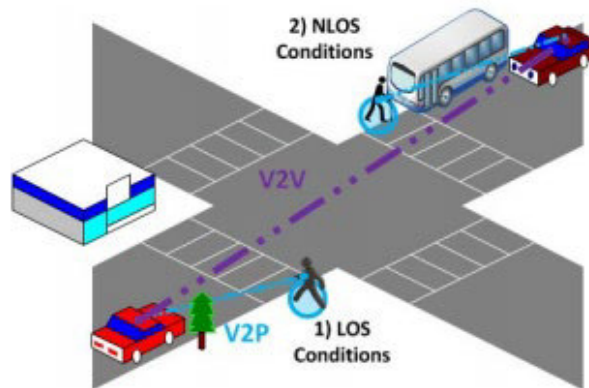
Hoy en día, las VANETs tienen algunas configuraciones las cuales se denominan generalmente como V2X (*Vehicule to Everything*) donde la X puede ser otro tipo de objeto, como otro vehículo generando una comunicación V2V (*Vehicule to Vehicule*). También, puede ser alguna infraestructura llamada V2I (*Vehicule to Infrastructure*) como una comunicación celular LTE (*Long Term Evolution*) o con un RSU (*Road Side Unit*), que es un nodo fijo similar a una antena que permite el intercambio de información de un tramo lejano de la vía [11]. Los vehículos pueden interactuar con los peatones al crear una comunicación V2P (*Vehicule to Pedestrian*) mediante el intercambio de información de los teléfonos móviles de los transeúntes con los vehículos que se encuentren cerca.

En la Figura 1.1 se visualizan algunos de los distintos tipos de comunicaciones que existe en una VANET, en la figura se aprecia la comunicación entre vehículos y la comunicación entre vehículos e infraestructuras.



**Figura 1.1.** Configuraciones de VANET [11].

Para la configuración V2P, los automóviles circundantes se comunican entre ellos y también con el peatón y en el caso de que un vehículo no tenga línea de vista directa con el peatón, otro vehículo que si tenga LOS (*Line Of Sighth*) avisará a todos los nodos cercanos de que existe un transeúnte cerca y que así se pueda evitar accidentes [12], como se observa en la Figura 1.2.



**Figura 1.2.** Configuración V2P [12].

En un escenario real, como una carretera de cualquier ciudad, los autos que por dicha carretera circulen, tienden a moverse y a cambiar su velocidad de forma aleatoria por lo que la topología de la VANET cambia constantemente. La topología cambiante que se tiene en las VANETs, genera desconexiones indeseadas debido a que los autos pierden cobertura entre sí, además que, distintos objetos circundantes como semáforos, construcciones, autos que no tienen la capacidad de conectarse a una VANET, etc. generan deterioro de la señal con la que los vehículos se comunican entre ellos [13].

### 1.3.1.1 Arquitectura VANET

Dentro de una VANET se tienen 3 principales componentes que se detallan a continuación:

- AU (*Application Unit*). - es el responsable de interactuar con los sensores y componentes con los que cuenta el vehículo, como por ejemplo el GPS (*Global Positioning System*), sensores de velocidad y frenado, etc. [3] y se encuentra dentro del nodo móvil que es el vehículo .
- OBU (*On Board Unit*). - es la unidad que se encarga de establecer la comunicación inalámbrica entre autos y se encuentra en el nodo móvil. El OBU envía la información recolectada por el AU como la velocidad del móvil y posición GPS hacia otros nodos móviles vecinos para saber cómo es la forma de la topología y como esta cambia [3].
- RSU (*Road Side Unit*). - es un nodo fijo que se coloca en un punto estratégico de la carretera y tiene como objetivo ampliar el rango de cobertura de la red. Gracias al RSU, los nodos móviles obtienen información de eventos en lugares distantes y con ello deciden cuál es la mejor ruta que se debe tomar [3]

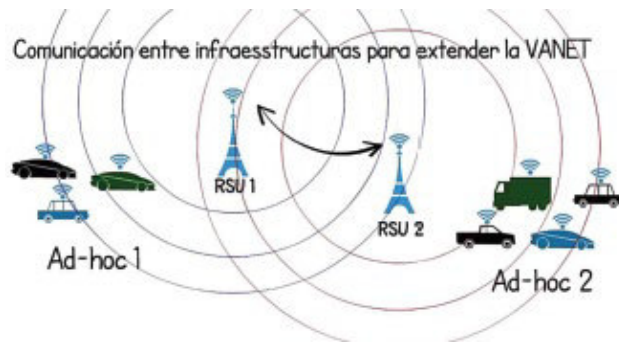
En la Figura 1.3 se muestra un diagrama de cómo se compone un nodo móvil, en donde se encuentra tanto el AU y el OBU. Estos dos módulos interactúan entre sí ya que el AU recolecta información de los sensores del vehículo y el OBU envía esa información hacia otros vehículos.



**Figura 1.3.** Componentes del nodo móvil [3].

En la Figura 1.4 se puede observar la comunicación V2V entre los vehículos y V2I con el RSU. En este escenario se observa que cada automóvil comparte su información mediante el OBU, tanto a los vehículos vecinos como a los RSU circundantes. Los RSU se comunican entre ellos para compartir información de los nodos móviles que se encuentran dentro de su rango de alcance. Con la información compartida simultáneamente, los vehículos deciden qué ruta tomar.





**Figura 1.4.** Comunicación entre nodos móviles y fijos [3].

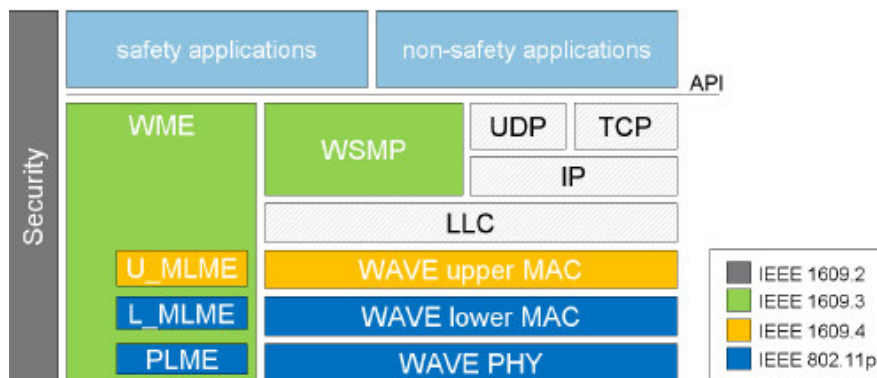
### 1.3.1.2 Estandarización de redes vehiculares

DSRC (*Dedicated Short-Range Communication*) se crea con el objetivo de que se desarrollen aplicaciones de seguridad pública las cuales permitan bajar el índice de accidentes de tránsito además de mejorar la movilidad [14].

WAVE nace cuando en Estados Unidos al estandarizar DSRC mediante la ASTM (*American Society for Testing and Materials*) 2313; sin embargo, en el 2004 el proyecto para estandarizar DSRC paso a un grupo que desarrollaba en ese tiempo el estándar IEEE 802.11 [14].

El cambio de grupo de investigación se produjo porque la tecnología de radio DSRC es similar a la de IEEE 802.11a pero difiere ligeramente en algunos aspectos, por lo que a DSRC se lo nombra IEEE 802.11p WAVE (*Wireless Access in Vehicular Environments*). IEEE 802.11p tiene implícito muchas de las características del estándar general IEEE 802.11 [14].

Como se muestra en la Figura 1.5 [14], WAVE es solo una parte del conjunto de estándares relacionados a todas las capas de protocolos de operación para DSRC. El estándar IEEE 802.11p está limitado por el alcance que tiene IEEE 802.11, que es estrictamente un estándar a nivel de capa MAC (*Medium Access Control*) y capa física. Todo lo relacionado con DSRC para las capas superiores está a cargo de los estándares IEEE 1609. El estándar IEEE 1609.3 contiene la configuración y administración de la conexión WAVE. El estándar IEEE 1609.4 está por arriba del IEEE 802.11p y este permite la multifunción de capas a través de varios canales, sin tener que conocer los parámetros físicos [14].

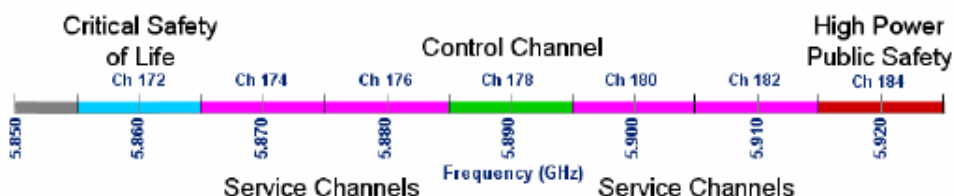


**Figura 1.5.** Conjunto de estándares DSRC y de comunicaciones [14].

### 1.3.1.3 Asignación de espectro DSRC

En Estados Unidos, la FCC (*Federal Communication Commission*) asignó en 1999 un ancho de banda de 75 MHz para que se utilice en las aplicaciones de DSRC y además define que la portadora sea de 5.9 GHz y pasaría a ser una frecuencia exclusiva para las VANETs.

En la Figura 1.6 se muestra que el espectro para DSRC se encuentra dividido en 7 canales de 10 MHz cada uno en donde el canal 178 se reserva solo para comunicaciones de seguridad. Los dos últimos canales al final del espectro se reservan para usos especiales. Los demás canales disponibles pueden ser utilizados para uso de aplicaciones de seguridad o multimedia [14], [15].



**Figura 1.6.** Espectro para DSRC en Estados Unidos [14].

A pesar de que la banda o frecuencia que utiliza DSRC es libre, esta se encuentra licenciada para el uso de comunicaciones vehiculares. A diferencia de las bandas de 900 MHz, 2.4 GHz y 5 GHz en donde se incrementa el acoplamiento de nuevos dispositivos sin restricción, en la banda de 5.9 GHz esto está más controlado por la FCC ya que ciertos canales se encuentran bloqueados y todos los dispositivos tienen que registrarse a un mismo estándar por lo que es imposible desarrollar una tecnología de radio diferente en esta frecuencia [14], [15].

En otras partes del mundo también se buscaba tener algo similar a DSRC, sin embargo, se rigen a sus propios estándares. En Europa se tiene un espectro de 30 MHz en la banda de 5 GHz para las redes vehiculares [14].

#### **1.3.1.4 IEEE 802.11p**

Como se observa en la Figura 1.5, DSRC se conforma por algunos estándares entre los cuales se encuentra el IEEE 802.11p.

IEEE 802.11p especifica la capa física y la capa MAC de DSRC y tiene mucha similitud con el estándar IEEE 802.11; sin embargo, por la topología cambiante de los vehículos ya que se encuentran en constante movimiento, se modificó el estándar IEEE 802.11 [16].

##### ❖ Capa Física.

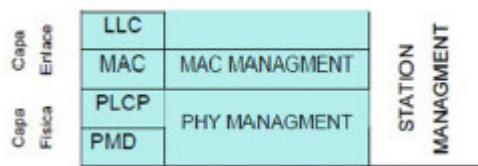
El estándar IEEE 802.11p tiene varias similitudes con el estándar IEEE 802.11 por lo que la capa física es la misma [17].

Los cambios se detallan a continuación:

- Reducción de tamaño de paquetes al momento de establecer la comunicación por el poco tiempo que tienen los vehículos de interactuar entre sí [16].
- Se incorpora el WBSS (*Wave Basic Service Set*) el cual posee un identificador fijo para transmitir tramas bajo demanda. Las tramas poseen información primordial para lograr la comunicación sin que sea necesario autenticarse [16].
- Utiliza OFDM (*Orthogonal Frequency-Division Multiplexing*) [16].
- El ancho de banda es de 10 MHz [16].
- Se divide el número de canales a 7 que se encuentran desde el canal 172 al 184 [16].

También es necesario explicar las características de la capa física y la capa MAC [16].

La capa física se encarga de los aspectos de transmisión de paquetes inalámbricamente. Se compone de 3 subcapas las cuales son PLCP (*Physical Layer Convergence Procedure*) y PMD (*Physical Medium Dependent*) [17]. Las capas y la subcapa se pueden apreciar en la Figura 1.7.



**Figura 1.7.** Arquitectura Capa Física IEEE 802.11 [17].

PLCP brinda convergencia para el cambio de la PDU MAC a la subcapa PMD, además provee un algoritmo de encapsulamiento idóneo de la PDU MAC para ser transmitida y receptada por medio de la interfaz inalámbrica.

En PMD se define codificación convolucional con tasas de  $\frac{1}{2}$ ,  $\frac{2}{3}$  y  $\frac{3}{4}$  además de las modulaciones BPSK, QPSK, 16-QAM y 64-QAM, también se define la técnica de transmisión OFDM.

❖ Subcapa MAC.

Esta subcapa se encarga de brindar mecanismos de seguridad y de acceso al medio, además de definir los tipos de tramas [18].

También se divide en 2 subcapas la cuales son la MAC y de administración [18].

✓ MAC.

Aquí se manejan 3 tipos de tramas entre las cuales están las tramas de control, tramas de datos y tramas de gestión [18]. En esta subcapa también se maneja mecanismos de acceso al medio entre los cuales están DCF (*Distribution Coordination Function*), PCF (*Point Coordination Function*), HCF (*Hybrid Coordination Function*) y MCF (*Mesh Coordination Function*).

✓ Administración.

Esta subcapa se compone de tres partes entre las cuales esta SME (*Station Management Entity*), PLME (*Physic Layer Management Entity*) y MLME (*MAC Layer Management Entity*) [18].

### 1.3.1.5 Caravanas de vehículos (*Platooning*)

Las caravanas de vehículos se definen como un conjunto de automóviles que viajan juntos de manera activa y coordinada con una formación fija [8]. Una caravana de vehículos se utiliza para mejorar la eficiencia o movilidad vehicular además de brindar seguridad y comodidad al conductor.

Varias organizaciones alrededor del mundo han propuesto diferentes formas de implementar caravanas de vehículos (*Platooning*) entre las cuales se encuentran SARTRE (*Safe Road Trains for the Environment*) [19] desarrollado en Europa, PATH (*Partners for Advanced Transportation Technology*) [20] desarrollado en California, GCDC (*Grand Cooperative Driving Challenge*) [21], caravana de vehículos (*Platooning*) desarrollado por SCANIA que es una concesionaria de camiones y Energy ITS que es un proyecto japonés para caravana de camiones (*Platooning*) [7].

Existen algunas variaciones de caravanas de vehículos (*Platooning*) que se definen según las metas que se tienen al utilizar este tipo de caravanas, la forma de implementar, la combinación de vehículos, la infraestructura y como se encuentra automatizado.

Según los objetivos de las caravanas, por ejemplo, SARTRE, PATH y *Energy ITS* sugieren un control longitudinal y lateral de los automóviles o camiones en la autopista, mientras que GCDC y SCANIA proponen algo distinto ya que el control automatizado solo se cumple de forma longitudinal [7].

Según el tipo de auto que forma la caravana de vehículos (*Platooning*), SARTRE y GCDC pretenden que distintos vehículos entre camiones y automóviles pequeños conformen la caravana. Por otro lado, PATH, SCANIA y *Energy ITS* argumentan que la conformación de la caravana de vehículos (*Platooning*) debe ser homogéneo y solo se conforma por un único tipo de vehículo [7].

SARTRE y SCANIA deciden no realizar cambios a la infraestructura de una VANET, mientras PATH plantea que existan vías dedicadas con marcadores de referencia instalados en la superficie de la carretera. Finalmente, GCDC dice que utilizará posicionamiento de alta precisión y *Energy ITS* empleará marcas de carril [7].

### **1.3.2 LTE (*LONG TERM EVOLUTION*)**

A pesar de que en la HETNET se utiliza LTE-A, se revisa todo lo concerniente a LTE ya que la arquitectura es la misma.

En el año 2009, el 3GPP (*The 3rd Generation Partnership Project*), grupo encargado de lanzar las especificaciones para comunicaciones móviles, publicó el Release 9 en el cual, dio a conocer las características de la red LTE (*Long Term Evolution*). Con el paso de los años, el 3GPP publicó el *Release 10* conocido como LTE-Advanced en el 2011 y el *Release 13* en el 2016 conocido como LTE-Advanced Pro [22]. El último *Release* publicado acerca de LTE es el número 14 lanzado en el 2017 en el cual se especifican algunas mejoras a LTE-Advanced Pro además de brindar un cimiento para la nueva tecnología de

comunicaciones móviles de quinta generación y también describe el soporte de LTE con los servicios V2x utilizados en VANETs [23].

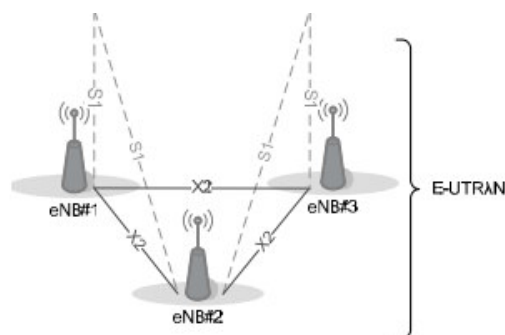
En las especificaciones de LTE se detalla que tanto el *Core*, llamada EPC (*Evolved Packet Core*), como la parte de Radio, llamada E-UTRAN (*Evolved-UTRAN*) evolucionaron en comparación con el sistema de comunicación celular de tercera generación.

Estas mejoras establecidas para LTE tanto en la parte de *Core*, como en la parte de Radio permiten tener más cobertura, velocidades de transmisión de datos altas, latencia reducida, tener soporte MIMO (*Multiple Input Multiple Output*), etc. LTE usa como técnica de transmisión OFDM (*Orthogonal Frequency Division Multiplexing*), con ancho de banda de 1.4, 3, 5, 10, 15 y 20 MHz y logra llegar a los 20 MHz. Gracias a OFDM se consigue mejorar benignamente los efectos del desvanecimiento multitrayecto mediante el uso de más portadoras para que la velocidad de transmisión aumente.

Otra de las características importantes que surge para LTE es que, ahora todo lo que se envía tiene que ser IP (*Internet Protocol*), a diferencia de tecnologías de comunicación celulares anteriores en donde prevalecía la conmutación de circuitos, conmutación de paquetes o un híbrido entre las dos. En los inicios de LTE, la voz no se podía enviar como paquetes IP por lo que se optó por utilizar CSFB (*Circuit-Switched Fallback*) que consiste en pasar a una red más accesible como 2G o 3G para entregar voz y SMS (*Short Message Services*). En el 2014 gracias al desarrollo de la tecnología LTE se envía voz sobre está dando paso a VoLTE (*Voice over LTE*) y con ello CSFB se descarta para enviar voz.

### 1.3.2.1 E-UTRAN (*Evolved UTRAN*)

Generalmente la E-UTRAN consta de un nodo (estación base) llamada eNode-B, el cual se conecta con otras estaciones base eNode-B por medio de una interfaz X2, mientras que para conectarse con el Core o EPC, el eNodeB utiliza la interfaz S1 [24], [25]. En la Figura 1.8 se observa la arquitectura de la E-UTRAN.



**Figura 1.8.** Arquitectura E-UTRAN [25].

A continuación, se explica los componentes de la E-UTRAN.

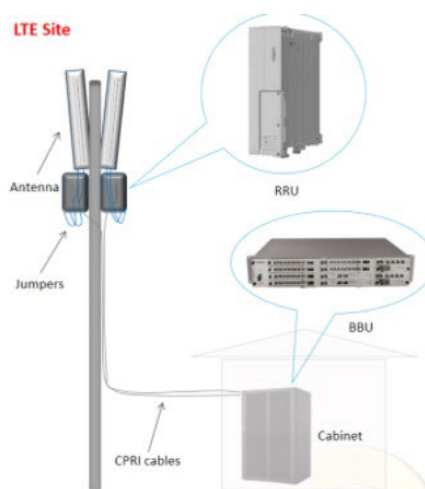
❖ eNode-B

El eNode-B es el elemento más complejo que tiene la E-UTRAN y toda la red LTE por lo que viene a ser la evolución del NodeB que es la estación base de UMTS (*Universal Mobile Telecommunication System*).

Dentro del eNode-B se tiene los componentes que se listan a continuación y que así mismo se muestran en la Figura 1.9.

- BBU (*Base Band Unit*), es el encargado de administrar la estación base, además de que supervisa algunos procesos como el de datos en uplink y downlink, señalización, operación y mantenimiento [24].
- RRU (*Remote Radio Unit*), tiene las funciones de transmisión y recepción señales banda base y también modula y demodula señales RF. Funciona como un amplificador de señal [24].
- CPRI (*Common Public Radio Interface*), permite que exista comunicación entre el BBU y RRU mediante una interfaz flexible además de brindar la posibilidad de reemplazar un conductor de cobre como medio de conexión entre una estación base y un transceptor de radio [24].
- *Jumper*, es el encargado de emisión y recepción de señales de RF (*Radio Frequency*). Este componente se encuentra conectado al RRU y al sistema de antena [24].

En la Figura 1.9 se describe cual es la estructura del eNode-B en donde también se muestran los elementos anteriormente descritos.



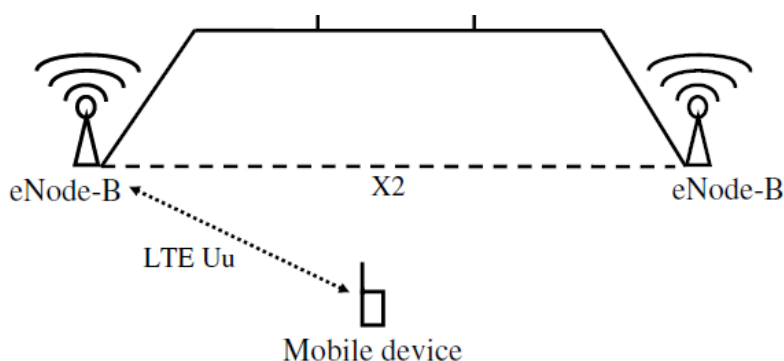
**Figura 1.9.** Estructura de eNode-B [24].

#### ❖ Interfaz X2

Una de las principales funciones que tiene la interfaz X2, se da en el proceso de *handover* en donde la comunicación entre celdas es posible mediante esta interfaz, solo sí se conoce y es posible acceder a la celda objetivo. En el caso de que no sea posible acceder a la celda deseada, la interfaz S1 junto con el EPC se encargan de realizar el *handover* [24].

La segunda función que tiene la interfaz X2 es coordinar interferencia, por ejemplo, cuando un UE (*User Equipment*) recibe varias señales con potencias iguales de varias estaciones base (e-NodeB) introduciendo ruido hacia el sistema. Con la ayuda de la interfaz X2 se mitiga este comportamiento indeseado mediante la detección de eNode-Bs vecinos y con ello determinar que señal es la que se debe tomar [24].

En la Figura 1.10 se muestra la conexión de dos estaciones base (e-NodeB) mediante la interfaz X2.



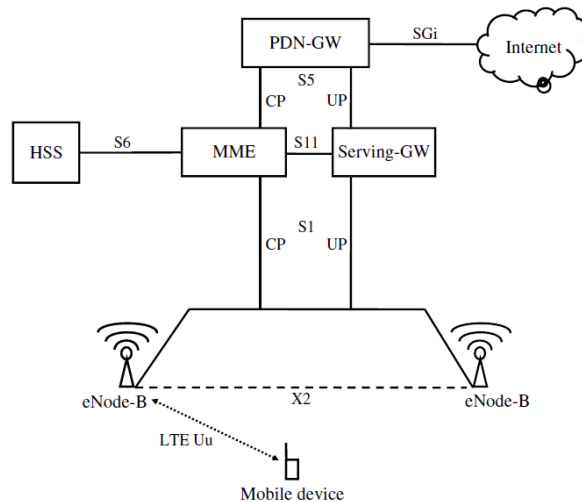
**Figura 1.10.** Conexión de eNode-Bs mediante interfaz X2 [22].

#### ❖ Interfaz S1

Mediante esta interfaz se realiza la conexión entre la estación base y la red central o *Core*. Dado que es necesario contar con una conexión que mínimo sea del orden de cientos de Mbps, generalmente se utilizan enlaces de fibra óptica o Ethernet sobre microondas[24].

La interfaz S1 tiene dos partes lógicas que utilizan la misma conexión. La primera parte se denomina S1-UP (*S1-User Plane*) o plano de usuario, el cual se basa netamente en paquetes IP y es el encargado de enviar los datos del usuario. La parte lógica se la conoce como S1-CP (*S1-Control Plane*) que es la sección encargada de intercambiar mensajes de señalización entre miembros o usuarios del sistema. Otra tarea que tiene S1-CP es la de conectarse con el *core* para dar aviso que existe esta red, notificar del estado y conexión de la red y también recibir los parámetros para configurar la EPC [24]. En la Figura 1.11 se aprecia la conexión entre los eNode-Bs y el EPC mediante la interfaz S1.





**Figura 1.11.** Conexión de eNode-B y EPC mediante interfaz S1 [24].

### 1.3.2.2 EPC (*Evolver Packet Core*)

Esta sección de red es en donde se coordinan varias de las funciones esenciales de las comunicaciones celulares de cuarta generación. A partir de la EPC es posible ejecutar tareas de *roaming* o permitir a los UE acceder a internet. Dentro de la EPC, se tienen algunos componentes entre bases de datos, *gateways* y servidores que son los responsables de que las acciones mencionadas en la oración anterior sean posibles [24].

A continuación, se detalla el nombre y la función de cada componente del EPC. Así mismo estos elementos se observan en la Figura 1.12 [24].

#### ❖ MME (*Mobility Management Entity*)

El MME se encarga de la señalización tanto de eNode-Bs y usuarios con el EPC. Para su conexión existen 2 interfaces llamadas S1-CP y S5-CP, las cuales se conectan a la parte de la e-UTRAN y al PDN-GW (*Packet Data Network Gateway*) respectivamente [24].

Este elemento es comparable con la SGSN (*Serving GPRS Support Node*) que pertenece a tecnologías de comunicación celulares más antiguas como GPRS o UMTS. La diferencia radica en que el MME delega las tareas reenvío de datos de usuario al S-GW (*Serving Gateway*) y únicamente se preocupa por la señalización a diferencia del SGSN en donde este manejaba las dos labores [24].

La MME tiene varias labores entre las cuales se encuentran tareas como de autenticación de los suscriptores a la red LTE, establecimiento de portadoras, gestión de movilidad NAS (*Non-Access Stratum*), soporte para *handovers*, interconexión con redes externas y soporte para SMS (*Short Message Service*) y voz [24].

#### ❖ S-GW (*Serving Gateway*)

A través del S-GW se envían los paquetes IP de los UE sirviendo como una especie de ancla de movilidad local para portadoras de datos en el caso de que un UE cambia de eNode-B [26].

Cuando el UE se queda temporalmente sin servicio, el S-GW es el encargado de guardar la información de los portadores además de retener temporalmente la información del *downlink* hasta que el MME pueda localizar al UE y reestablecer el enlace de portadoras [26].

Otra de las tareas que se le entrega a este elemento es la de realizar ciertas funciones de administración como por ejemplo recoger información para facturación aunque también se le designa el recabar volumen de datos enviados y recibidos por parte del suscriptor [26].

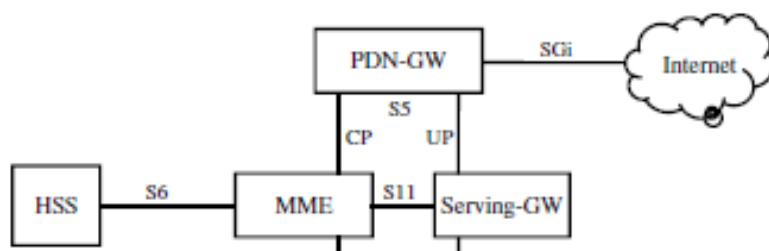
#### ❖ PDN-GW (*Packet Data Network Gateway*)

Este elemento es el responsable de crear un enlace hacia Internet u otra red externa ya sea una VPN (*Virtual Private Network*) o intranet. Además, también se encarga de asignar direcciones IP a los UE luego de la autenticación del usuario a la red. En el caso de que el UE cuente con la opción de tener VoLTE el PDN-GW entrega dos direcciones IP que cuentan con NAT (*Network Address Translation*) debido a la escasez de IPv4 [24].

#### ❖ HSS (*Home Subscriber Server*)

Este elemento es la evolución del HLR (*Home Location Register*) que es una base de datos de suscriptores. Para el intercambio de datos en HSS se utiliza el protocolo DIAMETER que se basa totalmente en IP y se lo considera como del tipo AAA (*Authentication, Authorization and Accounting*) [24].

Dentro de esta base de datos se encuentra, entre otros, el IMSI (*International Mobile Subscriber Identity*), el cual es responsable de dar al usuario una identificación única la cual tiene una copia en la SIM (*Subscriber Identity Module*) del suscriptor [24]. En la Figura 1.12 se observa un diagrama donde constan los elementos de la EPC junto con sus interfaces.



**Figura 1.12.** EPC de una red LTE [22].

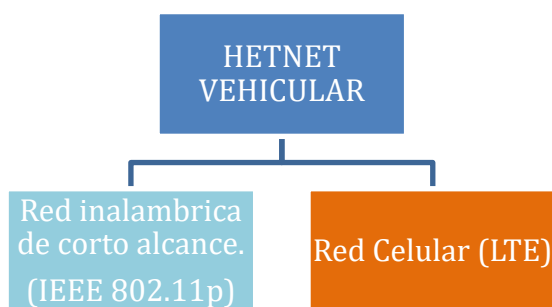
### 1.3.3 INTERACCION IEEE 802.11p CON LTE

En esta sección se describe brevemente lo que es una red HETNET. Además, se explica cómo interactuarán una red celular LTE y la red de comunicación inalámbrica de corto alcance regida por el estándar IEEE 802.11p para formar la red HETNET. También se explica de manera resumida la forma en que se busca que la red celular LTE no llegue a saturarse.

#### 1.3.3.1 HETNET Vehicular

Una HETNET se la conoce también como *heterogeneous network* o red heterogénea y se define como un tipo de red en donde convergen o funcionan distintos sistemas operativos y/o protocolos de comunicación [27].

Una red HETNET se conforma por dos redes, una red celular y una red inalámbrica de corto alcance como se muestra en el esquema de la Figura 1.13.



**Figura 1.13.** Esquema de una red HETNET vehicular.

Con el paso de los años, la posibilidad de utilizar V2X (*Vehicle to everything*) ha permitido muchas aplicaciones en esta tecnología, siendo una de las funciones que más se ha explorado es la seguridad vial; sin embargo, según los estudios de [6] "El 80% de las colisiones se evitarían si el 50% de las intersecciones estuvieran equipadas con un RSU (*Road-Side Unit*). Mientras que el 50% de las colisiones se evitarían si solo el 20% de las intersecciones estuvieran equipadas" [6].

En el caso de las comunicaciones celulares, la tecnología que se toma en cuenta para desarrollar una red híbrida heterogénea para VANETs es *LTE-Advanced*. Debido a el tiempo reducido en cuestión a autenticación y, además de la velocidad de transferencia que ofrece, se decide que LTE-A era óptima para lograr una comunicación V2I [6].

Para la RAN (*Radio Access Network*) se utiliza una macro celda como componente principal para una gran cobertura y con usuarios de alta movilidad; sin embargo, hoy en día se tiene un crecimiento exponencial de demanda de datos móviles gracias a que un usuario promedio tiene 2 o 3 dispositivos como mínimo conectados a internet ya sea por Wifi o comunicaciones celulares. Cuando la densidad de dispositivos que se conectan a la red es alta, no es tan viable depender únicamente de la macro celda. Para mitigar un poco la alta demanda de datos y mejorar la cobertura de la RAN se coloca estaciones base más pequeñas que ayudan con la eficiencia espectral en el área que brindan el servicio [6].

También se trabaja con micro celdas, pico celdas y femto celdas que cada día se hacen más populares entre los prestadores del servicio móvil en HETNETs.

En los siguientes literales se explica los estándares de comunicación IEEE 802.11p, LTE-A y además una explicación de cómo se vinculan las redes vehiculares comunes con LTE para formar la HETNET.

#### *1.3.3.1.1 Acceso a la red LTE-A para IEEE 802.11p*

Un punto para considerar es la reducción o limitación del número de *Gateways* o CH (*Car Header*) para bajar la sobrecarga de señalización de movilidad a pesar de que cada auto pueda por sí mismo conectarse a la radio base LTE-A. Esto se logra mediante un nuevo esquema que elige de manera eficiente los CHs móviles para transmitir el tráfico de los automóviles vecinos hacia las SC (*Small Cells*) [6]. SC básicamente son nodos intermedios para acceso a la red LTE, los cuales son de tamaño reducido y además en potencia de transmisión [28].

#### *1.3.3.1.2 Backhaul de SC LTE-A*

Al utilizar SCs, el rango de alcance de la señal se ve mermada lo que genera frecuentes trasposos o *handovers* entre celdas y aumenta la señalización hacia el EPC. Una alternativa para sobrellevar el problema que se especifica anteriormente es utilizar un esquema de movilidad local basado en anclajes de *k*-saltos que consta de tres procedimientos los cuales son intradominio, interdominio de *k*-saltos e interdominio [6].

Los saltos los realiza el o los paquetes que se dirigen hacia el *Car Header*.

#### 1.3.3.1.3 Selección de Car Header móvil para comunicaciones V2I.

Cuando se necesita precautelar la estabilidad y fiabilidad de una conexión, se define y aplica un algoritmo de agrupación de clústeres. En este caso el término clúster se acuña a un grupo de vehículos que se encuentran cerca.

Existen varios algoritmos para la selección del Car Header y al escoger alguno se toma en cuenta que se garantice la estabilidad de la comunicación y la vida útil del enlace que se crea entre los vehículos pertenecientes a un clúster. También se trata de disminuir la sobrecarga de control de enrutamiento para los automóviles que están dentro de distintos clústeres [6].

Se define un primer algoritmo, que actualmente es el más simple y utilizado, que trabaja con LET (*Link Expiration Time*) y RET (*Route Expiration Time*) estableciendo un enfoque descentralizado. LET viene a ser un algoritmo en donde un nodo envía un paquete si y solo si se sabe que es el destino correcto, luego recibe un ACK indicando que el paquete llega correctamente, para este tipo de algoritmo se tiene un tiempo de vencimiento del paquete [29]. RET viene a ser parecido a LET con la diferencia que el paquete tiene un determinado número de saltos definido por lo que, si no llega hacia su destino al saltar entre los nodos, el paquete desaparecerá [29].

Con este enfoque a un automóvil se añade una interfaz 802.11p junto a una con soporte a comunicaciones celulares ya sea UMTS o LTE. Se forman clústeres agrupándose dinámicamente los automóviles según métricas como RSS (*Receive Signal Strength*), el LET o el RET. Una vez formado el clúster se escoge un número mínimo de automóviles que actuarán como *Car Header* entre la VANET y la tecnología celular que en este caso será LTE. Las métricas tomadas en el primer algoritmo son en teoría fáciles de implementar, sin embargo, en cuestión a sobrecargas dan demasiados problemas. Es necesario tener una cantidad de paquetes de señalización muy grande para mantener los clústeres por lo que la red LTE podría llegar a saturarse [5], [6].

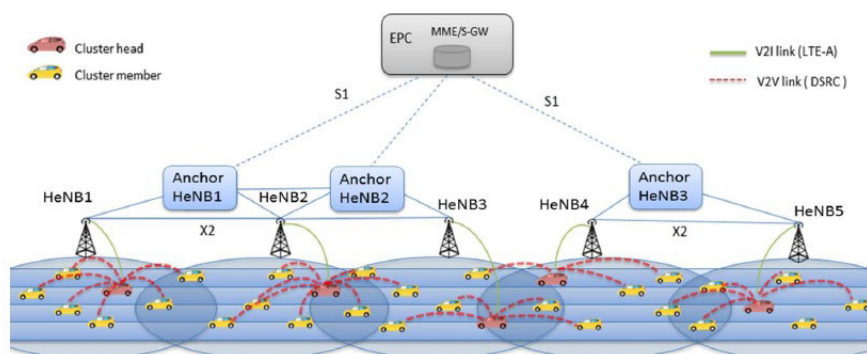
Zhioua en [6] propone un algoritmo basado en la selección de *Car Header* mediante lógica difusa para generar una red híbrida entre VANET y LTE-A. Este algoritmo también utiliza LET y RSS al igual que el algoritmo que se explica en tercer párrafo de este capítulo, pero añade clases de tráfico QoS (*Quality of Service*) y la carga que genera el *Car Header* para encontrar una puerta de enlace. Se esperaba que la sobrecarga de paquetes de señalización hacia la red sea menor, sin embargo, sucede todo lo contrario al tratar de obtener más información de datos para decidir qué vehículo será el *Car Header* [6].

El segundo algoritmo que se utiliza para escoger un *Car Header*, plantea tomar a los autobuses u otro vehículo que tenga una ruta fija cómo las puertas de enlace, sin embargo se tiene el inconveniente de que ya que no siempre están disponibles en las carreteras además que la calidad del canal no es de muy buena calidad [6].

El último enfoque conocido cómo centralizado tiene muy buenos resultados si se lo utiliza en situaciones de V2I (*Vehicle to Infrastructure*). Para tener menor sobrecarga, este enfoque aprovecha la macro celda de LTE con lo que se logra gestionar de mejor manera los clústeres. Uno de los investigadores de este algoritmo, llamado X. Duan en [6], muestra un algoritmo capaz de escoger un auto principal que se conoce como cabeza de grupo el cual toma el tráfico de vehículos vecinos para luego comunicarse con la estación base por medio del SDN (*Software-Defined Networking*). Se agrega redundancia en este enfoque ya que se tiene un nodo dual para ser cabezal del clúster, ya que con ello se tendría un respaldo en caso de fallas [6]. El algoritmo que se tiene en el enfoque centralizado tiene dos pasos que son los siguientes:

- Primero, se inicializa los clústeres donde los vehículos son asignados a cada clúster [6].
- Luego la cabeza del clúster es seleccionada usando programación lineal de acuerdo a la SNR (*Signal-to-Noise Ratio*) y a la proximidad de la velocidad del vehículo a la velocidad promedio del clúster [6].

En la Figura 1.14 se muestra un esquema de una HETNET en donde se forman grupos de vehículos los cuales tienen un *Car Header* que es de color rojo. Los demás vehículos de la caravana de color amarillo forman comunicaciones V2V con el *Car Header* y este a su vez forma comunicaciones V2I con el e-NodeB. Los e-NodeBs se comunican entre sí mediante la interfaz X2.



**Figura 1.14.** Conformación de HETNETs.

## 2 METODOLOGÍA

En el presente capítulo se detalla el diseño e implementación de la HETNET vehicular dentro de un sector del centro histórico de Quito para evaluar su rendimiento.

Primero, se explican brevemente el conjunto de simuladores que se requieren para implementar la HETNET. Posteriormente, se detalla la selección del área del centro histórico en donde se pretende que los automóviles circulen; y luego, se presentará cómo se configura la red heterogénea [4-6].

Posterior a la configuración de la HETNET, se insertan las cinco caravanas de automóviles (*Platooning*) y se diseña algunos escenarios en donde se cambian parámetros físicos y técnicos de la red. Una vez que se tengan los escenarios se simulan con el fin de que los resultados permitan el análisis del desempeño de la red HETNET.

Para llevar a cabo la simulación e implementación de los escenarios de la HETNET vehicular se utilizará los siguientes simuladores y software:

- OSM
- JOSM
- SUMO
- OMNET
- VEINS

Todo el software que se menciona corre en:

- Laptop Core i3, 2.20 [GHz] y RAM de 8 [GB].
- Ubuntu 16.04 instalado en una partición del disco duro.

### 2.1 SIMULADORES

A continuación, se resumen las características y funciones que cumplen los simuladores OMNET++ y SUMO; además, del software para generación de mapas OSM y del proyecto VEINS de OMNET++.

#### 2.1.1 OMNET++

Es un simulador de eventos discretos el cual se encuentra basado en C++ para crear librerías o entornos de trabajo de las simulaciones que se deseen realizar [30].

A partir de OMNET++ se han creado nuevos proyectos y librerías como INET FRAMEWORK, SIMU5G, SIMULte, VEINS [31], entre otros, los cuales contienen librerías

y archivos programables y editables para generar las simulaciones deseadas por cualquier usuario.

Actualmente, este software se encuentra en la versión 5.6.2 que fue lanzada en mayo del 2020 y por el momento es el más estable [32].

OMNET++ puede trabajar en diferentes sistemas operativos como WINDOWS, MAC OS y LINUX [32], sin embargo, al ser software libre este se desempeña de mejor manera corriendo en un sistema operativo basado en LINUX.

Este simulador ayudará en este proyecto a implementar las redes de comunicaciones inalámbricas de corto alcance del estándar IEEE 802.11p y la red celular LTE.

### **2.1.2 SUMO**

SUMO tiene las siglas "*Simulation of Urban Mobility*", y como dice su nombre es un simulador de movilidad en entornos urbanos que creado por el Instituto de Sistemas de Transporte en el Centro Aeroespacial de Alemania [33].

Este software es de código abierto y crea escenarios de movilidad en donde circulan automóviles, buses, motocicletas e incluso peatones [33]. La simulación de los escenarios se ejecuta a distinta velocidad con el objetivo de analizar de mejor manera la forma en que los vehículos se movilizan en los escenarios.

SUMO también cuenta con algunas herramientas que introducen tráfico vehicular según los requerimientos del programador ayudando a generar escenarios reales.

En el presente proyecto, este software permitirá crear la simulación vehicular dentro del sector del centro histórico de Quito.

### **2.1.3 OSM**

*Openstreetmap* es un proyecto de distribución libre, en donde se comparte información geográfica de todo el mundo. En [34] se indica que se inicia este proyecto para tener una alternativa a los mapas convencionales de grandes empresas los cuales tienen restricciones legales y técnicas.

OSM presenta características físicas de una zona territorial como carreteras, edificios, entre otros [34], los cuales son actualizados o modificados por cualquier usuario que se encuentre registrado en esta herramienta.

Este software permitirá crear el mapa del sector del centro histórico de Quito en donde se desea implementar la red HETNET, posteriormente el mapa se lo cargará en SUMO para generar la simulación vehicular.



### 2.1.4 VEINS

Es una herramienta de código abierto para simular entornos de redes vehiculares y une la simulación de tráfico vehicular de SUMO y la simulación de red que se tiene en OMNET++ [35].

La forma en que trabaja VEINS se resume en que la simulación de tráfico se la ejecuta mediante SUMO y la simulación de redes se la ejecuta en OMNET++, pero para unir las dos ejecuciones se utiliza una sub herramienta de VEINS, denominada MiXiM que se encarga de la parte de capa física y permite modelar distintos tipos de interferencia de radio y desvanecimiento a pequeña y a gran escala [35].

En la Figura 2.1 se detalla mediante un esquema como interactúan el simulador de redes OMNET++ y el simulador de tráfico vehicular SUMO gracias a VEINS.

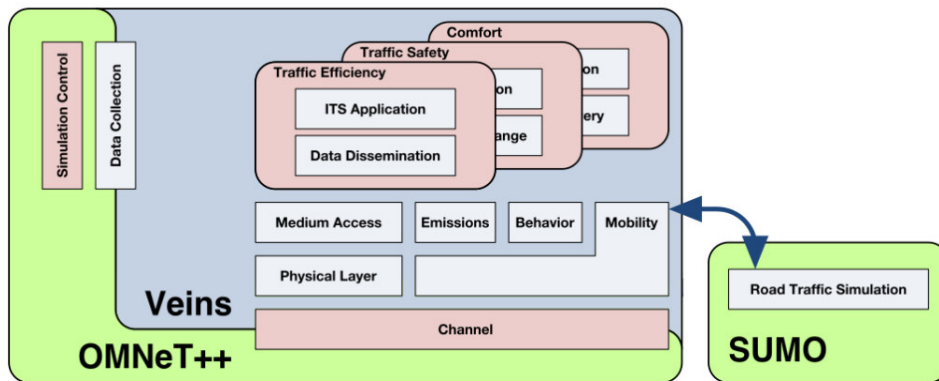


Figura 2.1. Interacción de VEINS, SUMO y OMNET++ [35].

## 2.2 INSTALACIÓN DE SOFTWARE

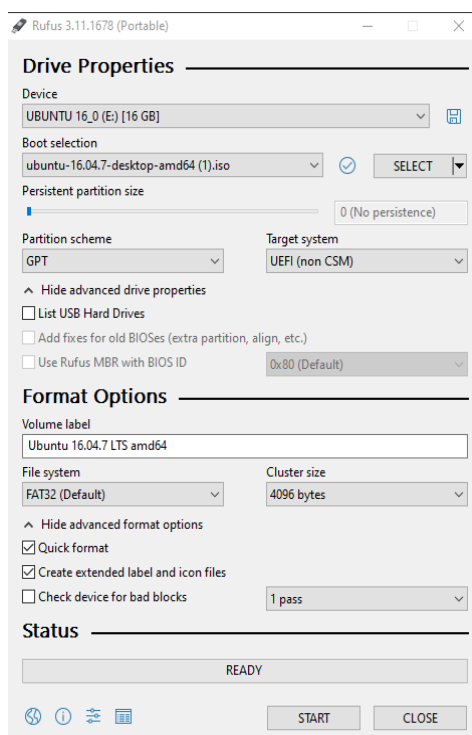
A continuación, se indica como se instala el sistema operativo en donde se ejecutarán los simuladores además de los pasos para instalar los mismos.

### 2.2.1 UBUNTU

Para trabajar con los simuladores se recomienda utilizar un sistema operativo basado en LINUX ya que se tiene más versatilidad al momento de ejecutar tareas; además, este tipo de sistema operativo es ligero ya que cuenta con menos procesos ejecutándose en segundo plano a diferencia de Windows.

También se puede instalar una máquina virtual de Ubuntu dentro de Windows, pero consume muchos recursos de hardware por lo que no se toma en cuenta esta opción.

Para instalar Ubuntu, dentro de una partición, se necesita una memoria extraíble de al menos 8 Gigabytes en donde se instalará sistema operativo. El software que gestiona el instalador de Ubuntu se denomina RUFUS en donde se carga la imagen ISO de Ubuntu como se indica en la Figura 2.2.

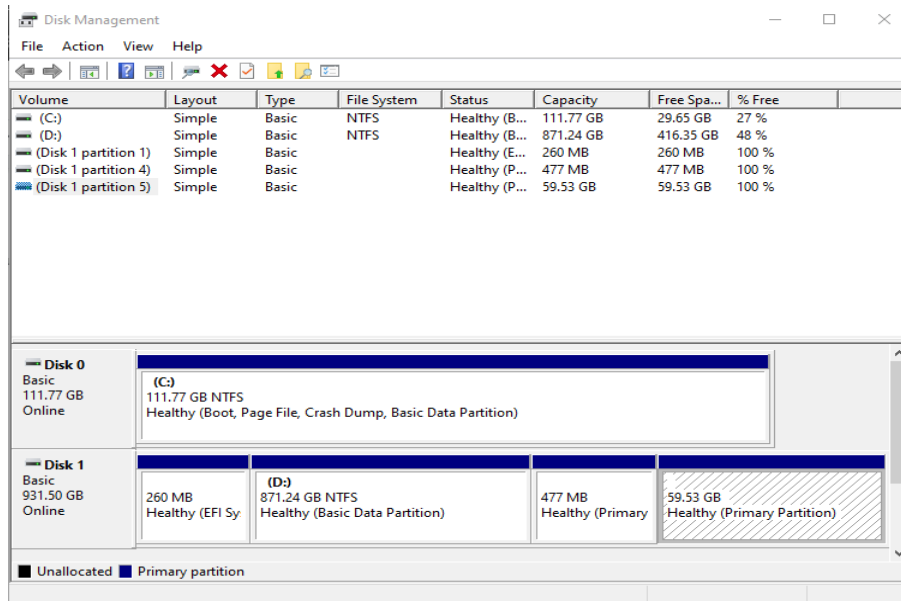


**Figura 2.2.** Configuración de RUFUS.

Una vez que se haya configurado RUFUS, se empieza la escritura del instalador en la memoria extraíble. Para este proyecto, la versión de Ubuntu a utilizar es la 16.04 ya que los compiladores y repositorios que se manejan son compatibles con la versión de SIMULte que se va a emplear.

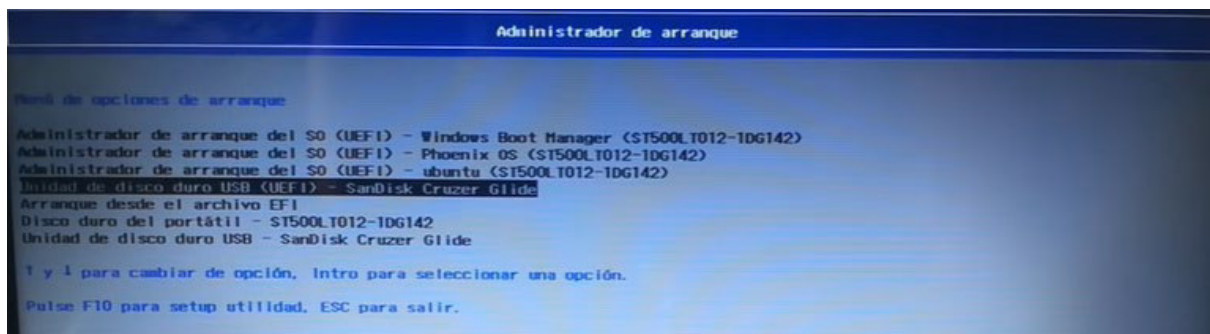
Con la memoria extraíble del instalador de Ubuntu, se prepara una partición en el disco duro del computador en donde se instalará Ubuntu. Para ello, se debe abrir la sección “Administrador de discos” en el panel de control de Windows y configurar una sección en blanco como se muestra en la Figura 2.3.

En este caso se crea una partición de 60 GB para la instalación de Ubuntu como se muestra en la Figura 2.3.



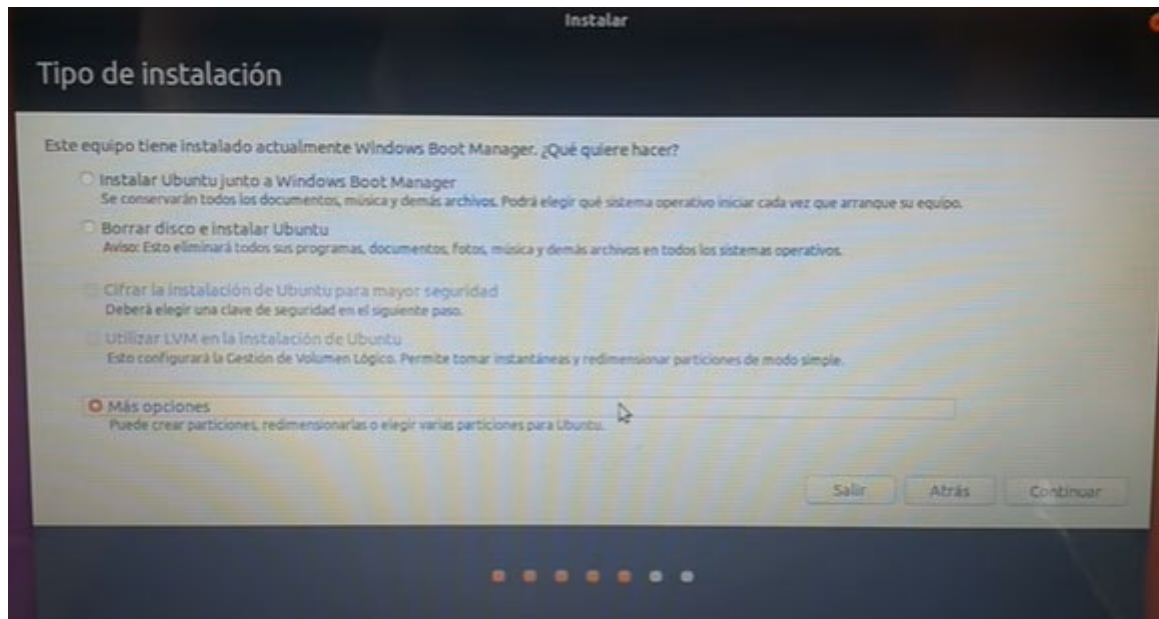
**Figura 2.3.** Configuración de disco duro.

Luego de tener la partición creada es necesario reiniciar el computador y entrar en la Bios del equipo para iniciar desde el instalador que se tiene en la memoria extraíble.



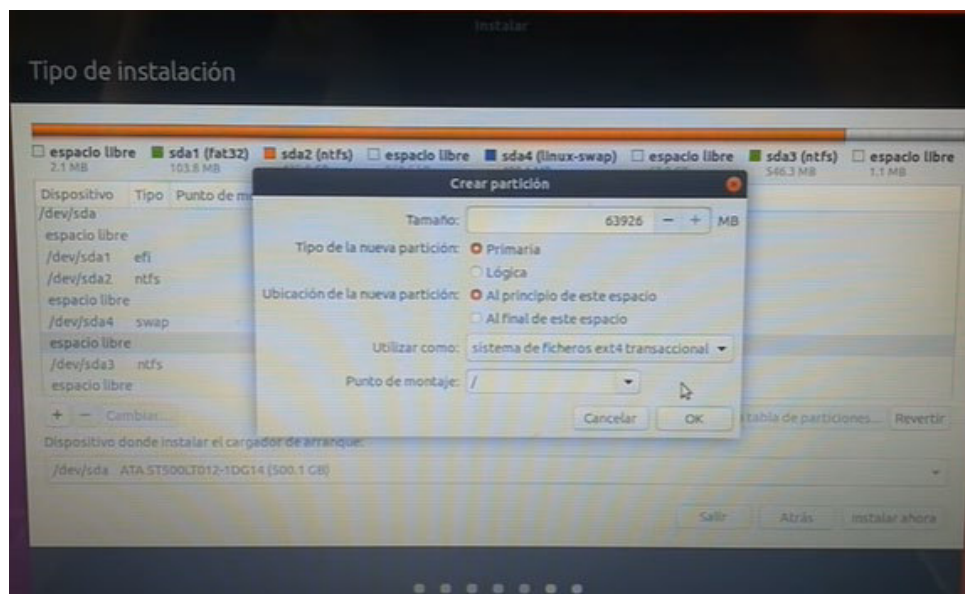
**Figura 2.4.** Arranque de instalador desde Bios del equipo.

Cuando ya se cargue el instalador de Ubuntu, se da clic en siguiente a todos los diálogos hasta llegar a lo que se tiene en la Figura 2.5.



**Figura 2.5.** Selección de instalación de Ubuntu junto a Windows.

En la ventana de tipo de instalación se selecciona “Más opciones” y ahí configurar el espacio vacío que se dejó en el disco duro para instalar Ubuntu como se muestra en la Figura 2.6.



**Figura 2.6.** Selección de espacio particionado.

Luego de seleccionar el tamaño de la partición, se da clic en “Instalar ahora” y después configurar la zona horaria, contraseña entre otros para que Ubuntu quede instalado junto a Windows en el equipo.

## 2.2.2 OMNET++

Para instalar OMNET++ en Ubuntu, descargar un archivo comprimido del *software* que se encuentra en <https://omnetpp.org/download/>. Escoger el sistema operativo dentro de la página como se muestra en la Figura 2.7.



**Figura 2.7.** Descarga de OMNET++.

En este caso se escoge la versión de OMNET++ 5.5.1 ya que esta es la más compatible con las librerías SIMULte y VEINS que son esenciales para la creación de la HETNET.

Luego de descargar OMNET++, se introducen los siguientes comandos dentro del directorio donde se descarga OMNET++. Estos comandos sirven para poder instalar OMNET++.

```
$ cd Downloads (Ingresa al directorio descargas)
$ tar xvfz omnetpp-5.5.1-src-linux.tgz (Descomprime el archivo de OMNET++)
$ cd omnetpp-5.5.1 (Ingresa al directorio de los archivos de OMNET++)
$ . setenv (Indica la ruta completa del directorio de OMNET++)
$ gedit ~/.bashrc (Abre un editor en donde se cambia propiedades y variables del sistema)
```

En el fichero ".bashrc" colocar la siguiente línea:

```
export PATH=$PATH:/home/user/Downloads/omnetpp-5.5.1/bin
```

Luego de añadir la línea en el fichero, se guardan los cambios.

Finalmente, se abre nuevamente una terminal de comandos y se ingresa los siguientes comandos. Los comandos permiten ingresar en el directorio de descarga y con ello cargar los archivos necesarios para que se ejecute OMNET++.

```
$ cd Downloads (Ingresa al directorio descargas)
$ cd omnetpp-5.5.1 (Ingresa al directorio de los archivos de OMNET++)
$ ./configure (Lee las configuraciones realizadas en el fichero .bashrc)
$ make (Instala los archivos necesarios para OMNET++)
$ omnetpp (Ejecuta OMNET++)
```

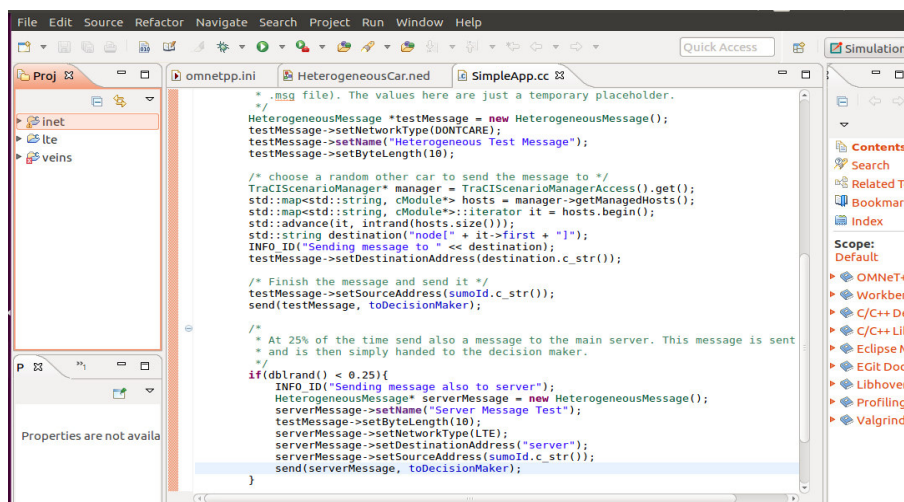


Figura 2.8. Inicialización de OMNET++.

Si no hay ningún problema al momento de la instalación, OMNET++ se ejecutarán y mostrarán una interfaz como se indica en la de la Figura 2.8.

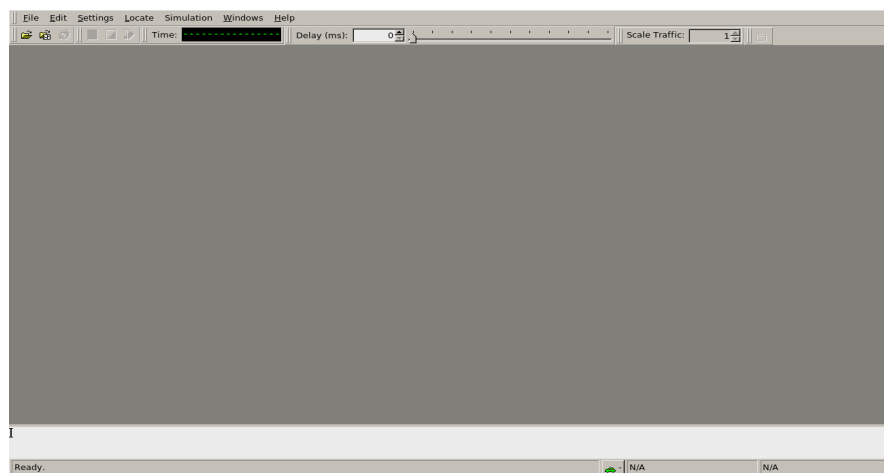
### 2.2.3 SUMO

Para instalar SUMO en Ubuntu, se abre una terminal de comandos y se ejecuta lo siguiente:

```
$ sudo apt update (Actualiza las librerías, compiladores y repositorios de Ubuntu)
$ sudo apt-get install cmake python g++ libxerces-c-dev libfox-1.6-dev libgdal-dev
libproj-dev libgl2ps-dev swig (Instala las librerías esenciales para el funcionamiento de
SUMO)
$ sudo apt install git (Instala la herramienta de Github)
$ git clone --recursive https://github.com/eclipse/sumo (Clona el repositorio de
SUMO desde Github)
```

```
$ export SUMO_HOME="$PWD/sumo" (Exporta la variable SUMO_HOME)
$ mkdir sumo/build/cmake-build && cd sumo/build/cmake-build (Crea e ingresa al directorio cmake-build)
$ cmake ../.. $] make -j8 (Instala los archivos necesarios para la ejecución de SUMO)
$ sumo-gui
```

Si no hay ningún error con el comando cmake, al ejecutar sumo-gui, se debe abrir una interfaz similar a la de la Figura 2.9.



**Figura 2.9.** Inicialización de SUMO.

## 2.2.4 LIBRERÍAS Y REPOSITORIOS EN OMNET++

A continuación, se muestra la instalación de las librerías INET FRAMEWORK, SimuLTE, VEINS y Veins\_INET.

Para instalar una librería en OMNET++, es necesario obtener el instalador en archivo comprimido y luego descomprimir en cualquier directorio dentro de Ubuntu.

Los enlaces de las librerías son los siguientes:

- INET FRAMEWORK: <https://inet.omnetpp.org/Download.html>
- SimuLTE: <https://github.com/inet-framework/simulte/releases>
- VEINS: <https://veins.car2x.org/download/>

En la Figura 2.10 se observan las versiones disponibles de INET FRAMEWORK para descargar. Es importante tener en cuenta la versión de OMNET++ que se haya instalado para seleccionar la versión de INET FRAMEWORK compatible.

## Stable Releases

Downloads of stable releases are available here. It is recommended that you use the latest stable release for projects.

Latest releases:

[Download latest stable version \(4.3.2, for OMNeT++ 6.0 pre 10 or later\)](#) (What's New)

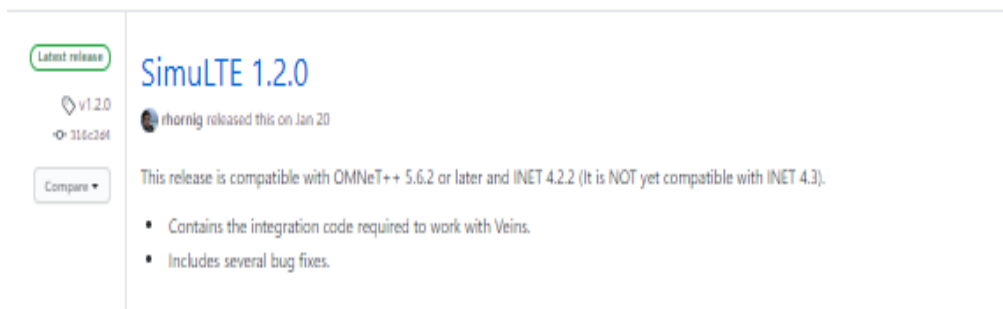
- [Download latest OMNeT++ 5.x compatible version \(4.2.5, for OMNeT++ 5.4.1 or later\)](#) (What's New)
- [Download latest legacy \(3.x\) version \(3.7.1, for OMNeT++ 5.3 or later\)](#) (What's New)

Older releases (please do not use them for new projects):

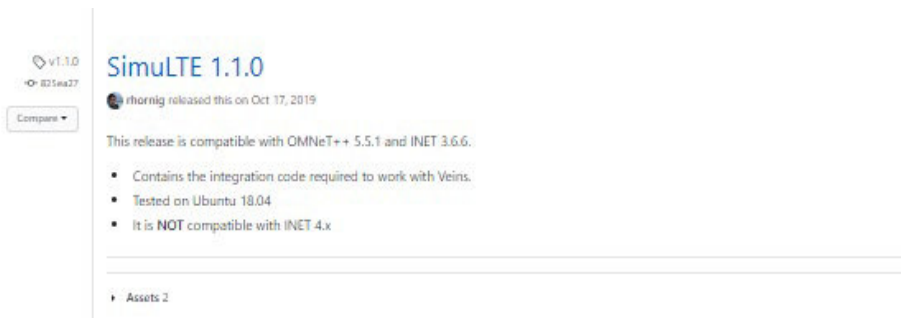
- [INET 4.1.2 for OMNeT++ 5.4.1](#) (What's New)
- [INET 4.0.0 for OMNeT++ 5.4.1](#) (What's New)
- [INET 3.6.8 for OMNeT++ 5.3](#) (What's New)
- [INET 3.5.0 for OMNeT++ 5.1](#) (What's New)
- [INET 3.4.0 for OMNeT++ 5.0](#) (What's New)
- [INET 3.3.0 for OMNeT++ 4.6](#) (What's New)
- [INET 3.2.4 for OMNeT++ 4.6](#) (What's New)
- [INET 3.1.1 for OMNeT++ 4.6](#) (What's New)
- [INET 3.0.0 for OMNeT++ 4.6](#) (What's New)
- [INET 2.6.0 for OMNeT++ 4.4](#) (What's New)
- [INET 2.5.0 for OMNeT++ 4.4](#) (What's New)
- [INET 2.4.0 for OMNeT++ 4.3](#) (What's New)
- [INET 2.3.0 for OMNeT++ 4.3](#) (What's New)
- [INET 2.2.0 for OMNeT++ 4.2](#) (What's New)
- [INET 2.1.0 for OMNeT++ 4.2](#) (What's New)
- [INET 2.0.0 for OMNeT++ 4.2](#) (What's New)
- [INET 20111118 for OMNeT++ 4.2](#)
- [INET 20110225 for OMNeT++ 4.1](#)
- [INET 20100723 for OMNeT++ 4.1](#)

**Figura 2.10.** Versiones disponibles de INET FRAMEWORK.

Al igual que con INET FRAMEWORK, se debe escoger una versión de SimuLTE compatible con la versión que se utilizará de OMNET++ , por ejemplo, para la versión de OMNET++ usada en este proyecto, las versiones compatibles son las que se muestran en la Figuras 2.11, 2.12 y 2.13.



**Figura 2.11.** Versiones de SimuLTE.



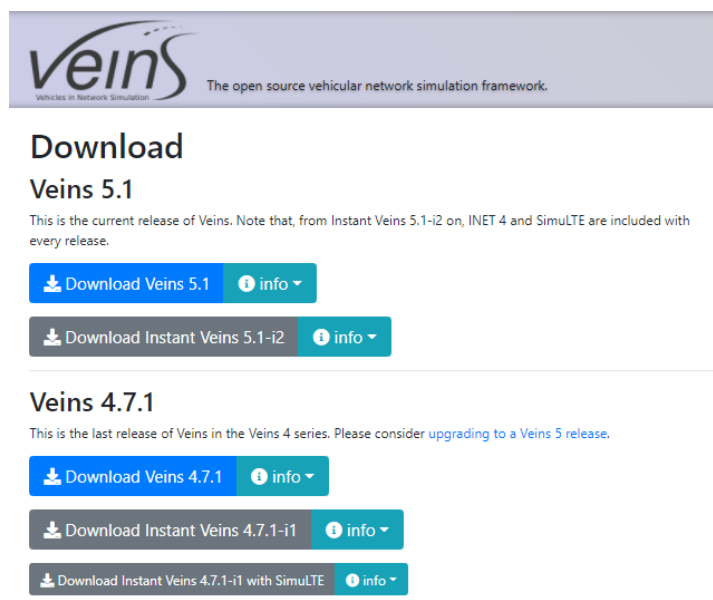
**Figura 2.12.** Versiones de SimuLTE.





**Figura 2.13.** Versiones de SimuLTE.

La última librería por añadir es VEINS, que al igual que con las demás librerías de OMNET++, cuenta con algunas versiones como se muestra en la Figura 2.14, y también hay que escoger una que sea compatible con la versión de OMNET++ que se tiene instalada.



**Figura 2.14.** Versiones de VEINS.

Una vez se tenga el archivo comprimido, es necesario descomprimirlo para luego poder importar a OMNET++ desde el IDE (*Integrated Development Environment*) de este software libre.

Para importar la librería se debe iniciar OMNET++ y dirigirse a la pestaña “File” como se muestra en la Figura 2.15.

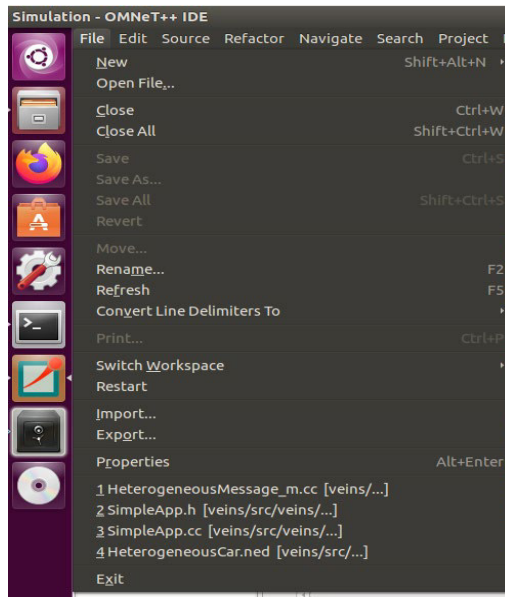


Figura 2.15. Selección de pestaña File.

Al desplegarse las opciones se selecciona “*Import*”.

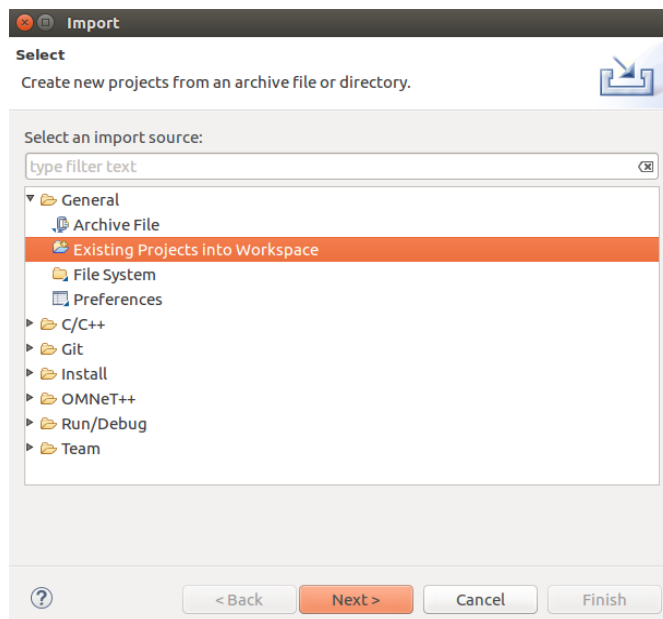
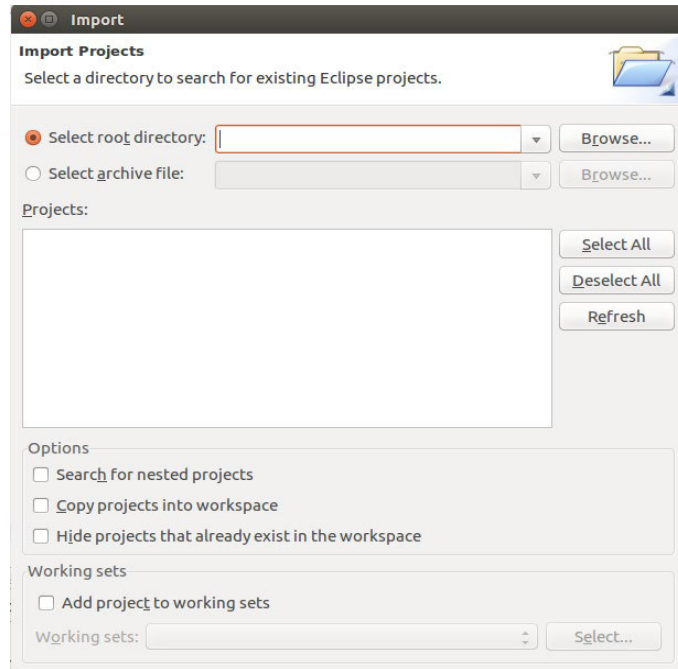


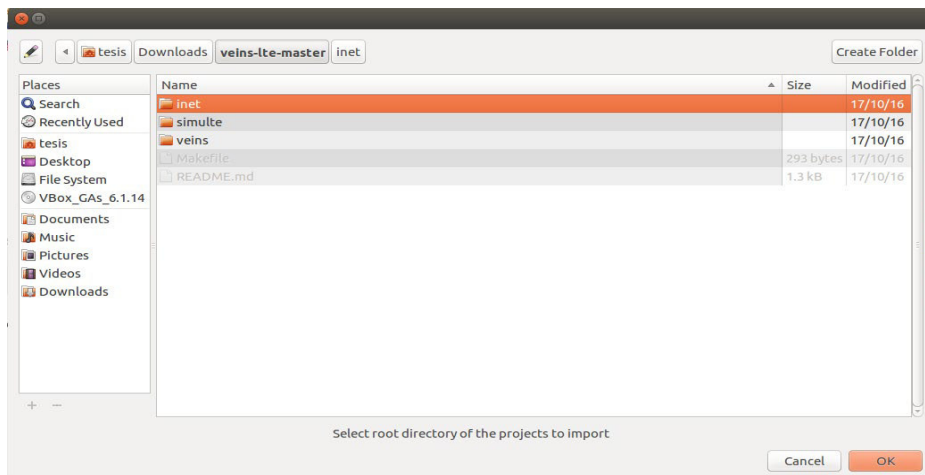
Figura 2.16. Menú de sección *Import*.

Al seleccionar “*import*”, se mostrarán más opciones en donde se debe seleccionar “*Existing Projects into workspace*” al igual que en la Figura 2.16 y luego dar clic en *Next*.



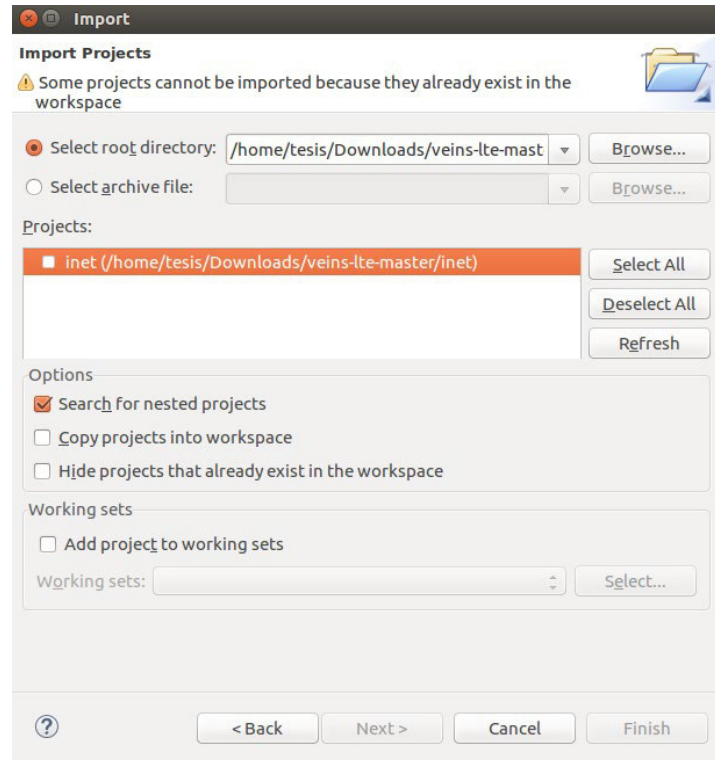
**Figura 2.17.** Selección de la ruta de librería.

En la Figura 2.18 se observa que aparece una nueva ventana, en donde se selecciona la ruta en donde se encuentra la librería de OMNET++ descomprimida, en este caso será INET FRAMEWORK.



**Figura 2.18.** Selección de la ruta de INET FRAMEWORK.

Cuando ya se seleccione la ruta y la carpeta de la librería, se selecciona “OK”, lo que hará que se abra otra pestaña como se indica en la Figura 2.19, en donde, finalmente se tiene que seleccionar “Finish”.



**Figura 2.19.** Importación final de la librería.

## 2.3 DISEÑO

A continuación, se explica en base a que parámetros se selecciona el área en donde se implementa la HETNET de caravanas de vehículos en el centro histórico de Quito. Una vez que se escoge el área donde se va a implementar la HETNET, se indica como se genera el mapa donde se inserta el tráfico vehicular.

Cuando ya se tiene el mapa, se inserta tráfico vehicular aleatorio con la ayuda de SUMO y se toma 5 de esas rutas aleatorias para definir caravanas de vehículos que tengan una misma trayectoria.

Luego, se busca un código beta de una HETNET el cual se modifica para que funcione según la teoría de las redes heterogéneas. Al tener el código de la HETNET diseñado y funcionando, se debe generar una interfaz visual que coincida con las calles del mapa para que la persona que observe el proyecto distinga como se mueven los vehículos.

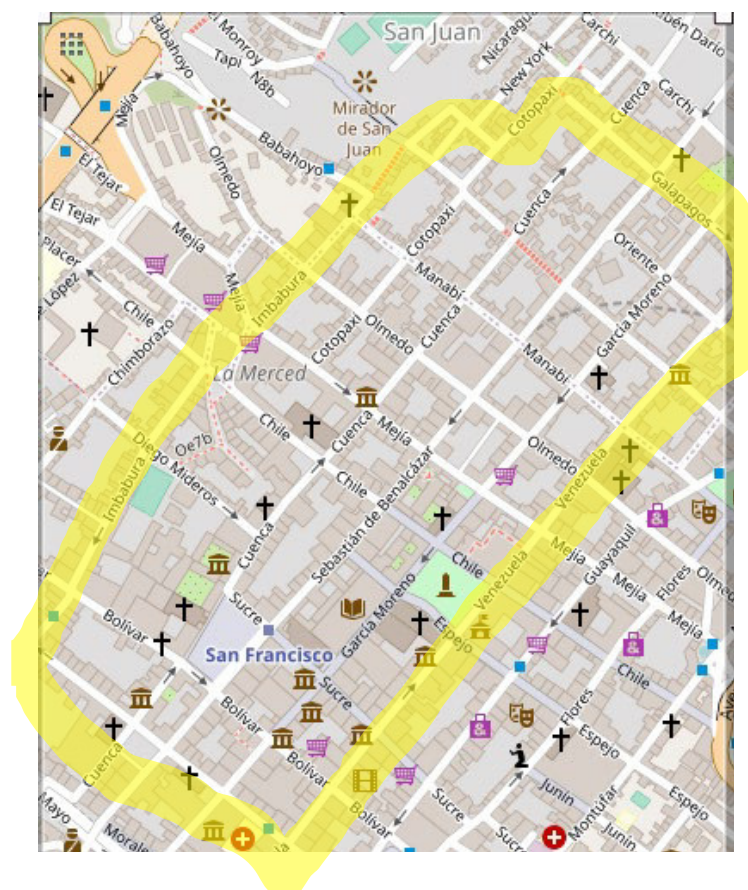
Con la HETNET ya diseñada y con la interfaz gráfica lista, se diseñan escenarios con los que se puedan evaluar el comportamiento de la HETNET cuando a esta se le cambia parámetros físicos o técnicos.

Finalmente, en este subcapítulo se indica como se vincula la simulación de la parte de redes de la HETNET junto con la simulación del tráfico vehicular para que se ejecuten simultáneamente.

### 2.3.1 SELECCIÓN DE ÁREA PARA IMPLEMENTAR LA HETNET

Un área idónea para implementar este tipo de HETNET vehicular, es aquella en donde exista cobertura LTE aceptable y tenga el problema de alto tráfico vehicular que se genera por la estrechez de sus calles, por lo que se considera el centro histórico de la ciudad de Quito con un perímetro de aproximadamente 3.6 km que tiene como límites las calles Galápagos al norte, Rocafuerte al sur, Imbabura al este y Venezuela al oeste.

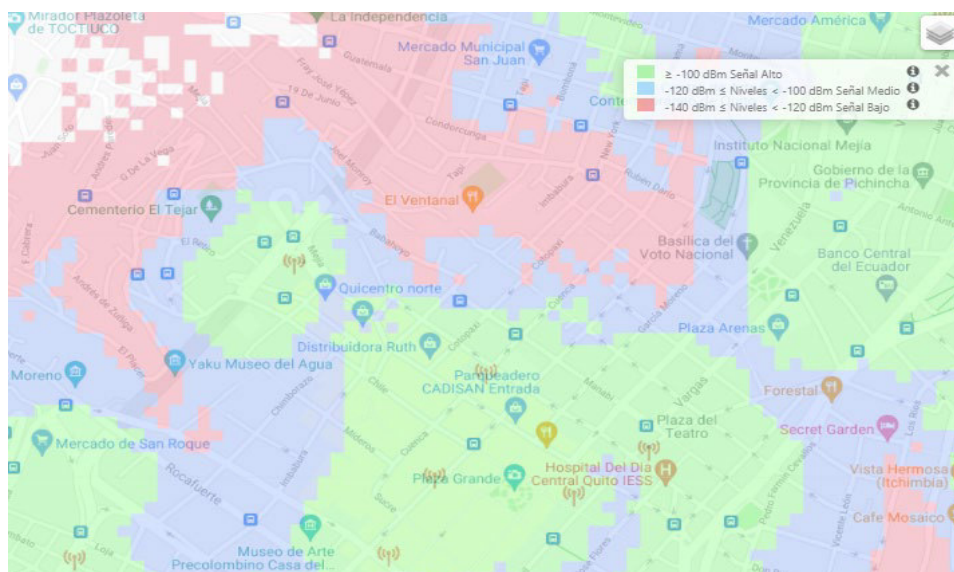
En la Figura 2.20 se muestra el área en donde se implementará la HETNET vehicular en el centro histórico de Quito.



**Figura 2.20.** Mapa aproximado en donde se pretende implementar la HETNET vehicular.

Para la red celular LTE se tomará la ubicación de las radios bases o *eNodeBs* de alguno de los operadores de telefonía celular de Ecuador. En este caso, se escoge a CONECEL S.A y se muestra el mapa de cobertura de la red celular en la Figura 2.21, en donde se

indica por colores que tan buena es la señal de la red. El color verde indica que en la zona se cuenta con una intensidad de la señal de la red celular mayor a los -100 dBm [36], por lo que la intensidad de la señal de la red celular es muy alta. Cuando el área del mapa de cobertura es de color azul, se indica que tiene una intensidad de la señal de la red celular entre -100 dBm y -120 dBm [36] por lo que se cuenta con la señal celular media. Finalmente, cuando el área del mapa de cobertura es de color rojo, se tiene la señal de la red celular baja ya que la intensidad de señal es menor a -120 dBm [36]. Además, en el mapa se encuentran unos iconos de antenas que indican la posición real de donde se encuentran aproximadamente los *eNodeBs* de la red en ese sector.



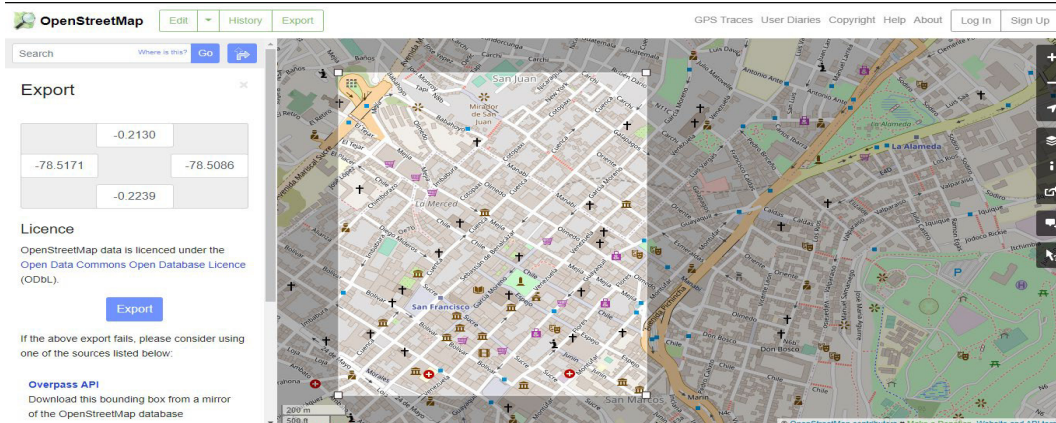
**Figura 2.21.** Cobertura LTE CONECEL S.A [36]

Una vez que se ha determinado el sector en donde se implementa la HETNET vehicular, se genera un mapa con la ayuda de la herramienta online OSM la cual se detalla en la siguiente sección.

### 2.3.2 GENERACIÓN DE MAPA

Con la herramienta online OSM se selecciona en el mapa de Ecuador (cargado por defecto en la herramienta OSM), el área que se usará en la evaluación de la HETNET, en la Figura

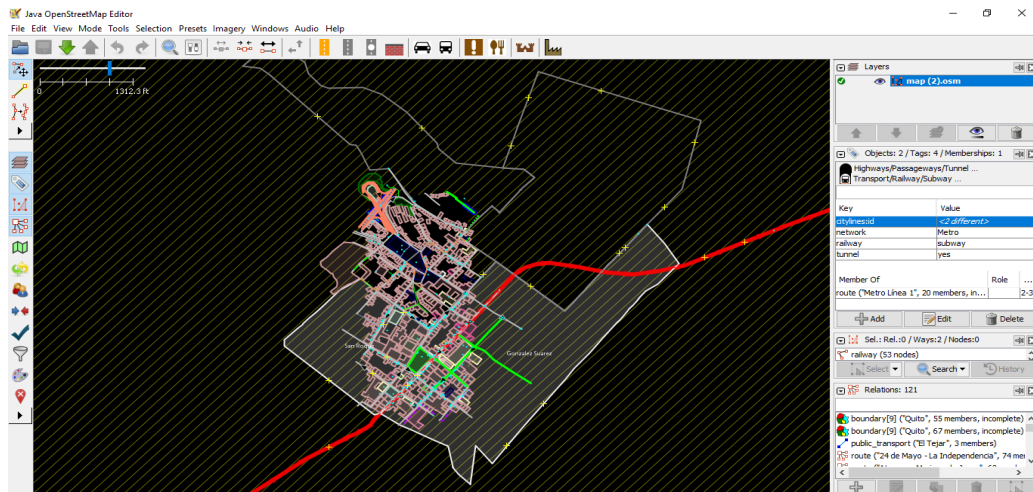
2.22 se muestra entre proceso, en donde se selecciona el área que corresponde al análisis.



**Figura 2.22.** Selección de sector con OSM [37].

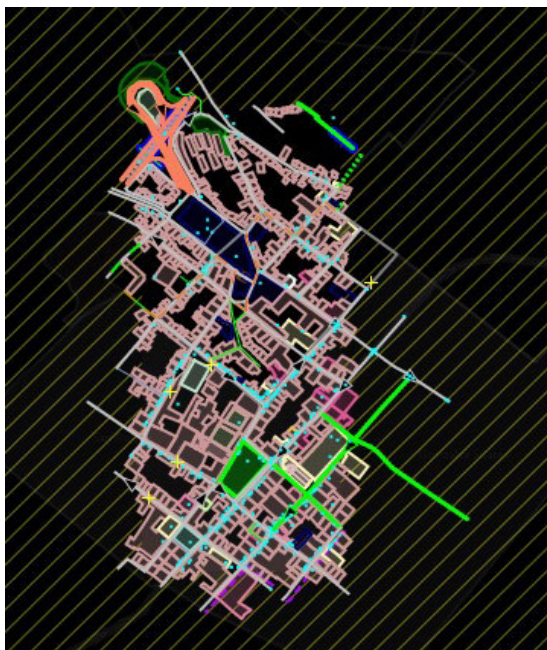
Posteriormente, se exporta el mapa en un archivo con extensión “.osm”, el cual puede ser editado con JOSM en el caso de que sea necesario eliminar alguna vía, objeto, etc. que no ayude con la simulación del flujo vehicular.

Al generar el mapa se observa que este tiene la vía subterránea del metro de Quito que se representa en color rojo como se muestra en la Figura 2.23. Esta vía no aporta en nada a la simulación del flujo vehicular y genera complicaciones cuando se inserta tráfico vehicular en el sector seleccionado del centro de Quito.



**Figura 2.23.** Mapa generado con OSM.

Se elimina la vía del metro con JOSM. El mapa resultante es el que se indica en la Figura 2.24.



**Figura 2.24.** Mapa editado con JOSM.

Con el mapa editado es posible introducir tráfico vehicular en la zona. Los detalles se explican en la siguiente sección.

### 2.3.3 INSERCIÓN DE FLUJO VEHICULAR

El software que introduce y simula el flujo vehicular es SUMO, pero se requieren algunos archivos con el formato adecuado para que funcione.

Al mapa modificado por JOSM con extensión “.osm”, se lo convierte a un archivo con extensión “.net.xml”, que es compatible con SUMO, para esto se utiliza el siguiente comando en una ventana terminal de Linux.

```
netconvert --osm-files centro.osm -o centro.net.xml
```

Para agregar los obstáculos, semáforos y demás elementos que existen en las vías, se crea un archivo con extensión “.poly.xml” a través del comando:

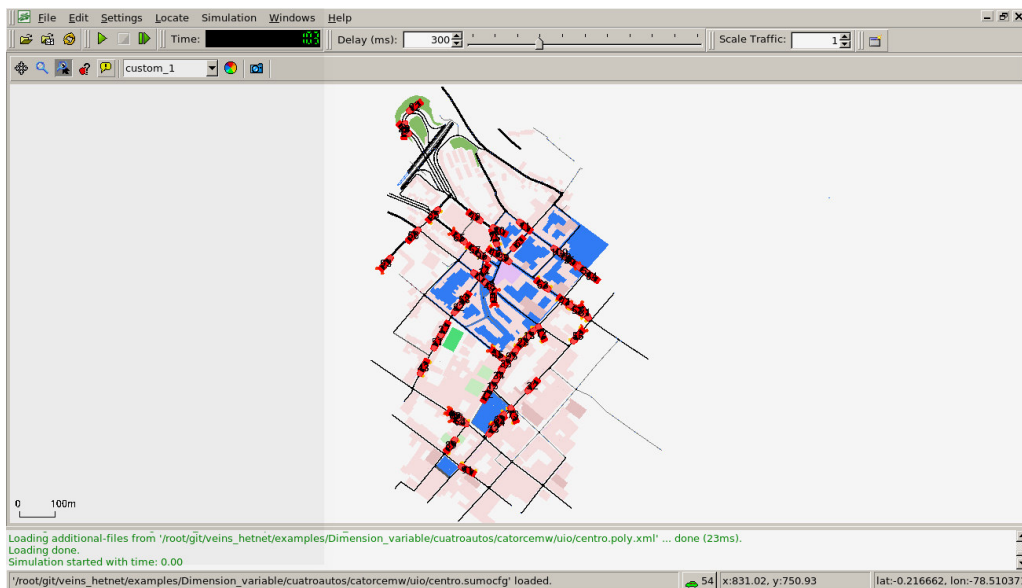
```
polyconvert --net-file centro.net.xml --osm-files centro.osm --type-file typemap.xml -  
o centro.poly.xml
```



Una vez se tengan los archivos con formato “net.xml” y “poly.xml”, ya es posible insertar el flujo vehicular. Para esto se utiliza el script de Python proporcionado por SUMO denominado “randomTrips.py”, el cual se ejecuta con los siguientes comandos.

```
python /home/veins/src/sumo/tools/randomTrips.py -n centro.net.xml -e 100 -l
python /home/veins/src/sumo/tools/randomTrips.py -n centro.net.xml -r
centro.rou.xml -e 100 -l
```

Con la línea de comandos anterior, se colocan 100 vehículos en la simulación, cada uno sigue una ruta distinta, y se obtiene un archivo con formato “.rou.xml”, en donde se tiene el tiempo de ingreso y dirección de cada automóvil. En la Figura 2.25 se observan algunos vehículos circulando en el mapa mientras que en la Figura 2.26 se comprueba que la cantidad de automóviles introducidos a la simulación llega a la cantidad de 100.



**Figura 2.25.** Simulación vehicular en SUMO.

network: Parameter			
Name	Value	Dynamic	
loaded vehicles [#]	100		
insertion-backlogged vehicles [#]	2		
departed vehicles [#]	73		
running vehicles [#]	54		
arrived vehicles [#]	19		
discarded vehicles [#]	0		
collisions [#]	0		

**Figura 2.26.** Cantidad de vehículos introducidos en la generación de tráfico vehicular.

### 2.3.4 SELECCIÓN DE RUTAS

Para configurar las rutas que seguirán las cinco (5) caravanas de vehículos, se modifican los archivos que se crean con las herramientas de SUMO.

El principal archivo que contiene las rutas termina en “.rou.xml” y contiene la identificación del automóvil, el tiempo de aparición en la simulación y las coordenadas de partida, llegada y por donde debe girar el vehículo.

En la Figura 2.27 se puede apreciar el código que genera SUMO en el archivo con extensión “.rou.xml”.

```
<vehicle id="0" depart="0.00">
  <route edges="351942413#0 351942413#1 579621131 31910404#0 31910404#1 31910404#2 -420910321 420910342#0 420910342#1 420910342#2
420910334 420910331 24679585" />
</vehicle>
<vehicle id="1" depart="1.00">
  <route edges="353552123#2 353552123#3 352907817#0 352907817#1 351942413#0 351942413#1 579621131 31910395#0 31910395#1 31910394#0
31910394#1 61897521" />
</vehicle>
<vehicle id="3" depart="3.00">
  <route edges="31297392 420910321 420910333 24918887#0 24918887#1 24918887#2 353552123#1 353552123#2 353552123#3 36739013 431750569#0" />
</vehicle>
<vehicle id="4" depart="4.00">
  <route edges="354423304#1 354423304#2 351942429#0 351942429#1 425625696 352907817#1 351942413#0 351942413#1 579621131 31910404#0
31910404#1 31910404#2 -420910321 420910342#0 420910342#1 24734053#0 24734053#1 24734053#2 420910337" />
</vehicle>
<vehicle id="5" depart="5.00">
  <route edges="128476278 431924426#0 421483699#2 351942429#1 425625696 352907817#1 351942413#0 351942413#1 351942413#2 351942414#0
351942414#1 425431676 425431688 31910453 -420910312 -420910337 -370247381" />
</vehicle>
<vehicle id="6" depart="6.00">
  <route edges="60355155#0 60355155#1 422450530#1 420910335 -420910342#2 -420910342#1 -420910342#0 420910321 420910333 24918887#0
24918887#1 351942414#1 425431676 425431688 31910441" />
</vehicle>
<vehicle id="7" depart="7.00">
  <route edges="31281345 579621131 31910404#0 31910404#1 31910404#2 -420910321 420910342#0 420910342#1 420910342#2 420910334 420910331
431718512" />
</vehicle>
<vehicle id="8" depart="8.00">
  <route edges="354423304#2 351942429#0 351942429#1 425625696 352907817#1 351942413#0 351942413#1 579621131" />
</vehicle>
<vehicle id="11" depart="11.00">
  <route edges="354423304#0 31910357#1 31910357#2 351942413#1 351942413#2 351942414#0 351942414#1 425431676 425431688 31910426 31910413" />
</vehicle>
```

Figura 2.27. Porción de código de archivo con extensión “.rou.xml”

Dentro del archivo “.rou.xml” existen los siguientes parámetros:

- **Id**: es el identificativo de un vehículo que se inserta en la simulación.
- **depart**: es el tiempo de aparición de un vehículo en la simulación.
- **edges**: indican las coordenadas de las esquinas de las calles por donde el vehículo debe curvar. En el parámetro **edges**, la primera coordenada que aparece indica el punto de partida del automóvil y así mismo la última coordenada indica el punto de llegada del automóvil.

Para cambiar las rutas, es decir editar los puntos de partida y llegada, se debe escoger solo aquellos automóviles dentro del archivo “.rou.xml” que no compartan coordenadas de inicio y llegada similares.

Una vez, se tengan identificadas las rutas a modificar, se pueden crear las caravanas de vehículos al modificar ciertos comandos dentro del archivo con extensión “.rou.xml” como se muestra en la Figura 2.28.

```

<route id="0" edges="24679580 422450527 422450530#0 422450530#1 420910335 -420910342#2 -420910342#1 -420910342#0 420910321 420910333
24918887#0 24918887#1 24918887#2"/>
<route id="1" edges="-61897521 420910317 24680727 31910391 24679590 354423304#0 354423304#1 354423304#2"/>
<route id="2" edges="36739013 431750569#0 431750569#1 541724221 431920752"/>
<route id="3" edges="353556189 425431687 24918887#1 24918887#2 353552123#1 353552123#2 353552123#3 36739013 128476278"/>
<route id="4" edges="354423304#2 351942429#0 351942429#1 351942429#2 431750569#0 431750569#1 541724221 431920751"/>

<flow id="0" begin="0" end="1" number="4" route="0" type="car"/>
<flow id="1" begin="0" end="1" number="4" route="1" type="car"/>
<flow id="2" begin="0" end="1" number="4" route="2" type="car"/>
<flow id="3" begin="0" end="1" number="4" route="3" type="car"/>
<flow id="4" begin="0" end="1" number="4" route="4" type="car"/>

```

**Figura 2.28.** Creación de caravanas de automóviles.

Se puede apreciar que algunos parámetros se eliminan o se reemplazan por otros, por ejemplo, ahora se coloca una identificación en las rutas y también en el flujo de vehículos.

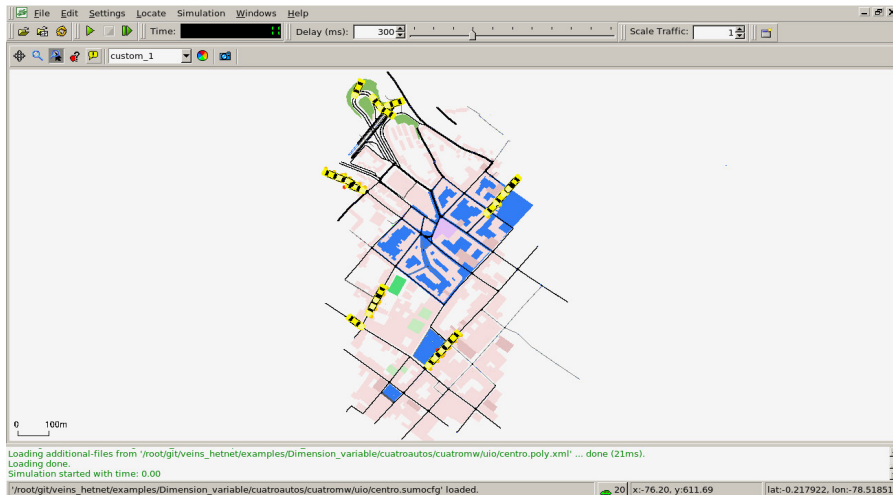
En las cinco (5) rutas que siguen las caravanas de vehículos, se tienen coordenadas distintas por donde los automóviles se van a desplazar.

Con respecto al código que permite crear las caravanas de vehículos, se tiene lo siguiente:

- *begin*: tiempo de aparición de los vehículos de ese flujo.
- *end*: intervalo de aparición entre vehículos del flujo.
- *number*: número de vehículos del flujo.
- *route*: ruta por donde se movilizarán los vehículos del flujo.
- *type*: tipo de vehículo.

Cada flujo de vehículos será una caravana de automóviles en donde todos los miembros compartan una ruta específica, además de un tiempo de aparición periódico.

En la Figura 2.29, se puede observar que existen 5 distintas caravanas de automóviles que se encuentran dispersadas en distintos puntos del mapa.



**Figura 2.29.** Creación de caravanas de automóviles.

En este ejemplo, se coloca cuatro automóviles dentro de cada caravana y esto se puede corroborar tanto visualmente en la simulación presentada en la Figura 2.29, así como también en el código del archivo “.rou.xml” que se encuentra en la Figura 2.28 en donde se coloca

“*number* = “4” ”.

Para cambiar la cantidad de vehículos dentro de cada caravana se deberá cambiar el valor del parámetro “*number*” del archivo “.rou.xml”.

### 2.3.5 DISEÑO E IMPLEMENTACIÓN DE HETNET

En el portal *GitHub*, el usuario Christoph Sommer [38] comparte el código de un proyecto que pretende generar una HETNET, en donde se envían y reciben paquetes de alerta en los vehículos por medio de las interfaces WLAN y LTE.

El código de la HETNET se encuentra en el archivo “omnet.ini” y se muestra en el anexo A.

Para llevar a cabo este proyecto, se toma como punto de partida el código de Sommer [35]. Se añaden características propias de este trabajo, en la HETNET vehicular uno solo de los vehículos del grupo recoge información de los automóviles de las caravanas por medio de WLAN y envía esa información a través de LTE hacia otras caravanas de vehículos.

El código completo modificado se lo expone en el Anexo I. En esta sección se deja una parte del código correspondiente a la configuración de las IPs.

```

#####
#                               NODE APP Transmitter node[0]           #
#####
*.car[*].numUdpApps = 1
*.car[*].udpApp[0].typename = "org.car2x.veins.subprojects.veins_hetvnet.HetVNetDemoApp"
*.car[*].udpApp[0].localPortLte = ${appcarport=1000}
*.car[*].udpApp[0].localPortWlan = 1001
*.car[*].udpApp[0].destAddressLte = "224.0.0.10"

###PRIMER CLUSTER#####
*.car[0].udpApp[0].destAddressWlan = "224.0.2.1"

###SEGUNDO CLUSTER#####
*.car[1].udpApp[0].destAddressWlan = "224.0.2.11"

###TERCER CLUSTER#####
*.car[2].udpApp[0].destAddressWlan = "224.0.2.21"

###CUARTO CLUSTER#####
*.car[3].udpApp[0].destAddressWlan = "224.0.2.31"

###QUINTO CLUSTER#####
*.car[4].udpApp[0].destAddressWlan = "224.0.2.41"

*.car[*].udpApp[0].DstAppServerPort = ${appserverport}
# user the upd app server port

#####

#                               MULTICAST                               #
#
#####

*.car[*].configuratorLte.interfaces = "wlan"
*.car[*].configuratorWlan.interfaces = "wlan0"

```

```

*.car[*].configuratorLte.addressBase = "10.0.0.0"
*.car[*].configuratorWlan.addressBase = "2.0.0.0"
*.car[*].configuratorLte.mcastGroups = "224.0.0.10" # Add nodes to the LTE multicast group

*.car[0].configuratorWlan.mcastGroups = "224.0.2.1" # Add nodes to the WLAN multicast group
*.car[1].configuratorWlan.mcastGroups = "224.0.2.11" # Add nodes to the WLAN multicast group
*.car[2].configuratorWlan.mcastGroups = "224.0.2.21" # Add nodes to the WLAN multicast group
*.car[3].configuratorWlan.mcastGroups = "224.0.2.31" # Add nodes to the WLAN multicast group
*.car[4].configuratorWlan.mcastGroups = "224.0.2.41" # Add nodes to the WLAN multicast group

```

En esta sección código se establece una dirección IP de *multicast* a uno de los automóviles de la caravana el cual es definido como CH (*Car Header*). Esto se encuentra en la sección “*MULTICAST*” y la IP se la define en el parámetro “\*.car[n].configuratorWlan.mcastGroups” donde n es el identificativo del automóvil que se desea establecer.

Para que los demás automóviles envíen la información hacia esta IP de *multicast* se lo hace configurando el parámetro “\*.car[n].udpApp[0].destAddressWlan = "IP destino" que se encuentra en la sección de “*NODE APP Transmitter*”.

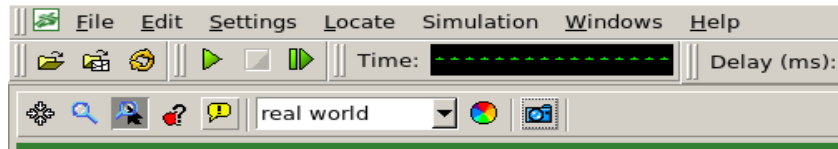
Para que se genere el reenvío LTE hacia el servidor se modifica el parámetro

“\*.car[n].udpApp[0].Forwarding\_nodeld = n “ en donde n es el identificativo del automóvil. En este parámetro se deberá colocar el número del automóvil CH que pertenece a cada caravana.

### 2.3.6 CONFIGURACIÓN DE LA HETNET

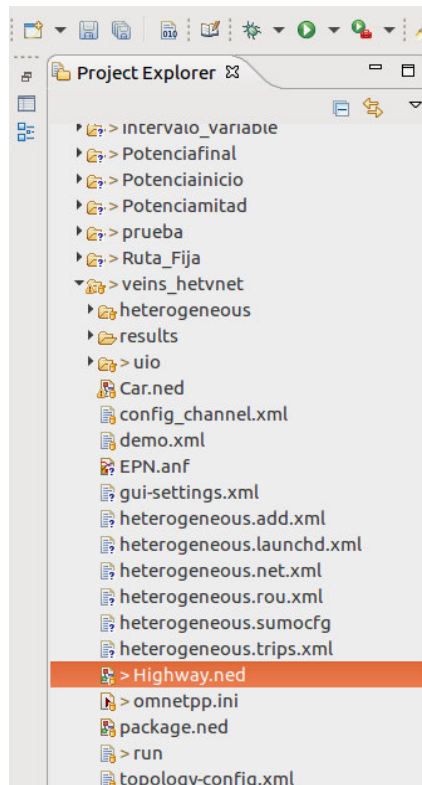
Para configurar a la HETNET de caravanas de vehículos dentro del sector del centro histórico de Quito, se debe considerar la Figura 2.21, en donde se indica el mapa de cobertura de la red LTE del operador CONECEL y la posición aproximada de los “*eNodeBs*”.

Adicionalmente, para gestionar y colocar de manera correcta las estaciones base LTE dentro de la zona de la simulación de la HETNET en OMNET++, se puede obtener una imagen del mapa con extensión .png a través de SUMO, para el cual se debe dar clic en el ícono de cámara denominado “*Save Snapshot*”, como se muestra en la Figura 2.30.



**Figura 2.30.** *Save Snapshot.*

Una vez se tiene la imagen del mapa con extensión .png, se ingresa a la carpeta de OMNET++ que contiene la HETNET y se busca el archivo “Highway.ned” como se indica en la Figura 2.31.

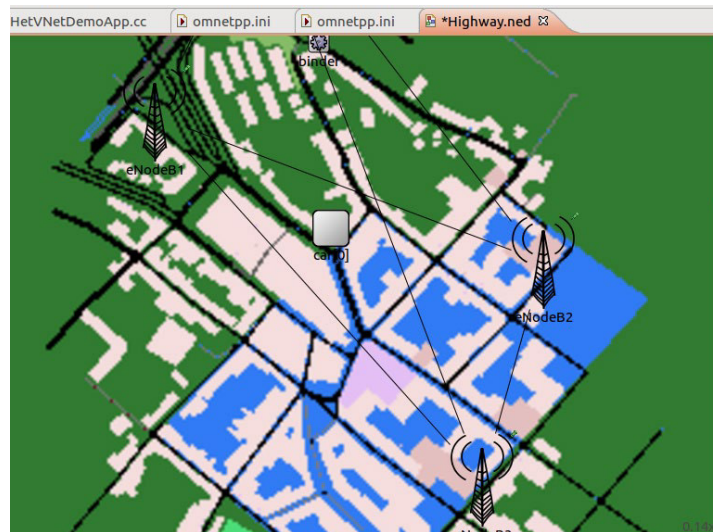


**Figura 2.31.** Highway.ned

En el archivo “Highway.ned” se coloca la ruta de la imagen del mapa con extensión png, en este caso, se encuentra dentro de la carpeta “*background*” del directorio *veins\_hetnet* y la imagen del mapa tiene el nombre “centro1.png”. En la siguiente porción de código, se observa el comando para añadir la imagen del mapa a OMNET++.

```
@display("i=block/network2;bgb=5583.7134,5139.554;bgi=background/centro1");
```

Cuando ya se carga la imagen del mapa a OMNET++, es posible configurar la posición de las estaciones bases mediante una interfaz gráfica como se indica en la Figura 2.32.



**Figura 2.32.** Interfaz gráfica de OMNET++.

También, es posible configurar los elementos de la HETNET mediante líneas de código, pero para el presente trabajo, se utiliza esta segunda opción para unir las estaciones base mediante interfaces X2 como se puede apreciar a continuación:

```
connections allowunconnected:

server.pppg++ <--> Eth10G <--> router.pppg++;
router.pppg++ <--> Eth10G <--> pgw.filterGate;
pgw.pppg++ <--> Eth10G <--> eNodeB1.ppp;
pgw.pppg++ <--> Eth10G <--> eNodeB2.ppp;
pgw.pppg++ <--> Eth10G <--> eNodeB3.ppp;

eNodeB1.x2++ <--> Eth10G <--> eNodeB2.x2++;
eNodeB1.x2++ <--> Eth10G <--> eNodeB3.x2++;

eNodeB2.x2++ <--> Eth10G <--> eNodeB3.x2++;
```

En la porción de código que se muestra arriba, se agregan los eNodeBs a la simulación y se conectan entre si mediante las interfaces X2.



Con las herramientas de SUMO, se obtienen algunos archivos que contienen la ruta de los vehículos y la posición de calles, avenidas, edificios y demás objetos que aparecen en la realidad en una ciudad.

En la sección 2.3.4 se generan y escogen cinco rutas por donde van a moverse los vehículos, entonces se importan esos archivos a OMNET++. Para ello se debe crear un archivo con extensión “.launch.xml” y colocar lo siguiente.

```
<launch>
  <copy file="centro.net.xml"/>
  <copy file="centro.rou.xml"/>
  <copy file="centro.poly.xml"/>
  <copy file="centro.sumocfg" type="config"/>
  <copy file="gui-settings.xml"/>

  <!-- Update SUMO files directory -->
  <basedir path="/root/git/veins_hetnet/examples/veins_hetvnet/ui/" />
</launch>
```

Con esto se cargan todos los archivos creados por las herramientas de SUMO y se especifica el directorio en donde se encuentran.

Una vez se haya creado el archivo “.launch.xml”, se lo invoca mediante el comando:

```
*.veinsManager.launchConfig = xmldoc("ui/centro.launchd.xml")
```

que se coloca en el archivo omnet.ini que contiene los comandos necesarios para la ejecución de la HETNET que se especifica en el punto 2.3.5 Diseño de la HETNET.

### 2.3.7 DISEÑO DE ESCENARIOS

Para evaluar el comportamiento de la HETNET de caravanas de vehículos en un determinado sector del centro histórico de Quito, es necesario crear escenarios en donde se modifiquen parámetros físicos y/o técnicos de la red.

En este trabajo de titulación, se crearán dos escenarios los cuales se detalla a continuación.

- Primer Escenario

Aquí los parámetros que cambiarán son:

- La potencia de transmisión de los vehículos para comunicación inalámbrica de corto alcance IEEE 802.11p.
- La posición del CH (*Car Header*) dentro de la caravana de vehículos.

Se escogen estos parámetros porque el éxito de transmisión de datos depende de las características del transmisor y el receptor. Si la potencia de transmisión y la sensibilidad del receptor no son las adecuadas, existirán pérdidas de paquetes considerables. Por lo que mediante la variación de la potencia de transmisión para el estándar IEEE 802.11p, se busca determinar el valor adecuado para una correcta transmisión de paquetes.

El segundo punto a consideración es la posición del CH dentro de la caravana. Es necesario determinar en qué posición de la caravana se debe situar al automóvil cabecera para que no exista paquetes perdidos que provienen de los vehículos circundantes.

Los dos parámetros mencionados anteriormente repercuten en el comportamiento LTE de la red, ya que si no llega cierta cantidad de paquetes IEEE 802.11p al CH, no se activará la retransmisión de información de la caravana hacia el servidor LTE y generará también una pérdida de paquetes en el estándar de comunicación celular.

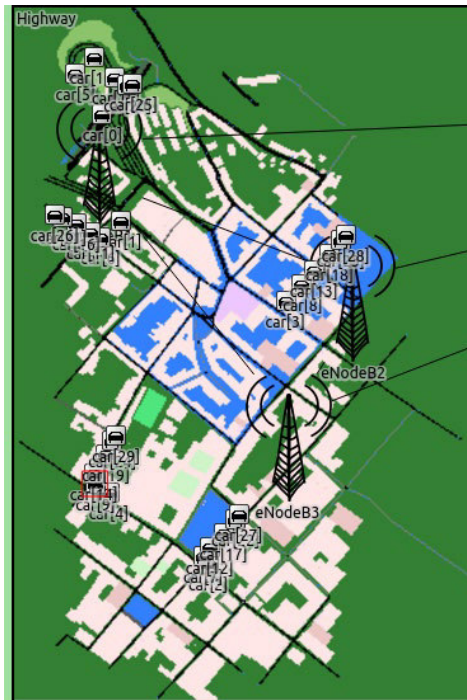
Para generar este escenario hay que modificar algunos comandos en el código principal que se menciona en la sección 2.3.5 Diseño de la HETNET.

Para modificar la potencia de transmisión, se debe modificar los valores numéricos de la siguiente línea de código que se encuentra en la sección CAR WLAN:

```
*.car[*].wlan[0].radio.transmitter.power = 14mW
```

En el caso de la posición del CH, el proceso es un poco más largo ya que se necesitan modificar algunas líneas de comando.

Primero, se debe identificar el número de automóvil que se desea colocar como CH dentro de la caravana, para ello se debe correr la simulación de la HETNET de caravanas de vehículos que se desplazan en un sector del centro histórico de Quito, como se observa en la Figura 2.33.



**Figura 2.33.** Simulación de HETNET vehicular.

Cuando ya se identifica al automóvil que cumplirá la función de CH, se coloca su número dentro del corchete en la siguiente línea de comando que se encuentra en la sección “MULTICAST”.

```
*.car[0].configuratorWlan.mcastGroups = "224.0.2.1"
```

Con esto se define una IP de *multicast* a donde los demás vehículos de la caravana deben enviar paquetes IEEE 802.11p, pero también hay que especificar y crear las caravanas. Esto se lo realiza en la sección “NODE APP Transmitter node[0]”.

Aquí se coloca el identificativo de todos los automóviles que conforman una caravana y se indica hacia que IP deben enviar los paquetes de información. La línea de comando que se modifica es la siguiente:

```
*.car[0].udpApp[0].destAddressWlan = "224.0.2.1"
```

El número de vehículo, se lo coloca dentro del corchete. Para ilustrar de mejor manera esta sección se coloca un ejemplo de la creación de una caravana.

```
###PRIMERA CARAVANA#####
*.car[0].udpApp[0].destAddressWlan = "224.0.2.1"
*.car[5].udpApp[0].destAddressWlan = "224.0.2.1"
```

```
*.car[10].udpApp[0].destAddressWlan = "224.0.2.1"  
*.car[15].udpApp[0].destAddressWlan = "224.0.2.1"
```

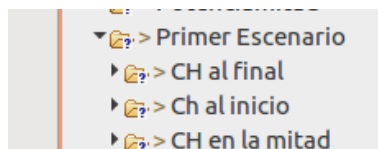
Los identificativos de los automóviles, se los obtiene al simular la HETNET como se muestra en la Figura 2.33 y la IP que se coloca en esta caravana es la misma, ya que el vehículo cabecera tiene esa IP.

Al crear una nueva caravana, se identifican los vehículos que pertenecen a dicha caravana y se coloca la IP del otro vehículo que fungirá como CH.

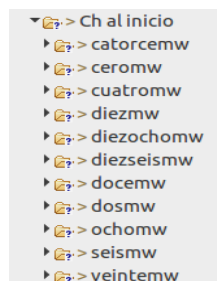
Para este escenario, se consideran 10 vehículos por caravana siempre, y primero se dejará la ubicación del CH fija, mientras varía la potencia de transmisión IEEE 802.11p que irá desde los 2 [mw] hasta los 20 [mw] saltando 2 [mw] en cada simulación.

Por ejemplo, se elige al vehículo del inicio de la caravana como CH y se simula esto con la potencia de transmisión variable desde los 2 [mw] hasta llegar a los 20 [mw]. Luego se cambia la posición del CH a la mitad de la caravana y se realizan las mismas simulaciones variando la potencia de transmisión. Finalmente, se cambia la posición del CH hacia el final de la caravana y se simula cambiando la potencia de transmisión.

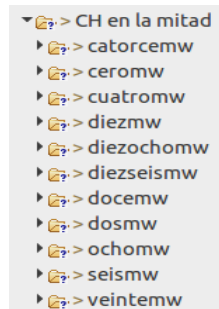
En las Figuras 2.34, 2.35, 2.36 y 2.37 se aprecia los directorios en donde se encuentran los archivos que permiten la simulación de los escenarios.



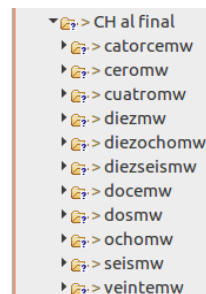
**Figura 2.34.** Establecimiento del primer escenario.



**Figura 2.35.** Establecimiento del primer escenario con CH al inicio de la caravana.



**Figura 2.36.** Establecimiento del primer escenario con CH en la mitad de la caravana.



**Figura 2.37.** Establecimiento del primer escenario con CH al final de la caravana.

Por cada posición del CH, se generan 10 simulaciones con un total de 30 simulaciones considerando que se cambia 3 veces la posición del CH.

También, se define en qué momento de la simulación aparece el CH con el objetivo de que él envío de paquetes empiece una vez aparezca dicho vehículo y no haya pérdidas de paquetes innecesarias debido a la constante transmisión de los automóviles de la caravana hacia ningún destino.

El tiempo en el que sale el CH en la simulación también es algo para tener en cuenta para que la simulación acabe y tampoco haya paquetes perdidos por no contar con el objetivo a donde enviar.

Tanto la premisa de empezar la transmisión de paquetes cuando el CH aparezca en la simulación y de parar la simulación cuando el CH salga de esta, brindará resultados dignos de análisis porque se contará con similitud de condiciones para las simulaciones.

- Segundo Escenario.

En este escenario, se toma como parámetros variables a:

- El intervalo de envío de mensajes IEEE 802.11p.
- La cantidad de vehículos que pertenecen a cada caravana de vehículos.

Se debe dejar en claro que para este escenario el CH permanecerá al inicio de cada caravana o grupo de vehículos.

En el estándar IEEE 802.11p, se considera el intervalo de tiempo para enviar paquetes de 500 ms [2], sin embargo, revisar el comportamiento de la HETNET con distintos valores de tiempo determinará la eficiencia de la red, ya que al enviar información en distintos intervalos, generará distinta cantidad de paquetes totales enviados que tienen distinto porcentaje de pérdida en el camino por motivo de desvanecimiento u otro efecto de las comunicaciones inalámbricas [39].

Es por ello, por lo que en este escenario se plantea tener simulaciones en donde se empieza con 100 [ms] y aumenta otros 100 [ms] en cada simulación hasta terminar en 1 [s]. Con esto se analiza el comportamiento de la red cuando se aumenta el tiempo del intervalo de envío de mensajes.

A la variación del intervalo de envío de mensajes se la combina con la variación de la cantidad de vehículos que pertenecen a cada caravana. Esto se simula con el objetivo de saber que tan eficiente puede ser la red dependiendo del número de vehículos que se vean involucrados, ya que mientras más vehículos existan en la simulación, más paquetes IEEE 802.11p se enviarán por lo que también sube la cantidad de paquetes perdidos.

Estas dos variaciones tendrán una repercusión en la red móvil LTE, porque al igual que se menciona en el primer escenario, mientras no llegue un paquete IEEE 802.11p de cada vehículo de la caravana hacia el CH, no existirá el reenvío de la información recolectada hacia el servidor LTE.

Para modificar el parámetro del intervalo de envío de mensajes IEEE 802.11p, se cambia el valor numérico del siguiente comando que se encuentra en el código principal:

```
*.car[*].udpApp[0].period = 0.1s
```

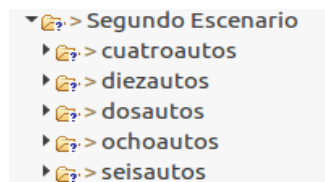
Para modificar la cantidad de vehículos que pertenecen a cada caravana, se lo hace en el código del archivo con extensión "rou.xml" que se obtiene de SUMO y que contiene la simulación vehicular.

```
<flow id="0" begin="0" end="1" number="4" route="0" type="car"/>  
<flow id="1" begin="0" end="1" number="4" route="1" type="car"/>  
<flow id="2" begin="0" end="1" number="4" route="2" type="car"/>  
<flow id="3" begin="0" end="1" number="4" route="3" type="car"/>  
<flow id="4" begin="0" end="1" number="4" route="4" type="car"/>
```

En el extracto de código mostrado arriba se disponen las 5 rutas que se definen anteriormente y en el parámetro “*number*” se encuentra la cantidad de automóviles que se agregan a cada ruta. Para cambiar esa cantidad de vehículos, se debe modificar el valor numérico de “*number*” lo que generara que haya más o menos vehículos en la simulación.

Al igual que en el primer escenario, se deja uno de los parámetros fijos momentáneamente, que en este caso es el número de automóviles por caravana. Se empieza con 2 vehículos por caravana y se incrementan en 2 hasta llegar a tener 10 automóviles. El parámetro que va a cambiar para cada caso es la variación de envío de mensajes IEEE 802.11p que empezará en 100 [ms] y se incrementará 100 [ms] en cada simulación hasta llegar a 1 [s].

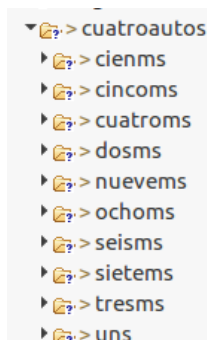
En la Figura 2.38, 2.39, 2.40, 2.41, 2.42 y 2.43 se puede observar los directorios en donde se encuentran los archivos necesarios para ejecutar las simulaciones de los escenarios.



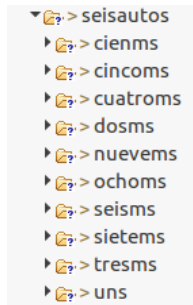
**Figura 2.38.** Establecimiento del segundo escenario.



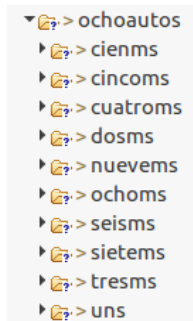
**Figura 2.39.** Establecimiento del segundo escenario para dos vehículos.



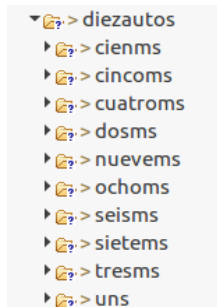
**Figura 2.40.** Establecimiento del segundo escenario para cuatro vehículos.



**Figura 2.41.** Establecimiento del segundo escenario para seis vehículos.



**Figura 2.42.** Establecimiento del segundo escenario para ocho vehículos.



**Figura 2.43.** Establecimiento del segundo escenario para diez vehículos.

Para cada caso del número de vehículos por caravana, se tiene un total de 10 simulaciones, lo que da en total 50 simulaciones, si se considera que hay dos, cuatro, seis, ocho y diez vehículos por caravana.

### 2.3.8 VINCULACIÓN DE SUMO CON OMNET++

Para lograr la vinculación de la simulación de tráfico vehicular con la simulación de la red, se necesita de la librería VEINS. Como ya se indicó en la sección 2.2.4, todas estas librerías ya se instalaron por lo que para su ejecución basta con ingresar una línea de comando que es la siguiente:

```
/root/sumo-launchd.py -vv /opt/sumo-1.5.0/bin/sumo
```

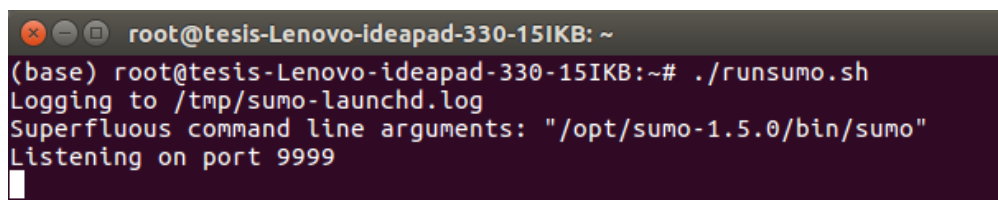


En donde primero, se especifica el directorio en donde se encuentra el script de *Python* que contiene los comandos necesarios para hacer que tanto SUMO y OMNET++ funcione al mismo tiempo, además, se especifica la carpeta en donde se encuentra el ejecutable de SUMO.

Para simplificar un poco esta línea de código, se puede generar un script de Linux que se encargue de ejecutar la línea de comando anteriormente presentada. Para ello en gedit, se procede a colocar la línea de comando que se necesita colocar y se guarda ese archivo con extensión “.sh”, en este caso el archivo se llamará “runsumo.sh” y para ejecutarlo basta con tipear

```
./runsumo.sh
```

Cuando se ejecute la línea de comando aparecerá una ventana que indica que el puerto 9999 se encuentra habilitado y escuchando a OMNET++. En la Figura 2.44 se evidencia lo del puerto 9999.



**Figura 2.44.** Activación del puerto 9999.

Al ejecutar la simulación en OMNET++, en el terminal de LINUX deberá aparecer una serie de líneas que indican que SUMO se encuentra también trabajando paralelamente junto con OMNET++.

Si se presenta lo de las Figuras 2.45 y 2.46, se puede determinar que la vinculación de la simulación vehicular y la simulación de red es exitosa.

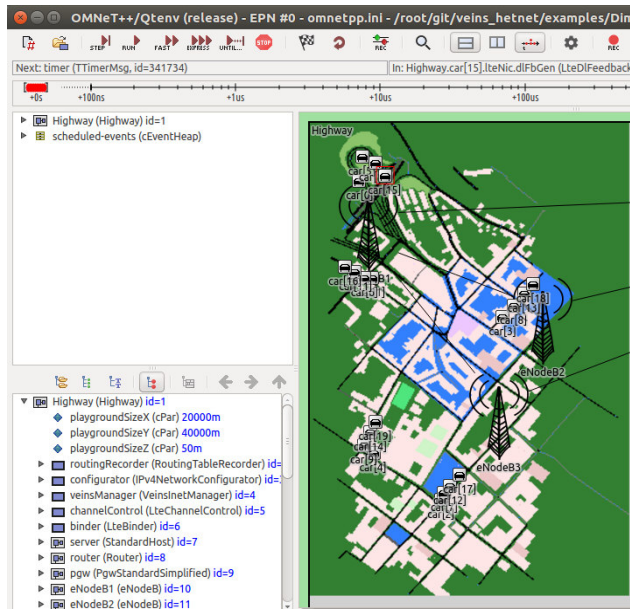


Figura 2.45. Vinculo exitoso SUMO - OMNET++.

```

root@tesis-Lenovo-ideapad-330-151KB: ~
(base) root@tesis-Lenovo-ideapad-330-151KB:~# ./runsumo.sh
Logging to /tmp/sumo-launchd.log
Superfluous command line arguments: "/opt/sumo-1.5.0/bin/sumo"
Listening on port 9999
Connection from 127.0.0.1 on port 56860
Handling connection from 127.0.0.1 on port 56860
Got TraCI message of length 2
Got TraCI command of length 1
Got TraCI command 0x0
Got CMD_GETVERSION
Got TraCI message of length 334
Got TraCI command of length 329
Got TraCI command 0x75
Got CMD_FILE_SEND for "sumo-launchd.launch.xml"
Got CMD_FILE_SEND with data "<launch>
<copy file="centro.net.xml"/>
<copy file="centro.rou.xml"/>
<copy file="centro.poly.xml"/>
<copy file="centro.sumocfg" type="config"/>
<copy file="gui-settings.xml"/>
<basedir path="/root/git/veins_hetnet/examples/Potenciamitad/catorcemw/uo"/>
<seed value="0"/>
</launch>"
Creating temporary directory...
Temporary dir is /tmp/sumo-launchd-tmp-MSRFed
Base dir is /root/git/veins_hetnet/examples/Potenciamitad/catorcemw/uo/
Seed is 0
Finding free port number...
Claiming lock on port
...found port 50451
Starting SUMO (sumo -c centro.sumocfg) on port 50451, seed 0
Connecting to SUMO (sumo -c centro.sumocfg) on port 50451 (try 1)
Error ([Errno 111] Connection refused)
Connecting to SUMO (sumo -c centro.sumocfg) on port 50451 (try 2)
Error ([Errno 111] Connection refused)
Connecting to SUMO (sumo -c centro.sumocfg) on port 50451 (try 3)
Releasing lock on port
Starting proxy mode

```

Figura 2.46. Vinculo exitoso SUMO - OMNET++.

### 3 RESULTADOS Y DISCUSIÓN

En este capítulo se detallan los resultados de las simulaciones de los escenarios descritos que se presentan en el subcapítulo 2.3.7 para su posterior análisis. Para presentar los resultados se desarrolló código en Python de tal manera que se pueda presentar de forma gráfica los resultados.

Para el Escenario 1 los resultados se presentan en términos de:

- Paquetes WLAN Tx vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.
- Paquetes WLAN Rx vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.
- % *Packet Delivery Ratio* WLAN vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.
- *Delay* WLAN vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.
- Paquetes LTE Tx vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.
- Paquetes LTE Rx vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.
- % Rendimiento LTE vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.
- *Delay* LTE vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.

En el Segundo escenario se tienen los resultados en función de:

- Paquetes WLAN Tx vs Intervalo de envío de mensajes IEEE 802.11p.
- Paquetes WLAN Rx vs Intervalo de envío de mensajes IEEE 802.11p.
- % *Packet Delivery Ratio* WLAN vs Intervalo de envío de mensajes IEEE 802.11p.
- *Delay* WLAN vs Intervalo de envío de mensajes IEEE 802.11p.
- Paquetes LTE Tx vs Intervalo de envío de mensajes IEEE 802.11p.
- Paquetes LTE Rx vs Intervalo de envío de mensajes IEEE 802.11p.
- % Rendimiento LTE vs Intervalo de envío de mensajes IEEE 802.11p.
- *Delay* LTE vs Intervalo de envío de mensajes IEEE 802.11p.

### 3.1 PROCESAMIENTO DE ESCENARIOS

En el capítulo 2 del presente trabajo de titulación se establecen dos escenarios dentro de los cuales se cambian parámetros físicos y/o técnicos de la HETNET y caravana de vehículos. En el ANEXO B se muestra un diagrama de cómo se ejecutan los escenarios y cómo cambian los parámetros de las simulaciones, además, se indica en función de que parámetros se generan las gráficas para evaluar la HETNET.

#### A) ESCENARIO 1

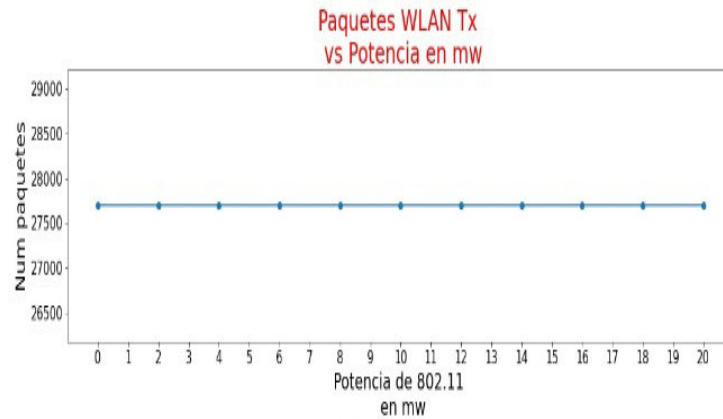
El escenario está formado por 5 caravanas de vehículos, cada caravana consta de 10 vehículos cada uno. En este escenario se considera como variables de estudio la posición del CH dentro de la caravana de vehículos y la potencia de transmisión de la OBU del vehículo, el cual se transmite por medio del estándar IEEE 802.11p. Para analizar la posición del CH se define que sea al inicio, en la mitad y al final de la caravana. Mientras que la potencia de transmisión se analiza que varía entre 0 mW hasta 20mW en pasos de 2mw. Al cambiar los parámetros se generan 30 simulaciones las cuales crean curvas que se presentarán en una misma gráfica.

Las curvas y las gráficas que se menciona anteriormente se las realiza por medio de un script de Python cuyo código se encuentra en el Anexo C.

Una vez que se tengan los resultados procesados de las 10 simulaciones se los coloca en una sola carpeta y se crea otro script de Python el cual unirá toda la información de los archivos con extensión .csv que se obtienen con el anterior código de Python y organizará estos datos con el fin de obtener un archivo con extensión .csv final que generará algunas gráficas estadísticas. El código de este script de *Python* se encuentra en el Anexo D.

Al ejecutar el script de *Python* se crean curvas como se puede observar en la Figura 3.1. El archivo de Python genera 7 figuras más similares a la Figura 3.1 pero con distintos parámetros en el eje de las ordenadas. Cabe resaltar que solo se presenta un ejemplo de cómo quedan las curvas al ejecutar el código de Python y más adelante se analiza de manera detallada estas curvas.

En este caso se realiza el ejemplo para la simulación en donde la posición del CH se mantiene fija al inicio de la caravana de vehículos y el parámetro que varía es la potencia del estándar IEEE 802.11p en cada simulación.

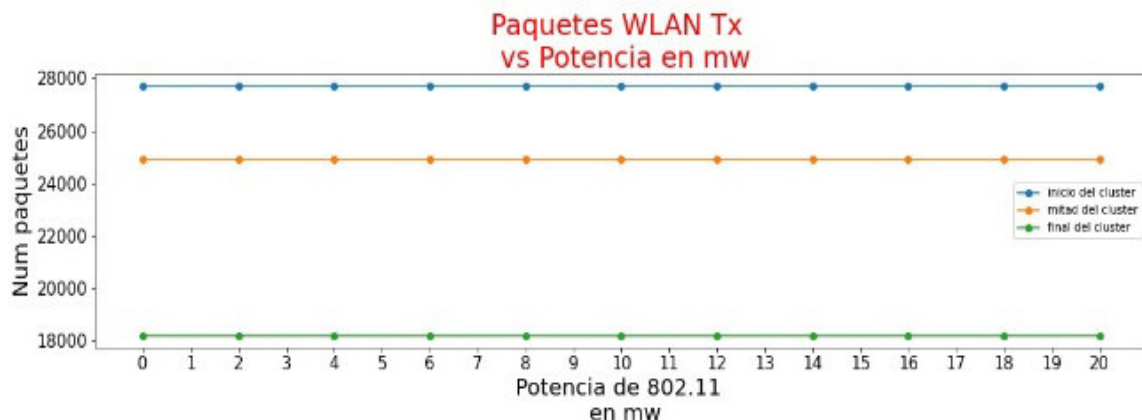


**Figura 3.1.** Paquetes WLAN Tx vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.

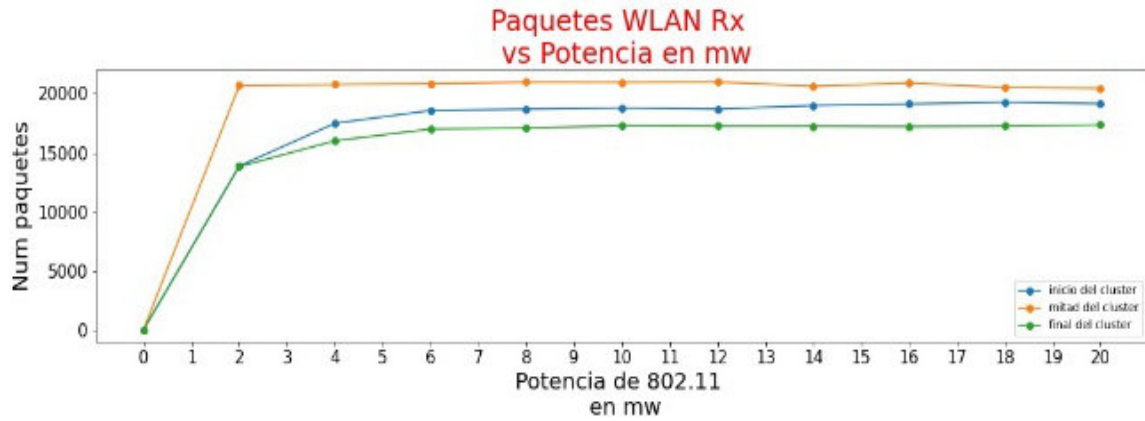
Para las simulaciones en donde se cambia la posición del CH a la mitad y al final de la caravana de vehículos el procedimiento se ejecuta con los mismos scripts, pero los resultados de las curvas van a ser distintos.

En el Anexo E se muestra el código del script de *Python* que permite superponer todas las curvas obtenidas de las simulaciones en una misma gráfica.

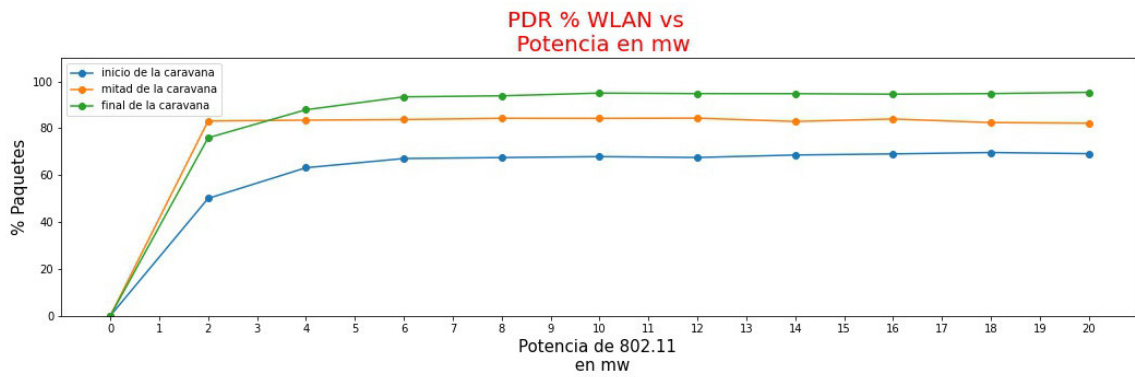
Cuando se ejecute el script se generan las Figuras 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8 y 3.9 que contienen las curvas superpuestas de los escenarios simulados. Se vuelve a recalcar que se muestran estas figuras solo como ejemplo ya que en las siguientes secciones se desglosa y se analiza de manera más detallada los resultados.



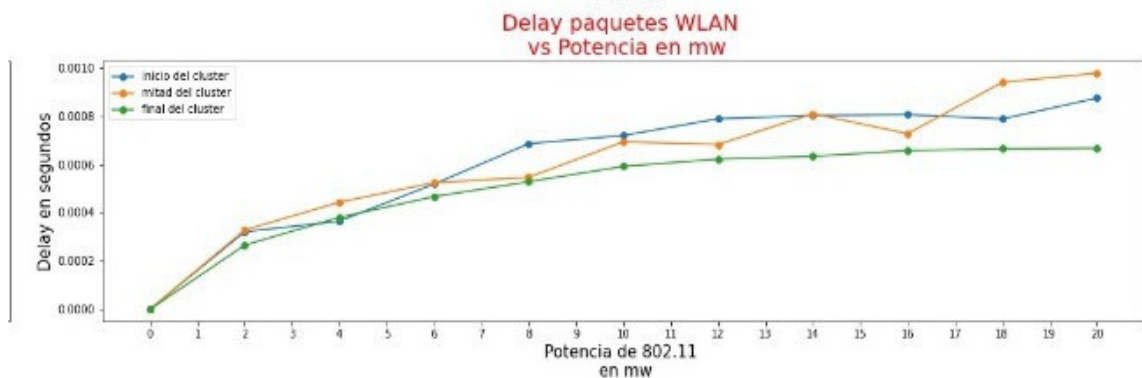
**Figura 3.2.** Paquetes WLAN Tx vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.



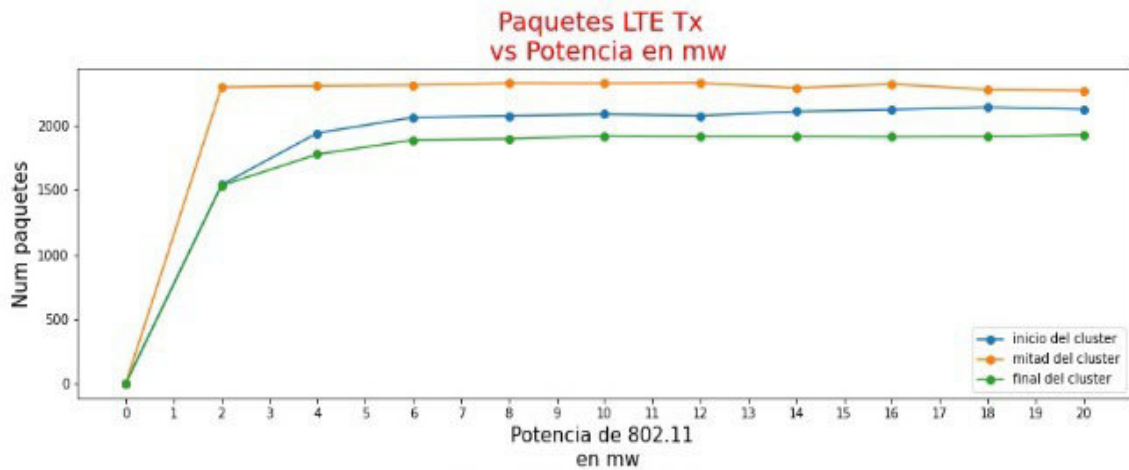
**Figura 3.3.** Paquetes WLAN Rx vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.



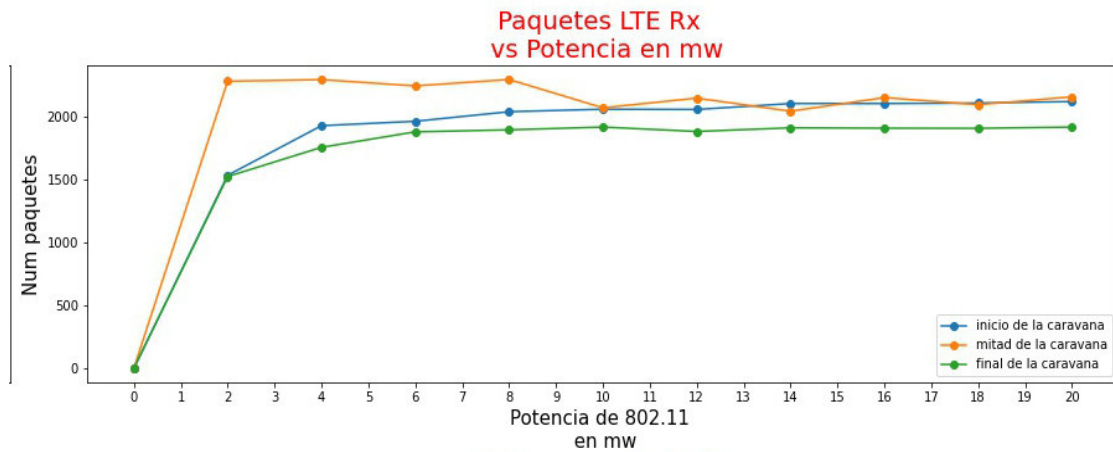
**Figura 3.4.** Packet Delivery Ratio WLAN vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.



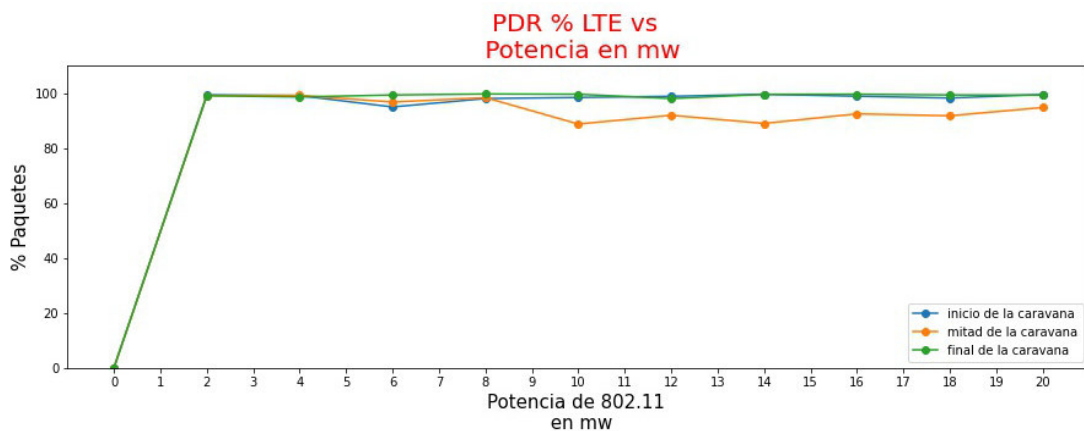
**Figura 3.5.** Delay WLAN vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.



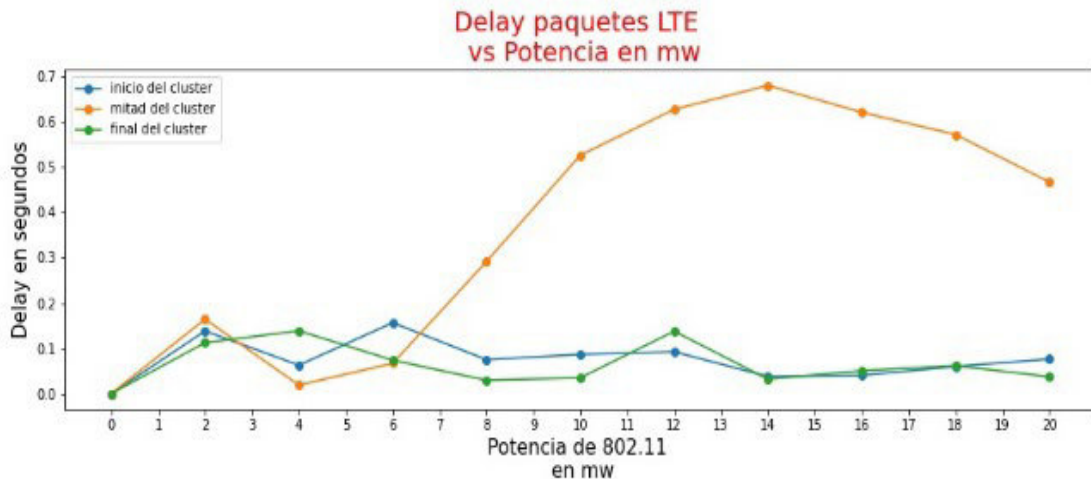
**Figura 3.6.** Paquetes LTE Tx vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.



**Figura 3.7.** Paquetes LTE Rx vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.



**Figura 3.8.** Packet Delivery Ratio LTE vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.



**Figura 3.9.** Delay LTE vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.

### B) ESCENARIO 2

Al igual que en el primer escenario, aquí también se tienen 5 caravanas de vehículos, pero, el número de vehículos en cada caravana es variable porque empieza con 2 vehículos y aumenta en 2 para cada simulación hasta llegar a tener 10 vehículos dentro de la caravana. El otro parámetro variable de este escenario es el intervalo de tiempo en que se envía los mensajes o paquetes IEEE 802.11p en donde se empieza en 100 ms y aumenta 100 ms en cada simulación hasta llegar a tener 1000 ms. Al modificar los parámetros se generan 50 simulaciones las cuales tendrán sus curvas que se mostrarán superpuestas para un mejor análisis.

El código utilizado para el primer script del segundo escenario se encuentra en el Anexo F.

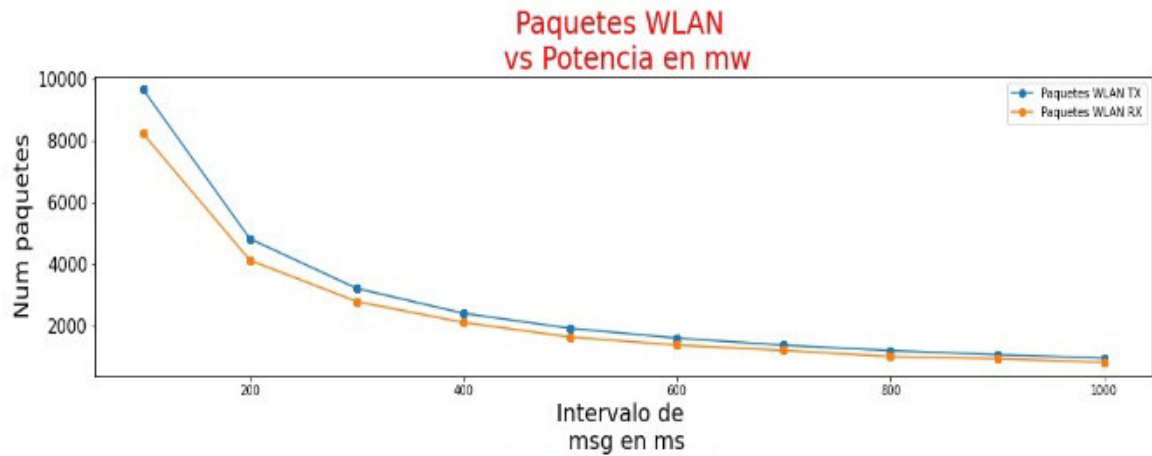
Luego de que se ejecute el script de *Python* se crea un nuevo archivo con extensión .csv con todos los parámetros ya detallados; y esos nuevos archivos con extensión .csv que se generan por cada simulación se los debe juntar en un mismo directorio con el fin de crear otro script de Python similar al segundo que se utilizó en el primer escenario, el cual es capaz de unir todos los datos que se tienen de los archivos con extensión .csv y así lograr graficar el comportamiento de la HETNET para cada simulación.

El código del segundo script para el segundo escenario se encuentra en el Anexo G.

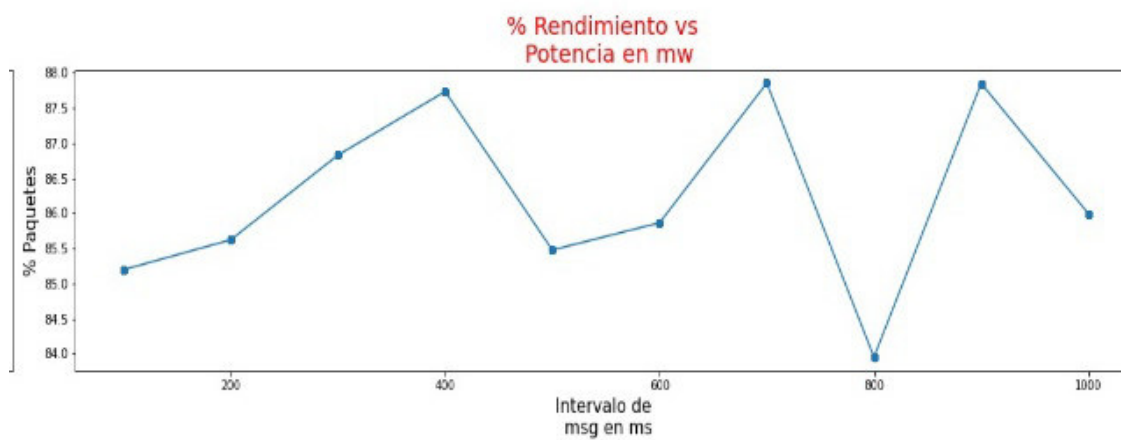
Cuando se ejecute el segundo script se generan las Figuras 3.10, 3.11, 3.12, 3.13, 3.14 y 3.15. En la Figura 3.10 se muestran en una gráfica superpuesta tanto la cantidad de paquetes WLAN transmitidos, como la cantidad de paquetes WLAN recibidos. En la Figura 3.11 se muestra el *Packet Delivery Ratio* WLAN y en la Figura 3.12 se muestra el *delay*



WLAN. Para la Figura 3.13 se muestran las curvas superpuestas de los paquetes LTE transmitidos y recibidos. En la Figura 3.14 se muestra el *Packet Delivery Ratio* LTE y finalmente en la Figura 3.15 se muestra el *delay* LTE. Se destaca que las curvas presentadas en las figuras son solo ejemplo, en posteriores secciones se analizan a profundidad las curvas de los resultados.



**Figura 3.10.** Paquetes WLAN vs Intervalo de envío de paquetes IEEE 802.11p.



**Figura 3.11.** *Packet Delivery Ratio* WLAN vs Intervalo de envío de paquetes IEEE 802.11p.

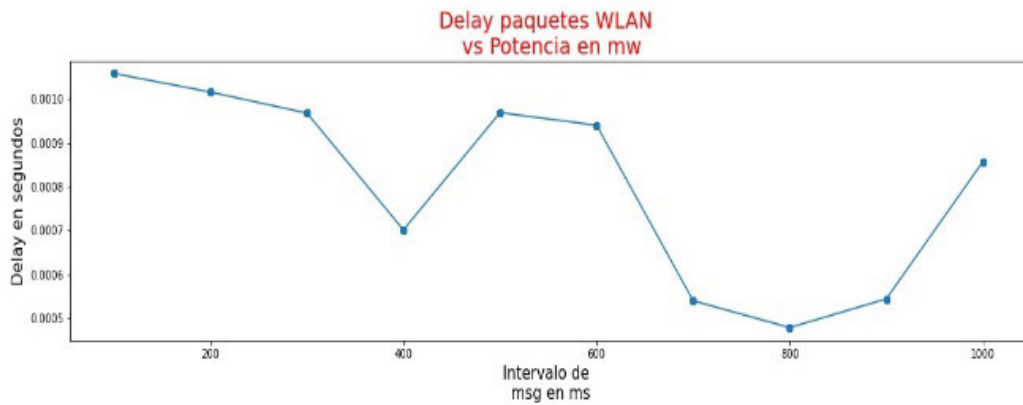


Figura 3.12. Delay WLAN vs Intervalo de envío de paquetes IEEE 802.11p.

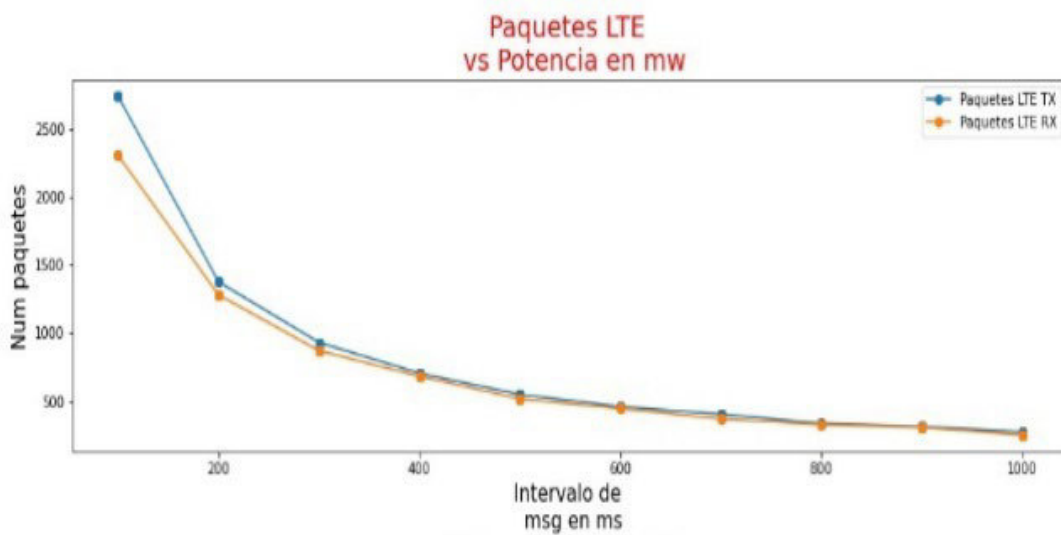


Figura 3.13. Paquetes LTE vs Intervalo de envío de paquetes IEEE 802.11p.

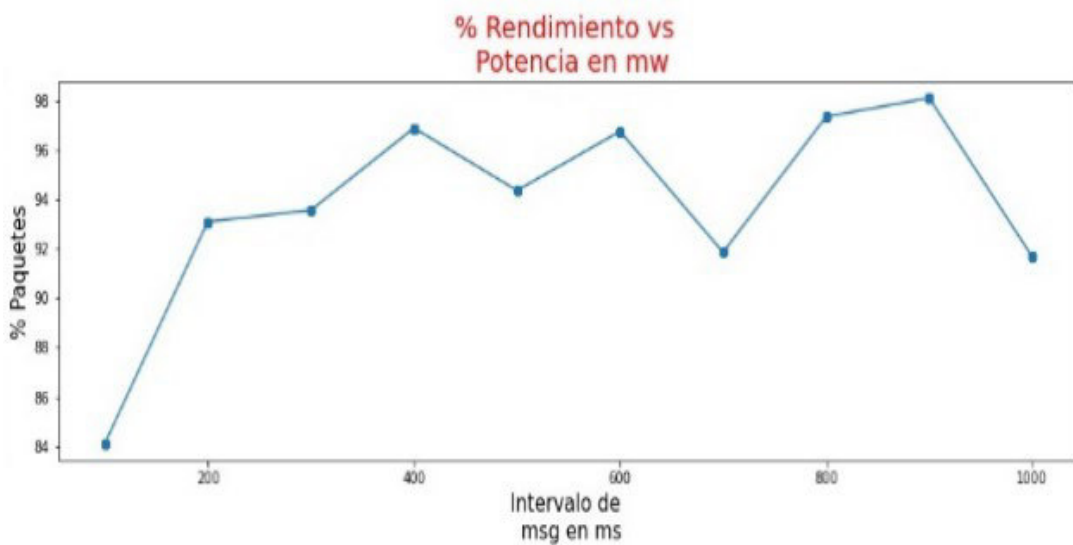
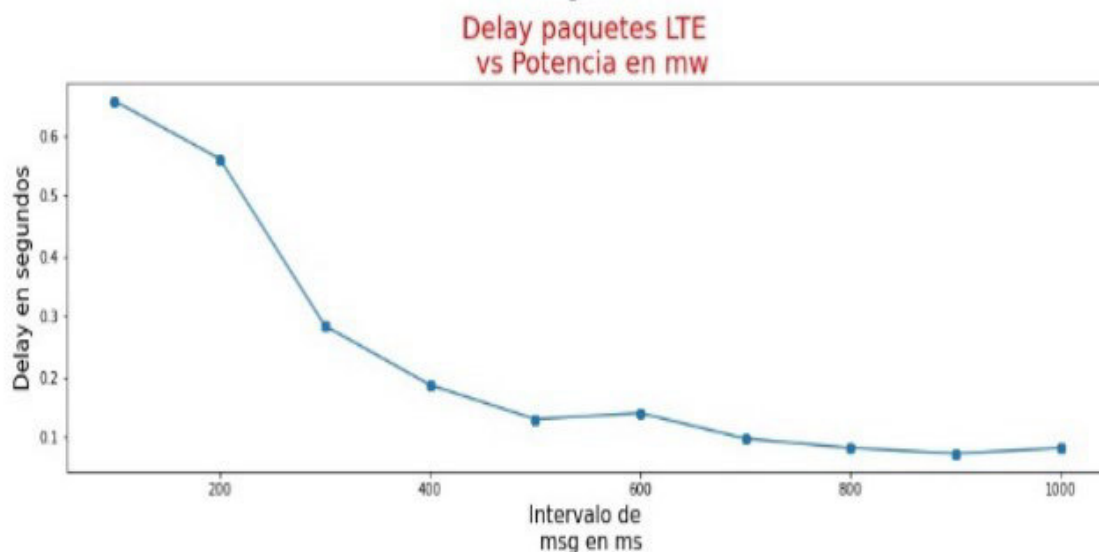


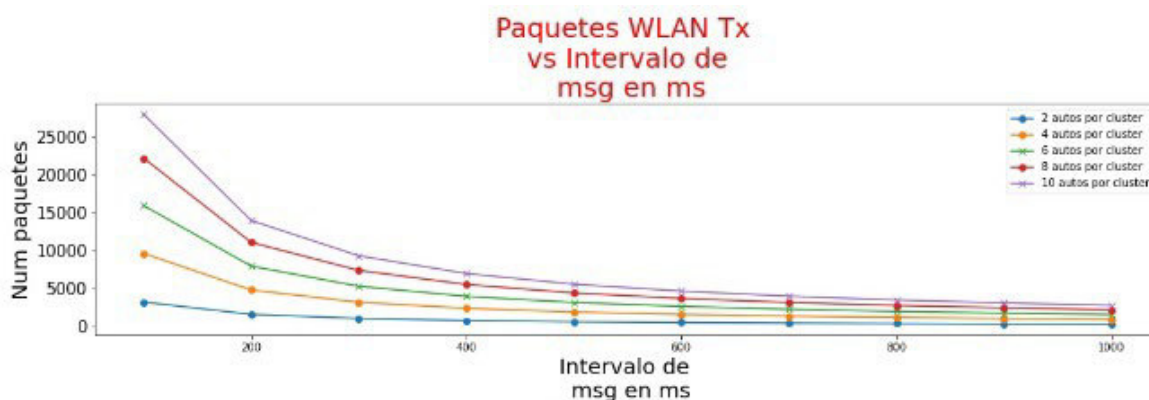
Figura 3.14. Packet Delivery Ratio LTE vs Intervalo de envío de paquetes IEEE 802.11p.



**Figura 3.15.** Delay WLAN vs Intervalo de envío de paquetes IEEE 802.11p.

Para analizar cómo se comporta la HETNET cuando a cada simulación se le atribuye distintas características, es necesario unir todas las curvas de las simulaciones y que con ello se aprecie la variación del comportamiento. Para ello se genera un script cuyo código se presenta en el Anexo H.

Cuando se ejecute el script para el segundo escenario se generan gráficas como la Figura 3.16, 3.17, 3.18, 3.19, 3.20, 3.21, 3.22 y 3.23 en donde se superpondrán las curvas de cada simulación con las cuales se pueden analizar el comportamiento de la HETNET al dotar a cada simulación con distintas características. El análisis se lo hace en secciones posteriores.



**Figura 3.16.** Paquetes WLAN Tx vs Intervalo de envío de paquetes IEEE 802.11p.

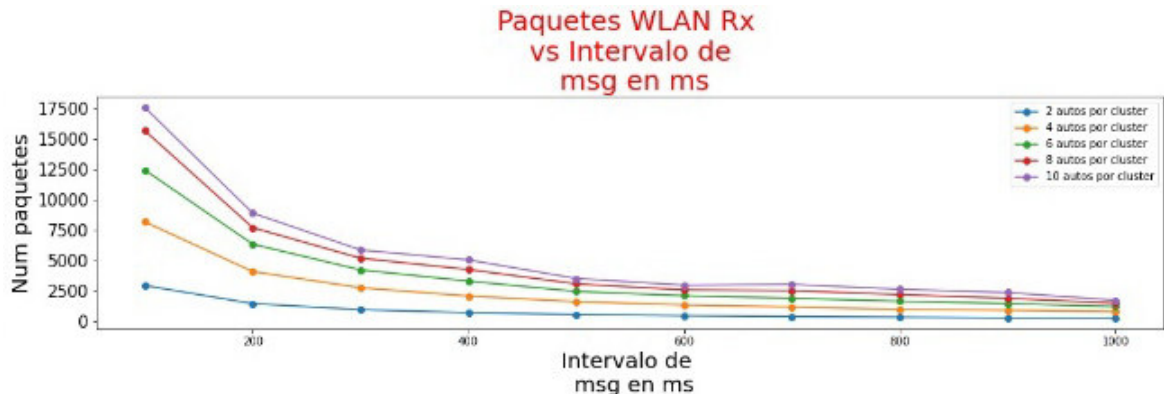


Figura 3.17. Paquetes WLAN Tx vs Intervalo de envío de paquetes IEEE 802.11p.

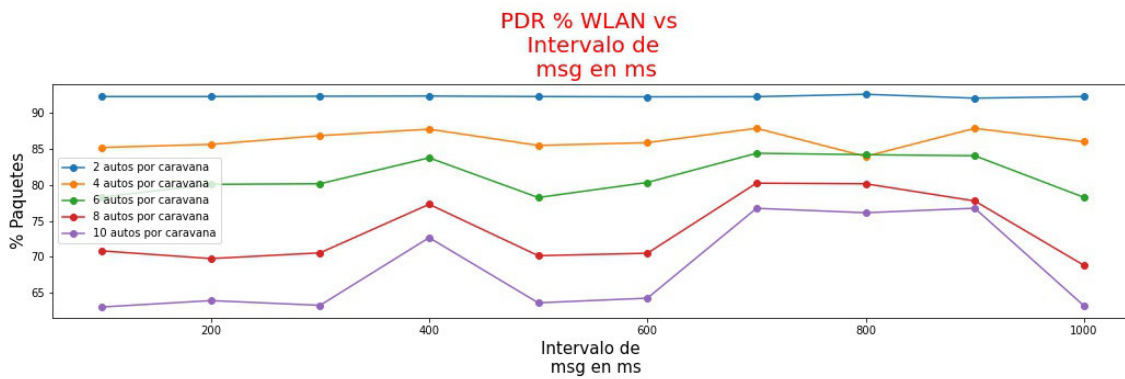


Figura 3.18. Packet Delivery Ratio WLAN vs Intervalo de envío de paquetes IEEE 802.11p.

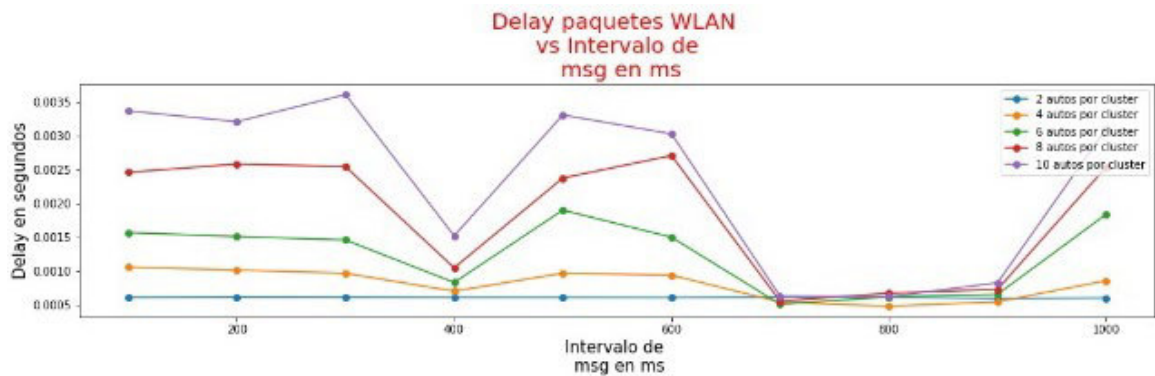
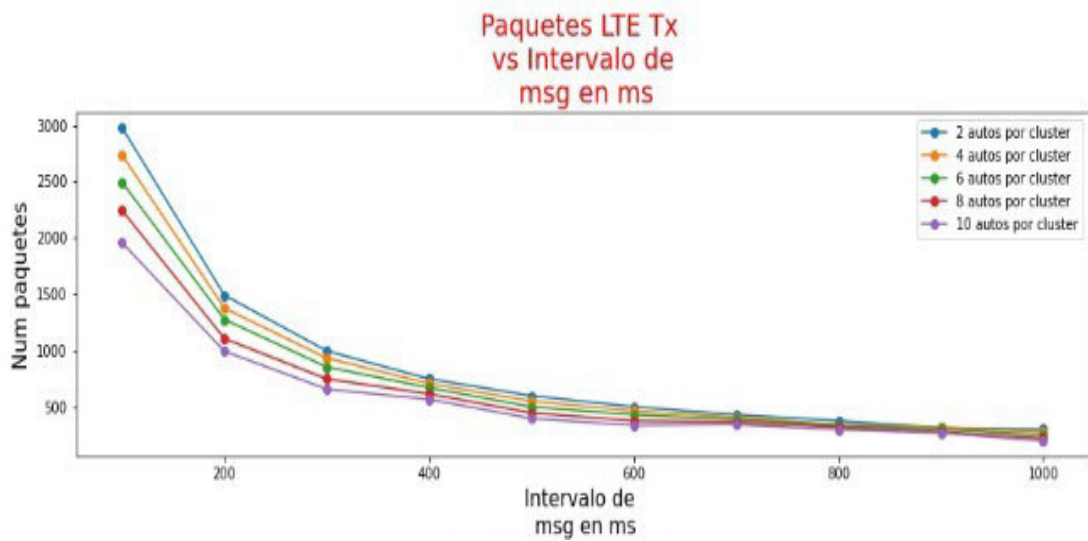
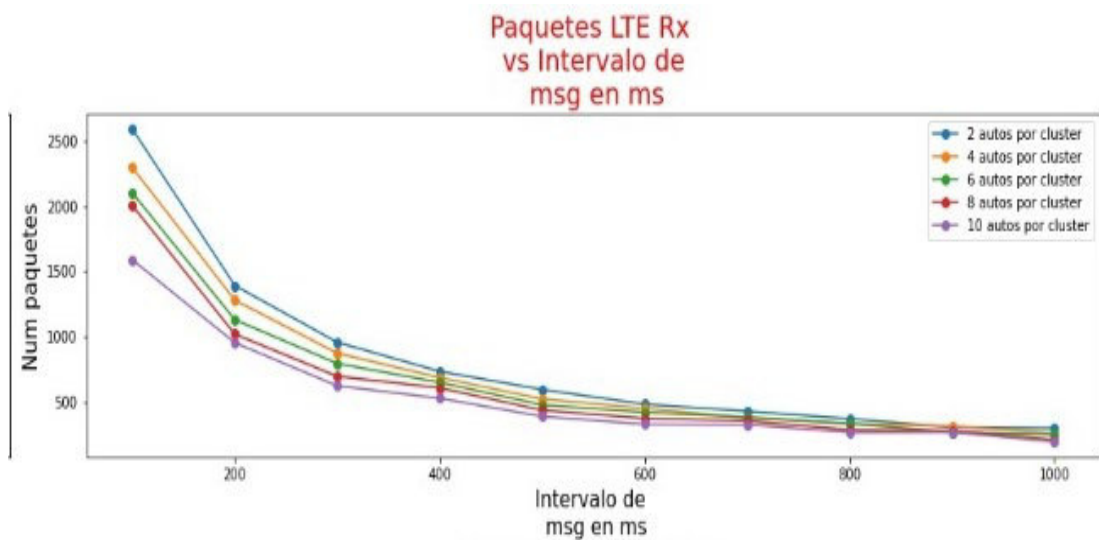


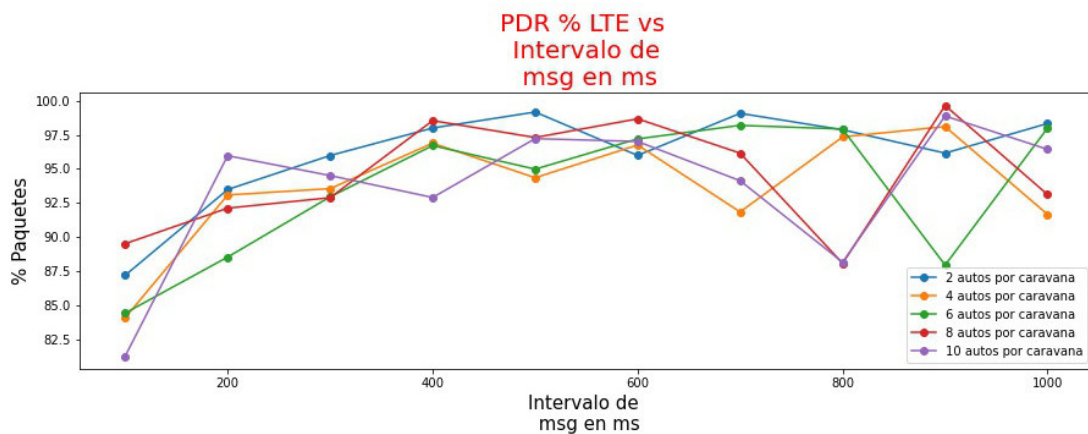
Figura 3.19. Delay WLAN vs Intervalo de envío de paquetes IEEE 802.11p.



**Figura 3.20.** Paquetes LTE Tx vs Intervalo de envío de paquetes IEEE 802.11p.



**Figura 3.21.** Paquetes LTE Rx vs Intervalo de envío de paquetes IEEE 802.11p.



**Figura 3.22.** Packet Delivery Ratio LTE vs Intervalo de envío de paquetes IEEE 802.11p.

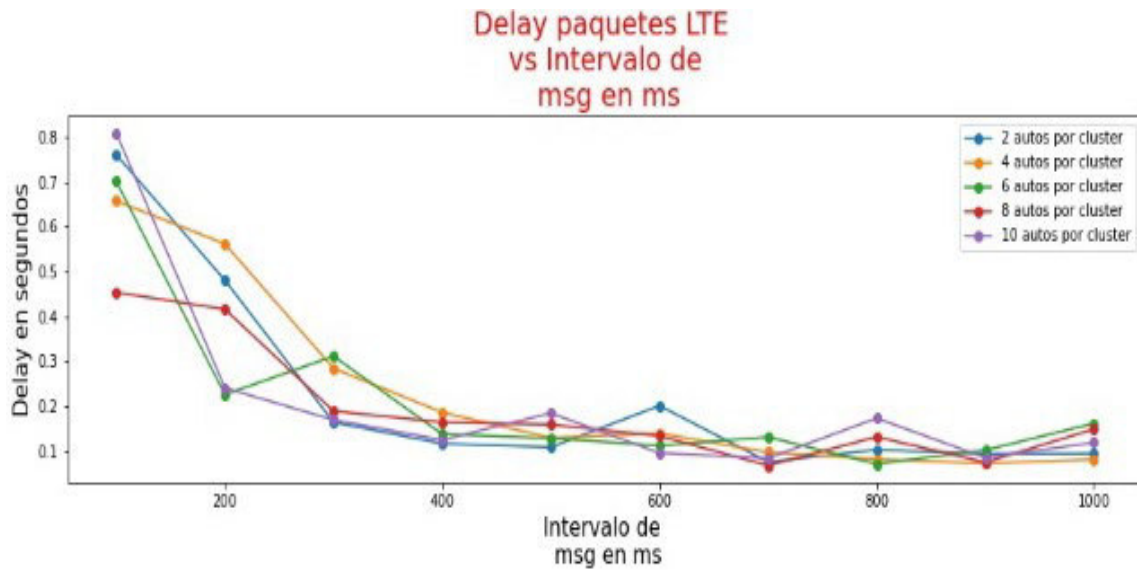


Figura 3.23. Delay WLAN vs Intervalo de envío de paquetes IEEE 802.11p.

## 3.2 ANÁLISIS DE RESULTADOS

En esta parte final del trabajo de titulación se analiza el comportamiento de la HETNET a través de las gráficas estadísticas obtenidas con Python en donde varían parámetros de la red y la caravana de vehículos.

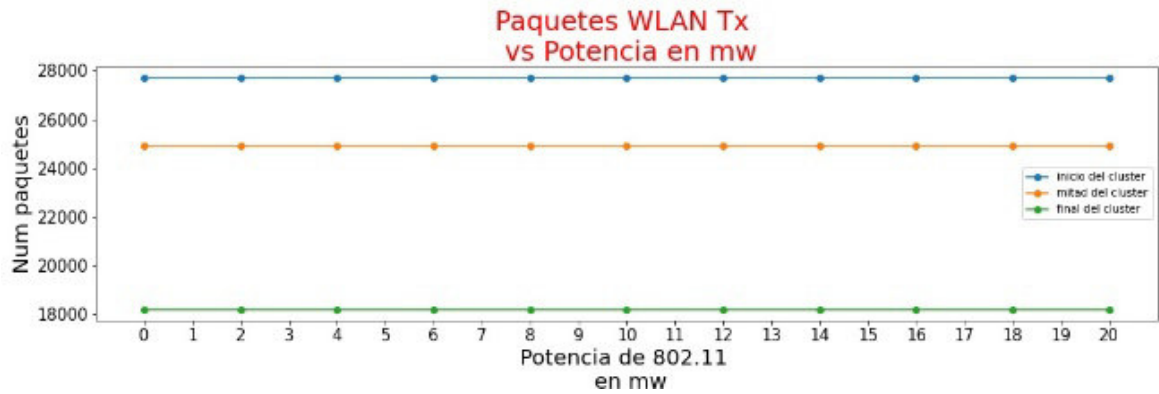
Los resultados de las simulaciones para el primer escenario se encuentran desde la Figura 3.24 hasta la 3.31. Las curvas de los resultados para el segundo escenario se encuentran desde la Figura 3.32 hasta la 3.39

### 3.2.1 PRIMER ESCENARIO

En las simulaciones planteadas se consideran que cuando el CH sale del mapa, la simulación se termina.

#### 3.2.1.1 Paquetes WLAN transmitidos vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p

En la Figura 3.24 se puede ver el comportamiento de la HETNET cuando se cambia la posición del CH dentro de la caravana de vehículos y también la potencia de transmisión de la OBU del vehículo para el estándar 802.11p. El análisis de la gráfica se da en función del número de paquetes WLAN transmitidos vs la potencia de transmisión de la OBU del vehículo para el estándar IEEE 802.11p.



**Figura 3.24.** Paquetes WLAN tx vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.

La cantidad de paquetes WLAN transmitidos, está ligado al tiempo en que el CH permanece en la simulación, es decir que dependerá del tiempo en que aparece y sale de la simulación.

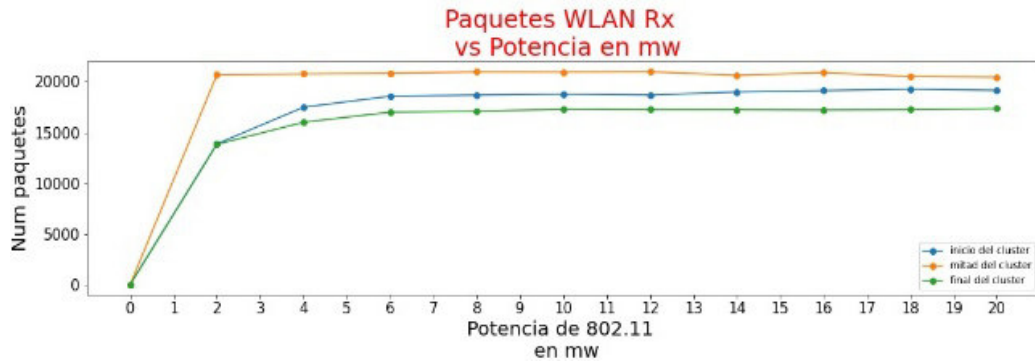
Siguiendo esa premisa se observa que cuando el CH se encuentra al inicio de la caravana, este vehículo aparecerá al inicio de la simulación, por lo que la cantidad de paquetes WLAN transmitidos por los demás automóviles que traten de alcanzar al CH será mayor porque la permanencia del CH en la simulación será superior a las demás.

Todo lo contrario, sucede cuando el CH se encuentra ubicado al final de la caravana, ya que tendrá menos tiempo en la simulación, además, existirán menos vehículos en cierto instante ya que estos empezarán a salir de la simulación y ya no transmitirán paquetes destinados al CH.

Con el CH en una posición intermedia se observa que a pesar de que el CH no aparece al inicio de la simulación, los demás vehículos si permanecen enviando paquetes constantemente hasta que estos salen de la simulación, sin embargo, en este escenario, cuando el CH sale de la simulación esta termina por lo que ya no se emitirán más paquetes.

### 3.2.1.2 Paquetes WLAN recibidos vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p

En la Figura 3.25 se indica una gráfica en donde se tiene la curva de Número de paquetes recibidos vs la potencia de transmisión de la OBU del vehículo para el estándar IEEE 802.11p.



**Figura 3.25.** Paquetes WLAN Rx vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.

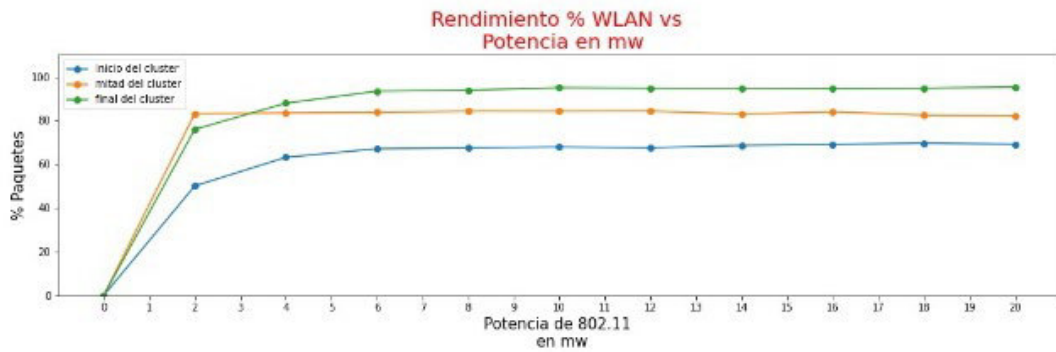
De los resultados expuestos en la Figura 3.25 se observa que, cuando la potencia de transmisión en el estándar IEEE 802.11p es cero, ningún paquete llega a su destino, sin embargo, cuando empieza a subir la potencia, los paquetes llegan a su destino hasta llegar a un punto en que parece que la cantidad de paquetes recibidos por el CH llega a ser constante.

Además, de la Figura 3.25 se concluye que, la cantidad de paquetes WLAN que recibe el CH, también depende de la posición del vehículo cabecera dentro de la caravana, ya que el CH recibe más paquetes cuando se encuentra en la mitad de la caravana porque su distancia a los demás automóviles es menor a que si se encontrará al inicio o al final de la caravana de vehículos. En el siguiente literal se hablará de las pérdidas de paquetes.

### 3.2.1.3 *Packet Delivery Ratio* WLAN vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p

La Figura 3.26 muestra el *Packet Delivery Ratio*, que es el rendimiento de la red, es decir, aquí se presenta el porcentaje de paquetes recibidos en función de la potencia de transmisión del estándar IEEE 802.11p.





**Figura 3.26.** *Packet Delivery Ratio* WLAN tx vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.

Como se observa en anteriores literales, la cantidad de paquetes WLAN transmitidos y recibidos se encuentran directamente relacionados al tiempo de simulación y a la posición del CH. Para este caso se observa que existe un rendimiento mayor cuando el CH está al final de la caravana ya que tiene aproximadamente el 90% de eficiencia debido a que el tiempo de simulación es menor que con las otras simulaciones y por lo tanto el envío y recepción de paquetes también es menor. Todo lo contrario, ocurre cuando el CH se encuentra al inicio de la caravana de vehículos, en donde la eficiencia es paupérrima con el 60% de eficiencia aproximadamente y esto se debe al tiempo de simulación, además de que se tiene una distancia considerable entre el CH y el primer vehículo de la caravana.

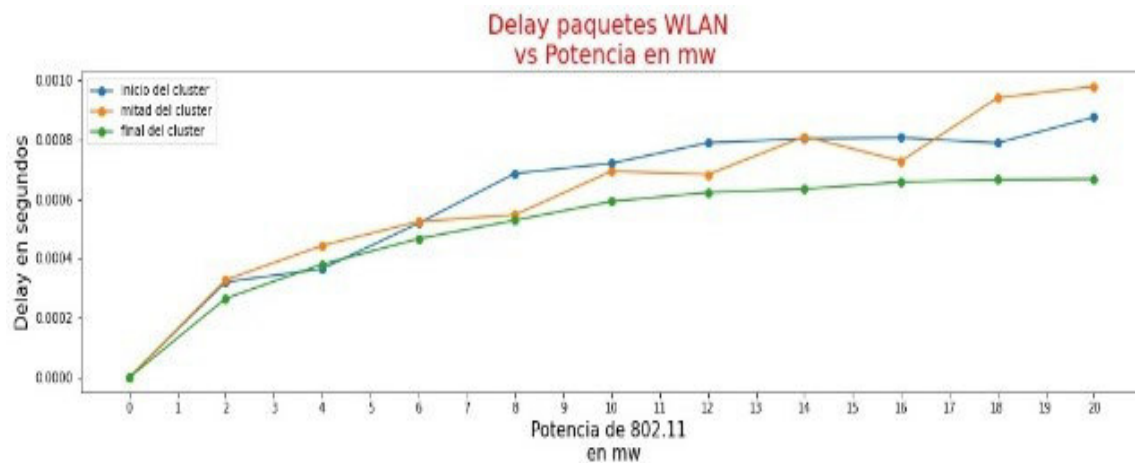
La simulación en donde se tiene al CH en la mitad de la caravana mostraría una situación que resulta favorable ya que el tiempo de simulación es equilibrado, además de que la posición del CH permitirá que se capte de mejor manera los paquetes IEEE 802.11p que los demás vehículos envían.

### 3.2.1.4 Delay WLAN vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p

En la Figura 3.27 se observa el *delay* o retardo en milisegundos en que un paquete IEEE 802.11p se demora en llegar al CH cuando se considera diferentes valores de la potencia de transmisión de los nodos/vehículos con el estándar IEEE 802.11p.

En la Figura 3.27 se observa que, el tiempo promedio en que llegan los mensajes IEEE 802.11p al CH está por debajo de 1 ms en todas las simulaciones, sin embargo, existe alguna variación que puede ser producto del tiempo de simulación y la posición del CH dentro de la caravana de vehículos. Cuando se tiene el CH al final de la caravana, el *delay* disminuye y esto se debe a que no se envían tantos paquetes como en las otras

simulaciones provocando que el *buffer* que almacena los mensajes no se sature y se distribuyan mejor los paquetes.



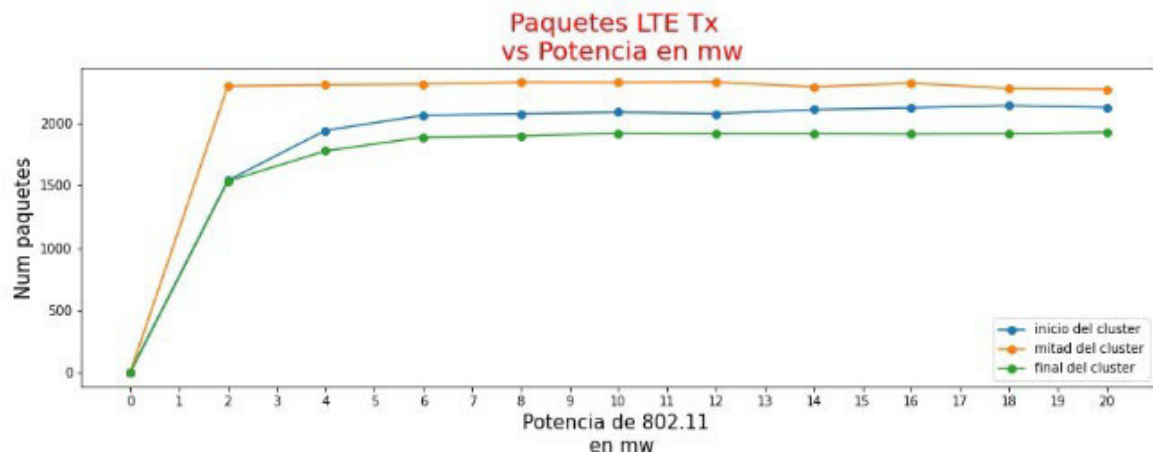
**Figura 3.27.** Delay WLAN tx vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.

### 3.2.1.5 Paquetes LTE transmitidos vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p

En la Figura 3.28 se establece la relación entre los paquetes LTE que se transmiten en función de la potencia de transmisión del estándar IEEE 802.11p.

El componente LTE esté ligado al estándar IEEE 802.11p, ya que se necesita que llegue cierta cantidad de paquetes WLAN al CH para que se pueda activar el reenvío de la información a través de la interfaz LTE.

En la Figura 3.28 se puede observar que existe mayor cantidad de reenvíos al servidor LTE y esto se debe a que la cantidad de paquetes WLAN recibidos por el CH es mayor cuando este se encuentra en la mitad de la caravana de vehículos. Si se hace un símil con la Figura 3.26 se puede observar que las gráficas son parecidas y esto es debido a que cada cierta cantidad de paquetes WLAN recibidos por parte del CH se realiza el forward hacia el servidor LTE.



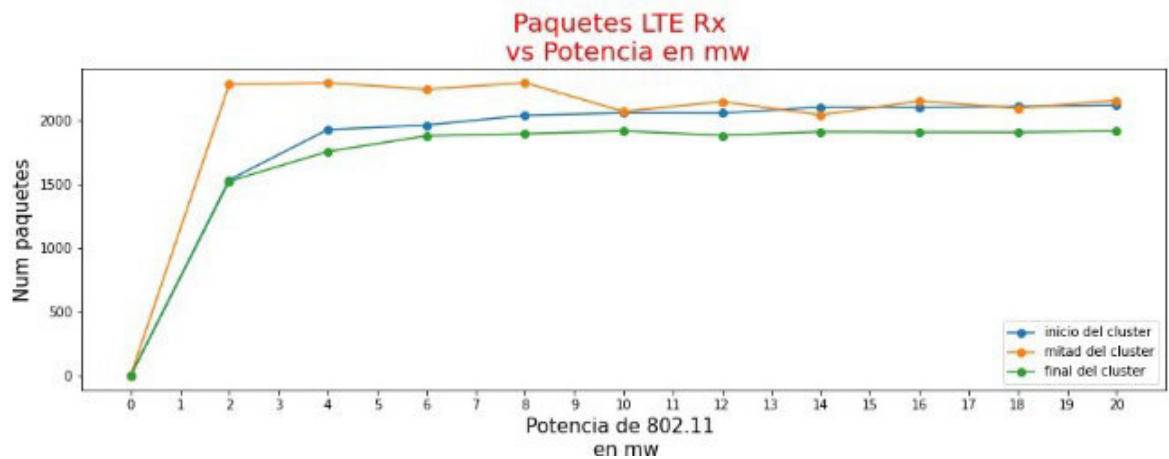
**Figura 3.28.** Paquetes WLAN tx vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.

### 3.2.1.6 Paquetes LTE recibidos vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p

En la Figura 3.29 se exhibe la cantidad de paquetes LTE que el servidor LTE recibe cuando la potencia del estándar IEEE 802.11p es variable.

La recepción LTE depende la posición del CH dentro de la zona del centro histórico de Quito escogida y que tan distante se encuentra de un *eNodeB* que pueda recoger esos paquetes. De acuerdo con el diseño realizado se colocan 3 *eNodeBs* repartidos en el mapa según la información que la página web de Claro Ecuador suministra. Para esta simulación dentro del código fuente de la simulación se estableció una potencia de 20 mW para el transmisor LTE, por lo que la cantidad de paquetes recibidos dependerá ahora de los obstáculos que existan en la zona y del desvanecimiento de la señal.

En esta Figura 3.29 se observa que, hasta cierto punto, cuando el CH se encuentra en la mitad de la caravana de vehículos, tiene una cantidad de paquetes recibidos superior a las demás simulaciones, pero llega a tomar un valor constante similar al de la simulación cuando el CH se encuentra al inicio de la caravana de vehículos.

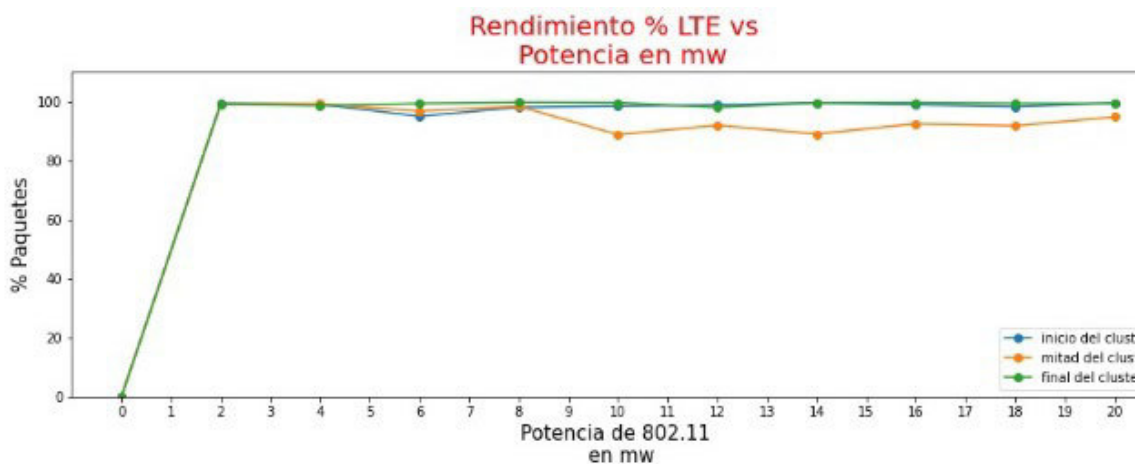


**Figura 3.29.** Paquetes LTE Rx vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.

### 3.2.1.7 Packet Delivery Ratio LTE vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p

En la figura 3.30 se observa el rendimiento LTE del escenario cuando la potencia de transmisión del estándar IEEE 802.11p es variable.

El rendimiento de la red celular para esta HETNET es superior al 80% en las 3 simulaciones ya que el rango de cobertura para LTE es mayor que el del estándar IEEE 802.11p, por lo que existirá una cantidad de paquetes perdidos menor; solo se observa una anomalía con respecto a la simulación de cuando el CH se encuentra en la mitad de la caravana de vehículos lo cual puede ser explicado por la interferencia y obstáculos que se tiene alrededor de los *eNodeBs*.



**Figura 3.30.** Packet Delivery Ratio LTE vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.

### 3.2.1.8 Delay LTE vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p

En la Figura 3.31 se presenta el retardo que tienen los paquetes LTE en llegar al servidor en función de la potencia de transmisión del estándar IEEE 802.11p.

Al igual que en los anteriores literales se observa una anomalía cuando el CH se encuentra en la mitad de la caravana de vehículos. Se reitera que esto puede ser producto de obstáculos, desvanecimiento u otro efecto que provoque que los paquetes tiendan a demorar en llegar a su destino.

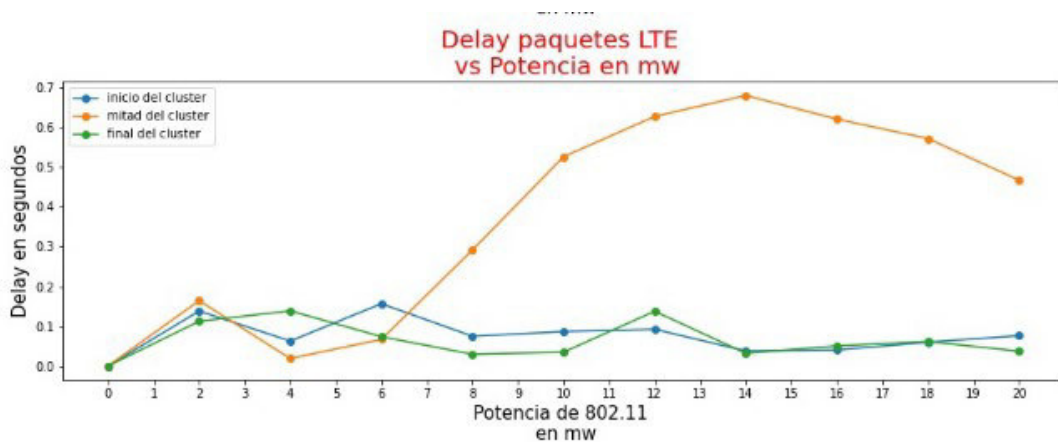


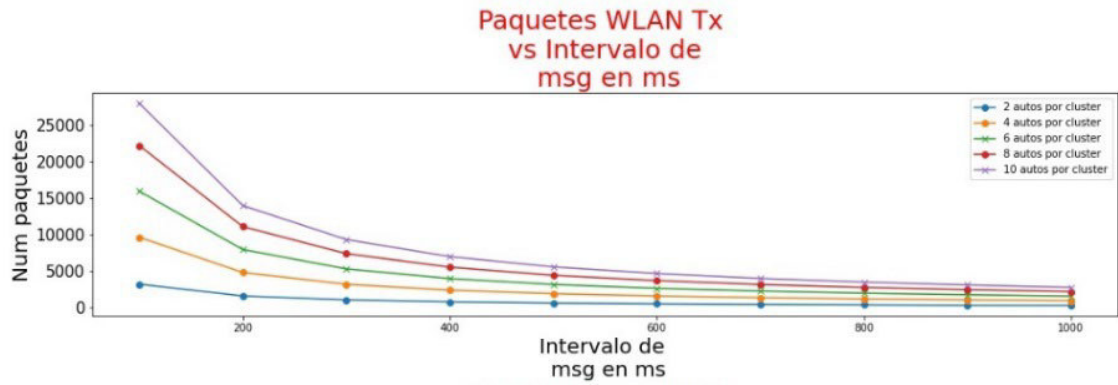
Figura 3.31. Delay LTE vs Potencia Tx de la OBU del vehículo en mW para IEEE 802.11p.

### 3.2.2 SEGUNDO ESCENARIO

#### 3.2.2.1 Paquetes WLAN transmitidos vs Intervalo de envío de mensajes para el estándar IEEE 802.11p

La Figura 3.32 muestra como varia el número de paquetes IEEE 802.11p transmitidos cuando existen parámetros variables como el intervalo de envío de mensajes y la cantidad de vehículos que forman una caravana o grupo de vehículos.

Se puede apreciar en la Figura 3.32 un comportamiento exponencial decreciente en donde, mientras existe mayor cantidad de vehículos en la simulación, la cantidad de paquetes que se generan con destino al CH es mayor, sin embargo, llega a un punto en que el tiempo entre mensajes enviados es bajo por lo que el número de paquetes enviados baja drásticamente hasta llegar al punto de ser casi constante.

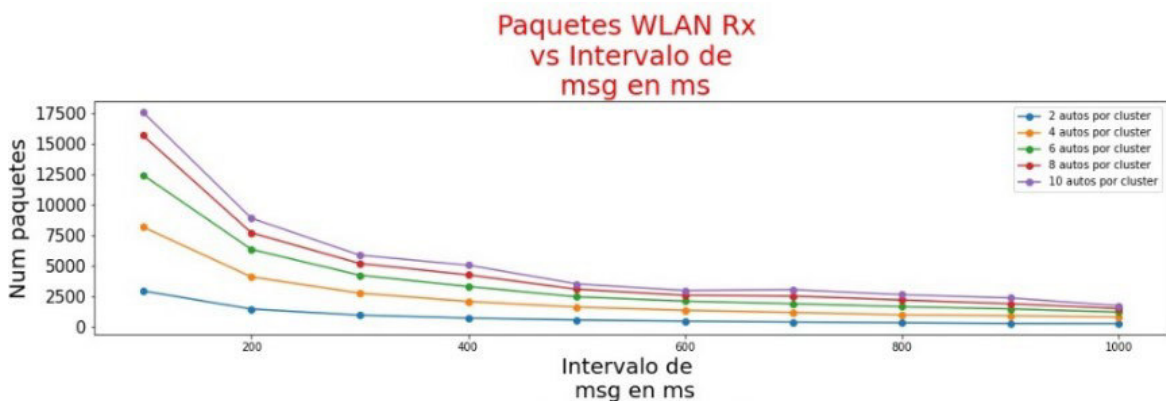


**Figura 3.32.** Paquetes WLAN tx vs Intervalo de mensajes para estándar IEEE 802.11p

### 3.2.2.2 Paquetes WLAN recibidos vs Intervalo de envío de mensajes para el estándar IEEE 802.11p

En la Figura 3.33 se observa el comportamiento de los paquetes recibidos cuando el intervalo de envío de mensajes IEEE 802.11p varía al igual que la cantidad de vehículos dentro de cada caravana de vehículos.

Si se compara la Figura 3.33 con la Figura 3.32 se observa que son similares, ya que cuando existe mayor cantidad de vehículos en la caravana se generan más paquetes IEEE 802.11p recibidos por el CH y decrece la cantidad de paquetes IEEE 802.11p de forma exponencial hasta llegar a un punto que parece que se estabiliza cuando el tiempo del intervalo de envío de mensajes es alto.



**Figura 3.33.** Paquetes WLAN Rx vs Intervalo de mensajes para estándar IEEE 802.11p

### 3.2.2.3 Packet Delivery Ratio WLAN vs Intervalo de envío de mensajes para el estándar IEEE 802.11p

La Figura 3.34 indica el rendimiento del estándar IEEE 802.11p cuando existen 2 parámetros variables en la simulación como lo son el intervalo de envío de mensajes IEEE 802.11p y la cantidad de vehículos que conforman una caravana de vehículos.

Cuando existe menor cantidad de vehículos se observa que la eficiencia es mayor y esto es entendible porque al no haber tantos paquetes en cola, la mayoría puede llegar a su destino que es el CH.

Sin embargo, cuando el número de vehículos dentro de una caravana aumenta, ya se puede observar que la eficiencia decae y en ciertos tramos tienen un comportamiento irregular que puede deberse a un encolamiento alto dentro del *buffer* que genera pérdida de paquetes.

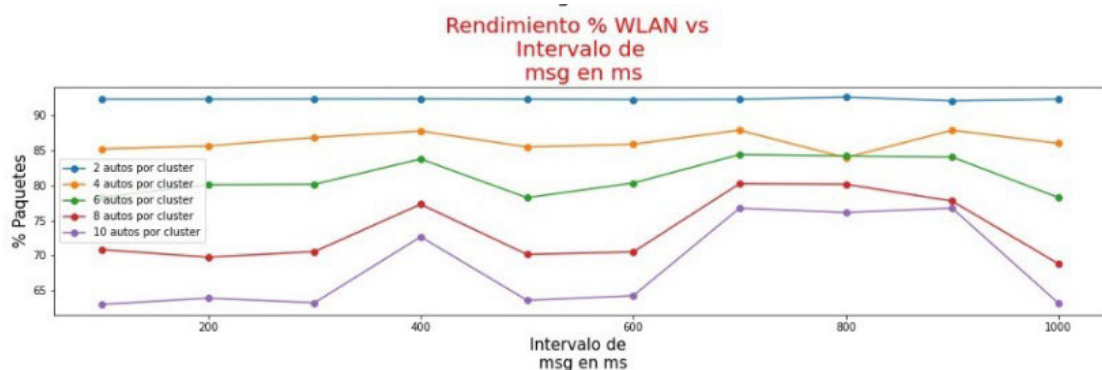


Figura 3.34. Packet Delivery Ratio WLAN tx vs Intervalo de mensajes para estándar IEEE 802.11p

### 3.2.2.4 Delay WLAN vs Intervalo de envío de mensajes para el estándar IEEE 802.11p

En la Figura 3.35 se muestra el retardo en promedio que los paquetes IEEE 802.11p tardan en llegar al CH cuando el intervalo de envío de mensajes IEEE 802.11p y la cantidad de vehículos dentro de la caravana es variable.

Al igual que con la eficiencia, aquí se observa que mientras menos automóviles se tengan en las caravanas, el tiempo que demora en llegar el paquete hacia el CH es menor.

Caso contrario con más vehículos, el tiempo aumenta y llega a tener fluctuaciones que pueden ser producto de encolamiento en el *buffer*.

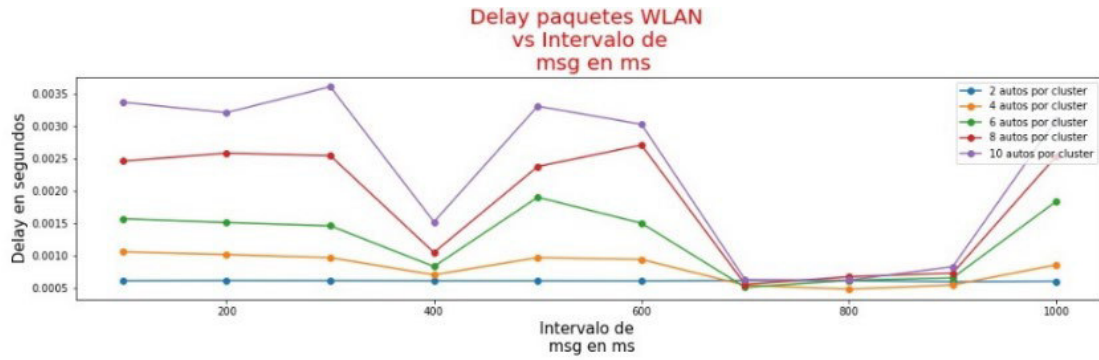


Figura 3.35. Delay WLAN tx vs Intervalo de mensajes para estándar IEEE 802.11p

### 3.2.2.5 Paquetes LTE transmitidos vs Intervalo de envío de mensajes para el estándar IEEE 802.11p

La Figura 3.36 presenta el comportamiento de los paquetes LTE que se transmiten cuando existen parámetros variables en la simulación como lo son el intervalo de envío de mensajes IEEE 802.11p y el número de vehículos dentro de las caravanas.

Al igual que con las comunicaciones de corto alcance entre los vehículos y el CH, en LTE se tiene un comportamiento exponencial decreciente, pero ahora se envían más paquetes LTE cuando el número de vehículos dentro de la caravana es menor. Esto puede ser consecuencia del alto rendimiento que se tiene en esas simulaciones ya que el envío de paquetes LTE depende de que cierta cantidad de paquetes lleguen correctamente al CH para generar el reenvío de la información hacia el *eNodeB*.

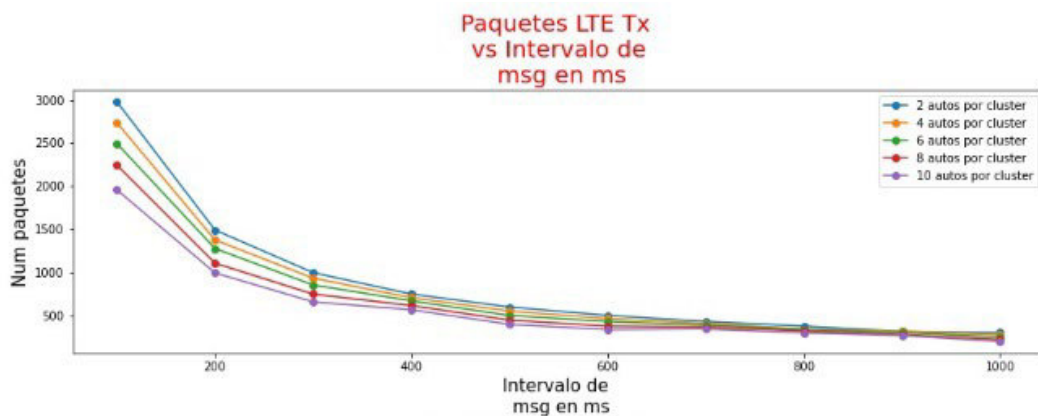


Figura 3.36. Paquetes WLAN tx vs Intervalo de mensajes para estándar IEEE 802.11p



### 3.2.2.6 Paquetes LTE recibidos vs Intervalo de envío de mensajes para el estándar IEEE 802.11p

La Figura 3.37 muestra cuantos paquetes LTE se recibe en el servidor y en los *eNodeBs* cuando se plantea condiciones variables como el intervalo de envío de mensajes IEEE 802.11p y la cantidad de vehículos que conforman las caravanas.

Esta gráfica muestra un comportamiento exponencial decreciente similar al de envío de paquetes LTE, sin embargo, se podrá verificar su verdadero comportamiento cuando se analice la eficiencia en el siguiente literal.

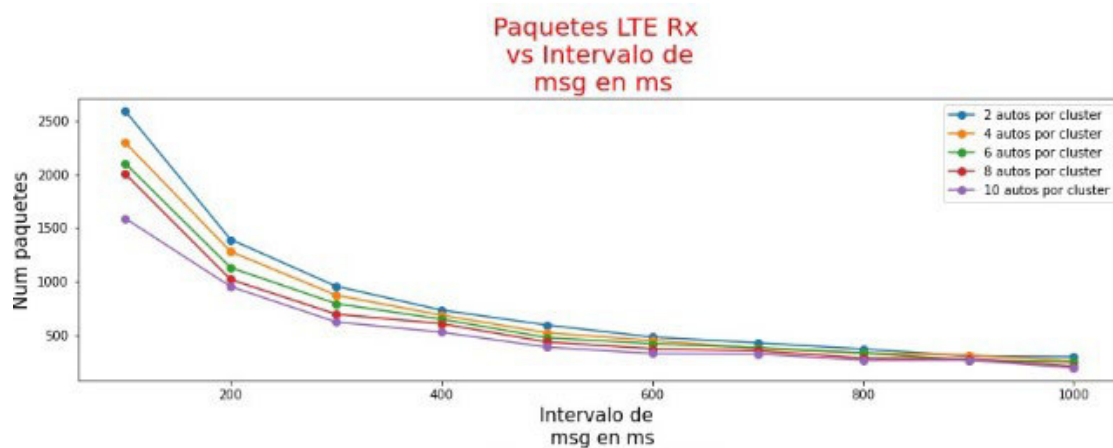


Figura 3.37. Paquetes LTE Rx vs Intervalo de mensajes para estándar IEEE 802.11p

### 3.2.2.7 Packet Delivery Ratio LTE vs Intervalo de envío de mensajes para el estándar IEEE 802.11p

La Figura 3.38 muestra la eficiencia LTE cuando se tiene parámetros variables como el intervalo de envío de mensajes IEEE 802.11p hacia el CH y la cantidad de vehículos presentes en cada caravana.

La gráfica no muestra ningún patrón, sin embargo, se puede apreciar que sea cual sea la variable que cambia, la eficiencia será mayor al 80%. Estas fluctuaciones e irregularidades en la eficiencia LTE puede ser producto de que existe desvanecimiento a gran o pequeña escala por el recorrido de los automóviles o por objetos que generen pérdidas de paquetes, además de que la posición de los *eNodeBs* no generaría el mismo rango de cobertura para los CHs.

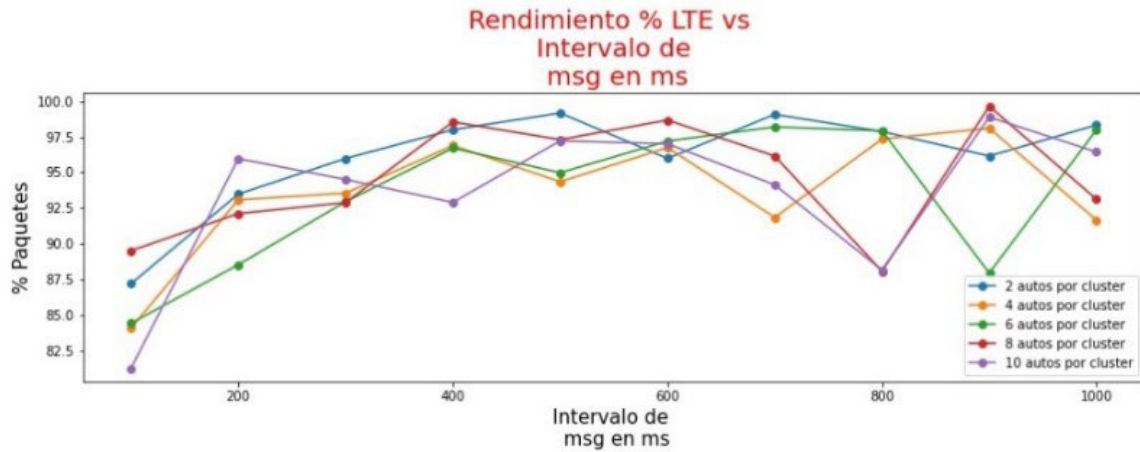


Figura 3.38. Packet Delivery Ratio LTE vs Intervalo de mensajes para estándar IEEE 802.11p

### 3.2.2.8 Delay LTE vs Intervalo de envío de mensajes para el estándar IEEE 802.11p

En la Figura 3.39 se muestra el tiempo en que tarda en llegar el paquete LTE hacia el servidor LTE cuando existen parámetros variables como el intervalo de envío de mensajes IEEE 802.11p y la cantidad de vehículos presentes en cada caravana.

En la Figura 3.39 se observa cierto patrón que parecería ser una exponencial decreciente indicando que cada que el intervalo de envío de mensajes IEEE 802.11p aumenta, el tiempo de retardo del paquete LTE disminuye.

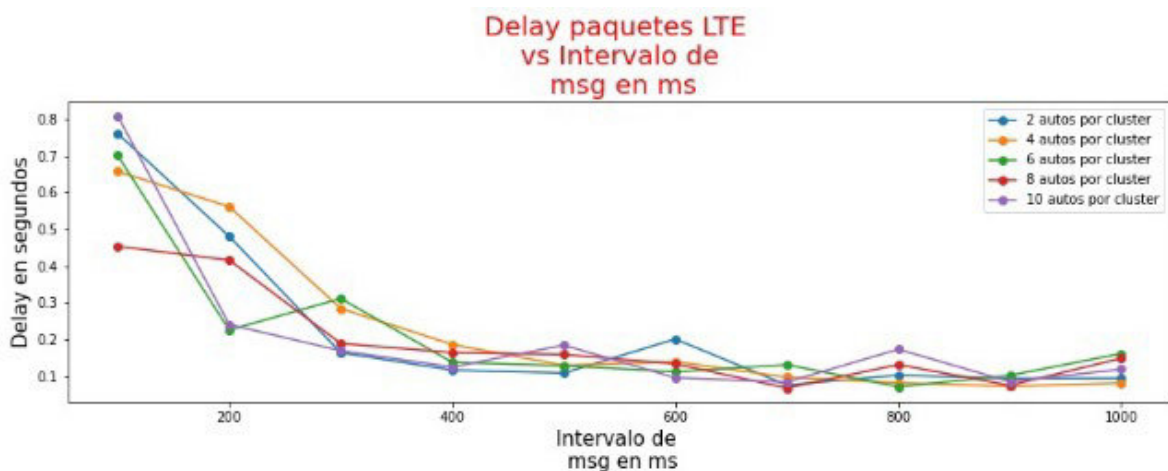


Figura 3.39. Delay LTE vs Intervalo de mensajes para estándar IEEE 802.11p

## 4 CONCLUSIONES Y RECOMENDACIONES

### 4.1 CONCLUSIONES

Una vez se analizan los resultados obtenidos en las simulaciones de los escenarios descritos para evaluar el rendimiento de la HETNET de caravanas de vehículos dentro del centro histórico de Quito se obtienen las siguientes conclusiones:

- La potencia de transmisión del estándar IEEE 802.11p tiene mucho más protagonismo en la eficiencia de la red ya que cuando esta potencia es baja se llegan a perder muchos paquetes por lo que la calidad de la red depende directamente de este parámetro.
- Se observa que, la posición del CH si juega un papel importante dentro de la HETNET, y al colocar el automóvil cabecera en la mitad de la caravana permitiría que los paquetes que lleguen de los vehículos en los extremos del grupo no se pierdan en mayor porcentaje.
- En el primer escenario se concluye que con una potencia de transmisión de la OBU para el estándar IEEE 802.11p de aproximadamente 10 mW y con el CH en la mitad de la caravana se logra una buena eficiencia WLAN que a su vez permitirá que la red LTE trabaje correctamente sin saturarse y presentando también una buena eficiencia y delay.
- Para el segundo escenario se observa que el comportamiento con respecto a los paquetes transmitidos y recibidos tanto para el estándar IEEE 802.11p como para LTE tiene un comportamiento exponencial decreciente ya que, de 2500 paquetes enviados, se reduce drásticamente el número de paquetes hasta llegar a menos de 500 paquetes.
- Al momento de obtener el *Packet Delivery Ratio* ya no se encuentra un patrón, sin embargo, la eficiencia de entrega de paquetes para el estándar IEEE 802.11p es superior cuando no existen muchos vehículos en las caravanas, sin embargo, decrece cuando existe mayor presencia de automotores. Esto se explica debido a que, una mayor cantidad de vehículos trata de comunicarse con el CH y muchos paquetes se pierden en ese proceso por temas de desvanecimiento o a encolamiento en buffers.
- En cuestión a LTE en el segundo escenario se observa que la eficiencia siempre es mayor al 80% independiente de la cantidad de vehículos presentes en las caravanas o al intervalo de envío de mensajes IEEE 802.11p. Esto se explica porque para generar un envío de paquetes LTE

por parte del CH, primero se debe llegar a tener cierta cantidad de paquetes IEEE 802.11p recibidos en el CH.

## **4.2 RECOMENDACIONES**

Para tener un mayor panorama de cómo se comporta una HETNET vehicular, e recomendable evaluar y generar otros escenarios en donde se cambie y se combine otros parámetros físicos o técnicos de la HETNET o caravanas de vehículos (*Platooning*).

Por ejemplo, se podría variar la potencia de los *eNodeBs*, el número de caravanas de vehículos, entre otros.

Otra recomendación es realizar este mismo análisis de la HETNET pero en otra locación de alto tráfico que puede ser en la ciudad de Quito o incluso otra ciudad de Ecuador o el mundo.

También se podría plantear un tema en donde la HETNET se la vincule con la red 5G que actualmente se esta desplegando en ciertos países del primer mundo.

## 5 REFERENCIAS BIBLIOGRÁFICAS

- [1] “Mientras el transporte público sea deficiente, el parque automotor seguirá engordando | Gestión”. [https://www.revistagestion.ec/sociedad-analisis/mientras-el-transporte-publico-sea-deficiente-el-parque-automotor-seguira#:~:text=EN QUITO%2C EL PARQUE AUTOMOTOR](https://www.revistagestion.ec/sociedad-analisis/mientras-el-transporte-publico-sea-deficiente-el-parque-automotor-seguira#:~:text=EN QUITO%2C EL PARQUE AUTOMOTOR,en continuo proceso de saturación), en continuo proceso de saturación (consultado may 31, 2021).
- [2] M. R. Ghori, K. Z. Zamli, N. Quosthoni, M. Hisyam, y M. Montaser, “Vehicular Ad-hoc Network ( VANET ): Review”, 2018 IEEE Int. Conf. Innov. Res. Dev., pp. 1–6, 2018.
- [3] M. D. Caiza Changoluisa, “Simulación de una VANET para un sector del centro histórico de la ciudad de Quito como apoyo a la movilidad vehicular”, Escuela Politécnica Nacional, 2017.
- [4] G. Rémy, S. M. Senouci, F. Jan, y Y. Gourhant, “LTE4V2X: LTE for a centralized VANET organization”, GLOBECOM - IEEE Glob. Telecommun. Conf., pp. 0–5, 2011, doi: 10.1109/GLOCOM.2011.6133884.
- [5] I. Ahmad et al., “VANET–LTE based heterogeneous vehicular clustering for driving assistance and route planning applications”, Comput. Networks, vol. 145, pp. 128–140, 2018, doi: 10.1016/j.comnet.2018.08.018.
- [6] A. S. Chekkouri, A. Ezzouhairi, y S. Pierre, “A new integrated VANET-LTE-A architecture for enhanced mobility in small cells HetNet using dynamic gateway and traffic forwarding”, Comput. Networks, vol. 140, pp. 15–27, 2018, doi: 10.1016/j.comnet.2018.04.013.
- [7] C. Bergenheim, H. Pettersson, E. Coelingh, C. Englund, S. Shladover, y S. Tsugawa, “Overview of platooning systems”, Proc. 19th ITS World Congr., 2012.
- [8] D. Lu, Z. Li, y D. Huang, “Platooning as a service of autonomous vehicles”, 18th IEEE Int. Symp. A World Wireless, Mob. Multimed. Networks, WoWMoM 2017 - Conf., 2017, doi: 10.1109/WoWMoM.2017.7974353.
- [9] The Institute of Electrical and Electronics Engineers, “IEEE Standard for Information technology -- Telecommunications and information exchange between systems LAN and MAN -- Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11-2016”, IEEE Std 802.11-2016 (Revision IEEE Std 802.11-2012), vol. 2016, pp. 1–3534, 2013, [En línea]. Disponible en: <http://ieeexplore.ieee.org/document/7786995/>.

- [10] E. C. Eze, S. Zhang, y E. Liu, "Vehicular ad hoc networks (VANETs): Current state, challenges, potentials and way forward", ICAC 2014 - Proc. 20th Int. Conf. Autom. Comput. Futur. Autom. Comput. Manuf., núm. September, pp. 176–181, 2014, doi: 10.1109/IConAC.2014.6935482.
- [11] I. Ku, Y. Lu, M. Gerla, R. L. Gomes, F. Ongaro, y E. Cerqueira, "Towards software-defined VANET: Architecture and services", 2014 13th Annu. Mediterr. Ad Hoc Netw. Work. MED-HOC-NET 2014, pp. 103–110, 2014, doi: 10.1109/MedHocNet.2014.6849111.
- [12] P. Merdrignac, O. Shagdar, y F. Nashashibi, "Fusion of Perception and V2P Communication Systems for the Safety of Vulnerable Road Users", IEEE Trans. Intell. Transp. Syst., vol. 18, núm. 7, pp. 1740–1751, 2017, doi: 10.1109/TITS.2016.2627014.
- [13] L. F. Urquiza-Aguiar, "Design and implementation of routing protocols with anonymity for vehicular ad-hoc networks in urban environments", p. 84, 2012, [En línea]. Disponible en: <http://upcommons.upc.edu/handle/2099.1/15903%5Cnhttp://repositorio.educacionsuperior.gob.ec/handle/28000/319>.
- [14] D. Jiang y L. Delgrossi, "IEEE 802.11p: Towards an international standard for wireless access in vehicular environments", IEEE Veh. Technol. Conf., pp. 2036–2040, 2008, doi: 10.1109/VETECS.2008.458.
- [15] R. Lopez y R. Guijarro, "Análisis del protocolo IEEE 802.11p en sistemas de comunicación vehiculares V2X", p. 85, 2019.
- [16] S. Eichler, "Performance evaluation of the IEEE 802.11p WAVE communication standard", IEEE Veh. Technol. Conf., pp. 2199–2203, 2007, doi: 10.1109/VETECF.2007.461.
- [17] D. Martha y C. Paredes, "Capa Física", doi: 10.1109/IEEESTD.2016.7786995.
- [18] D. Martha y C. Paredes, "Subcapa MAC", doi: 10.1109/IEEESTD.2016.7786995.
- [19] S. Pellé y B. Reber, "Towards Innovative Freight and Logistics", Towar. Innov. Freight Logist., vol. 3, núm. Lv, pp. 1–407, 2016, doi: 10.1002/9781119307785.
- [20] "Truck Platooning | California Partners for Advanced Transportation Technology". <https://path.berkeley.edu/research/connected-and-automated-vehicles/truck-platooning> (consultado oct. 08, 2021).

- [21] C. Englund et al., "The Grand Cooperative Driving Challenge 2016: Boosting the introduction of cooperative automated vehicles", *IEEE Wirel. Commun.*, vol. 23, núm. 4, pp. 146–152, 2016, doi: 10.1109/MWC.2016.7553038.
- [22] M. Sauter, *From GSM to LTE - Advanced Pro and 5G*. .
- [23] T. Specification y G. Services, "3gpp tr 21.914", vol. 0, núm. Release 14, 2017.
- [24] D. Martha y C. Paredes, "Comunicaciones Celulares".
- [25] S. Sesia, I. Toufik, y M. Baker, *LTE - The UMTS Long Term Evolution: From Theory to Practice: Second Edition*. John Wiley and Sons, 2011.
- [26] S. Sesia, *LTE -The UMTS Long Term Evolution. From Theory to Practice.*, vol. 7, núm. 2. 2557.
- [27] J. Han, M. Kamber, y J. Pei, "Data Mining Trends and Research Frontiers", en *Data Mining*, Elsevier, 2012, pp. 585–631.
- [28] V. Jungnickel et al., "The role of small cells, coordinated multipoint, and massive MIMO in 5G", *IEEE Commun. Mag.*, vol. 52, núm. 5, pp. 44–51, 2014, doi: 10.1109/MCOM.2014.6815892.
- [29] M. A. Uddin y Mamun-Or-Rashid, "Link expiration time-aware routing protocol for UWSNs", *J. Sensors*, vol. 2013, 2013, doi: 10.1155/2013/625274.
- [30] "A Quick Overview of the OMNeT++ IDE". <https://omnetpp.org/documentation/ide-overview/> (consultado jul. 04, 2021).
- [31] "What is OMNeT++?" <https://omnetpp.org/intro/> (consultado jul. 04, 2021).
- [32] "OMNeT++ Downloads". <https://omnetpp.org/download/> (consultado jul. 04, 2021).
- [33] "Documentation - SUMO Documentation". <https://sumo.dlr.de/docs/index.html> (consultado jul. 04, 2021).
- [34] "OpenStreetMap Wiki". [https://wiki.openstreetmap.org/wiki/Main\\_Page](https://wiki.openstreetmap.org/wiki/Main_Page) (consultado jul. 04, 2021).
- [35] "Documentation - Veins". <https://veins.car2x.org/documentation/> (consultado jul. 04, 2021).
- [36] "Claro". <https://www.claro.com.ec/personas/servicios/servicios-moviles/cobertura/> (consultado jul. 04, 2021).

- [37] “OpenStreetMap”. <https://www.openstreetmap.org/#map=7/-1.783/-78.132>  
(consultado nov. 17, 2021).
- [38] “GitHub - veins/veins\_hetvnet: Sample combination of Veins, INET, and SimuLTE”.  
[https://github.com/veins/veins\\_hetvnet](https://github.com/veins/veins_hetvnet) (consultado jul. 04, 2021).
- [39] S. Gadi, “BEP / SEP of Binary and M-ary Signals over  $k$ -  $\mu$  Shadowed Fading Channel”, núm. 2, pp. 3–6, 2016.



# ANEXOS

## ANEXO A

[General]

cmdenv-express-mode = false

cmdenv-autoflush = true

cmdenv-status-frequency = 1s

cmdenv-log-prefix = "[%c%?-%l] %G:%R %m %z::%u - "

\*\*cmdenv-log-level = trace

image-path = ../../veins/images

network = veins\_hetvnet.simulations.Highway

debug-on-errors = false

print-undisposed = false

sim-time-limit = 70s

\*\*sctp\*\*.scalar-recording = false

\*\*sctp\*\*.vector-recording = false

\*\*coreDebug = false

\*\*routingRecorder.enabled = false

\*.playgroundSizeX = 20000m

\*.playgroundSizeY = 20000m

\*.playgroundSizeZ = 50m

```

*.veinsManager.moduleType = "veins_hetvnet.simulations.Car"
*.veinsManager.moduleName = "car"
*.veinsManager.launchConfig = xmldoc("heterogeneous.launchd.xml")
*.veinsManager.updateInterval = 0.1s

*.car[*].mobilityType = "VeinsInetMobility"

**.deployer.positionUpdateInterval = 0.001s

**.channelControl.pMax = 10W
**.channelControl.alpha = 1.0
**.channelControl.carrierFrequency = 2100e+6Hz

**.lteNic.phy.channelModel=xmldoc("config_channel.xml")
**.feedbackComputation = xmldoc("config_channel.xml")

**.dynamicCellAssociation = true

**.car[*].masterId = 1
**.car[*].macCellId = 1
**.eNodeB1.macCellId = 1
**.eNodeB1.macNodeId = 1
**.eNodeB2.macCellId = 2
**.eNodeB2.macNodeId = 2
**.eNodeBCount = 2

**.rbAllocationType = "localized"
**.feedbackType = "ALLBANDS"
**.feedbackGeneratorType = "IDEAL"

```

```

** .maxHarqRtx = 3
** .numUe = ${numUEs=10}

** .deployer.ruRange = 50
** .deployer.ruTxPower = "50,50,50;"
** .deployer.antennaCws = "2;" # !!MACRO + RUS (numRus + 1)
** .deployer.numRbDL = 25
** .deployer.numRbUL = 25
** .numBands = 25
** .fbDelay = 1

** .enableHandover = true
** .broadcastMessageInterval = 0.5s

* .eNodeB*.numX2Apps = 1 # one x2App per peering eNodeB
* .eNodeB*.x2App[*].server.localPort = 5000 + ancestorIndex(1) # Server ports
(x2App[0]=5000, x2App[1]=5001, ...)
* .eNodeB1.x2App[0].client.connectAddress = "eNodeB2%x2ppp0"
* .eNodeB2.x2App[0].client.connectAddress = "eNodeB1%x2ppp0"
** .sctp.nagleEnabled = false # if true, transmission of small packets will be
delayed on the X2
** .sctp.enableHeartbeats = false

* .eNodeB*.d2dCapable = true
* .car[*].d2dCapable = true
** .amcMode = "D2D"

** .usePreconfiguredTxParams = true
** .d2dCqi = ${cqi=7}

```

```
*.car[*].numUdpApps = 1

*.car[*].udpApp[0].typename
"org.car2x.veins.subprojects.veins_hetvnet.HetVNetDemoApp"

*.car[*].udpApp[0].localPortLte = 1000
*.car[*].udpApp[0].localPortWlan = 1001
*.car[0].udpApp[0].startTime = uniform(0.01s,0.02s)
*.car[*].udpApp[0].destAddressLte = "224.0.0.10"
*.car[*].udpApp[0].destAddressWlan = "224.0.2.1"

*.car[*].configuratorLte.interfaces = "wlan"
*.car[*].configuratorWlan.interfaces = "wlan0"
*.car[*].configuratorLte.addressBase = "10.0.0.0"
*.car[*].configuratorWlan.addressBase = "2.0.0.0"
*.car[*].configuratorLte.mcastGroups = "224.0.0.10"
*.car[*].configuratorWlan.mcastGroups = "224.0.2.1"

*.car[*].wlan[0].mgmtType = "Ieee80211MgmtAdhoc"
*.car[*].wlan[0].bitrate = 6Mbps
*.car[*].wlan[0].radio.transmitter.power = 20mW
```

=

## Anexo B

```
cmdenv-status-frequency = 100s
cmdenv-log-prefix = "[%c%?-%l] %G:%R %m %z::%u - "
**.cmdenv-log-level = error
image-path = ../../veins/images
network = veins_hetvnet.simulations.Highway
debug-on-errors = true
print-undisposed = true
sim-time-limit = 83s
**.sctp.**.scalar-recording = false
**.sctp.**.vector-recording = false
**.coreDebug = false
**.routingRecorder.enabled = false
#####
#                               SCENARIO SIZE                               #
#####
*.playgroundSizeX = 20000m
*.playgroundSizeY = 40000m
*.playgroundSizeZ = 50m
#####
#                               VEINS MANAGER                               #
#####
*.veinsManager.moduleType = "veins_hetvnet.simulations.Car"
*.veinsManager.moduleName = "car"
*.veinsManager.updateInterval = 0.01s
#####
#                               MOBILITY                                   #
#####
*.car[*].mobilityType = "VeinsInetMobility"
**.deployer.positionUpdateInterval = 0.001s
#####
```

```

#                               Channel parameters                               #
#####

**.channelControl.pMax = 20W
**.channelControl.alpha = 1.0
**.channelControl.carrierFrequency = 2100e+6Hz
#####

#                               Physical layer parameters                       #
#####

**.lteNic.phy.channelModel=xmlDoc("config_channel.xml")
**.feedbackComputation = xmlDoc("config_channel.xml")

**.dynamicCellAssociation = true
#####

#                               MAC INITIALIZATION                               #
#####

**.car[*].masterId = 1
**.car[*].macCellId = 1
#####

#                               NUMBER OF eNBS                               #
#####

**.eNodeBCount = 2
**.eNodeB1.macCellId = 1
**.eNodeB1.macNodeId = 1
**.eNodeB2.macCellId = 2
**.eNodeB2.macNodeId = 2

#####

#                               AMC MODULE PARAMETERS                           #
#####

**.rbAllocationType = "localized"
**.feedbackType = "ALLBANDS"

```

```

**.feedbackGeneratorType = "IDEAL"

**.maxHarqRtx = 3

**.numUe = ${numUEs=50}

**.deployer.ruRange = 50
**.deployer.ruTxPower = "50,50,50;"
**.deployer.antennaCws = "2;" # !!MACRO + RUS (numRus + 1)
**.deployer.numRbDL = 25
**.deployer.numRbUL = 25
**.numBands = 25
**.fbDelay = 1

**.enableHandover = true
**.broadcastMessageInterval = 0.5s

#####
#                               #
#               eNBs Configurations               #
#                               #
#####

*.eNodeB*.mobility.initFromDisplayString = false
*.eNodeB1.mobility.initialX = 155m
*.eNodeB1.mobility.initialY = 279m
*.eNodeB2.mobility.initialX = 614m
*.eNodeB2.mobility.initialY = 529m

*.eNodeB*.numX2Apps = 2    # one x2App per peering eNodeB
*.eNodeB*.x2App[*].server.localPort = 5000 + ancestorIndex(1) # Server ports
(x2App[0]=5000, x2App[1]=5001, ...)

#####eNodeB1#####
*.eNodeB1.x2App[0].client.connectAddress = "eNodeB2%x2ppp0"

#####eNodeB2#####
*.eNodeB2.x2App[0].client.connectAddress = "eNodeB1%x2ppp0"

```

```

**sctp.nagleEnabled = false          # if true, transmission of small packets will be delayed
on the X2

**sctp.enableHeartbeats = false

#####

#                               D2D SETTINGS                               #
#####

*.eNodeB*.d2dCapable = false
*.car[*].d2dCapable = false
**.amcMode = "D2D"

#####

# Select Channel Quality Indicator (CQI) for D2D transmissions #
#####

**.usePreconfiguredTxParams = false
**.d2dCqi = ${cqi=7}

#####

#                               App Server                               #
#####

**.server.numUdpApps = 1
*.server.udpApp[0].typename = "org.car2x.veins.subprojects.veins_hetvnet.SimpleServerApp"
*.server.udpApp[0].localPort = ${appserverport=4242}          # udp app
server socket port -> this ports must be configured when sending a msg to server IP PORT in
hetvnetDemoAPP

*.server.udpApp[0].period = 0.1s
# send packetinterval for server

*.server.udpApp[0].startTime = uniform(10.01s,10.02s)
# server start sending message time

*.server.udpApp[0].send_N_packets = 1                          #
number of packets that node[0] send to multicastgroup

*.server.udpApp[0].DstAppCarPort = ${appcarport}
# server send to car app port

*.server.udpApp[0].DtsIpFwdCar = "10.0.1.86"   # TO DO MANUALLY destination IP of the
forward car

#####

```



```

#                               NODE APP Transmitter node[0]                               #
#####
*.car[*].numUdpApps = 1
*.car[*].udpApp[0].typename = "org.car2x.veins.subprojects.veins_hetvnet.HetVNetDemoApp"
*.car[*].udpApp[0].localPortLte = ${appcarport=1000}
*.car[*].udpApp[0].localPortWlan = 1001
*.car[*].udpApp[0].destAddressLte = "224.0.0.10"

###PRIMER CLUSTER#####
*.car[0].udpApp[0].destAddressWlan = "224.0.2.1"

###SEGUNDO CLUSTER#####
*.car[1].udpApp[0].destAddressWlan = "224.0.2.11"

###TERCER CLUSTER#####
*.car[2].udpApp[0].destAddressWlan = "224.0.2.21"

###CUARTO CLUSTER#####
*.car[3].udpApp[0].destAddressWlan = "224.0.2.31"

###QUINTO CLUSTER#####
*.car[4].udpApp[0].destAddressWlan = "224.0.2.41"

*.car[*].udpApp[0].DstAppServerPort = ${appserverport}
# user the upd app server port
#####

#                               MULTICAST                               #
#
#####

*.car[*].configuratorLte.interfaces = "wlan"
*.car[*].configuratorWlan.interfaces = "wlan0"

```

```

*.car[*].configuratorLte.addressBase = "10.0.0.0"
*.car[*].configuratorWlan.addressBase = "2.0.0.0"
*.car[*].configuratorLte.mcastGroups = "224.0.0.10" # Add nodes to the LTE multicast group

*.car[0].configuratorWlan.mcastGroups = "224.0.2.1" # Add nodes to the WLAN multicast
group
*.car[1].configuratorWlan.mcastGroups = "224.0.2.11" # Add nodes to the WLAN multicast
group
*.car[2].configuratorWlan.mcastGroups = "224.0.2.21" # Add nodes to the WLAN multicast
group
*.car[3].configuratorWlan.mcastGroups = "224.0.2.31" # Add nodes to the WLAN multicast
group
*.car[4].configuratorWlan.mcastGroups = "224.0.2.41" # Add nodes to the WLAN multicast
group

#####

#                               CAR WLAN
#

#####

*.car[*].wlan[0].mgmtType = "Ieee80211MgmtAdhoc"
*.car[*].wlan[0].bitrate = 6Mbps
*.car[*].wlan[0].radio.transmitter.power = 20mW
*.car[*].wlan[0].radio.carrierFrequency = 5900e+6Hz
*.car[*].wlan[0].radio.bandwidth = 75e+6Hz

#####

#                               STATISTICS
#

#####

*.car[*].udpApp[*].statistics          =                ${pathTocsv}          =
"/root/EPN/Dimensionvariable/ochoautos/diezochomw/statistics.csv"      # save tx/rx
messages for all the cars

*.server.udpApp[*].statistics = ${pathTocsv}
# save tx/rx messages for server

```

```
[Config EPN]
```

```
*.veinsManager.launchConfig = xmlDoc("uio/centro.launchd.xml") # SUMO scenario
```

```
*.car[5..39].udpApp[0].startTime = uniform(0.02s,0.03s)
```

```
*.car[5..39].udpApp[0].packetSize = 200
```

```
*.car[0..4].udpApp[0].packetSize = 3000
```

```
*.car[*].udpApp[0].send_N_packets = 1
```

```
*.car[*].udpApp[0].period = 0.1s
```

```
*.car[0].udpApp[0].Forwarding_nodeId = 0 # **node index within the platooning to forward packet. Notice that the node index must be in the multicast group
```

```
*.car[1].udpApp[0].Forwarding_nodeId = 1
```

```
*.car[2].udpApp[0].Forwarding_nodeId = 2
```

```
*.car[3].udpApp[0].Forwarding_nodeId = 3
```

```
*.car[4].udpApp[0].Forwarding_nodeId = 4
```

## ANEXO C

```
#!/usr/bin/env python
# coding: utf-8
import numpy as np
import pandas as pd #Lee las librerias necesarias para analisis de Datos e importacion
de archivos
import csv
import statistics as stats
#####AGREGANDO CABECERAS AL ARCHIVO CSV
with open('statistics.csv',newline='') as f:
    r = csv.reader(f)
    data = [line for line in r]
with open('estadisticas.csv','w',newline='') as f:
    w = csv.writer(f)
    w.writerow(['Fuente', 'Indice', 'Destino', 'Estado', 'ID de Fuente', 'Tipo',
'Transmisor', 'Tamanomensaje', 'ID del mensaje', 'Tiempo en el que se creo el
mensaje', 'Tiempo de Simulacion del mensaje' ])
    w.writerows(data)
content = pd.read_csv('estadisticas.csv') #Lee el archivo donde se tiene las
estadisticas
numpaqLTEtx = content[content.Estado.isin(['fd']) &
content.Tipo.isin(['O'])].count()['Fuente']
numpaqLTERx = content[content.Estado.isin(['rx']) &
content.Tipo.isin(['O'])].count()['Fuente']
numpaqWLANtx = content[content.Estado.isin(['tx']) &
content.Tipo.isin(['1'])].count()['Fuente']
numpaqWLANrx = content[content.Estado.isin(['rx']) &
content.Tipo.isin(['1'])].count()['Fuente']
rendimientoLTE = (numpaqLTERx/numpaqLTEtx)*100
rendimientoWLAN = (numpaqWLANrx/numpaqWLANtx)*100
servidor = content[content.Fuente.isin(['server'])]
delaylte = servidor.iloc[:,10] - servidor.iloc[:,9]
```

```

promediodelaylte = stats.mean(delaylte)
autos = content[content.Estado.isin(['rx']) & content.Tipo.isin(['1'])]
delaywlan = autos.iloc[:,10] - autos.iloc[:,9]
promediodelaywlan = stats.mean(delaywlan)
bytesltetx= ((content[content.Estado.isin(['fd']) &
content.Tipo.isin(['0'])]).iloc[:,7].sum())/1000
byteslterx= ((content[content.Estado.isin(['rx']) &
content.Tipo.isin(['0'])]).iloc[:,7].sum())/1000
byteswlanrx= ((content[content.Estado.isin(['tx']) &
content.Tipo.isin(['1'])]).iloc[:,7].sum())/1000
byteswlanrx= ((content[content.Estado.isin(['rx']) &
content.Tipo.isin(['1'])]).iloc[:,7].sum())/1000
datos =
{'Simulacion':[8], 'Potencia_mw':[14], 'Paquetes_LTE_Enviados':[numpaqLTETx], 'Paquetes_LTE_Recibidos':[numpaqLTERx], 'Bytes_LTE_tx':[bytesltetx], 'Bytes_LTE_rx':[byteslterx], 'Rendimiento_LTE':[rendimientoLTE], 'Delay_LTE':[promediodelaylte], 'Paquetes_WLAN_Enviados':[numpaqWLANtx], 'Paquetes_WLAN_Recibidos':[numpaqWLANrx], 'Bytes_WLAN_tx':[byteswlanrx], 'Bytes_WLAN_rx':[byteswlanrx], 'Rendimiento_WLAN':[rendimientoWLAN], 'Delay_WLAN':[promediodelaywlan]}
df = pd.DataFrame(datos, columns =
['Simulacion', 'Potencia_mw', 'Paquetes_LTE_Enviados',
'Paquetes_LTE_Recibidos', 'Bytes_LTE_tx', 'Bytes_LTE_rx', 'Rendimiento_LTE', 'Delay_LTE', 'Paquetes_WLAN_Enviados', 'Paquetes_WLAN_Recibidos', 'Bytes_WLAN_tx', 'Bytes_WLAN_rx', 'Rendimiento_WLAN', 'Delay_WLAN'])
df.to_csv('dosmw.csv')

```

## ANEXO D

```
#!/usr/bin/env python
# coding: utf-8
import numpy as np
import pandas as pd #Lee las librerias necesarias para analisis de Datos e importacion
de archivos
import csv
import statistics as stats
import glob
import os
import seaborn as sb
from matplotlib import pyplot as plt
extension = 'csv'
all_filenames = [i for i in glob.glob('*mw*.{0}'.format(extension))]

#combine all files in the list
combined_csv = pd.concat([pd.read_csv(f) for f in all_filenames ])
#export to csv
combined_csv.to_csv( "union_cuatroautos.csv", index=False, encoding='utf-8-sig')
tabla = pd.read_csv("union_cuatroautos.csv")
get_ipython().run_line_magic('matplotlib', 'inline')
orden = tabla.sort_values(['Simulacion'], ascending=True)
orden.to_csv("union_cuatroautos.csv", index=False, encoding='utf-8-sig')

#bins = orden.iloc[:,2]
bins = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]
mapeado = range(len(bins))

##### LTE #####
```

```

plt.figure(figsize=(25,10))
plt.subplot(2,3,1)
plt.plot(orden.Potencia_mw,orden.Paquetes_LTE_Enviados,"-o")
plt.xticks(mapeado,bins)
plt.xlabel('Potencia de 802.11 \n en mw', fontsize =15)
plt.title('Paquetes LTE Tx \n vs Potencia en mw',fontsize = 20,color = 'red')
plt.ylabel('Num paquetes', fontsize =15)
plt.subplot(2,3,2)
plt.plot(orden.Potencia_mw,orden.Paquetes_LTE_Recibidos,"-o")
plt.xticks(mapeado,bins)
plt.xlabel('Potencia de 802.11 \n en mw',fontsize =15)
plt.title('Paquetes LTE Rx \n vs Potencia en mw',fontsize =20,color = 'red')
plt.ylabel('Num paquetes',fontsize =15)
plt.subplot(2,3,3)
plt.plot(orden.Potencia_mw,orden.Bytes_LTE_tx,"-o")
plt.xticks(mapeado,bins)
plt.xlabel('Potencia de 802.11 \n en mw',fontsize =15)
plt.title('Bytes LTE Tx \n vs Potencia en mw',fontsize =20,color = 'red')
plt.ylabel('Kbytes',fontsize =15)
plt.subplot(2,3,4)
plt.plot(orden.Potencia_mw,orden.Bytes_LTE_rx,"-o")
plt.xticks(mapeado,bins)
plt.xlabel('Potencia de 802.11 \n en mw',fontsize =15)
plt.title('Bytes LTE Rx \n vs Potencia en mw',fontsize =20,color = 'red')
plt.ylabel('kbytes',fontsize =15)
plt.subplot(2,3,5)
plt.plot(orden.Potencia_mw,orden.Rendimiento_LTE,"-o")
plt.xticks(mapeado,bins)
plt.title('% Rendimiento vs \n Potencia en mw',fontsize =20,color = 'red')

```

```

plt.ylabel('% Paquetes',fontsize =15)
plt.xlabel('Potencia de 802.11 \n en mw',fontsize =15)
plt.subplot(2,3,6)
plt.plot(orden.Potencia_mw,orden.Delay_LTE,"-o")
plt.xticks(mapeado,bins)
plt.title('Delay paquetes LTE \n vs Potencia en mw',fontsize =20,color = 'red')
plt.ylabel('Delay en segundos',fontsize =15)
plt.xlabel('Potencia de 802.11 \n en mw',fontsize =15)
plt.tight_layout()
plt.savefig('LTE.jpeg')

##### WLAN#####
plt.figure(figsize=(30,10))
plt.subplot(2,3,1)
plt.plot(orden.Potencia_mw,orden.Paquetes_WLAN_Enviados,"-o")
plt.xticks(mapeado,bins,fontsize =15)
plt.yticks(fontsize =15)
plt.xlabel('Potencia de 802.11 \n en mw', fontsize =20)
plt.title('Paquetes WLAN Tx \n vs Potencia en mw',color = 'red', fontsize =25)
plt.ylabel('Num paquetes', fontsize =20)
plt.subplot(2,3,2)
plt.plot(orden.Potencia_mw,orden.Paquetes_WLAN_Recibidos,"-o")
plt.xticks(mapeado,bins,fontsize =15)
plt.yticks(fontsize =15)
plt.xlabel('Potencia de 802.11 \n en mw', fontsize =20)
plt.title('Paquetes WLAN Rx \n vs Potencia en mw',color = 'red', fontsize =25)
plt.ylabel('Num paquetes', fontsize =20)
plt.subplot(2,3,3)
plt.plot(orden.Potencia_mw,orden.Bytes_WLAN_tx,"-o")

```



```

plt.xticks(mapeado,bins)
plt.xlabel('Potencia de 802.11 \n en mw',fontsize =15)
plt.title('Bytes WLAN Tx \n vs Potencia en mw',fontsize =20,color = 'red')
plt.ylabel('Kbytes',fontsize =15)
plt.subplot(2,3,4)
plt.plot(orden.Potencia_mw,orden.Bytes_WLAN_rx,"-o")
plt.xticks(mapeado,bins)
plt.xlabel('Potencia de 802.11 \n en mw',fontsize =15)
plt.title('Bytes WLAN Rx \n vs Potencia en mw',fontsize =20,color = 'red')
plt.ylabel('kbytes',fontsize =15)
plt.subplot(2,3,5)
plt.plot(orden.Potencia_mw,orden.Rendimiento_WLAN,"-o")
plt.xticks(mapeado,bins)
plt.title('% Rendimiento vs \n Potencia en mw',fontsize =20,color = 'red')
plt.ylabel('% Paquetes',fontsize =15)
plt.xlabel('Potencia de 802.11 \n en mw',fontsize =15)
plt.subplot(2,3,6)
plt.plot(orden.Potencia_mw,orden.Delay_WLAN,"-o")
plt.xticks(mapeado,bins)
plt.title('Delay paquetes WLAN \n vs Potencia en mw',fontsize =20,color = 'red')
plt.ylabel('Delay en segundos',fontsize =15)
plt.xlabel('Potencia de 802.11 \n en mw',fontsize =15)
plt.tight_layout()
plt.savefig('WLAN.jpeg')

```

## ANEXO E

```
#!/usr/bin/env python
# coding: utf-8
import numpy as np
import pandas as pd #Lee las librerias necesarias para analisis de Datos e importacion
de archivos
import csv
import statistics as stats
import glob
import os
import seaborn as sb
from matplotlib import pyplot as plt
simuinicio = pd.read_csv("union_inicio.csv")
simumitad = pd.read_csv("union_mitad.csv")
simufinal = pd.read_csv("union_final.csv")
get_ipython().run_line_magic('matplotlib', 'inline')
#OBTIENE DATOS DE LA SIMULACION PARA EL EJE DE LAS X
#bins = simuinicio.iloc[:,2]
bins = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]
mapeado = range(len(bins))
plt.figure(figsize=(25,10))
plt.subplot(2,2,1)
plt.plot(simuinicio.Potencia_mw,simuinicio.Paquetes_LTE_Enviados,"-o",label = "inicio
del cluster")
plt.plot(simumitad.Potencia_mw,simumitad.Paquetes_LTE_Enviados,"-o", label = "mitad
del cluster")
plt.plot(simufinal.Potencia_mw,simufinal.Paquetes_LTE_Enviados,"-o", label = "final del
cluster")
plt.legend()
```

```

plt.xticks(mapeado,bins)
plt.xlabel('Potencia de 802.11 \n en mw' , fontsize =15)
plt.title('Paquetes LTE Tx \n vs Potencia en mw' ,fontsize = 20,color = 'red')
plt.ylabel('Num paquetes' , fontsize =15)
plt.subplot(2,2,2)
plt.plot(simuinicio.Potencia_mw,simuinicio.Paquetes_LTE_Recibidos,"-o",label = "inicio
del cluster")
plt.plot(simumitad.Potencia_mw,simumitad.Paquetes_LTE_Recibidos,"-o",label =
"mitad del cluster")
plt.plot(simufinal.Potencia_mw,simufinal.Paquetes_LTE_Recibidos,"-o",label = "final
del cluster")
plt.legend()
plt.xticks(mapeado,bins)
plt.xlabel('Potencia de 802.11 \n en mw' ,fontsize =15)
plt.title('Paquetes LTE Rx \n vs Potencia en mw' ,fontsize =20,color = 'red')
plt.ylabel('Num paquetes' ,fontsize =15)
plt.subplot(2,2,3)
plt.plot(simuinicio.Potencia_mw,simuinicio.Rendimiento_LTE,"-o",label = "inicio del
cluster")
plt.plot(simumitad.Potencia_mw,simumitad.Rendimiento_LTE,"-o",label = "mitad del
cluster")
plt.plot(simufinal.Potencia_mw,simufinal.Rendimiento_LTE,"-o",label = "final del
cluster")
plt.legend()
plt.xticks(mapeado,bins)
plt.title('Rendimiento % LTE vs \n Potencia en mw' ,fontsize =20,color = 'red')
plt.ylabel('% Paquetes' ,fontsize =15)
plt.ylim(0, 110)
plt.xlabel('Potencia de 802.11 \n en mw' ,fontsize =15)
plt.subplot(2,2,4)
plt.plot(simuinicio.Potencia_mw,simuinicio.Delay_LTE,"-o",label = "inicio del cluster")

```

```

plt.plot(simumidad.Potencia_mw,simumidad.Delay_LTE,"-o",label = "mitad del cluster")
plt.plot(simufinal.Potencia_mw,simufinal.Delay_LTE,"-o",label = "final del cluster")
plt.legend()
plt.xticks(mapeado,bins)
plt.title('Delay paquetes LTE \n vs Potencia en mw',fontsize =20,color = 'red')
plt.ylabel('Delay en segundos',fontsize =15)
plt.xlabel('Potencia de 802.11 \n en mw',fontsize =15)
plt.tight_layout()
plt.savefig('LTE.jpeg')
plt.figure(figsize=(30,10))
plt.subplot(2,2,1)
plt.plot(simuinicio.Potencia_mw,simuinicio.Paquetes_WLAN_Enviados,"-o",label =
"inicio del cluster")
plt.plot(simumidad.Potencia_mw,simumidad.Paquetes_WLAN_Enviados,"-o",label =
"mitad del cluster")
plt.plot(simufinal.Potencia_mw,simufinal.Paquetes_WLAN_Enviados,"-o",label = "final
del cluster")
plt.legend()
plt.xticks(mapeado,bins,fontsize =15)
plt.yticks(fontsize =15)
plt.xlabel('Potencia de 802.11 \n en mw', fontsize =20)
plt.title('Paquetes WLAN Tx \n vs Potencia en mw',color = 'red', fontsize =25)
plt.ylabel('Num paquetes', fontsize =20)
plt.subplot(2,2,2)
plt.plot(simuinicio.Potencia_mw,simuinicio.Paquetes_WLAN_Recibidos,"-o",label =
"inicio del cluster")
plt.plot(simumidad.Potencia_mw,simumidad.Paquetes_WLAN_Recibidos,"-o",label =
"mitad del cluster")
plt.plot(simufinal.Potencia_mw,simufinal.Paquetes_WLAN_Recibidos,"-o",label = "final
del cluster")
plt.legend()

```

```

plt.xticks(mapeado,bins,fontsize =15)
plt.yticks(fontsize =15)
plt.xlabel('Potencia de 802.11 \n en mw' , fontsize =20)
plt.title('Paquetes WLAN Rx \n vs Potencia en mw' ,color = 'red' , fontsize =25)
plt.ylabel('Num paquetes' , fontsize =20)
plt.subplot(2,2,3)
plt.plot(simuinicio.Potencia_mw,simuinicio.Rendimiento_WLAN,"-o",label = "inicio del
cluster")
plt.plot(simumitad.Potencia_mw,simumitad.Rendimiento_WLAN,"-o",label = "mitad del
cluster")
plt.plot(simufinal.Potencia_mw,simufinal.Rendimiento_WLAN,"-o",label = "final del
cluster")
plt.legend()
plt.xticks(mapeado,bins)
plt.title('Rendimiento % WLAN vs \n Potencia en mw' ,fontsize =20,color = 'red')
plt.ylabel('% Paquetes' ,fontsize =15)
plt.xlabel('Potencia de 802.11 \n en mw' ,fontsize =15)
plt.ylim(0, 110)
plt.subplot(2,2,4)
plt.plot(simuinicio.Potencia_mw,simuinicio.Delay_WLAN,"-o",label = "inicio del
cluster")
plt.plot(simumitad.Potencia_mw,simumitad.Delay_WLAN,"-o",label = "mitad del
cluster")
plt.plot(simufinal.Potencia_mw,simufinal.Delay_WLAN,"-o",label = "final del cluster")
plt.legend()
plt.xticks(mapeado,bins)
plt.title('Delay paquetes WLAN \n vs Potencia en mw' ,fontsize =20,color = 'red')
plt.ylabel('Delay en segundos' ,fontsize =15)
plt.xlabel('Potencia de 802.11 \n en mw' ,fontsize =15)
plt.tight_layout()
plt.savefig('WLAN.jpeg')

```

## ANEXO F

```
#!/usr/bin/env python
# coding: utf-8

import numpy as np

import pandas as pd #Lee las librerias necesarias para analisis de Datos e importacion
de archivos

import csv

import statistics as stats

#####AGREGANDO CABECERAS AL ARCHIVO CSV

with open('statistics.csv',newline='') as f:
    r = csv.reader(f)
    data = [line for line in r]

with open('estadisticas.csv','w',newline='') as f:
    w = csv.writer(f)

    w.writerow(['Fuente', 'Indice', 'Destino', 'Estado', 'ID de Fuente', 'Tipo',
'Transmisor', 'Tamanomensaje', 'ID del mensaje', 'Tiempo en el que se creo el
mensaje', 'Tiempo de Simulacion del mensaje' ])

    w.writerows(data)

content = pd.read_csv('estadisticas.csv') #Lee el archivo donde se tiene las
estadisticas

numpaqLTEtx = content[content.Estado.isin(['fd']) &
content.Tipo.isin(['O'])].count()['Fuente']

numpaqLTERx = content[content.Estado.isin(['rx']) &
content.Tipo.isin(['O'])].count()['Fuente']

numpaqWLANtx = content[content.Estado.isin(['tx']) &
content.Tipo.isin(['1'])].count()['Fuente']

numpaqWLANrx = content[content.Estado.isin(['rx']) &
content.Tipo.isin(['1'])].count()['Fuente']

rendimientoLTE = (numpaqLTERx/numpaqLTEtx)*100
rendimientoWLAN = (numpaqWLANrx/numpaqWLANtx)*100
servidor = content[content.Fuente.isin(['server'])]
```

```

delaylte = servidor.iloc[:,10] - servidor.iloc[:,9]
promediodelaylte = stats.mean(delaylte)

autos = content[content.Estado.isin(['rx']) & content.Tipo.isin(['1'])]
delaywlan = autos.iloc[:,10] - autos.iloc[:,9]
promediodelaywlan = stats.mean(delaywlan)

bytesltetx= ((content[content.Estado.isin(['fd']) &
content.Tipo.isin(['0'])]).iloc[:,7].sum())/1000
byteslterx= ((content[content.Estado.isin(['rx']) &
content.Tipo.isin(['0'])]).iloc[:,7].sum())/1000
byteswlanrx= ((content[content.Estado.isin(['tx']) &
content.Tipo.isin(['1'])]).iloc[:,7].sum())/1000
byteswlanrx= ((content[content.Estado.isin(['rx']) &
content.Tipo.isin(['1'])]).iloc[:,7].sum())/1000

datos =
{'Simulacion':[1], 'Intervalo_msg_ms':[100], 'Paquetes_LTE_Enviados':[numpaqLTERtx]
, 'Paquetes_LTE_Recibidos':[numpaqLTERrx], 'Bytes_LTE_tx':[bytesltetx], 'Bytes_LTE
_rx':[byteslterx], 'Rendimiento_LTE':[rendimientoLTE], 'Delay_LTE':[promediodelayl
te], 'Paquetes_WLAN_Enviados':[numpaqWLANtx], 'Paquetes_WLAN_Recibidos':[nu
mpaqWLANrx], 'Bytes_WLAN_tx':[byteswlanrx], 'Bytes_WLAN_rx':[byteswlanrx], '
Rendimiento_WLAN':[rendimientoWLAN], 'Delay_WLAN':[promediodelaywlan]}

df = pd.DataFrame(datos, columns =
['Simulacion', 'Intervalo_msg_ms', 'Paquetes_LTE_Enviados',
'Paquetes_LTE_Recibidos', 'Bytes_LTE_tx', 'Bytes_LTE_rx', 'Rendimiento_LTE', 'Del
ay_LTE', 'Paquetes_WLAN_Enviados', 'Paquetes_WLAN_Recibidos', 'Bytes_WLAN_t
x', 'Bytes_WLAN_rx', 'Rendimiento_WLAN', 'Delay_WLAN'])
df.to_csv('cienms.csv')

```

## ANEXO G

```
#!/usr/bin/env python
# coding: utf-8

import numpy as np

import pandas as pd #Lee las librerias necesarias para analisis de Datos e importacion
de archivos

import csv

import statistics as stats

import glob

import os

import seaborn as sb

from matplotlib import pyplot as plt

extension = 'csv'

all_filenames = [i for i in glob.glob('*s*.{0}'.format(extension))]

#combine all files in the list

combined_csv = pd.concat([pd.read_csv(f) for f in all_filenames ])

#export to csv

combined_csv.to_csv( "union_cuatroautos.csv", index=False, encoding='utf-8-sig')

tabla = pd.read_csv("union_cuatroautos.csv")

get_ipython().run_line_magic('matplotlib', 'inline')

orden = tabla.sort_values([' Simulacion' ], ascending=True)

orden.to_csv("union_cuatroautos.csv", index=False, encoding='utf-8-sig')

bins = [100,200,300,400,500,600,700,800,900,1000]

mapeado = range(len(bins))

plt.figure(figsize=(25,10))
```



```

plt.subplot(2,2,1)
plt.plot(orden.Intervalo_msg_ms,orden.Paquetes_LTE_Enviados,"-o")
plt.xlabel('Intervalo de \n msg en ms', fontsize =15)
plt.title('Paquetes LTE Tx \n vs Potencia en mw', fontsize = 20,color = 'red')
plt.ylabel('Num paquetes', fontsize =15)
plt.subplot(2,2,2)
plt.plot(orden.Intervalo_msg_ms,orden.Paquetes_LTE_Recibidos,"-o")
plt.xlabel('Intervalo de \n msg en ms', fontsize =15)
plt.title('Paquetes LTE Rx \n vs Potencia en mw', fontsize =20,color = 'red')
plt.ylabel('Num paquetes', fontsize =15)
plt.subplot(2,2,3)
plt.plot(orden.Intervalo_msg_ms,orden.Rendimiento_LTE,"-o")
plt.title('% Rendimiento vs \n Potencia en mw', fontsize =20,color = 'red')
plt.ylabel('% Paquetes', fontsize =15)
plt.xlabel('Intervalo de \n msg en ms', fontsize =15)
plt.subplot(2,2,4)
plt.plot(orden.Intervalo_msg_ms,orden.Delay_LTE,"-o")
plt.title('Delay paquetes LTE \n vs Potencia en mw', fontsize =20,color = 'red')
plt.ylabel('Delay en segundos', fontsize =15)
plt.xlabel('Intervalo de \n msg en ms', fontsize =15)
plt.tight_layout()
plt.savefig('LTE.jpeg')
plt.figure(figsize=(30,10))
plt.subplot(2,2,1)
plt.plot(orden.Intervalo_msg_ms,orden.Paquetes_WLAN_Enviados,"-o")
#plt.xticks(mapeado,bins,fontsize =15)
plt.yticks(fontsize =15)
plt.xlabel('Intervalo de \n msg en ms', fontsize =20)
plt.title('Paquetes WLAN Tx \n vs Potencia en mw', color = 'red', fontsize =25)

```

```

plt.ylabel('Num paquetes', fontsize =20)
plt.subplot(2,2,2)
plt.plot(orden.Intervalo_msg_ms,orden.Paquetes_WLAN_Recibidos,"-o")
#plt.xticks(mapeado,bins,fontsize =15)
plt.yticks(fontsize =15)
plt.xlabel('Intervalo de \n msg en ms', fontsize =20)
plt.title('Paquetes WLAN Rx \n vs Potencia en mw',color = 'red', fontsize =25)
plt.ylabel('Num paquetes', fontsize =20)
plt.subplot(2,2,3)
plt.plot(orden.Intervalo_msg_ms,orden.Rendimiento_WLAN,"-o")
#plt.xticks(mapeado,bins)
plt.title('% Rendimiento vs \n Potencia en mw',fontsize =20,color = 'red')
plt.ylabel('% Paquetes',fontsize =15)
plt.xlabel('Intervalo de \n msg en ms',fontsize =15)
plt.subplot(2,2,4)
plt.plot(orden.Intervalo_msg_ms,orden.Delay_WLAN,"-o")
#plt.xticks(mapeado,bins)
plt.title('Delay paquetes WLAN \n vs Potencia en mw',fontsize =20,color = 'red')
plt.ylabel('Delay en segundos',fontsize =15)
plt.xlabel('Intervalo de \n msg en ms',fontsize =15)
plt.tight_layout()
plt.savefig('WLAN.jpeg')

```

## ANEXO H

```
#!/usr/bin/env python
# coding: utf-8
import numpy as np
import pandas as pd #Lee las librerias necesarias para analisis de Datos e importacion
de archivos
import csv
import statistics as stats
import glob
import os
import seaborn as sb
from matplotlib import pyplot as plt
simudos = pd.read_csv("union_dosautos.csv")
simucuatro = pd.read_csv("union_cuatroautos.csv")
simuseis = pd.read_csv("union_seisautos.csv")
simuocho = pd.read_csv("union_ochoautos.csv")
simudiez = pd.read_csv("union_diezautos.csv")
get_ipython().run_line_magic('matplotlib', 'inline')

#OBTIENE DATOS DE LA SIMULACION PARA EL EJE DE LAS X
#bins = simuinicio.iloc[:,2]
bins = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]
mapeado = range(len(bins))
plt.figure(figsize=(25,10))
plt.subplot(2,2,1)
plt.plot(simudos.Intervalo_msg_ms,simudos.Paquetes_LTE_Enviados,"-o",label = "2
autos por cluster")
plt.plot(simucuatro.Intervalo_msg_ms,simucuatro.Paquetes_LTE_Enviados,"-o", label
= "4 autos por cluster")
```

```

plt.plot(simuseis.Intervalo_msg_ms,simuseis.Paquetes_LTE_Enviados,"-o", label = "6
autos por cluster")

plt.plot(simuocho.Intervalo_msg_ms,simuocho.Paquetes_LTE_Enviados,"-o", label = "8
autos por cluster")

plt.plot(simudiez.Intervalo_msg_ms,simudiez.Paquetes_LTE_Enviados,"-o", label = "10
autos por cluster")

plt.legend()

plt.xlabel('Intervalo de \n msg en ms', fontsize =15)

plt.title('Paquetes LTE Tx \n vs Intervalo de \n msg en ms',fontsize = 20,color = 'red')

plt.ylabel('Num paquetes', fontsize =15)

plt.subplot(2,2,2)

plt.plot(simudos.Intervalo_msg_ms,simudos.Paquetes_LTE_Recibidos,"-o",label = "2
autos por cluster")

plt.plot(simucuatro.Intervalo_msg_ms,simucuatro.Paquetes_LTE_Recibidos,"-o",label
= "4 autos por cluster")

plt.plot(simuseis.Intervalo_msg_ms,simuseis.Paquetes_LTE_Recibidos,"-o",label = "6
autos por cluster")

plt.plot(simuocho.Intervalo_msg_ms,simuocho.Paquetes_LTE_Recibidos,"-o",label = "8
autos por cluster")

plt.plot(simudiez.Intervalo_msg_ms,simudiez.Paquetes_LTE_Recibidos,"-o",label = "10
autos por cluster")

plt.legend()

plt.xlabel('Intervalo de \n msg en ms',fontsize =15)

plt.title('Paquetes LTE Rx \n vs Intervalo de \n msg en ms',fontsize =20,color = 'red')

plt.ylabel('Num paquetes',fontsize =15)

plt.subplot(2,2,3)

plt.plot(simudos.Intervalo_msg_ms,simudos.Rendimiento_LTE,"-o",label = "2 autos por
cluster")

plt.plot(simucuatro.Intervalo_msg_ms,simucuatro.Rendimiento_LTE,"-o",label = "4
autos por cluster")

plt.plot(simuseis.Intervalo_msg_ms,simuseis.Rendimiento_LTE,"-o",label = "6 autos
por cluster")

```

```

plt.plot(simuocho.Intervalo_msg_ms,simuocho.Rendimiento_LTE,"-o",label = "8 autos
por cluster")

plt.plot(simudiez.Intervalo_msg_ms,simudiez.Rendimiento_LTE,"-o",label = "10 autos
por cluster")

plt.legend()

plt.title('Rendimiento % LTE vs \n Intervalo de \n msg en ms',fontsize =20,color =
'red')

plt.ylabel('% Paquetes',fontsize =15)

plt.xlabel('Intervalo de \n msg en ms',fontsize =15)

plt.subplot(2,2,4)

plt.plot(simudos.Intervalo_msg_ms,simudos.Delay_LTE,"-o",label = "2 autos por
cluster")

plt.plot(simucuatro.Intervalo_msg_ms,simucuatro.Delay_LTE,"-o",label = "4 autos por
cluster")

plt.plot(simuseis.Intervalo_msg_ms,simuseis.Delay_LTE,"-o",label = "6 autos por
cluster")

plt.plot(simuocho.Intervalo_msg_ms,simuocho.Delay_LTE,"-o",label = "8 autos por
cluster")

plt.plot(simudiez.Intervalo_msg_ms,simudiez.Delay_LTE,"-o",label = "10 autos por
cluster")

plt.legend()

plt.title('Delay paquetes LTE \n vs Intervalo de \n msg en ms',fontsize =20,color =
'red')

plt.ylabel('Delay en segundos',fontsize =15)

plt.xlabel('Intervalo de \n msg en ms',fontsize =15)

plt.tight_layout()

plt.savefig('LTE.jpeg')

plt.figure(figsize=(30,10))

plt.subplot(2,2,1)

plt.plot(simudos.Intervalo_msg_ms,simudos.Paquetes_WLAN_Enviados,"-o",label = "2
autos por cluster")

plt.plot(simucuatro.Intervalo_msg_ms,simucuatro.Paquetes_WLAN_Enviados,"-
o",label = "4 autos por cluster")

```

```

plt.plot(simuseis.Intervalo_msg_ms,simuseis.Paquetes_WLAN_Enviados,"-x",label =
"6 autos por cluster")

plt.plot(simuocho.Intervalo_msg_ms,simuocho.Paquetes_WLAN_Enviados,"-o",label =
"8 autos por cluster")

plt.plot(simudiez.Intervalo_msg_ms,simudiez.Paquetes_WLAN_Enviados,"-x",label =
"10 autos por cluster")

plt.legend()

plt.yticks(fontsize =15)

plt.xlabel('Intervalo de \n msg en ms', fontsize =20)

plt.title('Paquetes WLAN Tx \n vs Intervalo de \n msg en ms',color = 'red', fontsize
=25)

plt.ylabel('Num paquetes', fontsize =20)

plt.subplot(2,2,2)

plt.plot(simudos.Intervalo_msg_ms,simudos.Paquetes_WLAN_Recibidos,"-o",label =
"2 autos por cluster")

plt.plot(simucuatro.Intervalo_msg_ms,simucuatro.Paquetes_WLAN_Recibidos,"-
o",label = "4 autos por cluster")

plt.plot(simuseis.Intervalo_msg_ms,simuseis.Paquetes_WLAN_Recibidos,"-o",label =
"6 autos por cluster")

plt.plot(simuocho.Intervalo_msg_ms,simuocho.Paquetes_WLAN_Recibidos,"-o",label =
"8 autos por cluster")

plt.plot(simudiez.Intervalo_msg_ms,simudiez.Paquetes_WLAN_Recibidos,"-o",label =
"10 autos por cluster")

plt.legend()

plt.yticks(fontsize =15)

plt.xlabel('Intervalo de \n msg en ms', fontsize =20)

plt.title('Paquetes WLAN Rx \n vs Intervalo de \n msg en ms',color = 'red', fontsize
=25)

plt.ylabel('Num paquetes', fontsize =20)

plt.subplot(2,2,3)

plt.plot(simudos.Intervalo_msg_ms,simudos.Rendimiento_WLAN,"-o",label = "2 autos
por cluster")

```

```

plt.plot(simucuatro.Intervalo_msg_ms,simucuatro.Rendimiento_WLAN,"-o",label = "4
autos por cluster")

plt.plot(simuseis.Intervalo_msg_ms,simuseis.Rendimiento_WLAN,"-o",label = "6 autos
por cluster")

plt.plot(simuocho.Intervalo_msg_ms,simuocho.Rendimiento_WLAN,"-o",label = "8
autos por cluster")

plt.plot(simudiez.Intervalo_msg_ms,simudiez.Rendimiento_WLAN,"-o",label = "10
autos por cluster")

plt.legend()

plt.title('Rendimiento % WLAN vs \n Intervalo de \n msg en ms',fontsize =20,color =
'red')

plt.ylabel('% Paquetes',fontsize =15)

plt.xlabel('Intervalo de \n msg en ms',fontsize =15)

plt.subplot(2,2,4)

plt.plot(simudos.Intervalo_msg_ms,simudos.Delay_WLAN,"-o",label = "2 autos por
cluster")

plt.plot(simucuatro.Intervalo_msg_ms,simucuatro.Delay_WLAN,"-o",label = "4 autos
por cluster")

plt.plot(simuseis.Intervalo_msg_ms,simuseis.Delay_WLAN,"-o",label = "6 autos por
cluster")

plt.plot(simuocho.Intervalo_msg_ms,simuocho.Delay_WLAN,"-o",label = "8 autos por
cluster")

plt.plot(simudiez.Intervalo_msg_ms,simudiez.Delay_WLAN,"-o",label = "10 autos por
cluster")

plt.legend()

plt.title('Delay paquetes WLAN \n vs Intervalo de \n msg en ms',fontsize =20,color =
'red')

plt.ylabel('Delay en segundos',fontsize =15)

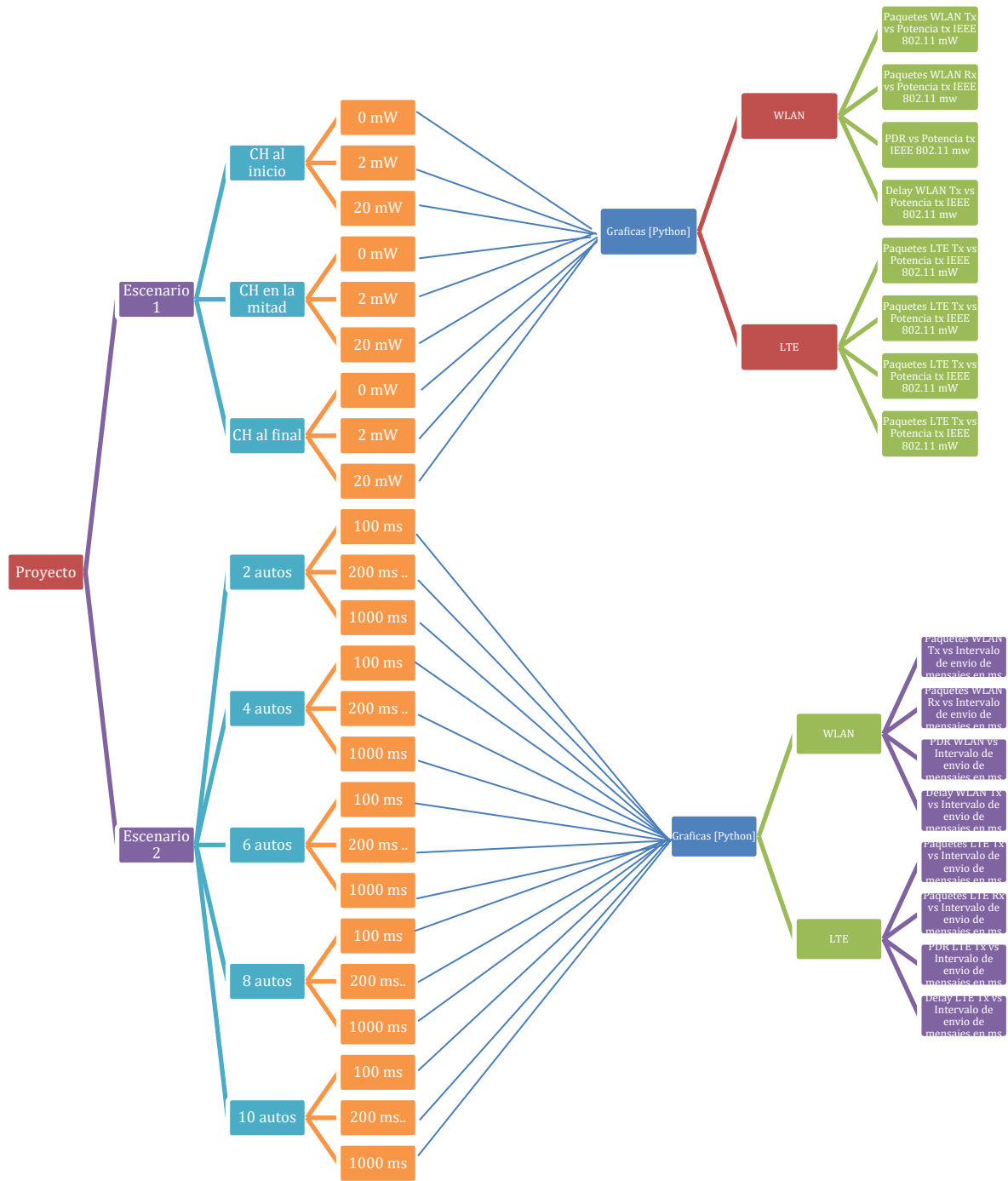
plt.xlabel('Intervalo de \n msg en ms',fontsize =15)

plt.tight_layout()

plt.savefig('WLAN.jpeg')

```

# ANEXO I



Anexo I. Esquema de ejecución de escenarios.



## ORDEN DE EMPASTADO