



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y  
TELECOMUNICACIONES**

**TRABAJO DE INTEGRACIÓN CURRICULAR**

Previo a la obtención del grado académico en  
**INGENIERO EN TELECOMUNICACIONES**

Evaluación del rendimiento de las comunicaciones vehiculares C –  
V2X

**AUTOR**

Carvajal Peña Jhostyn Iván

**DOCENTE TUTOR**

Ing. Chamba Macas Fernando Vinicio. Mgt

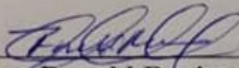
**LA LIBERTAD- Ecuador**

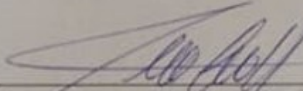
**Año 2025**

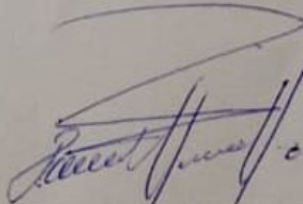


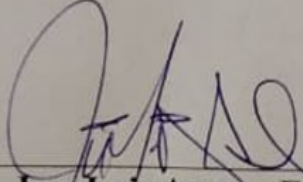
**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y  
TELECOMUNICACIONES**

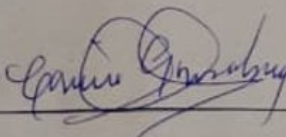
**TRIBUNAL DE SUSTENTACIÓN**

  
Ing. Ronald Rovira Jurado, PhD.  
DIRECTOR DE LA CARRERA

  
Ing. Fernando Chamba Macas, Mgt.  
DOCENTE TUTOR

  
Ing. Daniel Jaramillo Chamba, Mgr.  
DOCENTE ESPECIALISTA

  
Ing. Luis Amaya Fariño, Mgt.  
DOCENTE DE ÁREA

  
Ing. Corina Gonzabay De La A, MSc.  
DOCENTE ESPECIALISTA



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y  
TELECOMUNICACIONES**

## CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por Mariscal Méndez Daniela Nicole, como requerimiento para la obtención del título de Ingeniero en Telecomunicaciones.

La Libertad, a los días del mes de diciembre del año 2025

### TUTOR

A handwritten signature in black ink, appearing to read "F. Chamba Macas", is written over a horizontal line.

**Ing. Fernando Chamba Macas. Mgt**



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y  
TELECOMUNICACIONES**

**DECLARACIÓN DE RESPONSABILIDAD**

Yo, **CARVAJAL PEÑA JHOSTYN IVÁN**

**DECLARO QUE:**

El trabajo de Titulación, Evaluación de las comunicaciones vehiculares C-V2X previo a la obtención del título en Ingeniero en Telecomunicaciones, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los días del mes de diciembre del año 2025

**EL AUTOR**

---

**Carvajal Peña Jhostyn Iván**



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y  
TELECOMUNICACIONES**

## CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado Evaluación de las comunicaciones vehiculares C-V2X, presentado por el estudiante, Carvajal Peña Jhostyn Ivan fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al XX%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.

 CERTIFICADO DE ANÁLISIS  
magister

**TESIS FINAL MODIFICADA  
(1)**

**13%**  
Textos sospechosos

- 6% Similitudes  
< 1% similitudes entre comillas  
0% entre las fuentes mencionadas
- 8% Idiomas no reconocidos
- 14% Textos potencialmente generados por la IA (ignorado)

Nombre del documento: TESIS FINAL MODIFICADA (1).docx ID del documento: e619928de1ed7e524c17ef8f18e6b5c5cce44e5d Tamaño del documento original: 6,76 MB	Depositante: FERNANDO VINICIO CHAMBA MACAS Fecha de depósito: 2/12/2025 Tipo de carga: interface fecha de fin de análisis: 3/12/2025	Número de palabras: 18.221 Número de caracteres: 126.564
---	---	---

**TUTOR**

**Ing. Chamba Macas Fernando. Mgt**



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

## AUTORIZACIÓN

Yo, **CARVAJAL PEÑA JHOSTYN IVÁN**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.[1]

Cedo los derechos en línea patrimoniales de artículo profesional de alto nivel con fines de difusión pública, además apruebo la reproducción de este artículo académico dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, a los días del mes de diciembre del año 2025

**EL AUTOR**

---

**Jhostyn Iván Carvajal Peña**

# AGRADECIMIENTO

*Al final somos una mezcla de todas las personas que han sido parte de nuestro camino. Me alegra saber que algunas de ellas son parte esencial de quien soy hoy.*

*Jhostyn Iván Carvajal Peña*

# DEDICATORIA

A mi familia, por su apoyo incondicional y por creer siempre en mí.

A mis maestros, por compartir su conocimiento y guiarme en este camino académico.

A mis amigos, por su aliento constante y por hacer este recorrido más llevadero.

A todos aquellos que de una u otra forma contribuyeron a la realización de este trabajo.

Gracias por ser parte fundamental de este logro.

*Jhostyn Iván Carvajal Peña.*

## INDICE GENERAL

<b>TÍTULO DEL TRABAJO DE TITULACIÓN</b>	<b>;</b>	<b>Error!</b>	<b>Marcador no definido.</b>
<b>TRIBUNAL DE SUSTENTACIÓN</b>	<b>.....</b>	<b>;</b>	<b>Error! Marcador no definido.</b>
<b>CERTIFICACIÓN</b>	<b>.....</b>	<b>;</b>	<b>Error! Marcador no definido.</b>
<b>DECLARACIÓN DE RESPONSABILIDAD</b>	<b>.....</b>	<b>;</b>	<b>Error! Marcador no definido.</b>
<b>DECLARO QUE:</b>	<b>.....</b>	<b>;</b>	<b>Error! Marcador no definido.</b>
<b>CERTIFICACIÓN DE ANTIPLAGIO</b>	<b>.....</b>	<b>;</b>	<b>Error! Marcador no definido.</b>
<b>AUTORIZACIÓN</b>	<b>.....</b>	<b>;</b>	<b>Error! Marcador no definido.</b>
<b>AGRADECIMIENTO</b>	<b>.....</b>		<b>VII</b>
<b>DEDICATORIA</b>	<b>.....</b>		<b>1</b>
<b>INDICE GENERAL</b>	<b>.....</b>		<b>2</b>
<b>ÍNDICE DE FIGURAS</b>	<b>.....</b>		<b>5</b>
<b>ÍNDICE DE TABLAS</b>	<b>.....</b>		<b>6</b>
<b>CAPÍTULO I</b>	<b>.....</b>		<b>14</b>
<b>1. Fundamentación</b>	<b>.....</b>		<b>14</b>
<b>1.1 Antecedentes</b>	<b>.....</b>		<b>14</b>
<b>1.2. Descripción del proyecto</b>	<b>.....</b>		<b>15</b>
<b>1.3. Objetivos del proyecto</b>	<b>.....</b>		<b>16</b>
<b>1.3.1 Objetivo General</b>	<b>.....</b>		<b>16</b>
<b>1.3.2. Objetivos Específicos</b>	<b>.....</b>		<b>16</b>
<b>1.4. Justificación</b>	<b>.....</b>		<b>16</b>
<b>1.5. Alcance del proyecto</b>	<b>.....</b>		<b>17</b>
<b>1.6. Metodología</b>	<b>.....</b>		<b>18</b>
<b>CAPÍTULO II</b>	<b>.....</b>		<b>20</b>
<b>2.1. Marco Contextual</b>	<b>.....</b>		<b>20</b>
<b>2.2. Marco conceptual</b>	<b>.....</b>		<b>21</b>
<b>2.2.1. Comunicaciones Vehiculares</b>	<b>.....</b>		<b>21</b>
<b>2.2.2. Tecnologías de comunicación</b>	<b>.....</b>		<b>24</b>
<b>2.2.3. Asignación de Recursos</b>	<b>.....</b>		<b>29</b>

2.2.4. Frameworks para C-V2X.....	33
2.2.5 Simuladores de movilidad .....	38
2.2.6 Simuladores de eventos discretos.....	40
2.3 Marco teórico.....	41
<b>CAPÍTULO III.....</b>	<b>44</b>
<b>3.1. Introducción .....</b>	<b>44</b>
<b>3.2. Requisitos del Sistema y Consideraciones Previas.....</b>	<b>44</b>
3.2.1. Especificaciones de Hardware .....	44
3.2.2. Dependencias de Software y Compatibilidad .....	45
3.2.3. Bibliotecas Criptográficas y de Seguridad.....	45
3.2.4. Bibliotecas de Desarrollo y Herramientas Auxiliares.....	46
3.3. Preparación del Entorno de Desarrollo .....	48
3.4 Instalación y Configuración de OMNeT++ .....	50
3.4.1 Obtención e Instalación de OMNeT++.....	50
3.4.2 Configuración del Entorno de Variables .....	51
3.4.3 Proceso de Configuración de OMNeT++ .....	51
3.4.3 Compilación de OMNeT++ .....	52
3.4.5 Verificación de la Instalación de OMNeT++ .....	53
3.5 Instalación y Configuración de SUMO .....	54
3.6 Obtención y Preparación del Código Fuente de OpenCV2X.....	55
3.7 Compilación Integral de Componentes.....	57
3.7.1 Estrategia de Compilación Global.....	57
3.7.2 Configuración Específica de Veins .....	58
3.7.3 Compilación Modular de Componentes .....	59
3.8 Configuración del Sistema de Construcción CMake .....	60
3.8.1 Inicialización del Entorno de Construcción.....	60
3.8.2 Habilitación de Características Específicas .....	61
3.8.3 Construcción Final del Proyecto.....	62
3.9 Verificación Integral de la Instalación .....	63
3.9.1 Ejecución del Escenario de Validación .....	63
3.9.2 Análisis de Resultados Generados .....	64

<b>3.10 Diagnóstico y Resolución de Problemas Comunes</b> .....	65
<b>3.11 Creación de un escenario con SUMO</b> .....	68
<b>CAPÍTULO IV</b> .....	74
<b>4.1. Introducción</b> .....	74
<b>4.2. Escenario de Simulación</b> .....	74
<b>4.3. Análisis de Resultados</b> .....	76
<b>4.3.1. Distribución de Paquetes: Perdidos vs Exitosos por Distancia</b> .....	76
<b>4.3.2. Packet Delivery Ratio bajo Diferentes Condiciones de Densidad</b>	77
<b>4.4. Discusión de Resultados</b> .....	79
<b>4.5. Comparación con Trabajos Relacionados</b> .....	80
<b>4.6. Conclusiones del Capítulo</b> .....	80
<b>BIBLIOGRAFÍA</b> .....	82

## ÍNDICE DE FIGURAS

Figura 1 Red C-V2X.....	22
Figura 2 Casos de Uso de V2X. [15] .....	24
Figura 3. NR numerología y duración temporal [19].....	28
Figura 4. C-V2X modos de transmisión .....	31
Figura 5. Modos de transmisión en las comunicaciones 5G-v2x [21].....	32
Figura 6. Arquitectura de veins.....	37
Figura 7. Arquitectura de CARLA.....	40
Figura 8. Actualización de dependencias.....	48
Figura 9. Instalación de bibliotecas.....	49
Figura 10. Descarga de omnetpp.....	50
Figura 11. Configuración de omnetpp .....	52
Figura 12. Build de omnetpp.....	53
Figura 13. Verificación de omnetpp.....	54
Figura 14. Versión de sumo .....	55
Figura 15. Clonación del repositorio de opencv2x .....	56
Figura 16. Error 127 durante compilación .....	57
Figura 17. Compilación de inet.....	58
Figura 18. Instalación de veins.....	59
Figura 19. Configuración del proyecto principal .....	61
Figura 20. Habilitación de simulte .....	62
Figura 21. Compilación completa del proyecto general .....	63
Figura 22. Simulación en marcha.....	64
Figura 23. Resultados listados.....	65
Figura 24. Error con cryptopp.....	66
Figura 25. Error al ejecutar la simulación .....	67
Figura 26. creación de la red desde Netedit .....	68
Figura 27. Activando herramientas edge .....	69
Figura 28. Diseñando escenario .....	69
Figura 29. Route mode.....	70
Figura 30. Diseño de ruta .....	70
Figura 31. selección de vehículo .....	71
Figura 32. Archivos de sumo .....	71
Figura 33. Proyecto compilado. ....	73
Figura 34 Escenario de simulación en SUMO con tráfico vehicular bidireccional. .....	75
Figura 35 Distribución de paquetes: Perdidos vs Exitosos por Distancia.....	76
Figura 36 Packet Delivery Ratio vs Distancia bajo diferentes condiciones de densidad vehicular.....	77

## ÍNDICE DE TABLAS

Tabla 1. Características de LTE-V2X Y NR-V2X [18].....	27
Tabla 2. Numerología de NR-V2X.....	28
Tabla 3. Configuración General del Sistema .....	72
Tabla 4. Configuración del Canal de Radio .....	72
Tabla 5. Parámetros de Simulación Utilizados .....	75
Tabla 6. Tasa de Éxito de Paquetes por Rango de Distancia.....	77
Tabla 7. Comparación del PDR por Escenario de Densidad .....	78
Tabla 8. Comparación con Trabajos Relacionados.....	80

# GLOSARIO

## Glosario de Siglas

### A-D

- CP - Cyclic Prefix (Prefijo Cíclico)
- C-V2X - Cellular Vehicle-to-Everything (Comunicación Celular Vehículo a Todo)
- D2D - Device-to-Device (Dispositivo a Dispositivo)
- DL - Downlink (Enlace Descendente)
- DMRS - Demodulation Reference Signals (Señales de Referencia de Demodulación)
- DSRC - Dedicated Short Range Communication (Comunicación Dedicada de Corto Alcance)

### E-I

- eNB - Evolved Node B (Estación Base Evolucionada LTE)
- FR1 - Frequency Range 1 (Rango de Frecuencia 1)
- FR2 - Frequency Range 2 (Rango de Frecuencia 2)
- gNB - Next Generation Node B (Estación Base de Nueva Generación 5G)
- IEEE - Institute of Electrical and Electronics Engineers (Instituto de Ingenieros Eléctricos y Electrónicos)

### L-O

- LTE - Long Term Evolution (Evolución a Largo Plazo)
- LTE-V2X - LTE Vehicle-to-Everything (LTE Vehículo a Todo)
- MAC - Medium Access Control (Control de Acceso al Medio)

- MBMS - Multimedia Broadcast Multicast Services (Servicios de Difusión Multimedia Multicast)
- MCS - Modulation and Coding Scheme (Esquema de Modulación y Codificación)
- NR - New Radio (Nueva Radio 5G)
- NSA - Non-Standalone (No Autónomo)
- OFDM - Orthogonal Frequency Division Multiplexing (Multiplexación por División de Frecuencia Ortogonal)

#### P-S

- PHY - Physical Layer (Capa Física)
- PSSCH - Physical Sidelink Shared Channel (Canal Compartido Físico de Enlace Lateral)
- RB - Resource Block (Bloque de Recursos)
- RSU - Road-Side Unit (Unidad de Borde de Carretera)
- SA - Standalone (Autónomo)
- SC-FDMA - Single Carrier Frequency Division Multiple Access (Acceso Múltiple por División de Frecuencia de Portadora Única)
- SCS - Subcarrier Spacing (Espaciado entre Subportadoras)
- SI - Study Item (Elemento de Estudio)
- SL - Sidelink (Enlace Lateral)

#### T-Z

- TB - Transport Block (Bloque de Transporte)
- UE - User Equipment (Equipo de Usuario)
- UL - Uplink (Enlace Ascendente)

- V2I - Vehicle-to-Infrastructure (Vehículo a Infraestructura)
- V2N - Vehicle-to-Network (Vehículo a Red)
- V2P - Vehicle-to-Pedestrian (Vehículo a Peatón)
- V2V - Vehicle-to-Vehicle (Vehículo a Vehículo)
- V2X - Vehicle-to-Everything (Vehículo a Todo)
- VANET - Vehicular Ad Hoc Network (Red Vehicular Ad Hoc)
- WLAN - Wireless Local Area Network (Red de Área Local Inalámbrica)

#### Unidades de Medida

- GHz - Gigahertz (Gigahercio)
- kHz - Kilohertz (Kilohercio)
- MHz - Megahertz (Megahercio)
- ms - Millisecond (Milisegundo)

## RESUMEN

Esta tesis presenta una evaluación integral de las comunicaciones Cellular Vehicle-to-Everything (C-V2X) en modo 4 sidelink utilizando frameworks de simulación especializados. A medida que los vehículos conectados y autónomos continúan evolucionando, los sistemas de comunicación vehicular confiables y eficientes se vuelven críticos para los sistemas de transporte inteligente y aplicaciones de seguridad crítica. Esta investigación aborda las limitaciones de los sistemas de comunicación vehicular tradicionales, particularmente en cuanto a latencia, confiabilidad y adaptabilidad a condiciones ambientales variables.

Para examinar el desempeño de C-V2X bajo distintas configuraciones y condiciones operativas, la investigación utiliza una metodología de simulación con eventos discretos, haciendo uso de SUMO y OMNeT++. Para determinar su efecto sobre el rendimiento del sistema, se analizan de manera sistemática los parámetros esenciales, como la densidad vehicular, los esquemas de modulación y el espaciamiento de subportadoras. La investigación se basa en trabajos anteriores sobre las comunicaciones de pelotones vehiculares y las vulnerabilidades de los protocolos V2X, ampliando el análisis para aplicar un marco integral de simulación con el objetivo de evaluar C-V2X modo 4.

Los resultados evidencian que los métodos de simulación son eficaces para la evaluación de sistemas de comunicación vehicular, ya que ofrecen cifras cuantitativas sobre la latencia, la fiabilidad y la calidad del servicio. Los resultados se contrastan con investigaciones previas y se validan con estándares internacionales 3GPP, lo que ofrece visiones para mejorar la implementación de C-V2X en situaciones del mundo real. Esta labor aporta una guía práctica que permite la instalación del marco OpenCV2X y la generación de escenarios, sentando así un precedente para investigaciones posteriores en comunicaciones vehiculares avanzadas y el paso a redes NR-V2X.

**Palabras clave:** C-V2X, comunicaciones vehiculares, modo 4 sidelink, LTE-V2X, OMNeT++, SUMO, simulación por eventos discretos, sistemas de transporte inteligente

# ABSTRACT

This thesis presents a comprehensive evaluation of Cellular Vehicle-to-Everything (C-V2X) communications in sidelink mode 4 using specialized simulation frameworks. As connected and autonomous vehicles continue to evolve, reliable and efficient vehicular communication systems become critical for intelligent transportation systems and safety-critical applications. This research addresses the limitations of traditional vehicular communication systems, particularly regarding latency, reliability, and adaptability to varying environmental conditions.

To examine C-V2X performance under different configurations and operational conditions, the research employs a discrete-event simulation methodology, utilizing SUMO and OMNeT++. Essential parameters such as vehicular density, modulation schemes, and subcarrier spacing are systematically analyzed to determine their effect on system performance. The research builds upon previous work on vehicular platoon communications and V2X protocol vulnerabilities, extending the analysis to apply a comprehensive simulation framework aimed at evaluating C-V2X mode 4.

The results demonstrate that simulation methods are effective for evaluating vehicular communication systems, as they provide quantitative metrics on latency, reliability, and quality of service. The findings are compared with previous research and validated against 3GPP international standards, offering insights for improving C-V2X implementation in real-world scenarios. This work provides a practical guide that enables the installation of the OpenCV2X framework and scenario generation, thus establishing a precedent for further research in advanced vehicular communications and the transition to NR-V2X networks.

**Keywords:** C-V2X, vehicular communications, Mode 4 sidelink, LTE-V2X, OMNeT++, SUMO, discrete event simulation, intelligent transportation system

# INTRODUCCIÓN

El desarrollo de sistemas de transporte inteligente representa uno de los desafíos más significativos de la ingeniería moderna, impulsado por la necesidad de mejorar la seguridad vial, reducir la congestión vehicular y optimizar el uso de recursos en las redes de transporte. Las comunicaciones vehiculares surgen, en este marco, como un componente esencial que permite la comunicación entre vehículos, infraestructura y otros componentes del ambiente de tránsito, sentando el fundamento para la futura generación de movilidad conectada y autónoma.

Desde sus inicios con Wi-Fi (802.11p) hasta sus actuales soluciones celulares avanzadas, las tecnologías de comunicación Vehicle-to-Everything (V2X) han tenido un desarrollo importante. La tecnología Cellular V2X (C-V2X) aparece como una solución con mucho potencial que saca provecho de la infraestructura celular ya existente y proporciona un mayor soporte, escalabilidad y cobertura para aplicaciones que requieren alta exigencia. Su importancia radica en la habilidad de permitir comunicaciones directas entre vehículos (V2V), con la infraestructura (V2I), con los peatones (V2P) y con la red (V2N).

La puesta en práctica adecuada de C-V2X es especialmente importante, ya que la Organización Mundial de la Salud señala que cada año ocurren más de 1.19 millones de muertes y 50 millones de lesiones en todo el mundo debido a accidentes viales. No obstante, las restricciones actuales de los sistemas de comunicación entre vehículos, sobre todo en lo que respecta a su adaptabilidad y optimización del empleo de recursos en circunstancias cambiantes, obstaculizan su implementación real en aplicaciones críticas de seguridad.

El presente trabajo de investigación se centra en la evaluación sistemática del rendimiento de las comunicaciones C-V2X en modo 4 sidelink mediante simulación por eventos discretos utilizando herramientas especializadas como OMNeT++ y SUMO. Este enfoque permite analizar el impacto de diferentes parámetros del sistema, incluyendo espaciamiento de subportadoras, esquemas de modulación y densidad vehicular, sobre métricas de rendimiento críticas. La investigación contribuye mediante la evaluación comprensiva de configuraciones

que ofrecen rendimiento óptimo validado contra estándares 3GPP, estableciendo una base sólida para futuras investigaciones en comunicaciones vehiculares avanzadas y proporcionando una guía práctica para la implementación del framework OpenCV2X que facilita la reproducibilidad de experimentos en este campo emergente.

# CAPÍTULO I

## 1. Fundamentación

### 1.1 Antecedentes

En la actualidad, el desarrollo de comunicaciones vehiculares eficientes y confiables representa un desafío significativo para la implementación masiva de vehículos conectados y autónomos. Los sistemas de comunicación vehicular tradicionales presentan limitaciones en términos de latencia, confiabilidad y capacidad de adaptación a condiciones variables del entorno, lo que dificulta su aplicación efectiva en aplicaciones críticas de seguridad y la integración en sistemas de transporte inteligente avanzados.

Desde las primeras redes de cable hasta las tecnologías celulares avanzadas actuales, la comunicación V2X (Vehicle-to-Everything) ha experimentado una notable evolución. Las comunicaciones vehiculares se efectuaban, al principio, a través de tecnologías fundamentadas en Wi-Fi, particularmente 802.11p, que sentó las bases para las redes con fines vehiculares. No obstante, con el progreso de la tecnología celular y la demanda de aplicaciones más exigentes, ha aparecido como una opción más poderosa y flexible la tecnología Celular V2X (C-V2X).

Debido a la creciente necesidad de aplicaciones vehiculares avanzadas que exigen comunicación en tiempo real con niveles elevados de fiabilidad, es imprescindible evaluar y mejorar las comunicaciones C-V2X [2]. En este contexto, han surgido tecnologías como Long-Term Evolution V2X (LTE-V2X) y New Radio V2X (NR-V2X) como soluciones alentadoras, siendo NR-V2X la que ofrece mejores prestaciones para comunicaciones vehiculares según investigaciones recientes.

En investigaciones previas, los autores han desarrollado marcos de trabajo comprensivos para el análisis de comunicaciones vehiculares utilizando simuladores especializados como OMNET++. Se han realizado estudios comparativos de diferentes topologías de flujo de información (IFTs) en sistemas de platooning vehicular, incluyendo estrategias Car-to-Server, Multi-Hop y One-

Hop sobre redes 5G. Los resultados demostraron que la estrategia One-Hop, combinada con algoritmos de asignación de recursos como Maximum of Carrier to Interference Ratio (MaxC/I), proporciona el mejor rendimiento en términos de calidad de servicio [2]. Sin embargo, estos estudios se enfocaron principalmente en modelos simplificados que consideraban ya sea arquitecturas de comunicación detalladas con modelos de platoon simplificados, o sistemas de control detallados con modelos de comunicación simplificados [2].

En otros trabajos de investigación, los autores han abordado las vulnerabilidades y limitaciones de diferentes protocolos de comunicación vehicular, incluyendo C-V2X, DSRC, ITS-G5 y 5G [3]. Estas investigaciones han empleado instrumentos de simulación para llevar a cabo análisis comparativos, tomando como referencia criterios tales como el tiempo de respuesta, la sincronización, la latencia y la selección de recursos. Se ha determinado que el tiempo de respuesta es un elemento crucial que define la eficacia y la robustez de un protocolo de comunicación entre vehículos. No obstante, muchos de estos análisis se han restringido a evaluaciones teóricas que no incluyen la aplicación de algoritmos adaptativos capaces de mejorar el desempeño en situaciones reales [4].

La evaluación del desempeño de las comunicaciones vehiculares que utilizan la tecnología C-V2X es el objetivo de este trabajo de investigación. Se examinarán diversas configuraciones y situaciones por medio de simulaciones para entender el funcionamiento del sistema bajo distintas condiciones. Esta evaluación posibilitará reconocer los elementos que afectan la eficacia de la comunicación y sugerir una nueva perspectiva antes de ponerla en práctica en aplicaciones vehiculares reales.

## **1.2. Descripción del proyecto**

El presente proyecto tiene como objetivo evaluar de forma sistemática el rendimiento de las comunicaciones vehiculares celulares C-V2X (Cellular Vehicle-to-Everything) mediante simulaciones. A través de este análisis se buscará comparar diferentes configuraciones y parámetros clave como el espaciamiento de subportadoras, los diferentes modulaciones y las condiciones variables del entorno vehicular con el fin de identificar su impacto en el desempeño del sistema. Para

ello, se utilizarán herramientas de simulación avanzadas que permitan recolectar y analizar métricas de rendimiento, con miras a verificar que las condiciones sean óptimas y estén dentro de los estándares internacionales.

### **1.3. Objetivos del proyecto**

#### **1.3.1 Objetivo General**

- Evaluar el rendimiento de las comunicaciones vehiculares C-V2X en modo 4 mediante herramientas de simulación de redes utilizando frameworks especializados en redes vehiculares.

#### **1.3.2. Objetivos Específicos**

- Conocer el estado del arte de frameworks y herramientas de simulación para redes vehiculares.
- Diseñar e implementar un escenario de simulación C-V2X en modo 4 sidelink.
- Desarrollar la simulación de red vehicular utilizando Opencv2x y Omnet++.
- Evaluar el rendimiento de las comunicaciones C-V2X mediante métricas específicas.

### **1.4. Justificación**

En la actualidad, las comunicaciones vehiculares están transformando la manera en que los vehículos interactúan entre sí y con su entorno, impulsando soluciones innovadoras en el ámbito del transporte inteligente. C-V2X (Cellular Vehicle-to-Everything) es una de estas tecnologías que sobresale por su capacidad para incrementar de manera notable la seguridad en las vías y la eficacia del tráfico, gracias a un intercambio constante y fiable de información entre vehículos e infraestructura [5][6].

Desde un punto de vista social, la puesta en marcha eficaz de C-V2X tiene el potencial de disminuir los accidentes e, inevitablemente, mejorar la calidad de vida de las personas que lo utilizan. Una comunicación más precisa y de baja latencia facilita la detección temprana de riesgos y situaciones peligrosas en carretera. Este aspecto es especialmente relevante si se considera que, según la Organización

Mundial de la Salud, los accidentes de tránsito provocan más de 1.19 millones de muertes y 50 millones de lesiones al año [7]. Asimismo, las comunicaciones vehiculares confiables permiten avanzar hacia sistemas de conducción cooperativa y autónoma, que dependen de una infraestructura de comunicación robusta para su funcionamiento seguro.

La justificación de este trabajo radica en las limitaciones actuales de los sistemas de comunicación vehicular, particularmente en cuanto a la adaptabilidad y optimización del uso de recursos bajo condiciones variables. Muchos esquemas existentes de asignación de recursos no consideran adecuadamente los distintos requerimientos de los vehículos según su contexto operativo, asumiendo demandas uniformes que afectan negativamente el rendimiento global [8][9][10].

En este contexto, se propone el desarrollo de un sistema de evaluación comprensiva de las comunicaciones C-V2X, basado en simulaciones, que permita analizar y comparar distintas configuraciones con el fin de identificar aquellas que ofrecen un rendimiento óptimo. Esta evaluación no solo permitirá mejorar la eficiencia y confiabilidad de las comunicaciones vehiculares, sino que también servirá como base para impulsar nuevas investigaciones en el área de las comunicaciones avanzadas para transporte. La relevancia del estudio también se sustenta en la tendencia global hacia la adopción de C-V2X sobre otros estándares como DSRC, y en el creciente interés por la evolución hacia NR-V2X como parte de las redes 5G, debido a su mayor capacidad y rendimiento [11].

### **1.5. Alcance del proyecto**

El alcance de este proyecto de investigación cubre las siguientes áreas:

- Se analizarán los frameworks y software de simulación existentes.
- Se obtendrán métricas de evaluación de un escenario sidelink modo 4 LTE V2X
- Se realizará guía de como instalar Opencv2x, como crear un escenario con sumo y como vincularlo a nuestro framework para su uso.
- Se comparará los resultados obtenidos con otros proyectos similares y se registrará por el estándar 3gpp.

## 1.6. Metodología

**Método práctico proyectual:** Se trata de una metodología flexible y adaptable que tiene como objetivo gestionar el proceso de evaluación y análisis de sistemas de comunicación vehículo-a-todo (CV2X) de manera eficaz y eficiente. Este procedimiento es un proceso iterativo que fusiona conceptos teóricos de comunicaciones inalámbricas con experimentación práctica para desarrollar proyectos de evaluación de tecnologías CV2X.

Esta metodología se enfoca en la elaboración, planificación y ejecución de protocolos de evaluación específicos para comunicaciones vehiculares. Para ello, sigue un ciclo iterativo que incluye el diseño experimental, la implementación de pruebas y la revisión de resultados. De esta manera, garantiza que la evaluación progrese y mejore constantemente a partir de los datos obtenidos sobre el rendimiento y la retroalimentación recibida [9].

En el proyecto de investigación CV2X, este procedimiento permitirá la creación de los protocolos de prueba para examinar parámetros de comunicación y del marco inicial de evaluación. Asimismo, se llevarán a cabo evaluaciones en contextos reales y controlados para determinar la latencia, fiabilidad, cobertura de transmisión y calidad del servicio de las comunicaciones CV2X. Esto hará posible que los métodos de evaluación sean calibrados y validados con estándares reconocidos en el sector de las comunicaciones vehiculares.

**Método de simulación por eventos discretos:** La simulación de eventos discretos se utiliza como un método para modelar con el propósito de reflejar sistemas dinámicos en los que las transformaciones de estado suceden en momentos determinados. Este método adopta un enfoque computacional que posibilita modelar la complejidad tanto temporal como espacial de las comunicaciones vehiculares, lo que hace posible una evaluación exhaustiva y contextualizada del comportamiento del sistema en condiciones cambiantes. [6].

En el artículo [9] se describe la teoría de simulación por eventos discretos aplicada a redes de comunicación vehicular, la cual está basada en la modelización temporal de eventos de comunicación desde la perspectiva del sistema completo. En donde

se busca comprender el comportamiento dinámico de las comunicaciones vehiculares, más allá de análisis estáticos o simplificados.

En el contexto del trabajo de investigación, este método servirá para el modelado y simulación de escenarios vehiculares realistas utilizando herramientas como OMNET++ y Sumo. También permitirá hacer una evaluación temporal de las comunicaciones, las cuales tendrán un contexto dinámico de movilidad vehicular y condiciones de canal variables, permitiendo comprender como los diferentes factores temporales y espaciales influyen en el rendimiento de las comunicaciones C-V2X. Este enfoque permitirá crear modelos de simulación que no solo sean precisos, sino también representativos y relevantes para condiciones reales de operación.

## CAPÍTULO II

### 2.1. Marco Contextual

El transporte es uno de los aspectos más importantes en la vida de las personas y el desarrollo de las sociedades, ya que afecta directamente su movilidad, seguridad y calidad de vida. La capacidad de comunicar y gestionar el tráfico vehicular de manera efectiva es crucial para prevenir accidentes, mejorar la eficiencia del transporte y reducir los costos sociales de la congestión. Las comunicaciones vehiculares, como las tecnologías V2V (Vehicle-to-Vehicle), V2I (Vehicle-to-Infrastructure) y V2X (Vehicle-to-Everything), son elementos clave del funcionamiento seguro y eficiente del transporte moderno. Estos sistemas pueden proporcionar información valiosa sobre las condiciones del tráfico, el estado de los vehículos y alertar sobre posibles situaciones peligrosas.

Según el estudio [6] se determina que la capacidad de transmitir información crítica de manera precisa y en tiempo real puede mejorar significativamente la gestión del tráfico vehicular, permitir la detección temprana de situaciones de riesgo y reducir la carga sobre los sistemas de transporte mediante la conducción cooperativa y autónoma.

La puesta en marcha de comunicaciones C-V2X optimizadas se enfocará en los sistemas de transporte inteligentes que afrontan problemas de eficiencia energética, congestión del tráfico y seguridad vial. El uso de estas tecnologías posibilitará una comunicación ininterrumpida y segura entre vehículos e infraestructura, la cual podrá unirse con centros de gestión del tráfico con el objetivo de coordinarse adecuadamente a partir de datos en tiempo real. Esta tecnología supondrá un cambio radical en el ámbito del transporte inteligente, puesto que hará más sencillo poner en marcha sistemas de conducción cooperativa y vehículos autónomos, lo cual mejorará considerablemente la seguridad en las vías y la eficacia del transporte.

El propósito del proyecto es mejorar las comunicaciones de los vehículos conectados y de los sistemas de transporte inteligente en autopistas y áreas urbanas. Las tecnologías C-V2X analizadas cambiarán la forma en que se realiza la comunicación entre vehículos, proporcionando a los usuarios del transporte y a los

operadores de sistemas viales una solución eficaz, confiable y eficiente. Se prevé que el sistema NR de quinta generación (5G) sobrepase a LTE C-V2X en cuanto a la obtención de un rendimiento más alto en situaciones con un throughput elevado, latencia baja y confiabilidad aumentada [12]. Por lo tanto, estas tecnologías se ubican como claves para el porvenir del transporte inteligente y la movilidad autónoma.

## **2.2. Marco conceptual**

El marco conceptual de este proyecto abarca diversos temas fundamentales para la evaluación de las comunicaciones vehiculares C-V2X. Se explorarán las tecnologías de comunicación vehicular, así como sus parámetros de rendimiento y como estos varían según las condiciones del entorno y configuraciones del sistema. Además, abordaremos las diferentes arquitecturas que se presentan en la implementación de C-V2X en sistemas de transporte inteligente. Finalmente, se analiza los diferentes tipos de métricas de evaluación que las comunicaciones vehiculares ofrecen para determinar cuáles son las más viables para la implementación real del sistema. Este marco conceptual proporciona una base teórica y técnica esencial para el desarrollo del proyecto de investigación en comunicaciones vehiculares avanzadas.

### **2.2.1. Comunicaciones Vehiculares**

La comunicación vehicular consiste en el intercambio de información que se establece entre diferentes vehículos. La finalidad primordial de implementar VANET es disminuir la cantidad de accidentes de tránsito. Esta tecnología tiene un impacto significativo tanto en la protección de los pasajeros como en facilitar una conducción fluida para los conductores en zonas urbanas. Dado que el número de vehículos crece continuamente cada día, también se incrementa la frecuencia de accidentes, lo que hace indispensable establecer sistemas de comunicación entre vehículos [12].

Como respuesta a este problema, se ha sugerido la implementación de Redes Ad-hoc Vehiculares (VANET), una solución tecnológica que posibilita que los vehículos intercambien información de manera eficaz. Estas redes se han adaptado

específicamente para soportar la naturaleza muy dinámica de los vehículos como nodos de red, satisfaciendo así sus necesidades de movilidad. Sin embargo, debido a la variación constante en la topología de la red, tienen limitaciones en términos de cobertura efectiva.

A la par, los productores de equipos originales (OEM) de automóviles han creado un nuevo paradigma revolucionario llamado "Conducción Autónoma", que consiste en vehículos diseñados para funcionar de manera totalmente autónoma, sin ninguna intervención humana. Es esencial equipar estos sistemas autónomos con conciencia situacional del ambiente para que operen de forma eficaz y segura. Esto se consigue al integrar las habilidades de comunicación vehicular con sensores avanzados y algoritmos de procesamiento de datos en tiempo real[13].

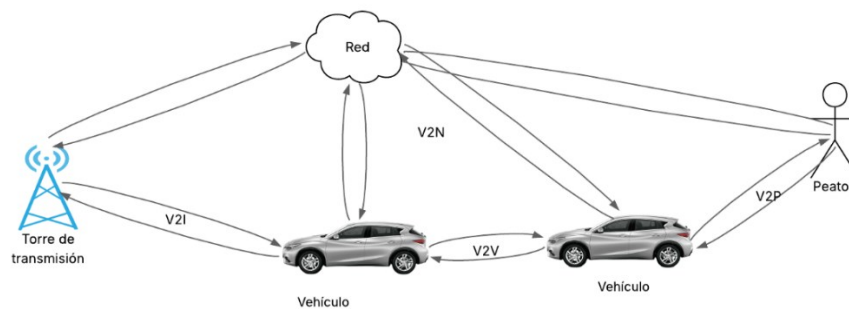


Figura 1. Red C-V2X.

En la figura 1 podemos ver de manera ilustrativa como se conectan los tipos de comunicaciones en una red C-V2X de una manera superficial pero efectiva para entender la base de esta tecnología.

#### ***2.2.1.1. Tipos de comunicaciones vehiculares***

Por las características de los mensajes que se transmiten, principalmente relacionados con la seguridad, es fundamental contar con comunicaciones de respuesta rápida y gran confiabilidad. Estas exigencias representan retos significativos para las infraestructuras de red actuales. No obstante, el progreso tecnológico nos está permitiendo avanzar hacia el cumplimiento y la superación de estas dificultades.

Las redes vehiculares ad hoc (VANETs) funcionan con infraestructura permanente mínima o inexistente, y se distinguen por las siguientes características:

1. Elevada movilidad de los nodos
2. Topología de red basada en vías de tránsito establecidas
3. Velocidades y comportamientos de tráfico que pueden anticiparse
4. Restricciones mínimas en términos de energía y capacidad de almacenamiento

El modelo de comunicación V2X implementado en las VANETs facilita diversos escenarios de intercambio de información:

Comunicación vehículo a vehículo (V2V): Se establece el intercambio de información entre automóviles cercanos. Puede realizarse de forma directa entre dos vehículos (unicast) o desde un vehículo hacia múltiples receptores (multicast/broadcast), modalidad frecuentemente utilizada para compartir datos de posicionamiento, alertas de tráfico y otras advertencias viales.

Comunicación vehículo a infraestructura (V2I): Consiste en el intercambio de datos entre un automóvil y elementos de infraestructura vial, como las Unidades de Borde de Carretera (RSUs) establecidas en el estándar DSRC.

Comunicación vehículo a red (V2N): Permite el intercambio directo de mensajes entre el vehículo y la red de telecomunicaciones celular.

Comunicación vehículo a peatón (V2P): Facilita la comunicación directa entre un automóvil y uno o varios peatones que se encuentren en las proximidades. Esta interacción puede realizarse de manera directa o mediante el uso de infraestructura de red como intermediario.

La Figura 2 ilustra cómo funcionan estos escenarios en un caso real.



Figura 2. Casos de Uso de V2X. [14]

### 2.2.2. Tecnologías de comunicación

Tras las investigaciones realizadas sobre comunicaciones de corto alcance entre vehículos durante las décadas de 1980 y 1990, la asignación de un ancho de banda específico en Estados Unidos en 1999 marcó sin duda el primer hito importante. Se destinaron siete canales de 10 MHz cada uno en la frecuencia de aproximadamente 5.9 GHz para lo que se denominó comunicación dedicada de corto alcance (DSRC).

#### 2.2.2.1. DSRC

DSRC fue la primera tecnología V2X incorporada en el estándar IEEE 802.11p del Instituto de Ingenieros Eléctricos y Electrónicos. Esta tecnología utiliza sistemas WLAN para establecer canales de comunicación dedicados de corto alcance, permitiendo que los vehículos se conecten directamente con otras entidades a distancias cortas y medias.

Dado que el propósito inicial de DSRC era transmitir mensajes básicos de seguridad entre vehículos en un rango limitado (aproximadamente 300 metros), no fue diseñado para gestionar las altas demandas de ancho de banda que requieren las aplicaciones V2N. Además, debido a las características de su diseño original,

presenta limitaciones para evolucionar y adaptarse a los nuevos requerimientos más complejos que están siendo considerados en la actualidad [4].

#### **2.2.2.2. LTE-V2X**

La versión Release 14 estableció el LTE-V2X de corto alcance basándose en la comunicación dispositivo a dispositivo (D2D) del Release 12. También se conoce como LTE-V2X sidelink (en contraposición a downlink/uplink) y utiliza la interfaz de comunicación PC5.

En las capas físicas (PHY) y de control de acceso al medio (MAC), el LTE-V2X sidelink emplea el mismo acceso múltiple por división de frecuencia de portadora única (SC-FDMA) que el enlace ascendente LTE:

En el dominio de frecuencia: el espaciado entre subportadoras se fija en 15 kHz, y estas se agrupan en conjuntos de 12 (equivalente a 180 kHz).

En el dominio temporal: 14 símbolos conforman una subtrama de 1 ms, también denominada intervalo de tiempo de transmisión. Los esquemas de modulación y codificación (MCS) son diversos en LTE-V2X.

El V2X celular [15] utiliza conectividad móvil celular estándar 3GPP 4G LTE o 5G para transmitir y recibir señales desde un vehículo hacia otros vehículos, peatones u objetos fijos del entorno, como semáforos. Comúnmente opera en el espectro de frecuencia de 5.9 GHz.

Cada subtrama emplea 14 símbolos OFDM con prefijo cíclico normal. Típicamente, nueve de estos símbolos se destinan a la transmisión de datos, mientras que cuatro se utilizan para enviar señales de referencia de demodulación (DMRS) para la estimación del canal y la minimización del efecto Doppler a altas velocidades. El símbolo restante funciona como símbolo de guarda, permitiendo que los vehículos alteren entre transmisión y recepción a través de las subtramas y realicen ajustes de temporización.

#### **2.2.2.3 NR-V2X**

El C-V2X del Release 14 (es decir, LTE-V2X) constituirá la base fundamental para las comunicaciones básicas de seguridad, mientras que las futuras versiones de 5G

NR incorporarán mejoras para dar soporte a servicios avanzados [16]. A diferencia de Wi-Fi, este enfoque no mantiene compatibilidad hacia atrás, sino que añade una segunda interfaz con mayor rendimiento en diferentes canales.

El enlace lateral NR V2X puede operar en frecuencias dentro de dos rangos específicos:

- Rango de frecuencia 1 (FR1): 410 MHz – 7.125 GHz
- Rango de frecuencia 2 (FR2): 24.25 GHz – 52.6 GHz

Gracias a la multiplexación por división de frecuencia ortogonal (OFDM) multiportadora habilitada por la adaptabilidad de la numerología NR, el 5G-V2X mejora el LTE-V2X incorporando las capacidades de la capa física 5G NR y permite aplicaciones V2X sofisticadas con diversos tipos de requisitos.

Los modos de operación de comunicación V2X compatibles con la arquitectura del sistema 5G incluyen la comunicación V2X sobre el punto de referencia o interfaz PC5 y la comunicación V2X sobre el punto de referencia o interfaz Uu. Para NR y LTE, la interfaz PC5 posibilita las comunicaciones SL V2X. Bajo implementaciones NR No Autónomas (NSA) y Autónomas (SA), son posibles las comunicaciones V2X sobre Uu para transmisiones UL y DL. En el Release 16, solo se admiten comunicaciones unicast en la comunicación V2X sobre Uu. Sin embargo, el Release 17 incluye un estudio en curso que analiza las mejoras de transmisión multicast y broadcast de 5G. Los mensajes V2X enviados a través de LTE-Uu también pueden transmitirse en DL utilizando Servicios de Difusión Multimedia Multicast (MBMS).

Una de las metas de este elemento de trabajo del Release 17 es mejorar el modo 2 de asignación de recursos del NR V2X del Release 16. Algunas de las mejoras propuestas se centran en reducir el consumo de energía y mejorar indicadores clave de rendimiento como la confiabilidad y la latencia. Con el NR V2X del Release 16, todas las actualizaciones deben poder compartir el mismo conjunto de recursos (coexistencia en el mismo canal). En el NR V2X del Release 16, se desarrollaron

el Modo 1 y Modo 2 para equipos de usuario con requisitos de baja potencia, como vehículos o RSUs.

En la tabla 1 podemos ver la comparación de las características de LTE-V2X y NR-V2X.

*Tabla 1. Características de LTE-V2X Y NR-V2X [17]*

<b>CARACTERÍSTICAS</b>	<b>LTE-V2X</b>	<b>NR-V2X</b>
Sub-carrier spacing	15 kHz	15, 30, 60, 120 kHz
Communications modes	Broadcast	Unicast, Groupcast, Broadcast
MCS	QPSK, 16QAM	QPSK, 16QAM, 64QAM
Waveform	SC-FDMA	OFDM
PSCCH and PSSCH	FDM	TDM
Feedback Channel	No	PSFCH
DMRS / sub-frame	4	Flexible
Sidelink modes	3, 4	1, 2

#### **2.2.2.4. Numerología**

Las transmisiones NR V2X SL utilizan la forma de onda de multiplexación por división de frecuencia ortogonal (OFDM) con prefijo cíclico. La estructura de trama del enlace lateral se organiza en tramas de radio (también denominadas marcos) con una duración de 10 milisegundos cada una. Cada trama de radio se subdivide en diez subtramas de un milisegundo de duración.

En NR V2X, tanto la cantidad de ranuras por subtrama como el espaciado entre subportadoras (SCS) para la forma de onda OFDM pueden ajustarse. Para NR V2X se está estudiando una numerología OFDM escalable basada en el NR Uu del Release 15, con el fin de satisfacer diversos requisitos y frecuencias de operación en FR1 y FR2. Cada numerología OFDM se define mediante un SCS y un Prefijo Cíclico (CP).

NR V2X admite múltiplos de 15 kHz (es decir, el SCS utilizado en LTE V2X) para la transmisión de datos.

Numerologías de NR-V2X:

Tabla 2. Numerología de NR-V2X

$\mu$	SCS	Rango de frecuencia	Símbolos por ranura	Duración de ranura
0	15 kHz	FR1	14	1 ms
1	30 kHz	FR1	14	0.5 ms
2	60 kHz	FR1, FR2	14	0.25 ms

Mediante un SCS escalable de  $2\mu \times 15$  kHz, donde  $\mu$  representa el factor de configuración del SCS, se pueden lograr diferentes numerologías OFDM. El factor de configuración del SCS para NR V2X puede establecerse en 0, 1, 2 o 3, resultando en valores de SCS de 15 kHz, 30 kHz, 60 kHz o 120 kHz respectivamente. En FR1 se admiten SCS de 15 kHz, 30 kHz y 60 kHz, mientras que en FR2 se soportan 60 kHz y 120 kHz.

El soporte de un SCS más elevado mejora la robustez de la forma de onda OFDM frente a deterioros de frecuencia ocasionados por efectos Doppler, desviaciones de frecuencia portadora y ruido de fase del hardware, que son más significativos en FR2.

En la figura 3 podemos observar la estructura de un subframe de manera más detallada.

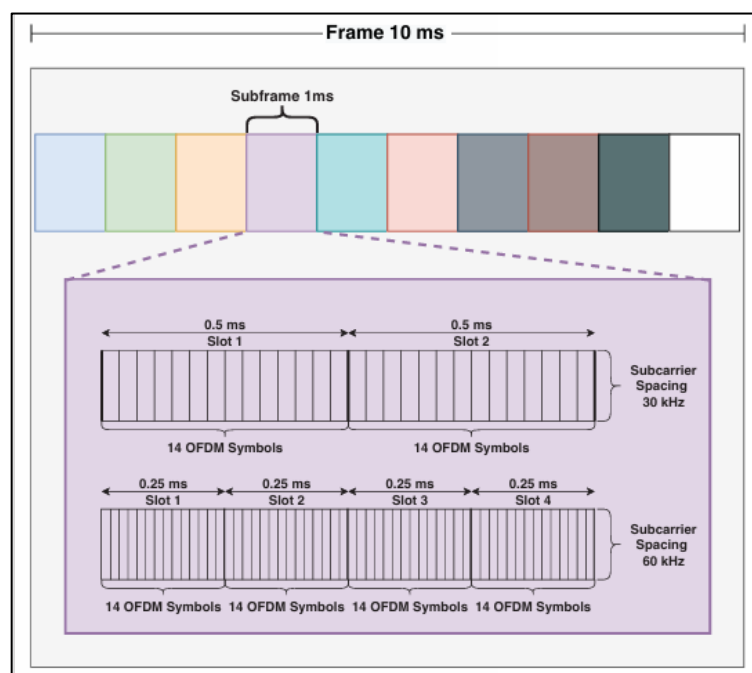


Figura 3. NR numerología y duración temporal [18]

### **2.2.3. Asignación de Recursos**

La distribución de recursos comúnmente se vincula con un mecanismo de programación semipersistente (SPS), considerando que LTE-V2X fue diseñado fundamentalmente para facilitar el servicio de conciencia cooperativa, el cual implica la transmisión periódica de mensajes por parte de cada vehículo con el fin de informar acerca de su estado y desplazamientos.

#### **2.2.3.1. Sidelink**

El Sidelink es una tecnología celular que permite la comunicación directa entre dispositivos, ya sea con o sin la asistencia de estaciones base celulares. Desde la versión 12, el 3GPP ha estado desarrollando esta tecnología [17]. Se han diseñado canales específicos de capa física para facilitar el descubrimiento de dispositivos, la sincronización y la transmisión sidelink. Estos incluyen el Canal Compartido Físico Sidelink (PSSCH), el Canal de Control Físico Sidelink (PSCCH), el Canal de Difusión Físico Sidelink (PSBCH) y la Información de Control Sidelink (SCI) [18].

Los canales sidelink se organizan en una estructura de rejilla de recursos tiempo-frecuencia, compuesta por bloques de recursos (RB) contiguos en el dominio de frecuencia y subtramas de 1 ms en el dominio temporal. Un conjunto de RBs dentro de la misma subtrama se denomina subcanal (mostrado en la Figura 3). La cantidad de RBs en cada subcanal puede variar según la configuración. La programación de recursos tiempo-frecuencia de los RBs consiste en secuencias recurrentes de hipertramas llamadas Asignación de Programación (SA) o período PSCCH. El período SA se utiliza para comunicación directa en ambos Modos sidelink 1 y 2 [17,18].

Cada período SA controla los PSCCH, que son transportados por PSSCH. La SCI transportada por PSCCH contiene información sobre el esquema de modulación y codificación, los RBs utilizados y el período de reserva de recursos. Un paquete destinado para transmisión PSSCH se contiene en un Bloque de Transporte (TB).

En los Modos 3 y 4, la transmisión TB puede ocupar RBs adyacentes o no adyacentes en la misma subtrama, mientras que la SA se difunde utilizando RBs

específicos en el subcanal. Pueden surgir conflictos de recursos cuando múltiples dispositivos seleccionan ocasionalmente los mismos recursos del pool común [18,19]. Por tanto, estas selecciones se coordinan mediante técnicas apropiadas de resolución de colisiones.

Cada nodo puede seleccionar los recursos tiempo-frecuencia a emplear en el algoritmo de programación semi-persistente actualmente en uso. Los nodos competidores deben utilizar un mecanismo basado en detección para encontrar los recursos tiempo-frecuencia menos utilizados, función que el algoritmo no proporciona [20,21].

La detección de dispositivos y la asignación de recursos, dos características fundamentales de la tecnología sidelink, han sido objeto de extensos requisitos regulatorios e investigación académica [22,23]. El descubrimiento se refiere a la capacidad de usar la interfaz de radio sidelink (PC5) para localizar otro dispositivo cercano. Esto puede realizarse directamente por los Equipos de Usuario (UE) o a nivel de red central. El concepto de proximidad puede expandirse para incluir factores como calidad de señal, retardo, rendimiento, carga de red y calidad del canal de radio experimentada por los pares de UEs comunicándose [7,24].

#### ***2.2.3.2. Modo 3 y Modo 4 de C-V2X***

El estándar 3GPP establece dos enfoques para la asignación de recursos: Modo 3 y Modo 4, que se diferencian según quién controla dicha asignación. En el Modo 3, la infraestructura celular (eNB) supervisa las comunicaciones V2X del enlace lateral, incluyendo la determinación y configuración de los recursos de comunicación (subcanales). Por el contrario, el Modo 4 opera de manera independiente sin requerir infraestructura celular, permitiendo que los vehículos seleccionen, gestionen y configuren los subcanales de forma autónoma.

Existen dos esquemas de programación definidos: programación dinámica y Programación Semi-Persistente (SPS). Con la programación dinámica, los vehículos deben solicitar subcanales al eNB para cada bloque de transporte (TB). En la programación SPS, el eNB reserva subcanales de antemano para que un vehículo pueda transmitir múltiples TBs. Dado que la programación de

transmisiones se centraliza en el eNB, el Modo 3 puede superar al Modo 4 en términos de rendimiento. Sin embargo, esto conlleva la necesidad de cobertura de red y genera complicaciones adicionales.

En la Figura 4 se puede ver de mejor manera el funcionamiento del modo 3 y modo 4 de C-V2X



*Figura 4. C-V2X modos de transmisión*

### **2.2.3.3. Modo 1 de NR-V2X**

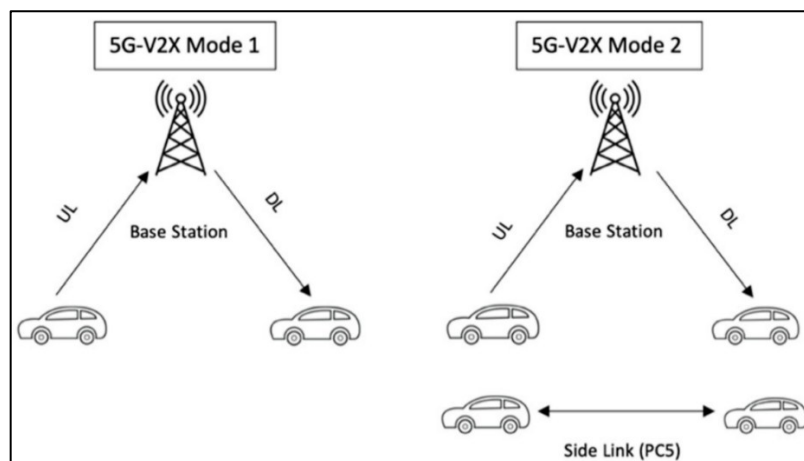
La modalidad 1 en NR V2X es un sistema de comunicación de vehículos que se distingue por el control centralizado, desde la estación base (gNB o eNB), sobre la gestión y asignación de recursos de radio para las comunicaciones entre vehículos (V2V). En este modo, los equipos de usuario (UE) necesitan estar dentro del alcance de la red para funcionar, porque la estación base es la encargada de gestionar totalmente la repartición de recursos del espectro radioeléctrico por medio de la interfaz Uu de LTE o NR. El Modo 1 establece dos sistemas de asignación de recursos fundamentales: la asignación dinámica (Dynamic Grant - DG), En la asignación configurada (Configured Grant - CG), la estación base preasigna conjuntos de recursos periódicos según los datos de asistencia del UE sobre las propiedades del tráfico anticipado, como periodicidad, tamaño máximo de bloques de transporte y necesidades de calidad de servicio (QoS). Esto disminuye la latencia para aplicaciones con patrones de transmisión previsibles. En la otra modalidad, cada vehículo tiene que pedir a la estación base recursos específicos para cada transmisión mediante peticiones programáticas (Scheduling Request), lo que asegura un uso eficaz de recursos para tráfico irregular, pero aumenta la latencia.

Una característica fundamental del Modo 1 es su capacidad de coordinación con otros modos de operación, donde los vehículos informan a otros equipos sobre los recursos que utilizarán mediante la señalización de control de primera etapa (1st-stage SCI), evitando colisiones en el espectro y permitiendo una coexistencia eficiente entre diferentes esquemas de comunicación vehicular [19].

#### 2.2.3.4. Modo 2 de NR-V2X

Al igual que el Modo 4, en el Modo 2 los equipos de usuario eligen de forma independiente los recursos del enlace lateral para sus transmisiones sin asistencia de la red. El sistema de programación y asignación dinámica para tráfico no periódico es introducido específicamente por 5G-V2X como complemento al esquema SB-SPS (Semi-Persistent Scheduling basado en sensado). En el método dinámico, el recurso seleccionado solamente se requiere para una única transmisión, y pueden asignarse recursos exclusivamente para sus propias retransmisiones. Por consiguiente, utilizando el método dinámico, es necesario elegir nuevos recursos para cada transferencia de datos [13].

En la *Figura 5* podemos ver cómo funcionan los modos 1 y 2 y hacer más notoria tanto la similitud como la diferencia con los modos 3 y 4 de LTE-V2X haciendo énfasis en que el modo 2 puede trabajar con o sin ayuda de la infraestructura.



*Figura 5. Modos de transmisión en las comunicaciones 5G-v2x [20]*

#### **2.2.4. Frameworks para C-V2X**

Los frameworks de C-V2X proporcionan la arquitectura tecnológica fundamental que define cómo los vehículos pueden comunicarse de manera confiable y en tiempo real con su entorno. Estos marcos de trabajo abarcan desde las especificaciones de las capas físicas de comunicación hasta los protocolos de aplicación de alto nivel, estableciendo los estándares necesarios para garantizar la interoperabilidad, seguridad y rendimiento de los sistemas vehiculares conectados.

##### **2.2.4.1. *Opencv2x***

OpenCV2X es una implementación de código abierto del estándar 3GPP CV2X (Release 14) Modo 4. Está basado en una versión extendida del simulador SimuLTE de OMNeT++ que habilita simulaciones de redes LTE. Están disponibles dos variantes: la primera se integra con el framework Artery para proporcionar estandarización completa ITS-G5 a través de todo el stack de comunicación, y la segunda se integra únicamente con Veins [21].

[21] presentan OpenCV2X como el primer modelo de simulación de código abierto disponible públicamente del sidelink Cellular Vehicle to Everything (C-V2X) 3GPP Release 14, que forma la base para el Modo 2 de 5G NR bajo versiones posteriores. Este modelo es completamente compatible con las capas de servicio y aplicación vehicular existentes, incluyendo conjuntos de mensajes definidos por las comunidades automotriz y de estándares, proporcionando un modelo de comunicación cross-layer completamente estandarizado [22].

En su trabajo seminal, los autores presentan el primer modelo de simulación de código abierto disponible públicamente del estándar 3GPP Release 14 Cellular Vehicle to Everything (C-V2X). El modelo aborda específicamente el Modo 4, en el cual los vehículos se comunican directamente en modo peer-to-peer utilizando un algoritmo de asignación de recursos distribuido [23]. Importantemente, el modelo es completamente compatible con las capas de servicio y aplicación existentes ya definidas por las comunidades automotriz y de estándares, proporcionando un modelo de comunicación cross-layer completamente

estandarizado que abarca desde la capa de aplicación basada en el estándar ITS-G5 hasta la capa física del nuevo estándar C-V2X Modo 4 [23].

El modelo de simulación ha sido implementado extendiendo SimuLTE, un stack de comunicación celular de OMNeT++ que habilita la simulación de redes LTE, así como extendiendo e integrando Artery, una implementación ITS-G5 compatible con ETSI [23]. OpenCV2X habilita la simulación de escenarios V2X basados en el estándar 3GPP CV2X Modo 4 a través de una extensión del framework SimuLTE. También permite la simulación con las capas de aplicación del estándar ETSI ITS-G5 mediante una integración con Artery que implementa estos estándares para OMNeT++ [21].

Una limitación clave identificada es que OpenCV2X implementó el estándar C-V2X en SimuLTE, que está construido sobre OMNeT++ para soportar comunicación LTE, habiendo extendido Veins para soportar modelado C-V2X [7]. Utilizando este modelo, se demuestra cómo el mecanismo actual de programación sidelink tiene un rendimiento deficiente al programar aplicaciones con características de comunicación altamente aperiódicas, como los Mensajes de Conciencia Cooperativa (CAMs) de ETSI [22].

#### **2.2.4.2. Simu5G**

Simu5G es la evolución del popular simulador de redes 4G SimuLTE que incorpora acceso 5G New Radio. Basado en el framework OMNeT++, está escrito en C++ y es completamente personalizable con una interfaz pluggable simple. También se pueden desarrollar nuevos módulos implementando nuevos algoritmos y protocolos [24]. Simu5G es un simulador de código abierto bajo licencia LGPL, basado en el framework OMNeT++ y representa un modelo de simulación de plano de usuario 5G NR y LTE/LTE-A para OMNeT++ & INET [25].

La idea detrás de Simu5G es permitir a los investigadores simular y evaluar sus soluciones en un framework fácil de usar. Se toma prestado el concepto de modularidad de OMNeT++, por lo que es fácil de extender. Además, puede integrarse con otros módulos del Framework INET. Ofrece soporte para

herramientas de optimización (por ejemplo, solucionadores de optimización como CPLEX) [26].

[27] proporcionan una descripción detallada de Simu5G en "Simu5G – An OMNeT++ library for end-to-end performance evaluation of 5G networks" publicado en IEEE Access, que recibió el Best Paper Award en SIMULTECH 2020. Simu5G permite a los usuarios simular el plano de datos de despliegues 5G New Radio, en una perspectiva end-to-end e incluyendo todas las capas de protocolo, convirtiéndolo en una herramienta valiosa para investigadores y profesionales interesados en la evaluación de rendimiento de redes y servicios 5G [28].

Simu5G es compatible con SimuLTE y permite simular escenarios de red donde 4G y 5G coexisten, tanto en despliegues StandAlone (SA) como E-UTRA/NR Dual Connectivity (ENDC). Además, hereda la compatibilidad de SimuLTE con otras librerías basadas en OMNeT++, por ejemplo, Veins para movilidad vehicular [26].

[29] describieron cómo usar Simu5G como un emulador en tiempo real en "Scalable Real-Time Emulation of 5G Networks With Simu5G" publicado en IEEE Access, demostrando que redes con cientos de usuarios simulados y decenas de celdas pueden emularse en una sola máquina de escritorio. Otras funcionalidades incluyen capacidades de emulación de red en tiempo real y soporte para aplicaciones Multi-access Edge Computing (MEC) [30].

#### **2.2.4.3. WILABv2Xsim**

WiLabV2Xsim es un simulador dinámico desarrollado en MATLAB que moderniza LTEV2Vsim para soportar sidelink 5G-V2X. Está diseñado para la investigación de asignación de recursos en redes basadas en sidelink C-V2X, con enfoque en el servicio de conciencia cooperativa, aunque también permite simular IEEE 802.11p/ITS-G5 [31]. El simulador es distribuido bajo licencia GNU GPLv3 y ha sido desarrollado por la Universidad de Bologna, CNR y WiLab/CNIT en Italia [31].

Este framework se ha utilizado extensivamente en investigación académica. Estudios recientes incluyen trabajos sobre "Improving Resource Allocation for beyond 5G V2X Sidelink Connectivity" presentado en ASILOMAR Conference on

Signals, Systems and Computers 2021 [32], y "On Wireless Blind Spots in C-V2X Sidelink" publicado en IEEE Transactions on Vehicular Technology [33]. La investigación más reciente aborda NOMA (Non-Orthogonal Multiple Access) en el modo autónomo de C-V2X sidelink, como se documenta en "Towards 6G-V2X Sidelink: Non-Orthogonal Multiple Access in the Autonomous Mode" publicado en IEEE Vehicular Technology Magazine [34].

#### **2.2.4.3. 5G-LENA**

Es un simulador de red 5G New Radio (NR) de código abierto, diseñado como un módulo conectable para ns-3. Su desarrollo, inicialmente financiado por InterDigital, está abierto a la comunidad bajo licencia GPLv2 para fomentar la adopción temprana, contribuciones de socios industriales y académicos, desarrollo colaborativo y reproducibilidad de resultados [35].

LENA y 5G-LENA combinan un canal simulado y modelo de capa física con una implementación completa del stack de protocolos LTE/NR RAN y EPC que sigue de cerca las especificaciones 3GPP. Los módulos han sido desarrollados en CTTC en estrecha colaboración con la industria [35]. El simulador ns-3 5G-LENA es un simulador de nivel de sistema NR end-to-end de código abierto que permite realizar investigación extensiva [36].

La calibración del simulador es fundamental para su validez académica. [36] realizaron la calibración del simulador ns-3 5G-LENA según los resultados de referencia 3GPP para despliegues NR en exteriores, validando que los resultados obtenidos sean comparables con simuladores industriales y redes reales [37]. Investigaciones recientes incluyen trabajos sobre QoS para tráfico XR en 5G NR y modelos MIMO de usuario único compatibles con 3GPP para simulaciones de redes móviles de alta fidelidad

#### **2.2.4.5. Veins**

Veins constituye una biblioteca de modelos y conjunto de herramientas para OMNeT++, diseñada para asistir a investigadores en la realización de simulaciones que involucran vehículos terrestres con capacidades de comunicación. Puede emplearse tanto como elemento central de investigación (por ejemplo, en Redes

Vehiculares Ad Hoc - VANETs) como componente auxiliar (en Sistemas Inteligentes de Transporte - ITS). Se distribuye bajo licencia de código abierto, permitiendo su descarga, modificación y uso gratuito.

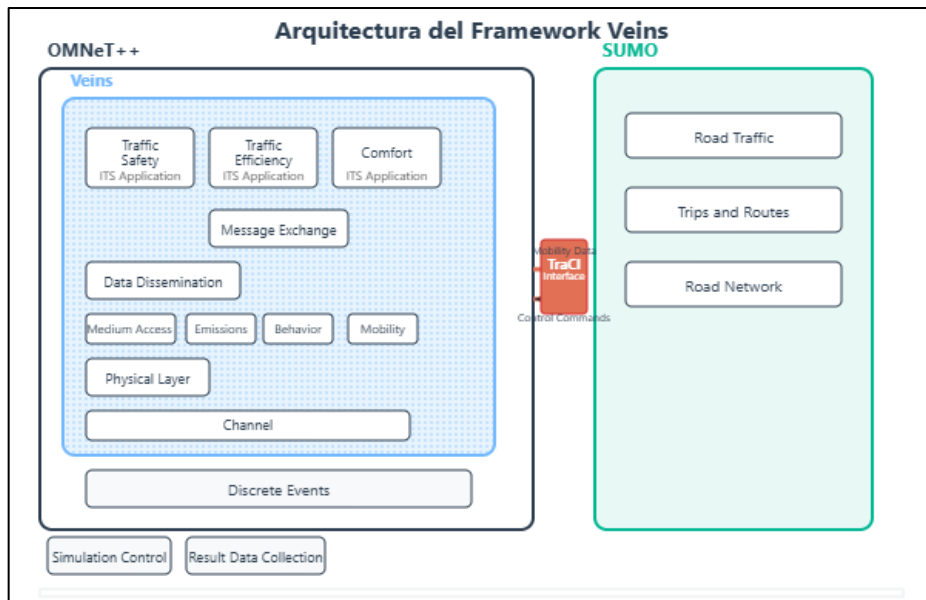


Figura 6. Arquitectura de veins

La biblioteca incorpora un conjunto completo de modelos de simulación para el estudio de vehículos e infraestructura comunicados. En su versión 4.7, se enfoca principalmente en automóviles y camiones que utilizan tecnologías basadas en Redes de Área Local Inalámbricas (WLAN). Para ello, Veins integra un modelo avanzado de los componentes de la capa MAC del estándar IEEE 802.11, empleado por protocolos como IEEE Wireless Access in Vehicular Environments (WAVE) (del cual incluye un modelo de simulación básico).

Dado su diseño modular, Veins puede adaptarse igualmente para modelar otros tipos de nodos móviles como peatones, bicicletas, trenes y Vehículos Aéreos No Tripulados (UAVs), así como para otras tecnologías de comunicación tales como Long Term Evolution (LTE) de banda ancha móvil y Comunicación por Luz Visible [38].

## **2.2.5 Simuladores de movilidad**

### **2.2.5.1 Sumo**

"Simulation of Urban MObility" es un paquete de simulación de tráfico microscópico y continuo de código abierto, altamente portable, diseñado para manejar grandes redes. Permite simulación intermodal incluyendo peatones y viene con un amplio conjunto de herramientas para la creación de escenarios. Es desarrollado principalmente por empleados del Instituto de Sistemas de Transporte del Centro Aeroespacial Alemán [39].

Krajzewicz (2010) describe SUMO como una simulación microscópica de tráfico vial disponible como código abierto bajo licencia GPL. La suite completa incluye herramientas para importar redes viales, generar rutas desde diferentes fuentes, y dos versiones de la simulación de tráfico en sí, una iniciada desde línea de comandos y otra incluyendo una interfaz gráfica de usuario [40]. La simulación utiliza el modelo microscópico, espacio-continuo y tiempo-discreto de seguimiento de vehículos desarrollado por S. Krauß y un modelo de cambio de carril desarrollado dentro del trabajo en la simulación [41].

SUMO permite el modelado de sistemas de tráfico intermodal incluyendo vehículos de carretera, transporte público y peatones. Incluido con SUMO está una gran cantidad de herramientas de apoyo que manejan tareas como búsqueda de rutas, visualización, importación de redes y cálculo de emisiones. SUMO puede ser mejorado con modelos personalizados y proporciona varias APIs para controlar remotamente la simulación [41].

Además de las aplicaciones principales (sumo, sumo-gui, netedit, netconvert, etc.), hay más de 250 herramientas adicionales. Cubren temas desde análisis de redes de tráfico, generación de demanda, modificación de demanda hasta análisis de salida. La mayoría están escritas en Python [42].

Guastella et al. (2023) proporcionan una guía paso a paso para generar y simular un escenario de tráfico usando SUMO, describiendo cómo importar datos de tráfico existentes en un modelo para lograr precisión en la simulación de tráfico [43].

### **2.2.5.2 CARLA**

Dosovitskiy et al. (2017) introdujeron CARLA como un simulador de código abierto para investigación en conducción autónoma. Fue desarrollado desde cero para soportar desarrollo, entrenamiento y validación de sistemas de conducción urbana autónoma. Además del código y protocolos de código abierto, CARLA proporciona activos digitales abiertos (diseños urbanos, edificios, vehículos) que fueron creados para este propósito y pueden usarse libremente [44].

La plataforma de simulación soporta especificación flexible de suites de sensores y condiciones ambientales. CARLA expone una API poderosa que permite a los usuarios controlar todos los aspectos relacionados con la simulación, incluyendo generación de tráfico, comportamientos de peatones, climas, sensores y mucho más. Entre las características principales se incluyen: escalabilidad vía arquitectura servidor multi-cliente donde múltiples clientes en el mismo o diferentes nodos pueden controlar diferentes actores, y una suite de sensores para conducción autónoma que permite configurar diversas suites de sensores incluyendo LIDARs, múltiples cámaras, sensores de profundidad y GPS entre otros [44].

El trabajo utiliza CARLA para estudiar el rendimiento de tres enfoques para conducción autónoma: un pipeline modular clásico, un modelo end-to-end entrenado via aprendizaje por imitación, y un modelo end-to-end entrenado via aprendizaje por refuerzo [44]. CARLA es un simulador poderoso que abarca herramientas para desarrollar, entrenar y validar sistemas en escenarios controlados [45].

Hassan et al. (2022) utilizaron el simulador CARLA para entrenar y evaluar redes neuronales convolucionales para predicción de ángulo de dirección en conducción autónoma end-to-end, recolectando datos del simulador Car Learning to Act (CARLA) para entrenar arquitecturas CNN ligeras [46]. Gutiérrez-Moreno et al. (2022) proporcionaron un enfoque de Deep Reinforcement Learning para manejo de intersecciones que combina DRL con Curriculum Learning, utilizando CARLA como entorno de simulación realista donde la dinámica vehicular es modelada con precisión [47].

CARLA también proporciona capacidades específicas como simulación rápida para planificación y control (este modo deshabilita el renderizado para ofrecer una ejecución rápida de simulación de tráfico y comportamientos viales para los cuales los gráficos no son requeridos), generación de mapas, simulación de escenarios de tráfico, integración con ROS, y líneas base de conducción autónoma incluyendo un agente AutoWare y un agente de Aprendizaje por Imitación Condicional.

En la figura 7 se observa la arquitectura del framework.

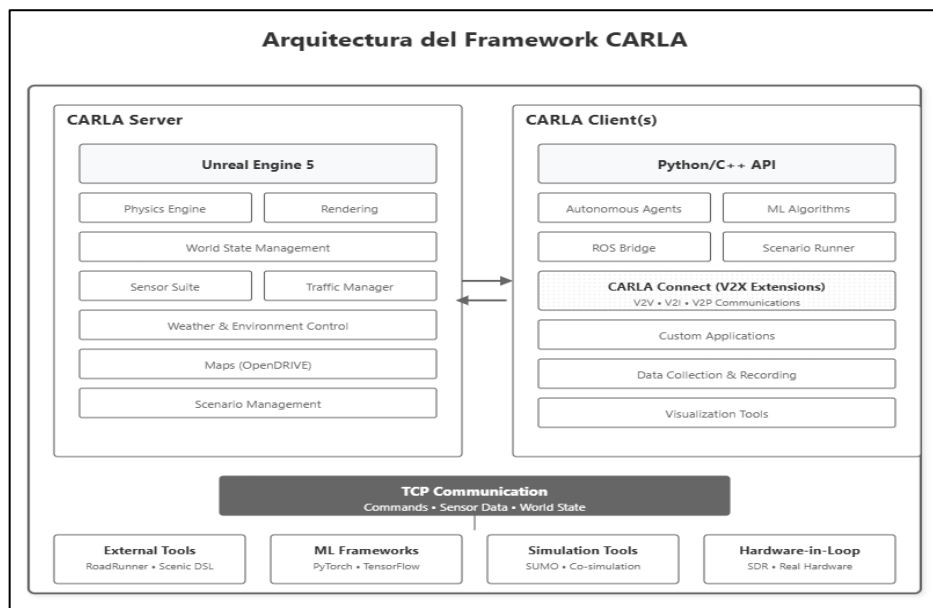


Figura 7. Arquitectura de CARLA

## 2.2.6 Simuladores de eventos discretos

### 2.2.6.1 NS-3 (Network Simulator 3)

NS-3 es un simulador de red de eventos discretos que, a medida que las redes de dispositivos informáticos crecen en tamaño y complejidad, se vuelve crítico tener tecnologías de simulación de red altamente precisas y escalables. A pesar del surgimiento de testbeds a gran escala para investigación de redes, la simulación aún juega un papel vital en términos de escalabilidad (tanto en tamaño como en velocidad experimental), reproducibilidad, prototipado rápido y educación [48].

NS-3 es un instrumento potente y eficaz para simular redes de comunicación; esto se debe a la alta flexibilidad que ofrece y a su detallado modelado de protocolos de red. Para situaciones de movilidad elevada, son más indicados SUMO y NS-3:

SUMO permite la simulación microscópica del tráfico, lo cual es esencial para poner en marcha y analizar métodos de comunicación V2V (vehículo a vehículo) y V2I (vehículo a infraestructura). NS-3 es capaz de soportar simulaciones de protocolos y estrategias de comunicación en configuraciones dinámicas.

#### **2.2.6.2 OMNeT++**

El entorno de simulación de eventos discretos OMNeT++ ha estado disponible públicamente desde 1997. Fue creado con la simulación de redes de comunicación, multiprocesadores y otros sistemas distribuidos en mente como área de aplicación, pero en lugar de construir un simulador especializado, OMNeT++ fue diseñado para ser lo más general posible [49]. OMNeT++ es una biblioteca de simulación C++ basada en componentes modulares y extensibles y un framework, principalmente para construir simuladores de red [50].

OMNeT++ es una biblioteca de simulación C++ basada en componentes modulares y extensibles y un framework que también incluye un entorno de desarrollo integrado y un entorno de tiempo de ejecución gráfico. La funcionalidad específica del dominio (soporte para simulación de redes de comunicación, redes de colas, evaluación de rendimiento, etc.) es proporcionada por frameworks de modelos, desarrollados como proyectos independientes [50].

#### **2.2.6.3 MATLAB/Simulink**

MATLAB es un entorno de computación técnica que combina cálculo numérico, visualización y un lenguaje de programación de alto nivel. Simulink es una extensión gráfica de MATLAB para representar funciones matemáticas y sistemas en forma de diagrama de bloques, y simular la operación de estos sistemas [51].

### **2.3 Marco teórico**

Actualmente las comunicaciones vehiculares se han convertido en un área de investigación fundamental debido al creciente desarrollo de sistemas de transporte inteligente y la necesidad de mejorar la seguridad vial y la eficiencia del tráfico. Las tecnologías Vehicle-to-Everything (V2X) representan una evolución significativa en este campo, permitiendo la comunicación entre vehículos, infraestructura y otros elementos del ecosistema de transporte en tiempo real. Por consiguiente, se describe algunos trabajos de investigación relacionados con la

evaluación del rendimiento de las comunicaciones vehiculares CV2X, los cuales fueron utilizados por el autor como referencia para el desarrollo del proyecto de titulación.

En el trabajo de investigación presentado por los autores se describe el análisis y evaluación de las tecnologías 802.11p, LTE-V2X modo 4 y NR-V2X modo 2. Este estudio aborda la problemática de la necesidad de evaluar el rendimiento de estas tecnologías para cumplir con los requisitos de aplicaciones V2X avanzadas. Los investigadores realizaron una revisión exhaustiva de estas tecnologías y la asignación de recursos de radio, posteriormente ejecutaron simulaciones para verificar el rendimiento de las tecnologías, específicamente en términos de la Relación de Entrega de Paquetes (PDR) basada en la distancia, con el objetivo de determinar el rango de cobertura para transmisión confiable. Los resultados demuestran que la tecnología 802.11p es más susceptible a variaciones en la densidad del tráfico y al aumento en el número de paquetes por segundo, mientras que las tecnologías C-V2X como LTE-V2X y NR-V2X muestran mejor rendimiento en varios escenarios de prueba que involucran cambios en la frecuencia de paquetes y densidad vehicular, concluyendo que es esencial considerar el rendimiento, latencia y rango de cobertura para cumplir con los requisitos de aplicaciones de comunicación V2X avanzadas [16].

En otra investigación relevante se presenta Artery-C, un marco de simulación para la evaluación del rendimiento de protocolos Cellular V2X y aplicaciones V2X. Los autores destacan que la tecnología Cellular V2X, introducida con LTE Release 14, habilita la comunicación de un dispositivo a otro dispositivo para soportar aplicaciones de seguridad vial y eficiencia del tráfico. Su simulador se basa en el framework de simulación SimuLTE y lo extiende sustancialmente implementando planos de control y usuario, una interfaz de enlace lateral dedicada con enfoque específico en el cambio dinámico de modo y algunas características avanzadas de redes móviles 5G, tales como numerologías variables. Artery-C se integra perfectamente en el framework de simulación Artery, lo cual permite la simulación de mensajes V2X estandarizados en la capa de facilidades, así como el acoplamiento al simulador de movilidad SUMO. Para demostrar las capacidades de

Artery-C, los investigadores evaluaron el platooning basado en V2X como un caso de uso representativo y presentaron resultados para modo 3, modo 4 y cambio de modo en un escenario de autopista, proporcionando una herramienta valiosa para la evaluación integral de sistemas de comunicación vehicular [23].

Estos trabajos de investigación establecen un fundamento sólido para comprender las diferentes tecnologías de comunicación vehicular, sus métodos de evaluación y las herramientas de simulación disponibles, aspectos que son fundamentales para el desarrollo de una evaluación comprehensiva del rendimiento de las comunicaciones vehiculares CV2X en el contexto de aplicaciones de transporte inteligente modernas.

## **CAPÍTULO III**

### **3.1. Introducción**

La implementación de OpenCV2X utilizando el framework Artery representa una evolución significativa en el ámbito de las simulaciones de comunicaciones vehiculares. Esta combinación tecnológica permite la creación de entornos de simulación robustos y escalables que integran las capacidades avanzadas de comunicación celular C-V2X (Cellular Vehicle-to-Everything) con las funcionalidades de simulación de redes vehiculares que caracterizan a Artery.

Este capítulo busca ofrecer una guía completa para instalar y configurar OpenCV2X con Artery, describiendo minuciosamente cada uno de los pasos requeridos para establecer un entorno de desarrollo totalmente operativo. La metodología expuesta se ha creado para asegurar que el proceso de instalación sea reproducible, sin importar cuál sea el nivel de experiencia técnica del usuario.

La arquitectura resultante de esta instalación proporcionará las bases necesarias para la investigación en sistemas de transporte inteligentes, permitiendo la simulación de escenarios complejos que involucran comunicaciones vehiculares avanzadas. Esta plataforma constituye una herramienta fundamental para el desarrollo y evaluación de nuevos protocolos y aplicaciones en el contexto de las comunicaciones V2X.

### **3.2. Requisitos del Sistema y Consideraciones Previas**

#### **3.2.1. Especificaciones de Hardware**

El desarrollo exitoso de simulaciones utilizando OpenCV2X con Artery requiere de un hardware adecuado que pueda soportar las demandas computacionales inherentes a este tipo de aplicaciones. Se aconseja un procesador Intel Core i5 como mínimo, pero para un óptimo desempeño es mejor usar uno de la serie Intel Core i7 o más. La memoria RAM es otro elemento fundamental, siendo 8GB el mínimo absoluto necesario, aunque se aconseja tener por lo menos 16GB para asegurar que las simulaciones complejas se realicen sin problemas.

En cuanto al almacenamiento, es necesario disponer de un mínimo de 20GB de espacio libre en disco para acomodar todos los componentes del sistema, las dependencias y los archivos de trabajo. La arquitectura del sistema debe ser x86\_64, ya que las bibliotecas y componentes utilizados han sido optimizados específicamente para esta arquitectura.

### **3.2.2. Dependencias de Software y Compatibilidad**

La instalación de OpenCV2X con Artery depende de un ecosistema complejo de software que debe ser cuidadosamente coordinado para garantizar la compatibilidad entre componentes. Ubuntu 16.04 o versiones superiores constituyen la base del sistema operativo recomendado, aunque es importante señalar que la instalación ha sido probada específicamente en estas versiones y su funcionamiento en otras distribuciones de Linux no está garantizado.

SUMO (Simulation of Urban Mobility) versión 1.0 o superior es fundamental para las simulaciones de tráfico vehicular. Este simulador proporciona las capacidades de modelado de movilidad que son esenciales para crear escenarios realistas de comunicaciones vehiculares. CMake versión 3.13 o superior actúa como el sistema de construcción principal, gestionando las dependencias y la compilación de los diversos componentes del proyecto.

El marco fundamental para la realización de simulaciones de comunicaciones vehiculares es OMNeT++ versión 5.4 o superior, que representa el núcleo del ambiente de simulación de redes. El compilador que se necesita es GNU GCC versión 7.3, el cual generalmente está preinstalado en las versiones de Ubuntu mencionadas. La versión 1.65.1 de Boost ofrece bibliotecas de C++ extendidas que los diversos elementos del sistema emplean en gran medida.

### **3.2.3. Bibliotecas Criptográficas y de Seguridad**

Las comunicaciones vehiculares requieren implementaciones robustas de algoritmos criptográficos para garantizar la seguridad e integridad de los mensajes intercambiados. Crypto++ (también conocida como CryptoPP) constituye la biblioteca fundamental para estas funcionalidades, proporcionando

implementaciones optimizadas de algoritmos de cifrado simétrico y asimétrico, funciones hash criptográficas y generadores de números pseudoaleatorios seguros.

Para que Vanetza, el componente que implementa los protocolos de comunicación ETSI ITS-G5, funcione correctamente, se necesita específicamente la biblioteca Crypto++ versión 5.6.5 o una superior. Esta biblioteca ofrece las primitivas criptográficas requeridas para poner en práctica los mecanismos de seguridad establecidos en las normas IEEE 1609.2 y ETSI TS 103 097, lo que incluye la creación y verificación de firmas digitales ECDSA y el cifrado de mensajes mediante algoritmos simétricos como AES.

Adicionalmente, OpenSSL versión 1.1.0 o superior complementa las capacidades criptográficas del sistema, proporcionando funcionalidades adicionales para el manejo de certificados digitales y la implementación de protocolos de comunicación segura. La integración de estas bibliotecas criptográficas asegura que las simulaciones puedan modelar accurately los aspectos de seguridad que son críticos en implementaciones reales de sistemas V2X.

#### **3.2.4. Bibliotecas de Desarrollo y Herramientas Auxiliares**

El entorno de desarrollo requiere bibliotecas adicionales que soportan diversas funcionalidades especializadas del sistema. Entre estas, la biblioteca Boost merece atención especial debido a su papel fundamental en el ecosistema de OpenCV2X. Las bibliotecas Boost proporcionan implementaciones de alta calidad de estructuras de datos avanzadas, incluyendo smart pointers que gestionan automáticamente la memoria, contenedores especializados como boost:circular\_buffers utilizados para el manejo eficiente de datos de simulación, y bibliotecas de serialización que permiten el intercambio de objetos complejos entre diferentes componentes del sistema.

En concreto, Vanetza hace un amplio uso de Boost.Asio para implementar comunicaciones en red asíncronas. Por su parte, Boost.Date\_Time ofrece habilidades avanzadas de gestión del tiempo que son esenciales para la sincronización exacta de eventos en simulaciones vehiculares. Boost.Filesystem hace más sencillas las operaciones para manipular directorios y archivos, lo que

permite manejar de manera más sencilla los archivos de configuración y los resultados de simulación.

GeographicLib constituye otra biblioteca fundamental que proporciona algoritmos geodésicos de alta precisión esenciales para simulaciones vehiculares realistas. Esta biblioteca implementa cálculos precisos de distancias geodésicas utilizando el elipsoide WGS84, conversiones entre diferentes sistemas de coordenadas geográficas (incluyendo UTM, MGRS y coordenadas geográficas decimales), y cálculos de rumbo y distancia entre puntos geográficos. En el contexto de OpenCV2X, GeographicLib es particularmente importante para la conversión precisa entre las coordenadas utilizadas por SUMO en las simulaciones de tráfico y las coordenadas geográficas reales utilizadas en los protocolos de comunicación vehicular.

La precisión de GeographicLib es crucial para simulaciones que involucran escenarios geográficamente distribuidos o que requieren interoperabilidad con datos geográficos reales. Para poder calcular con exactitud las métricas de comunicación que dependen de la distancia física entre vehículos, es fundamental que los algoritmos aplicados en esta biblioteca aseguren una precisión submétrica en el cálculo de distancias.

Las bibliotecas de desarrollo de Qt5 son imprescindibles para que los elementos de la interfaz gráfica del usuario en OMNeT++ se puedan visualizar de forma interactiva y se pueda acceder a herramientas que analizan en tiempo real. Para ofrecer un entorno gráfico totalmente operativo que admita visualizaciones 2D y 3D de las simulaciones, se necesitan, en concreto, los paquetes qt5-default, libqt5opengl5-dev y qttools5-dev. Para procesar archivos de configuración XML que establecen los parámetros de simulación y los escenarios de red, la biblioteca libxml2-dev es esencial. Esta biblioteca ofrece las capacidades de análisis y validación XML que todos los elementos del sistema emplean con frecuencia para interpretar archivos de configuración complejos. Estos incluyen la configuración de redes OMNeT++, los parámetros de movilidad SUMO y la configuración particular de protocolos Vanetza.

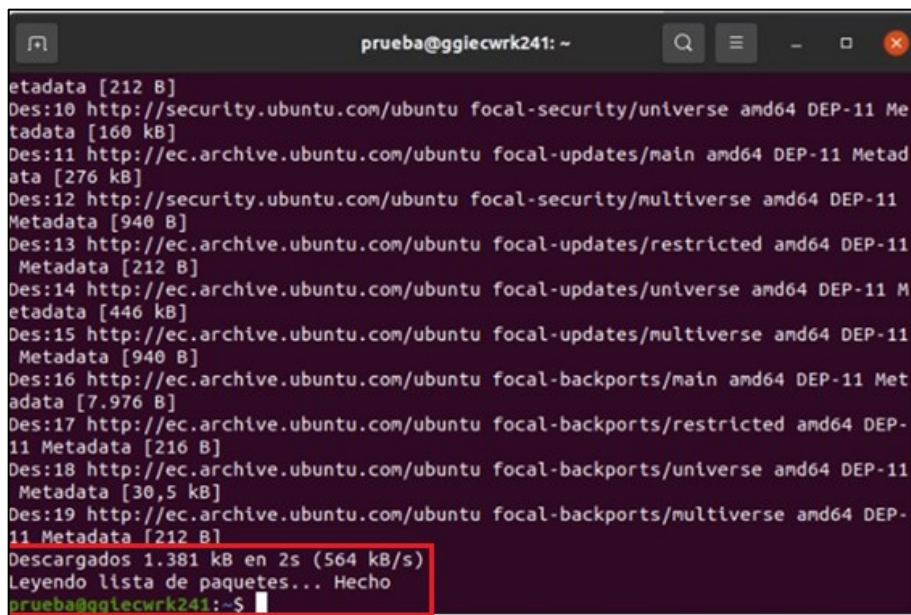
### 3.3. Preparación del Entorno de Desarrollo

La preparación adecuada del entorno de desarrollo constituye un paso fundamental que determinará el éxito de todo el proceso de instalación. Este proceso comienza con la actualización completa del sistema operativo para asegurar que todas las bibliotecas y dependencias del sistema estén en sus versiones más recientes y seguras.

La actualización del sistema se realiza mediante la ejecución del comando:

```
$ sudo apt update && sudo apt upgrade -y
```

que actualiza la base de datos de paquetes disponibles y posteriormente instala las versiones más recientes de todos los paquetes instalados en el sistema. Durante este proceso, el usuario observará en la consola una serie de mensajes que indican el progreso de la descarga e instalación de las actualizaciones.



```
prueba@ggilecwrk241: ~  
etadata [212 B]  
Des:10 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Me  
tadata [160 kB]  
Des:11 http://ec.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metad  
ata [276 kB]  
Des:12 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11  
Metadata [940 B]  
Des:13 http://ec.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 DEP-11  
Metadata [212 B]  
Des:14 http://ec.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 M  
etadata [446 kB]  
Des:15 http://ec.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11  
Metadata [940 B]  
Des:16 http://ec.archive.ubuntu.com/ubuntu focal-backports/main amd64 DEP-11 Met  
adata [7.976 B]  
Des:17 http://ec.archive.ubuntu.com/ubuntu focal-backports/restricted amd64 DEP-  
11 Metadata [216 B]  
Des:18 http://ec.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11  
Metadata [30,5 kB]  
Des:19 http://ec.archive.ubuntu.com/ubuntu focal-backports/multiverse amd64 DEP-  
11 Metadata [212 B]  
Descargados 1.381 kB en 2s (564 kB/s)  
Leyendo lista de paquetes... Hecho  
prueba@ggilecwrk241:~$
```

Figura 8. Actualización de dependencias

Luego, es preciso poner en marcha las herramientas esenciales de desarrollo que se emplearán a lo largo del proceso de compilación. Esto abarca el paquete build-essential, que incluye las herramientas y compiladores necesarios para desarrollar software, CMake para manejar la construcción del proyecto, Git para controlar las versiones y wget y curl como utilidades para bajar archivos de la red.

```
$ sudo apt install -y build-essential cmake git wget curl
```

La instalación de estas dependencias se ejecuta mediante el comando:

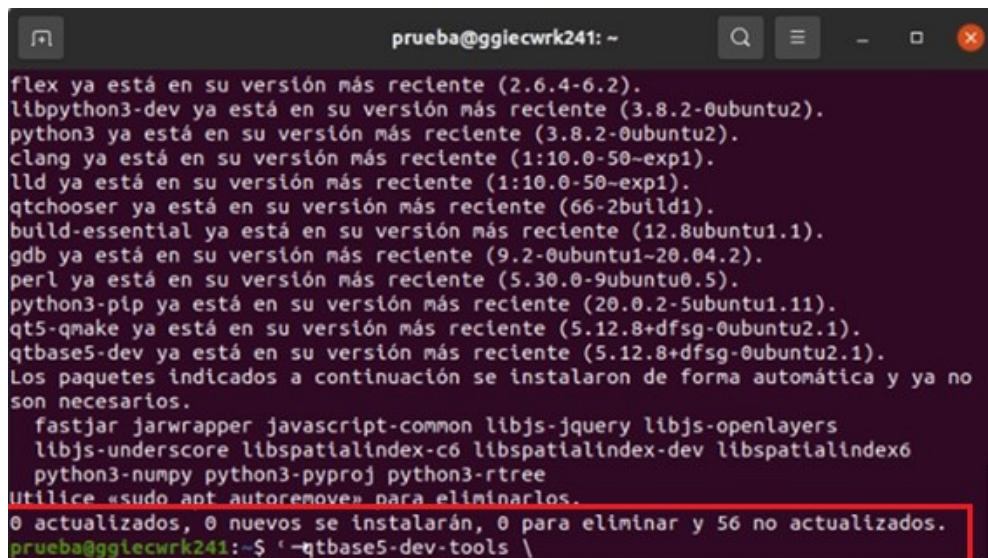
```
$ sudo apt install -y libboost-all-dev  
$ sudo apt install -y python3 python3-pip
```

seguido por la instalación de las bibliotecas fundamentales. Las bibliotecas Boost se instalan con:

y las bibliotecas criptográficas y auxiliares a través de:

```
$ sudo apt install -y libcrypto++-dev libssl-dev libgeographic-dev libxml2-dev.
```

Además, para respaldar los elementos gráficos de OMNeT++, es imprescindible que instale las bibliotecas de desarrollo de Qt5. Esto se lleva a cabo usando `sudo apt install -y qt5-default libqt5opengl5-dev qttools5-dev qtbase5-dev`. La consola indicará el avance de cada instalación, con datos sobre los paquetes que se descargan y configuran, además de las dependencias adicionales que se instalan de manera automática.



```
prueba@gglicwrk241: ~  
flex ya está en su versión más reciente (2.6.4-6.2).  
libpython3-dev ya está en su versión más reciente (3.8.2-0ubuntu2).  
python3 ya está en su versión más reciente (3.8.2-0ubuntu2).  
clang ya está en su versión más reciente (1:10.0-50-exp1).  
lld ya está en su versión más reciente (1:10.0-50-exp1).  
qtchooser ya está en su versión más reciente (66-2build1).  
build-essential ya está en su versión más reciente (12.8ubuntu1.1).  
gdb ya está en su versión más reciente (9.2-0ubuntu1-20.04.2).  
perl ya está en su versión más reciente (5.30.0-9ubuntu0.5).  
python3-pip ya está en su versión más reciente (20.0.2-5ubuntu1.11).  
qt5-qmake ya está en su versión más reciente (5.12.8+dfsg-0ubuntu2.1).  
qtbase5-dev ya está en su versión más reciente (5.12.8+dfsg-0ubuntu2.1).  
Los paquetes indicados a continuación se instalaron de forma automática y ya no  
son necesarios.  
fastjar jarwrapper javascript-common libjs-jquery libjs-openlayers  
libjs-underscore libspatialindex-c6 libspatialindex-dev libspatialindex6  
python3-numpy python3-pyproj python3-rtree  
Utilice «sudo apt autoremove» para eliminarlos.  
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 56 no actualizados.  
prueba@gglicwrk241:~$ 'qtbase5-dev-tools \
```

Figura 9. Instalación de bibliotecas

### 3.4 Instalación y Configuración de OMNeT++

#### 3.4.1 Obtención e Instalación de OMNeT++

OMNeT++ es el componente central del ambiente de simulación y, para que todo el sistema funcione correctamente, es fundamental instalarlo bien. La descarga del código es el primer paso del procedimiento.

origen desde el repositorio oficial de GitHub. La descarga se efectúa con wget para conseguir el archivo comprimido, después se extrae y se coloca en el directorio /opt/ del

sistema. El siguiente comando descarga el archivo fuente:

```
$ cd ~/Downloads
$ tar -xzf omnetpp-5.7-src-linux.tgz
$ sudo mv omnetpp-5.7 /opt/
```

Finalmente, mueve la instalación a su ubicación definitiva. Durante este proceso, la consola mostrará el progreso de la descarga, indicando el tamaño del archivo y la velocidad de transferencia.

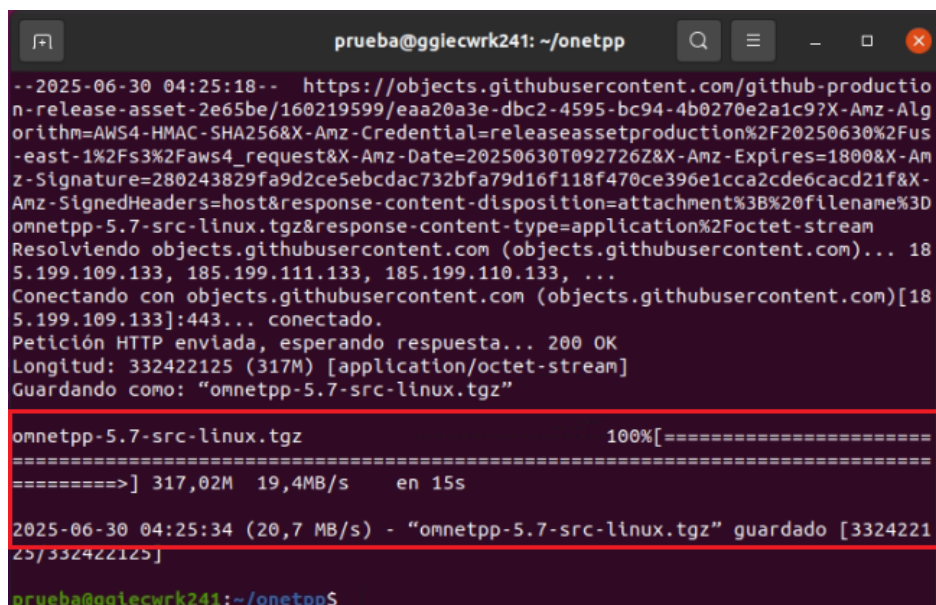


Figura 10. Descarga de omnetpp

### 3.4.2 Configuración del Entorno de Variables

La configuración correcta de las variables de entorno es esencial para que el sistema pueda localizar y utilizar OMNeT++ adecuadamente. Este proceso involucra la modificación del archivo `.bashrc` del usuario para incluir las rutas necesarias en las variables `PATH` y `LD_LIBRARY_PATH`.

Los siguientes comandos agregan las rutas necesarias al archivo de configuración del Shell:

```
$ echo 'export PATH=$PATH:/opt/omnetpp-5.7/bin' >>
~/.bashrc $ echo 'export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/omnetpp-5.7/lib' >>
~/.bashrc
```

```
$ source ~/.bashrc
```

Posteriormente se recarga la configuración para que los cambios tomen efecto inmediatamente en la sesión actual.

### 3.4.3 Proceso de Configuración de OMNeT++

La configuración de OMNeT++ requiere atención especial debido a problemas conocidos con ciertos componentes opcionales. Específicamente, las bibliotecas `osgEarth` y `OSG` pueden causar conflictos durante la ejecución de simulaciones, por lo que se desactivan explícitamente durante el proceso de configuración.

Navegando al directorio de OMNeT++ con:

```
$ cd /opt/omnetpp-5.7
```

Y ejecutando:

```
$ source setenv

$ ./configure WITH_OSGEARTH=no WITH_OSG=no
```

Se inicia el proceso de configuración. Este script analiza el sistema, verifica la disponibilidad de las dependencias necesarias y genera los archivos de configuración apropiados para la compilación. La consola mostrará información detallada sobre cada componente que está siendo verificado y configurado.

```
prueba@gglecwrk241: ~/opencv2x/omnetpp-5.7.1
checking for PTHREAD with CFLAGS=" -fPIC " LIBS="-lpthread"... yes
checking for LibXML XML parser with CFLAGS=" -fPIC " LIBS="-lxml2"... no
checking for LibXML XML parser with CFLAGS=" -fPIC -I/usr/include/libxml2" LIBS=
"-lxml2"... yes
configure: Using LibXML for XML parsing
checking for zlib with CFLAGS=" -fPIC " LIBS="-lz"... yes
checking for Akaroa with CFLAGS=" -fPIC -I/usr/local/akaroa/include" LIBS="-L/us
r/local/akaroa/lib -lakaroa -lfl"... no
configure: WARNING: Optional package Akaroa not found
configure: creating ./config.status
config.status: creating Makefile.inc
config.status: creating src/qtenv/qtenv.pri

WARNING: The configuration script could not detect the following packages:

    MPI (optional) Akaroa (optional)

Scroll up to see the warning messages (use shift+PgUp), and search config.log
for more details. While you can use OMNeT++ in the current configuration,
be aware that some functionality may be unavailable or incomplete.

Configuration phase finished. Use 'make' to build OMNeT++.
prueba@gglecwrk241:~/opencv2x/omnetpp-5.7.1$
```

Figura 11. Configuración de omnetpp

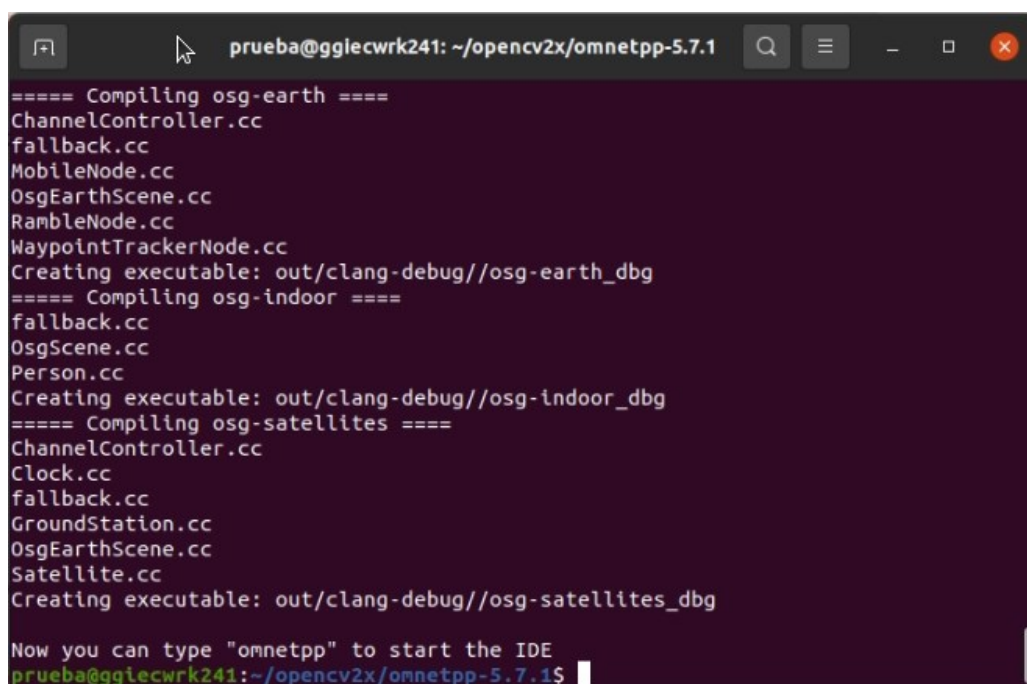
### 3.4.3 Compilación de OMNeT++

```
$ make -j$(nproc)
```

La compilación de OMNeT++ es un proceso intensivo que puede requerir varios minutos dependiendo de las capacidades del hardware disponible. El siguiente comando:

Inicia la compilación utilizando todos los núcleos de procesamiento disponibles para acelerar el proceso. El parámetro `-j$(nproc)` instruye a `make` para utilizar tantos procesos paralelos como núcleos tenga el procesador.

Durante la compilación, la consola mostrará el progreso de la construcción de cada componente, incluyendo información sobre archivos fuente que están siendo compilados, bibliotecas que están siendo enlazadas y cualquier advertencia o error que pueda surgir. Un proceso de compilación exitoso concluirá sin errores críticos y resultará en la generación de todos los ejecutables y bibliotecas necesarias.



```
prueba@ggicwrk241: ~/opencv2x/omnetpp-5.7.1
==== Compiling osg-earth ====
ChannelController.cc
fallback.cc
MobileNode.cc
OsgEarthScene.cc
RambleNode.cc
WaypointTrackerNode.cc
Creating executable: out/clang-debug//osg-earth_dbg
==== Compiling osg-indoor ====
fallback.cc
OsgScene.cc
Person.cc
Creating executable: out/clang-debug//osg-indoor_dbg
==== Compiling osg-satellites ====
ChannelController.cc
Clock.cc
fallback.cc
GroundStation.cc
OsgEarthScene.cc
Satellite.cc
Creating executable: out/clang-debug//osg-satellites_dbg

Now you can type "omnetpp" to start the IDE
prueba@ggicwrk241:~/opencv2x/omnetpp-5.7.1$
```

*Figura 12. Build de omnetpp*

### 3.4.5 Verificación de la Instalación de OMNeT++

```
$ omnetpp
```

La verificación de que OMNeT++ se ha instalado correctamente se realiza mediante la ejecución del comando:

Que debe iniciar el entorno de desarrollo integrado (IDE) de OMNeT++. La aparición de la interfaz gráfica del IDE confirma que la instalación se ha completado exitosamente y que todas las dependencias están correctamente configuradas.

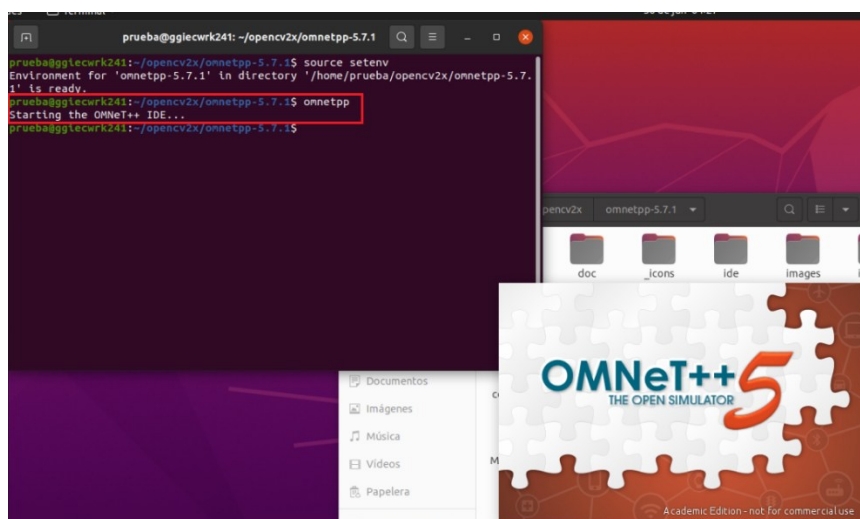


Figura 13. Verificación de omnetpp

### 3.5 Instalación y Configuración de SUMO

SUMO (Simulación de movilidad urbana) ofrece las habilidades necesarias para simular el tráfico vehicular, las cuales son esenciales para desarrollar situaciones realistas de comunicación entre vehículos. La instalación de SUMO puede hacerse directamente desde los repositorios oficiales de Ubuntu, lo cual hace que el procedimiento sea mucho más sencillo y asegura la compatibilidad con el sistema operativo.

El comando:

```
$ sudo apt install -y sumo sumo-tools sumo-doc
```

Instala no solo el simulador principal sino también las herramientas auxiliares y la documentación completa. Esto proporciona un conjunto comprehensivo de utilidades para la creación, edición y análisis de simulaciones de tráfico.

La configuración de las variables de entorno para SUMO sigue un proceso similar al utilizado para OMNeT++. Los comandos a continuación establecen las rutas necesarias:

```
$ echo 'export SUMO_HOME=/usr/share/sumo' >> ~/.bashrc y echo 'export PATH=$PATH:$SUMO_HOME/bin' >> ~/.bashrc
```

```
git clone https://github.com/brianmc95/OpenCV2X.git
```

Mientras que:

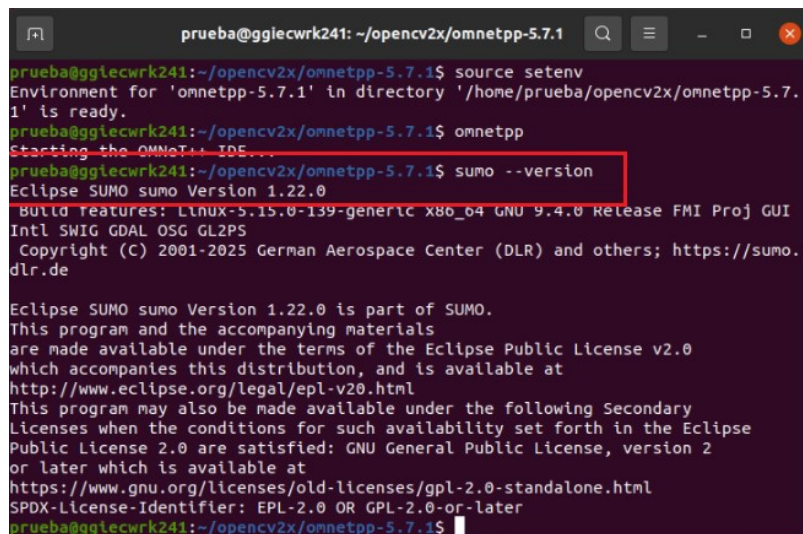
```
$ source ~/.bashrc
```

Aplican los cambios a la sesión actual.

La verificación de la instalación se realiza mediante:

```
$ sumo --version
```

La cual, debe mostrar información detallada sobre la versión instalada de SUMO. Esta información incluye el número de versión, la fecha de compilación y las características habilitadas en la construcción específica.



```
prueba@ggtecwrk241: ~/opencv2x/omnetpp-5.7.1
prueba@ggtecwrk241:~/opencv2x/omnetpp-5.7.1$ source setenv
Environment for 'omnetpp-5.7.1' in directory '/home/prueba/opencv2x/omnetpp-5.7.1' is ready.
prueba@ggtecwrk241:~/opencv2x/omnetpp-5.7.1$ omnetpp
Starting the OMNeT++ IDE...
prueba@ggtecwrk241:~/opencv2x/omnetpp-5.7.1$ sumo --version
Eclipse SUMO sumo Version 1.22.0
Build Features: Linux-5.15.0-139-generic x86_64 GNU 9.4.0 Release FMI Proj GUI
Intl SWIG GDAL OSG GL2PS
Copyright (C) 2001-2025 German Aerospace Center (DLR) and others; https://sumo.dlr.de

Eclipse SUMO sumo Version 1.22.0 is part of SUMO.
This program and the accompanying materials
are made available under the terms of the Eclipse Public License v2.0
which accompanies this distribution, and is available at
http://www.eclipse.org/legal/epl-v20.html
This program may also be made available under the following Secondary
Licenses when the conditions for such availability set forth in the Eclipse
Public License 2.0 are satisfied: GNU General Public License, version 2
or later which is available at
https://www.gnu.org/licenses/old-licenses/gpl-2.0-standalone.html
SPDX-License-Identifier: EPL-2.0 OR GPL-2.0-or-later
prueba@ggtecwrk241:~/opencv2x/omnetpp-5.7.1$
```

Figura 14. Versión de sumo

### 3.6 Obtención y Preparación del Código Fuente de OpenCV2X

La obtención del código fuente de OpenCV2X marca el inicio de la fase específica de instalación del sistema de comunicaciones vehiculares. Este proceso comienza con la clonación del repositorio principal y la inicialización de todos los submódulos que contienen las dependencias específicas del proyecto.

El comando:

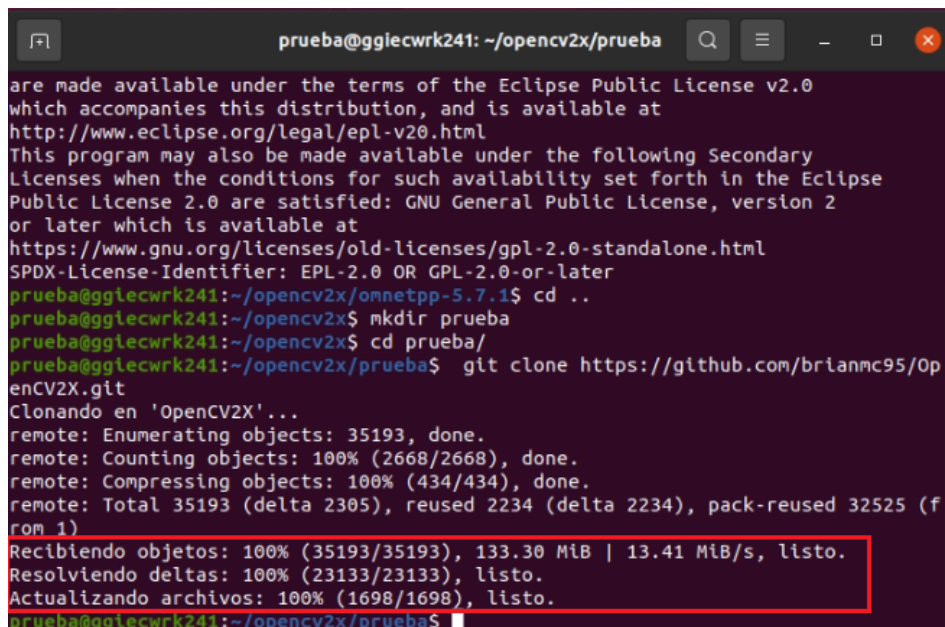
Descarga el repositorio principal al directorio local.

Posteriormente, navegando al directorio del proyecto con `cd OpenCV2X` y ejecutando:

```
git submodule update --init --recursive,
```

Se descargan e inicializan todos los submódulos necesarios. Estos submódulos incluyen componentes como INET, Veins, SimuLTE y Vanetza, cada uno de los cuales proporciona funcionalidades específicas para la simulación de comunicaciones vehiculares.

El proceso de clonación de submódulos puede ser extenso debido a la cantidad de código que debe descargarse. La consola mostrará el progreso de cada submódulo individual, incluyendo información sobre los repositorios que están siendo clonados y el progreso de la descarga.



```
prueba@ggilecwrk241: ~/opencv2x/prueba
are made available under the terms of the Eclipse Public License v2.0
which accompanies this distribution, and is available at
http://www.eclipse.org/legal/epl-v20.html
This program may also be made available under the following Secondary
Licenses when the conditions for such availability set forth in the Eclipse
Public License 2.0 are satisfied: GNU General Public License, version 2
or later which is available at
https://www.gnu.org/licenses/old-licenses/gpl-2.0-standalone.html
SPDX-License-Identifier: EPL-2.0 OR GPL-2.0-or-later
prueba@ggilecwrk241:~/opencv2x/omnetpp-5.7.1$ cd ..
prueba@ggilecwrk241:~/opencv2x$ mkdir prueba
prueba@ggilecwrk241:~/opencv2x$ cd prueba/
prueba@ggilecwrk241:~/opencv2x/prueba$ git clone https://github.com/brianmc95/OpenCV2X.git
Clonando en 'OpenCV2X'...
remote: Enumerating objects: 35193, done.
remote: Counting objects: 100% (2668/2668), done.
remote: Compressing objects: 100% (434/434), done.
remote: Total 35193 (delta 2305), reused 2234 (delta 2234), pack-reused 32525 (from 1)
Recibiendo objetos: 100% (35193/35193), 133.30 MiB | 13.41 MiB/s, listo.
Resolviendo deltas: 100% (23133/23133), listo.
Actualizando archivos: 100% (1698/1698), listo.
prueba@ggilecwrk241:~/opencv2x/prueba$
```

Figura 15. Clonación del repositorio de `opencv2x`

## 3.7 Compilación Integral de Componentes

### 3.7.1 Estrategia de Compilación Global

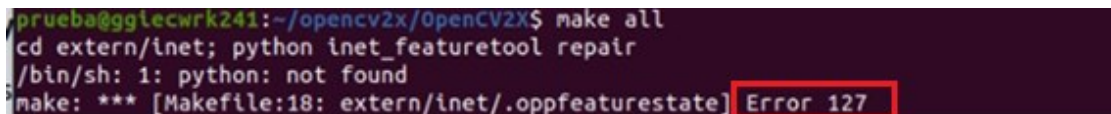
Para evitar errores de dependencia, la compilación de OpenCV2X con Artery requiere que se compilen varios elementos interdependientes en un orden determinado. El proyecto ofrece un Makefile principal que automatiza en gran medida este proceso; sin embargo, es fundamental entender las dependencias subyacentes para solucionar cualquier inconveniente que pueda presentarse.

El siguiente comando:

```
$ make all
```

Cuando es ejecutado desde el directorio raíz del proyecto inicia la compilación automática de todos los componentes en el orden correcto. Este proceso puede requerir considerable tiempo, especialmente en la primera ejecución, ya que involucra la compilación de múltiples bibliotecas y frameworks complejos.

Ya que el programa está pensado para usarse con python2 al ejecutar make all nos dará el error 127:

A terminal window screenshot with a dark background. The text shows a user running 'make all' in a directory. The output shows a shell prompt, a directory change to 'extern/inet', and an attempt to run 'python inet\_featuretool repair'. This results in an error: '/bin/sh: 1: python: not found'. The 'make' command then reports: 'make: \*\*\* [Makefile:18: extern/inet/.oppfeaturestate] Error 127'. The 'Error 127' text is highlighted with a red rectangular box.

```
prueba@ggglecwrk241:~/opencv2x/OpenCV2X$ make all
cd extern/inet; python inet_featuretool repair
/bin/sh: 1: python: not found
make: *** [Makefile:18: extern/inet/.oppfeaturestate] Error 127
```

*Figura 16. Error 127 durante compilación*

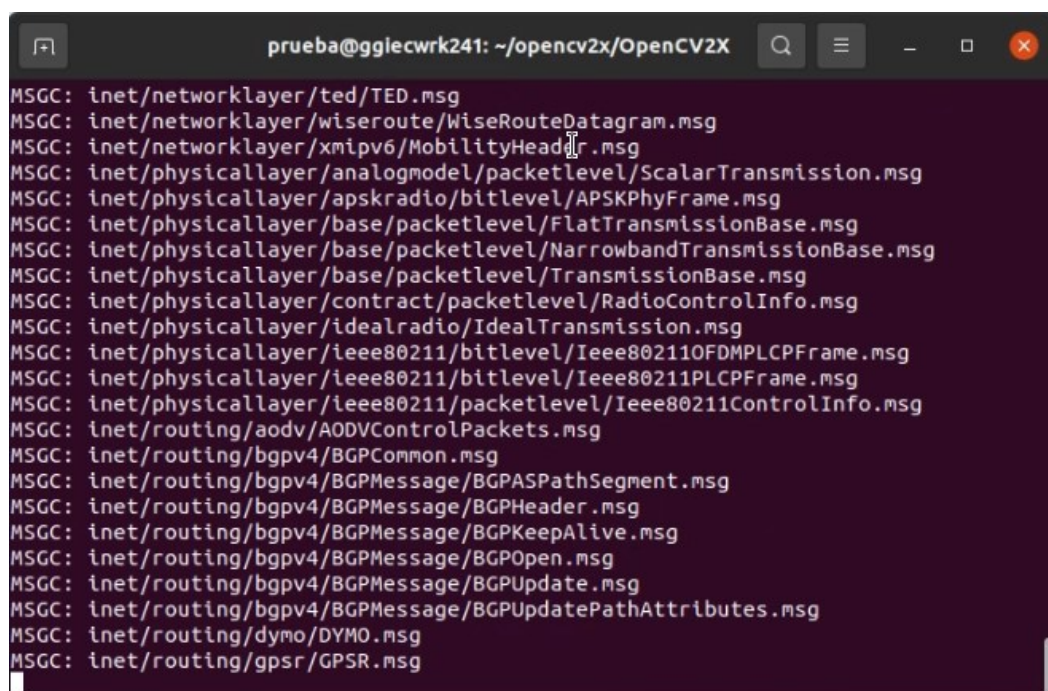
El cual se resuelve usando el comando:

```
$ sudo ln -s /usr/bin/python3 /usr/bin/python
```

Esto crea un enlace simbólico para que el programa haga la request de Python el sistema lo dirija hacia Python3. Una vez solucionado este error procedemos con

```
$ make all
```

Dando como resultado:



```
MSGC: inet/networklayer/ted/TED.msg
MSGC: inet/networklayer/wiseroute/WiseRouteDatagram.msg
MSGC: inet/networklayer/xmipv6/MobilityHead.msg
MSGC: inet/physicallayer/analogmodel/packetlevel/ScalarTransmission.msg
MSGC: inet/physicallayer/apskradio/bitlevel/APSKPhyFrame.msg
MSGC: inet/physicallayer/base/packetlevel/FlatTransmissionBase.msg
MSGC: inet/physicallayer/base/packetlevel/NarrowbandTransmissionBase.msg
MSGC: inet/physicallayer/base/packetlevel/TransmissionBase.msg
MSGC: inet/physicallayer/contract/packetlevel/RadioControlInfo.msg
MSGC: inet/physicallayer/idealradio/IdealTransmission.msg
MSGC: inet/physicallayer/ieee80211/bitlevel/Ieee80211FDMLCPFrame.msg
MSGC: inet/physicallayer/ieee80211/bitlevel/Ieee80211PLCPFrame.msg
MSGC: inet/physicallayer/ieee80211/packetlevel/Ieee80211ControlInfo.msg
MSGC: inet/routing/aodv/AODVControlPackets.msg
MSGC: inet/routing/bgpv4/BGPCommon.msg
MSGC: inet/routing/bgpv4/BGPMessage/BGPASPathSegment.msg
MSGC: inet/routing/bgpv4/BGPMessage/BGPHeader.msg
MSGC: inet/routing/bgpv4/BGPMessage/BGPKeepAlive.msg
MSGC: inet/routing/bgpv4/BGPMessage/BGPOpen.msg
MSGC: inet/routing/bgpv4/BGPMessage/BGPUpdate.msg
MSGC: inet/routing/bgpv4/BGPMessage/BGPUpdatePathAttributes.msg
MSGC: inet/routing/dymo/DYMO.msg
MSGC: inet/routing/gpsr/GPSR.msg
```

Figura 17. Compilación de inet

### 3.7.2 Configuración Específica de Veins

Veins requiere un proceso de configuración particular que debe realizarse manualmente debido a su arquitectura específica. Navegando al directorio `extern/veins` y ejecutando:

```
$ ./configure
```

Al igual que se mencionó anteriormente este proyecto fue diseñado para funcionar con Python2, por lo tanto, para poder compilar Veins, necesitaremos instalar Python2 para lo cual emplearemos el siguiente comando:

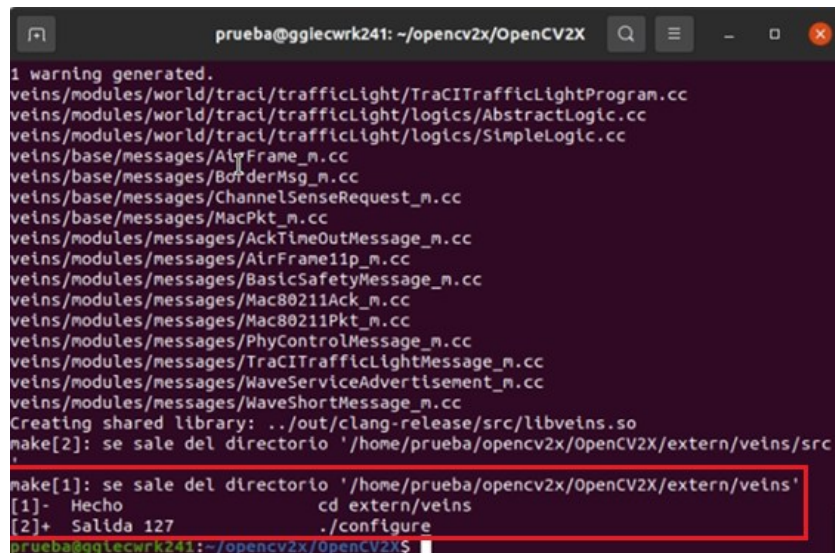
```
$ sudo apt install python2
```

Luego de esto, se inicia el script de configuración específico de Veins que analiza el entorno y configura los parámetros de compilación apropiados.

Posteriormente, la ejecución de:

```
$ make -j$(nproc)
```

Compila Veins utilizando todos los núcleos de procesamiento disponibles. La consola mostrará información detallada sobre la configuración detectada y el progreso de la compilación de cada componente de Veins.



```
prueba@ggglecwrk241: ~/opencv2x/OpenCV2X
1 warning generated.
veins/modules/world/traci/trafficLight/TraCITrafficLightProgram.cc
veins/modules/world/traci/trafficLight/logics/AbstractLogic.cc
veins/modules/world/traci/trafficLight/logics/SimpleLogic.cc
veins/base/messages/AirFrame_n.cc
veins/base/messages/BorderMsg_n.cc
veins/base/messages/ChannelSenseRequest_n.cc
veins/base/messages/MacPkt_n.cc
veins/modules/messages/AckTimeoutMessage_n.cc
veins/modules/messages/AirFrame11p_n.cc
veins/modules/messages/BasicSafetyMessage_n.cc
veins/modules/messages/Mac80211Ack_n.cc
veins/modules/messages/Mac80211Pkt_n.cc
veins/modules/messages/PhyControlMessage_n.cc
veins/modules/messages/TraCITrafficLightMessage_n.cc
veins/modules/messages/WaveServiceAdvertisement_n.cc
veins/modules/messages/WaveShortMessage_n.cc
Creating shared library: ../out/clang-release/src/libveins.so
make[2]: se sale del directorio '/home/prueba/opencv2x/OpenCV2X/extern/veins/src'
make[1]: se sale del directorio '/home/prueba/opencv2x/OpenCV2X/extern/veins'
[1]- Hecho          cd extern/veins
[2]+ Salida 127    ./configure
prueba@ggglecwrk241:~/opencv2x/OpenCV2X$
```

Figura 18. Instalación de veins

### 3.7.3 Compilación Modular de Componentes

Para casos donde se requiere mayor control sobre el proceso de compilación o cuando surgen problemas con la compilación automática, es posible compilar cada componente individualmente. Esta aproximación permite identificar y resolver problemas específicos con componentes individuales.

La secuencia:

```
$ make inet, make vanetza, make simulte
```

Compila cada componente en orden. Es crucial que INET se compile antes que SimuLTE debido a las dependencias entre estos componentes. Vanetza puede compilarse independientemente ya que no tiene dependencias críticas con los otros componentes.

## 3.8 Configuración del Sistema de Construcción CMake

### 3.8.1 Inicialización del Entorno de Construcción

CMake actúa como el sistema de construcción principal para Artery y OpenCV2X, gestionando las dependencias complejas y las configuraciones de compilación. El proceso comienza con la creación de un directorio de construcción separado para mantener organizados los archivos generados.

La creación del directorio con:

```
$ mkdir build
```

Y la navegación con:

```
$ cd build/
```

Establecen el entorno de trabajo.

Posteriormente con:

```
$ cmake ..
```

Ejecuta la configuración inicial, analizando el proyecto y generando los archivos de construcción necesarios. Durante este proceso, CMake verifica la disponibilidad de todas las dependencias y configura los parámetros de compilación apropiados.

```
prueba@gglegcwrk241: ~/opencv2x/OpenCV2X/build
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Failed
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
-- Found CryptoPP: /usr/include (found suitable version "5.6.4", minimum required is "5.6.4")
-- SimulTE integration disabled
-- Found Boost: /usr/lib/x86_64-linux-gnu/cmake/Boost-1.71.0/BoostConfig.cmake (found suitable version "1.71.0", minimum required is "1.59") found components: date_time system
-- Enable Veins integration
-- Enable INET integration
-- Found PythonLibs: /usr/lib/x86_64-linux-gnu/libpython3.8.so (found version "3.8.10")
-- Found PythonInterp: /usr/bin/python3 (found version "3.8.10")
-- Found PythonLibs: /usr/lib/x86_64-linux-gnu/libpython3.8.so
-- pybind11 v2.2.4
-- Configuring done
-- Generating done
-- Build files have been written to: /home/prueba/opencv2x/OpenCV2X/build
prueba@gglegcwrk241:~/opencv2x/OpenCV2X/build$
```

Figura 19. Configuración del proyecto principal

### 3.8.2 Habilitación de Características Específicas

La configuración de OpenCV2X requiere la habilitación explícita de la opción WITH\_SIMULTE para integrar las capacidades de comunicación celular. Esta configuración se realiza utilizando:

```
$ ccmake .
```

El cual es una interfaz de texto para la configuración de CMake.

La ejecución de ccmake . abre la interfaz de configuración donde el usuario puede navegar hasta la opción WITH\_SIMULTE y cambiar su valor a ON. El proceso involucra presionar 'c' para configurar los cambios y 'g' para generar los archivos de construcción actualizados. La interfaz mostrará todas las opciones de configuración disponibles y permitirá su modificación según las necesidades específicas del proyecto.

```
prueba@gglecwrk241: ~/opencv2x/OpenCV2X/build
Page 2 of 2
PYBIND11_TEST OFF
RUN_FLAGS
SCENARIO_CONFIG Base
SUMO_LAUNCHD_ARGUMENTS --daemon --kill
USE_PYTHON_INCLUDE_DIR OFF
VALGRIND_COMMAND VALGRIND_COMMAND-NOTFOUND
Vaneza_DIR /home/prueba/opencv2x/OpenCV2X/extern/vaneza
Veins_DIR /home/prueba/opencv2x/OpenCV2X/extern/veins
WITH_ENVMOD ON
WITH_SIMULTE ON
WITH_STORYBOARD ON
WITH_TESTBED OFF
WITH_TRANSFUSION OFF

WITH_SIMULTE: Build Artery with SimuLTE integration
Press [enter] to edit option Press [d] to delete an entry CMake Version 3.16.3
Press [c] to configure
Press [h] for help Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently off)
```

Figura 20. Habilitación de simulte

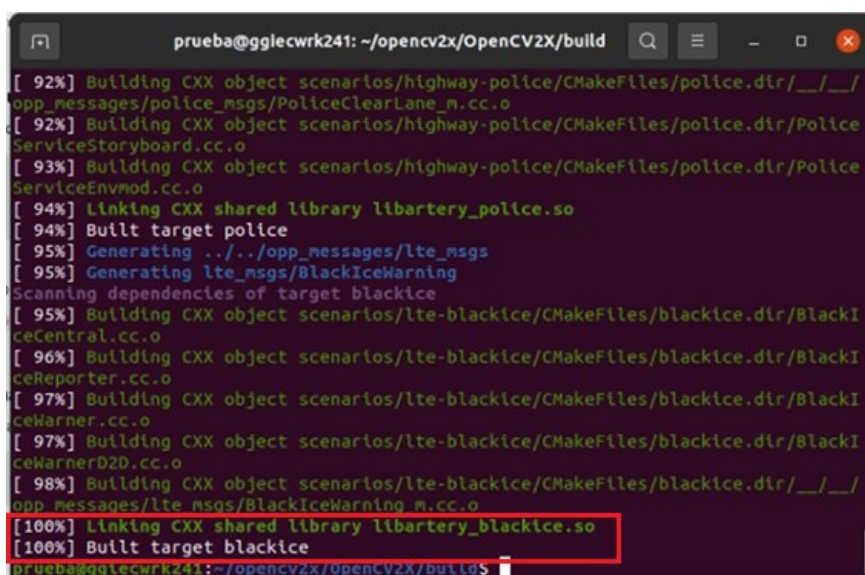
### 3.8.3 Construcción Final del Proyecto

Una vez completada la configuración, la construcción final se realiza mediante

```
$ cmake --build .
```

El cual utiliza el sistema de construcción configurado para compilar todo el proyecto. Este proceso integra todos los componentes previamente compilados y genera los ejecutables finales necesarios para ejecutar las simulaciones.

La consola mostrará el progreso de la construcción, incluyendo información sobre archivos que están siendo enlazados y bibliotecas que están siendo generadas. Un proceso exitoso resultará en la generación de todos los targets de construcción sin errores críticos.



```
prueba@ggglecwrk241: ~/opencv2x/OpenCV2X/build
[ 92%] Building CXX object scenarios/highway-police/CMakeFiles/police.dir/___/
opp_messages/police_msgs/PoliceClearLane_n.cc.o
[ 92%] Building CXX object scenarios/highway-police/CMakeFiles/police.dir/Police
ServiceStoryboard.cc.o
[ 93%] Building CXX object scenarios/highway-police/CMakeFiles/police.dir/Police
ServiceEnvmod.cc.o
[ 94%] Linking CXX shared library libartery_police.so
[ 94%] Built target police
[ 95%] Generating ../../opp_messages/lte_msgs
[ 95%] Generating lte_msgs/BlackIceWarning
Scanning dependencies of target blackice
[ 95%] Building CXX object scenarios/lte-blackice/CMakeFiles/blackice.dir/BlackI
ceCentral.cc.o
[ 96%] Building CXX object scenarios/lte-blackice/CMakeFiles/blackice.dir/BlackI
ceReporter.cc.o
[ 97%] Building CXX object scenarios/lte-blackice/CMakeFiles/blackice.dir/BlackI
ceWarner.cc.o
[ 97%] Building CXX object scenarios/lte-blackice/CMakeFiles/blackice.dir/BlackI
ceWarnerD2D.cc.o
[ 98%] Building CXX object scenarios/lte-blackice/CMakeFiles/blackice.dir/___/
opp_messages/lte_msgs/BlackIceWarning_n.cc.o
[100%] Linking CXX shared library libartery_blackice.so
[100%] Built target blackice
prueba@ggglecwrk241:~/opencv2x/OpenCV2X/build$
```

Figura 21. Compilación completa del proyecto general

### 3.9 Verificación Integral de la Instalación

#### 3.9.1 Ejecución del Escenario de Validación

La verificación de que toda la instalación se ha completado correctamente se realiza mediante la ejecución de un escenario de simulación predeterminado. El comando:

```
$ cmake --build build --target run_simulte-cars
```

Ejecuta el escenario de prueba "simulte-cars" que incluye simulaciones de comunicaciones vehiculares utilizando tecnología LTE.

Este escenario de prueba está diseñado para ejercitar todos los componentes principales del sistema, incluyendo la simulación de movilidad vehicular con SUMO, las comunicaciones de red con OMNeT++, y las características específicas de LTE con SimuLTE. La ejecución exitosa de este escenario confirma que toda la cadena de herramientas está funcionando correctamente.

```
prueba@gglecwrk241: ~/opencv2x/OpenCV2X
Loading NED files from /home/prueba/opencv2x/OpenCV2X/extern/veins/examples/veins: 1
Loading NED files from /home/prueba/opencv2x/OpenCV2X/extern/veins/src/veins: 38
Loading NED files from /home/prueba/opencv2x/OpenCV2X/extern/inet/src: 666
Loading NED files from /home/prueba/opencv2x/OpenCV2X/extern/inet/examples: 174
Loading NED files from /home/prueba/opencv2x/OpenCV2X/extern/inet/tutorials: 19
Loading NED files from /home/prueba/opencv2x/OpenCV2X/extern/inet/showcases: 33
Loading NED files from /home/prueba/opencv2x/OpenCV2X/extern/simulte/simulations: 13
Loading NED files from /home/prueba/opencv2x/OpenCV2X/extern/simulte/src: 47

Preparing for running configuration Base, run #0...
Assigned runID=Base-0-20250629-07:51:33-363163
Setting up network "artery.lte.Mode4World"...
Initializing...

Running simulation...
** Event #0 t=0 Elapsed: 1e-05s (0m 00s) 0% completed (0% total)
Speed: ev/sec=0 simsec/sec=0 ev/simsec=0
Messages: created: 3 present: 3 in FES: 2
Warning: Environment variable SUMO_HOME is not set properly, disabling XML validation. Set 'auto' or 'always' for web lookups.
```

Figura 22. Simulación en marcha

### 3.9.2 Análisis de Resultados Generados

Una simulación exitosa debe generar varios archivos de resultados que contienen datos sobre el comportamiento de la red, estadísticas de comunicación y métricas de rendimiento. Estos archivos se almacenan típicamente en el directorio `scenarios/cars/results/` y pueden incluir archivos con extensiones, `.vec` (vectores) y `.vci` (índices de vectores).

La verificación de estos archivos mediante:

```
$ ls -la scenarios/cars/results/
```

Y debe mostrar la presencia de archivos de resultados con timestamps recientes, confirmando que la simulación se ejecutó completamente y generó los datos esperados.

```
prueba@ggilecwrk241: ~/opencv2x/OpenCV2X/scenarios/cars...
prueba@ggilecwrk241:~/opencv2x$ cd prueba/
prueba@ggilecwrk241:~/opencv2x/prueba$ git clone https://github.com/brianmc95/OpenCV2X.git
Clonando en 'OpenCV2X'...
remote: Enumerating objects: 35193, done.
remote: Counting objects: 100% (2668/2668), done.
remote: Compressing objects: 100% (434/434), done.
remote: Total 35193 (delta 2305), reused 2234 (delta 2234), pack-reused 32525 (from 1)
Recibiendo objetos: 100% (35193/35193), 133.30 MiB | 13.41 MiB/s, listo.
Resolviendo deltas: 100% (23133/23133), listo.
Actualizando archivos: 100% (1698/1698), listo.
prueba@ggilecwrk241:~/opencv2x/prueba$ cd ..
prueba@ggilecwrk241:~/opencv2x$ cd OpenCV2X/
prueba@ggilecwrk241:~/opencv2x/OpenCV2X$ cd escenarios/
prueba@ggilecwrk241:~/opencv2x/OpenCV2X/scenarios$ cd cars/
prueba@ggilecwrk241:~/opencv2x/OpenCV2X/scenarios/cars$ cd results/
prueba@ggilecwrk241:~/opencv2x/OpenCV2X/scenarios/cars/results$ ls
Base
prueba@ggilecwrk241:~/opencv2x/OpenCV2X/scenarios/cars/results$ cd Base/
prueba@ggilecwrk241:~/opencv2x/OpenCV2X/scenarios/cars/results/Base$ ls
Base-0-20250629-07:51:33-363163.sca Base-0-20250629-07:51:33-363163.vec
Base-0-20250629-07:51:33-363163.vci
prueba@ggilecwrk241:~/opencv2x/OpenCV2X/scenarios/cars/results/Base$
```

Figura 23. Resultados listados

### 3.10 Diagnóstico y Resolución de Problemas Comunes

Durante el proceso de instalación de un sistema tan complejo como OpenCV2X con Artery, es común encontrar diversos tipos de problemas que pueden requerir intervención manual para su resolución. La experiencia acumulada en múltiples instalaciones ha identificado varios patrones de problemas recurrentes junto con sus soluciones correspondientes.

En caso de que el comando make all cause errores, es preferible instalar modulo por modulo usando los comandos make nombre\_modulo, reemplazando esta última sección por el nombre del módulo a instalar.

Los errores relacionados con las bibliotecas Boost constituyen uno de los problemas más frecuentes, típicamente manifestándose como errores de enlace durante la compilación o errores de biblioteca no encontrada durante la ejecución. La solución generalmente involucra la reinstalación completa de las bibliotecas Boost mediante:

```
$ sudo apt install -y libboost-all-dev
```

Seguido por:

```
$ sudo ldconfig
```

Para actualizar la caché de bibliotecas del sistema.

Los problemas con bibliotecas criptográficas, particularmente Crypto++, pueden manifestarse como errores de compilación en componentes de Vanetza relacionados con funcionalidades de seguridad. Estos errores frecuentemente indican que la biblioteca no está instalada o que se está utilizando una versión incompatible. La solución involucra la instalación o reinstalación de las bibliotecas criptográficas mediante `sudo apt install -y libcrypto++-dev libssl-dev` y la verificación de que las versiones instaladas son compatibles con los requisitos del proyecto. En caso de incurrir en el error 1

```
CMake Error at /usr/share/cmake-3.16/Modules/FindPackageHandleStandardArgs.cmake:146 (message):
  Could NOT find CryptoPP: Found unsuitable version "..", but required is at least "5.6.1" (found /usr/local/include)
Call Stack (most recent call first):
  /usr/share/cmake-3.16/Modules/FindPackageHandleStandardArgs.cmake:391 (_FPHSA_FAILURE_MESSAGE)
  cmake/FindCryptoPP.cmake:11 (find_package_handle_standard_args)
  CMakeLists.txt:52 (find_package)

-- Configuring incomplete, errors occurred!
See also "/home/prueba/opencv2x/OpenCV2X/extern/vanetza/build/CMakeFiles/CMakeOutput.log".
See also "/home/prueba/opencv2x/OpenCV2X/extern/vanetza/build/CMakeFiles/CMakeError.log".
make: *** [Makefile:44: extern/vanetza/build/Makefile] Error 1
prueba@ggglecwrk241:~/opencv2x/OpenCV2X$
```

Figura 24. Error con cryptopp

Se debe borrar manualmente las carpetas de la librería cryptopp lo cual haremos con los siguientes comandos:

```
$ sudo rm -rf /usr/local/include/cryptopp
$ sudo rm -rf /usr/local/lib/cryptopp*
$ sudo ldconfig
```

Esto nos dejara un entorno limpio para volver a instalar la librería de manera correcta con los comandos:

```
$ sudo apt install --reinstall libcrypto++-dev
```

Luego de seguir estos pasos podremos continuar con nuestra instalación.

Los errores de compilación de OMNeT++ frecuentemente están relacionados con la presencia de bibliotecas osgEarth conflictivas en el sistema o con la ausencia de bibliotecas Qt5 necesarias para los componentes gráficos. Los problemas gráficos se manifiestan como errores de compilación en componentes de interfaz de usuario y se resuelven instalando las bibliotecas Qt5 de desarrollo mediante `sudo apt install -y qt5-default libqt5opengl5-dev qttools5-dev`. Los problemas con osgEarth se resuelven asegurando que la configuración incluya explícitamente las opciones:

```
$ ./configure WITH_OSGEARTH=no WITH_OSG=no
```

Los problemas de permisos pueden surgir cuando ciertos directorios o archivos no tienen los permisos apropiados para el usuario actual. La solución típica involucra la corrección de la propiedad de archivos mediante:

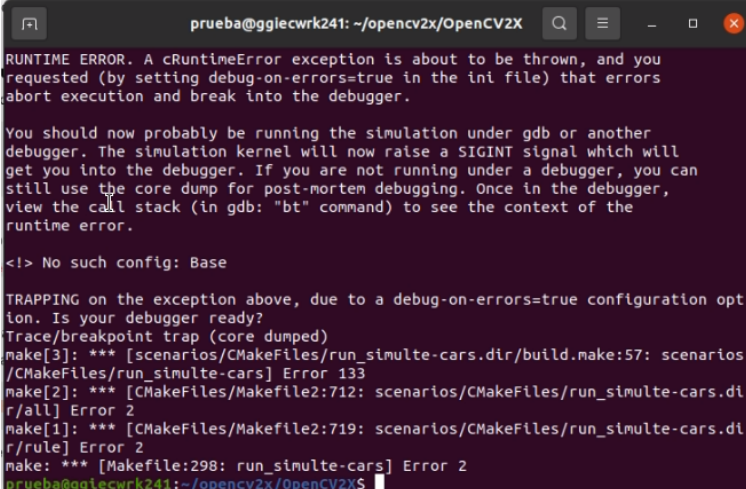
```
$ sudo chown -R $USER:$USER /opt/omnetpp-5.7
```

Con el establecimiento de permisos de ejecución apropiados con:

```
$ chmod +x /opt/omnetpp-5.7/bin/*
```

Luego de instalar todo correctamente es probable que no encontremos con el error 2 al ejecutar el comando de pruebas:

```
$ cmake --build build --target run_simulte-cars
```



```
prueba@gglecwrk241: ~/opencv2x/OpenCV2X
RUNTIME ERROR. A cRuntimeError exception is about to be thrown, and you
requested (by setting debug-on-errors=true in the ini file) that errors
abort execution and break into the debugger.

You should now probably be running the simulation under gdb or another
debugger. The simulation kernel will now raise a SIGINT signal which will
get you into the debugger. If you are not running under a debugger, you can
still use the core dump for post-mortem debugging. Once in the debugger,
view the call stack (in gdb: "bt" command) to see the context of the
runtime error.

<|> No such config: Base

TRAPPING on the exception above, due to a debug-on-errors=true configuration opt
ion. Is your debugger ready?
Trace/breakpoint trap (core dumped)
make[3]: *** [scenarios/CMakeFiles/run_simulte-cars.dir/build.make:57: scenarios
/CMakeFiles/run_simulte-cars] Error 133
make[2]: *** [CMakeFiles/Makefile2:712: scenarios/CMakeFiles/run_simulte-cars.dl
r/all] Error 2
make[1]: *** [CMakeFiles/Makefile2:719: scenarios/CMakeFiles/run_simulte-cars.dl
r/rule] Error 2
make: *** [Makefile:298: run_simulte-cars] Error 2
prueba@gglecwrk241:~/opencv2x/OpenCV2X$
```

Figura 25. Error al ejecutar la simulación

Para solucionar este error tendremos que modificar el archivo .ini que se encuentra en nuestro directorio raíz Opencv2x/scenarios/cars, el archivo a modificar se llama omnetpp.ini, al inicio de este cambiaremos el parámetro debugs-on-error a off y buscaremos en nuestro código una sección que tenga este encabezado [config highwayFast] el cual cambiaremos a [config Base].

### 3.11 Creación de un escenario con SUMO

Para poder realizar una simulación utilizando opencv2x y omnetpp primero debemos crear un escenario en SUMO para esto se tiene que seguir los pasos listados a continuación:

Abrimos la terminal y usamos el comando:

```
$ netedit
```

Esto hará que se abra la interfaz de la herramienta Netedit.

Procedemos a crear una nueva red desde la pestaña File como se muestra en la Figura 26.

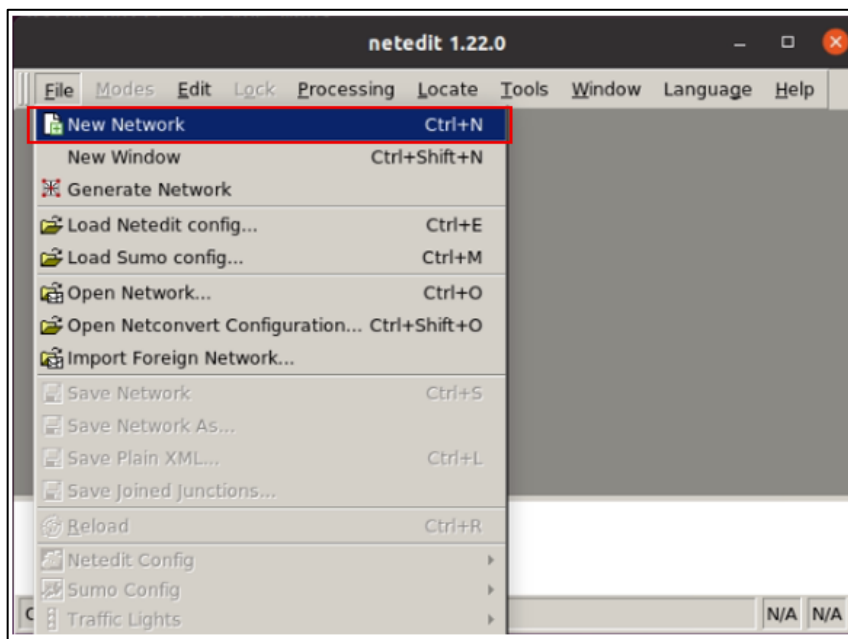
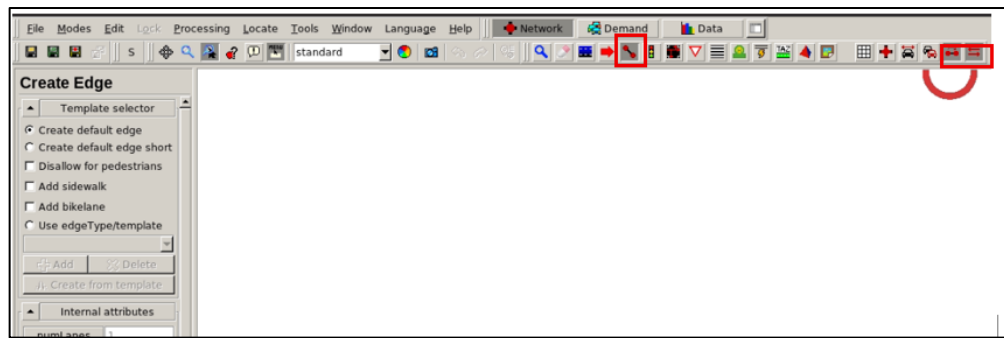


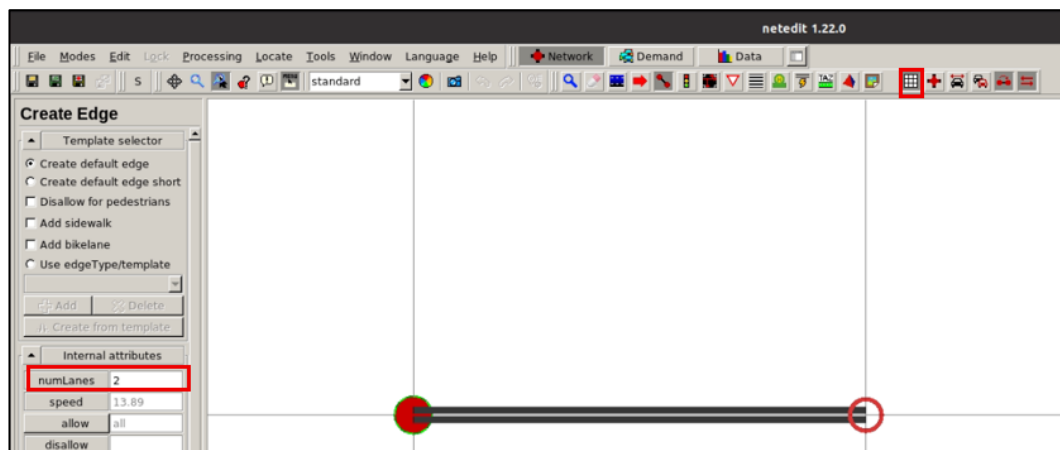
Figura 26. creación de la red desde Netedit

Una vez que se abrió nuestra nueva interfaz procedemos activando el Edge mode, el edge chain y el edge opposite direction como se muestra en la figura 26



*Figura 27. Activando herramientas edge*

Luego de esto activamos el grid y asignamos el numLanes en 2 lo que indicaría que tendríamos dos carriles por cada sentido en nuestra carretera, a su vez procedemos a darle forma a nuestro escenario haciendo click desde el punto donde queremos que inicie hasta el final como se muestra en la Figura 28.



*Figura 28. Diseñando escenario*

Una vez que le damos la forma deseada a nuestro escenario cambiamos la pestaña a Demand y activamos el route mode como se muestra en la Figura 29.

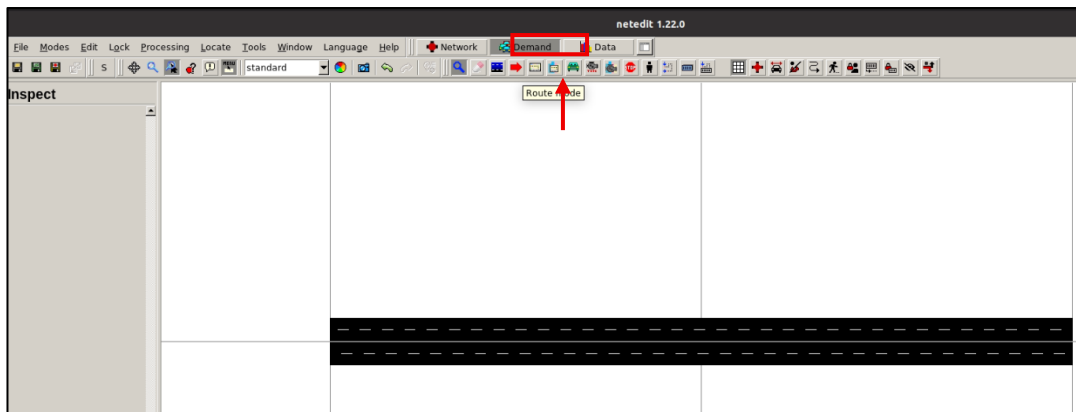


Figura 29. Route mode

Procedemos a crear la ruta para nuestro escenario como se muestra en la Figura 30.

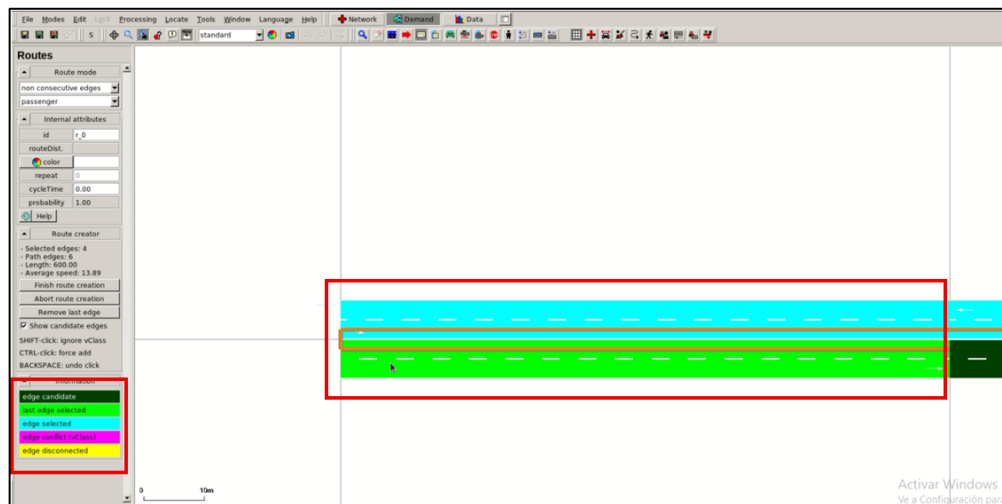


Figura 30. Diseño de ruta

Una vez se tenga el diseño de la ruta se procede a agregar los vehículos para esto accedemos al vehicle mode y elegimos el modo vehicle over route como se muestra en la figura 31, de allí solo se agrega el vehículo sobre la línea amarilla y se procede a guardar los archivos.

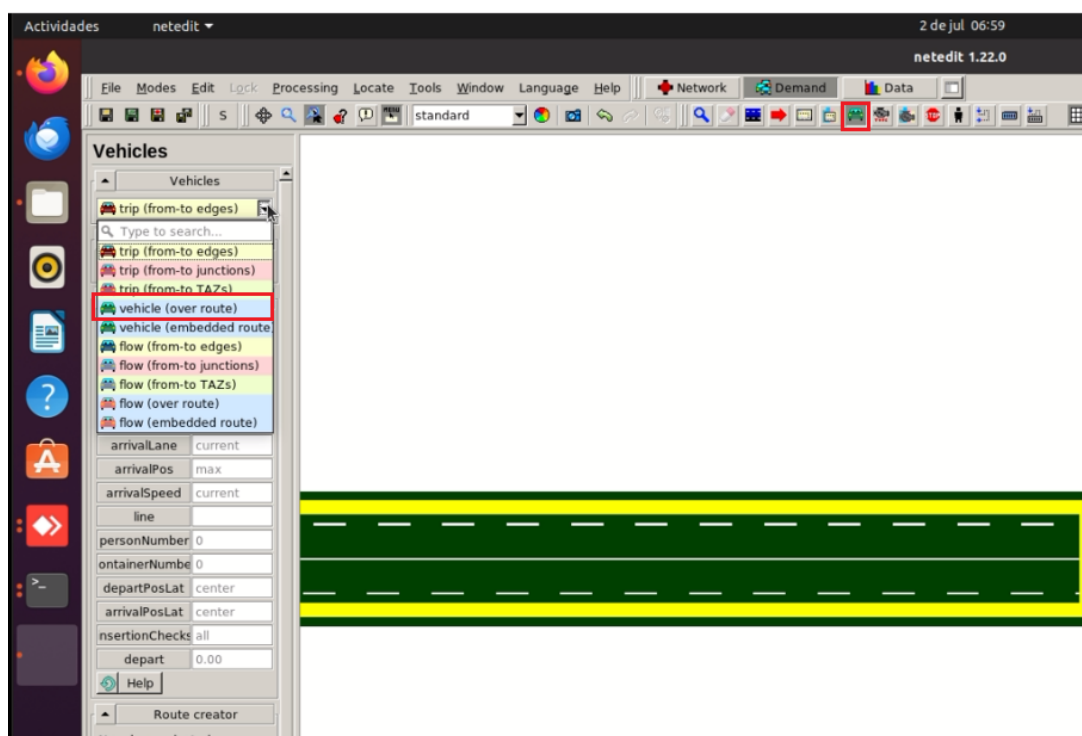


Figura 31. selección de vehículo

### 3.11.1 integración de escenario sumo con opencv2x

Para poder integrar el escenario que creamos es necesario acceder a nuestro directorio raíz, de allí a escenarios y creamos una carpeta con el nombre de nuestra simulación en este caso se utilizó el nombre ruta.

Tenemos que asegurarnos de guardar 3 archivos que provienen de nuestro escenario de sumo, cuyas extensiones son .sumocfg .rou.xml .net.xml como se muestra en la figura 32, estos deben guardarse en el siguiente directorio Opencv2x/scenarios/ruta



Figura 32. Archivos de sumo

Luego de esto procedemos a crear el archivo CmakeLists.txt desde consola utilizando el comando desde el directorio de nuestra simulación:

```
$ gedit
```

Donde escribiremos el siguiente código:

```
$ add_opp_run(ruta CONFIG omnetpp.ini)
```

Para la simulación de comunicaciones vehiculares utilizaremos un archivo omnetpp.ini preconfigurado del proyecto cars, que implementa las especificaciones 3GPP LTE Mode 4 D2D para comunicaciones V2X. También emplearemos el archivo services.xml para definir los servicios de aplicación.

*Tabla 3. Configuración General del Sistema*

Parámetro	Valor	Descripción
network	artery.lte.Mode4World	Red de simulación para LTE Mode 4
sim-time-limit	512s	Duración de la simulación
ser-set	0	Semilla para generador de números aleatorios
repeat	1	Número de repeticiones
playgroundSizeX/Y	2000 m	Dimensiones del área de simulación
playgroundSizeZ	50 m	Altura del área de simulación

*Tabla 4. Configuración del Canal de Radio*

Parámetro	Valor	Descripción
channelControl.pMax	10W	Potencia máxima de transmisión
channelControl.carrierFrequency	6000e+6Hz	Frecuencia portadora (6 GHz)
channelControl.alpha	1.0	Factor de pérdida de trayectoria
ueTxPower	23	Potencia de transmisión del UE (dBm)

El siguiente paso es modificar el archivo CmakeList.txt que se encuentra en el directorio Opencv2x/scenarios y agregar la siguiente línea de código en el inicio:

```
$ add_subdirectory(ruta)
```

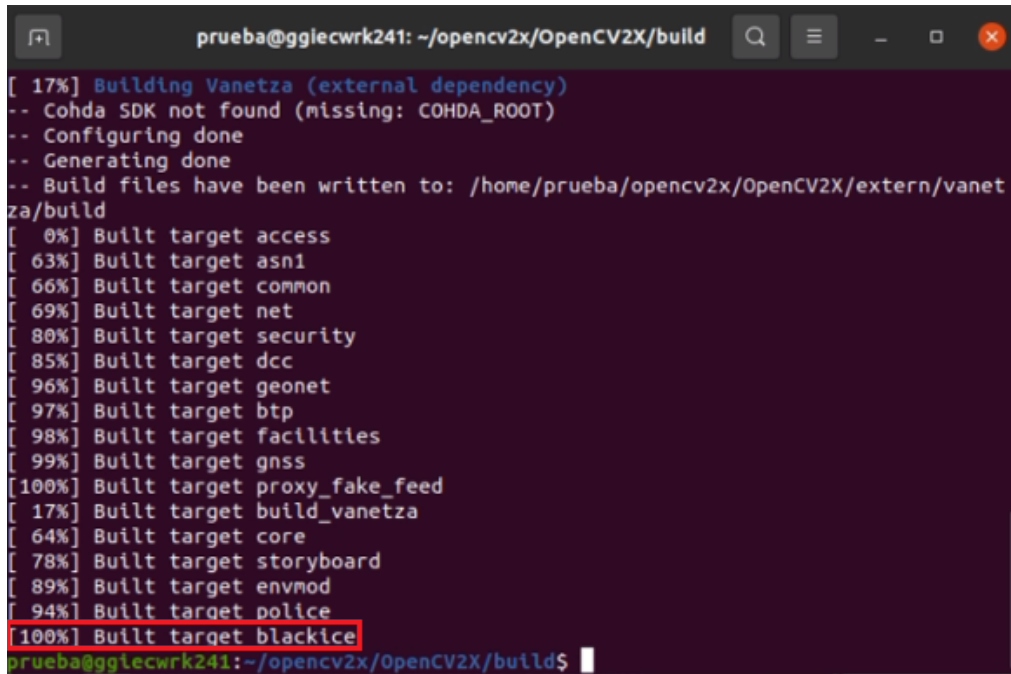
Por último, solo queda volver a compilar nuestra carpeta build, para esto desde el terminal en el directorio opencv2x/build ejecutamos:

```
$ cmake ..
```

Esto hará que se configure el proyecto incluyendo nuestra simulación, ahora queda compilarlo usando:

```
$ cmake --build .
```

Si todo se hizo correctamente veremos una salida en consola como se muestra en la Figura 33.



```
prueba@ggtecwrk241: ~/opencv2x/OpenCV2X/build
[ 17%] Building Vanetza (external dependency)
-- Cohda SDK not found (missing: COHDA_ROOT)
-- Configuring done
-- Generating done
-- Build files have been written to: /home/prueba/opencv2x/OpenCV2X/extern/vanetza/build
[  0%] Built target access
[ 63%] Built target asn1
[ 66%] Built target common
[ 69%] Built target net
[ 80%] Built target security
[ 85%] Built target dcc
[ 96%] Built target geonet
[ 97%] Built target btp
[ 98%] Built target facillites
[ 99%] Built target gnss
[100%] Built target proxy_fake_feed
[ 17%] Built target build_vanetza
[ 64%] Built target core
[ 78%] Built target storyboard
[ 89%] Built target envnod
[ 94%] Built target police
[100%] Built target blackice
prueba@ggtecwrk241:~/opencv2x/OpenCV2X/build$
```

Figura 33. Proyecto compilado.

# CAPÍTULO IV

## 4.1. Introducción

Los hallazgos adquiridos a través de las simulaciones efectuadas con el marco OpenCV2X, que se integra con OMNeT++ y SUMO, son presentados en este capítulo. Las simulaciones se llevaron a cabo de acuerdo con la configuración que se describe en el Capítulo III, utilizando el modo sidelink LTE-V2X Mode 4, de acuerdo con las especificaciones del 3GPP. Los resultados se examinan en función de las principales métricas de rendimiento para la comunicación entre vehículos: Distribución de paquetes exitosos frente a los perdidos (Packet Delivery Ratio, PDR) y el desempeño del sistema en distintas situaciones de densidad vehicular.

Se llevó a cabo la evaluación teniendo en cuenta un escenario de autopista con varios vehículos que intercambiaban mensajes de aplicaciones de seguridad vial, lo que imitaba situaciones reales de comunicación entre vehículos. Los datos recolectados hacen posible describir el comportamiento del sistema y verificar la aplicación del marco para estudios venideros en comunicaciones vehiculares C-V2X.

## 4.2. Escenario de Simulación

El escenario de simulación implementado corresponde a una autopista de cuatro carriles (dos por sentido) con tráfico vehicular bidireccional. La Figura 34 muestra la visualización del escenario en SUMO, donde se observan los vehículos (representados en color cian y amarillo según su dirección) circulando por la vía. El escenario fue diseñado utilizando la herramienta Netedit de SUMO, siguiendo la metodología descrita en la sección 3.11.

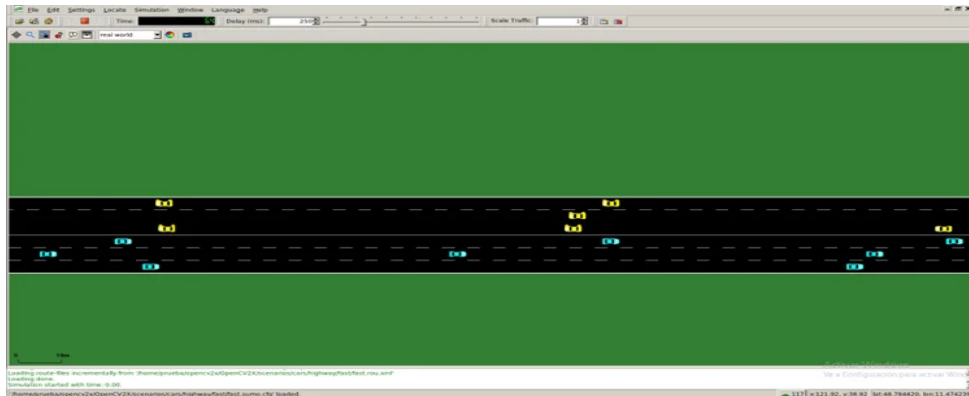


Figura 34. Escenario de simulación en SUMO con tráfico vehicular bidireccional.

La Tabla 5 resume los parámetros de configuración utilizados en la simulación, los cuales fueron establecidos conforme a las especificaciones del estándar 3GPP para comunicaciones LTE-V2X Mode 4.

Tabla 5. Parámetros de Simulación Utilizados

Parámetro	Valor	Descripción
Tiempo de simulación	512 s	Duración total de la simulación
Área de simulación	2000 x 2000 m	Dimensiones del playground
Frecuencia portadora	6 GHz	Banda de frecuencia C-V2X
Potencia de transmisión UE	23 dBm	Potencia del equipo de usuario
Potencia máxima canal	10 W	Potencia máxima del control de canal
Factor de pérdida ( $\alpha$ )	1.0	Exponente de pérdida de trayectoria
Red de simulación	artery.lte.Mode4World	Modelo de red LTE Mode 4
Número de carriles	4 (2 por sentido)	Configuración de autopista

### 4.3. Análisis de Resultados

#### 4.3.1. Distribución de Paquetes: Perdidos vs Exitosos por Distancia

La Figura 35 presenta la distribución porcentual de paquetes exitosos (verde) versus paquetes perdidos (rojo) en función de la distancia entre transmisor y receptor. Esta métrica es fundamental para caracterizar el alcance efectivo de las comunicaciones C-V2X y la confiabilidad del enlace sidelink.

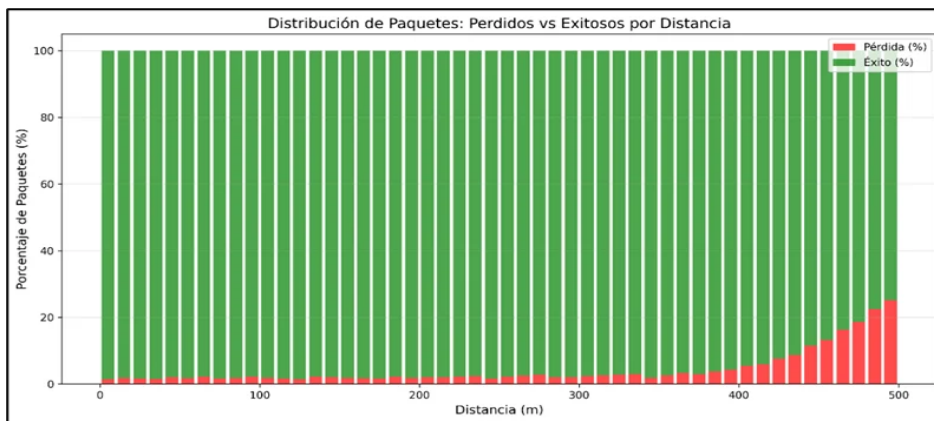


Figura 35 Distribución de paquetes: Perdidos vs Exitosos por Distancia.

Los resultados muestran un comportamiento altamente favorable para distancias de comunicación hasta aproximadamente 400 metros, donde la tasa de éxito se mantiene cercana al 100%. A partir de los 400 metros, se observa un incremento gradual en la pérdida de paquetes, alcanzando aproximadamente un 25% de pérdida a los 500 metros. Este comportamiento es consistente con los modelos de propagación en entornos vehiculares y las características del canal sidelink LTE-V2X.

La degradación observada a distancias mayores puede atribuirse a múltiples factores: la atenuación por distancia según el modelo de pérdida de trayectoria configurado ( $\alpha = 1.0$ ), las condiciones de propagación del canal, y posibles colisiones en el acceso al medio cuando múltiples vehículos transmiten simultáneamente. La Tabla 6 resume los resultados cuantitativos observados en diferentes rangos de distancia.

Tabla 6. Tasa de Éxito de Paquetes por Rango de Distancia

Rango de Distancia	Tasa de Éxito (%)	Tasa de Pérdida (%)
0 - 100 m	≈ 99.5%	≈ 0.5%
100 - 200 m	≈ 98.5%	≈ 1.5%
200 - 300 m	≈ 97%	≈ 3%
300 - 400 m	≈ 95%	≈ 5%
400 - 450 m	≈ 88%	≈ 12%
450 - 500 m	≈ 75%	≈ 25%

### 4.3.2. Packet Delivery Ratio bajo Diferentes Condiciones de Densidad

La Figura 36 presenta la comparación del Packet Delivery Ratio (PDR) en función de la distancia para tres escenarios de densidad vehicular: baja (Low), media (Medium) y congestionada (Congested). Esta métrica representa el porcentaje de paquetes que son recibidos exitosamente respecto al total de paquetes transmitidos.

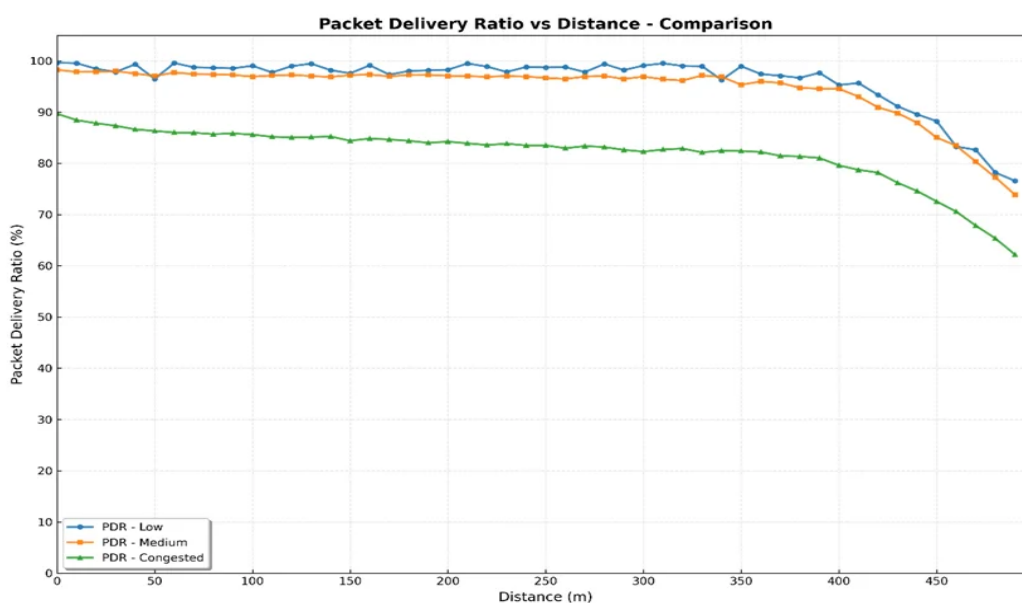


Figura 36. Packet Delivery Ratio vs Distancia bajo diferentes condiciones de densidad vehicular.

Los resultados revelan patrones significativos en el comportamiento del PDR según la densidad vehicular:

- **Baja densidad (PDR - baja):** Hasta los 350 metros, aproximadamente, muestra el mejor desempeño, conservando un PDR que supera el 95%. La curva presenta un deterioro progresivo, llegando a cerca del 75 % a los 500 metros. Esta conducta es un reflejo de las condiciones óptimas en las que la interferencia entre automóviles es mínima.
- **Densidad Media (PDR - Medium):** Muestra un desempeño algo más bajo que el escenario de baja densidad, con un PDR inicial próximo al 98%, que disminuye hasta alrededor del 73% a los 500 metros. A distancias superiores a los 350 metros, la diferencia con el escenario de baja densidad se vuelve más notoria.
- **Densidad Congestionada (PDR - Congested):** Demuestra el efecto más importante que tiene la densidad vehicular en el desempeño. El PDR inicial está en torno al 90%, un porcentaje significativamente menor que el de los demás escenarios, y disminuye hasta cerca del 62% a los 500 metros. Las colisiones en el acceso a los medios y la interferencia entre las transmisiones simultáneas son las causas principales de esta degradación.

*Tabla 7. Comparación del PDR por Escenario de Densidad*

<b>Distancia</b>	<b>PDR Low (%)</b>	<b>PDR Medium (%)</b>	<b>PDR Congested (%)</b>
0 m	≈ 100	≈ 98	≈ 90
100 m	≈ 98	≈ 97	≈ 86
200 m	≈ 99	≈ 97	≈ 84
300 m	≈ 98	≈ 97	≈ 82
400 m	≈ 92	≈ 94	≈ 80
500 m	≈ 75	≈ 73	≈ 62

## 4.4. Discusión de Resultados

Los resultados obtenidos demuestran que la implementación del framework OpenCV2X con el modo LTE-V2X Mode 4 proporciona un rendimiento consistente con las especificaciones del estándar 3GPP y con resultados reportados en la literatura científica. El análisis de los datos permite extraer las siguientes observaciones clave:

**Alcance Efectivo de Comunicación:** El sistema demuestra una alta confiabilidad (>95% PDR) para distancias de hasta 350-400 metros en condiciones de baja y media densidad vehicular. Este alcance es adecuado para la mayoría de aplicaciones de seguridad vial que requieren el intercambio de mensajes de tipo Cooperative Awareness Messages (CAM) entre vehículos cercanos.

**Impacto de la Densidad Vehicular:** La comparación de escenarios muestra que el aumento en la densidad vehicular tiene un impacto importante en el desempeño del sistema. En el contexto congestionado, se aprecia que el PDR inicial disminuye alrededor de un 10% en comparación con el escenario de baja densidad. Esta degradación se debe al sistema de acceso del modo 4, en el que la selección autónoma de recursos puede provocar colisiones si varios vehículos escogen los mismos subcanales.

**Comportamiento a Distancias Largas:** En todos los casos, la degradación del PDR es más significativa a distancias mayores a 400 metros. Se prevé esta conducta a causa de la combinación de elementos: una atenuación más elevada por distancia, un índice de señal a ruido (SNR) en el receptor más bajo y una probabilidad superior de que interfieran otras transmisiones.

**Validación del Framework:** Los resultados obtenidos validan la correcta implementación del framework OpenCV2X para la evaluación de comunicaciones C-V2X. Las tendencias observadas son coherentes con estudios previos como los reportados por Molina-Masegosa y Gozalvez [53], McCarthy et al. [22], y los análisis de rendimiento del LTE-V2X Mode 4 documentados en la literatura [32].

## 4.5. Comparación con Trabajos Relacionados

Los resultados obtenidos en este trabajo son consistentes con los reportados en investigaciones previas sobre evaluación de rendimiento de LTE-V2X Mode 4. La Tabla 8 presenta una comparación con estudios similares encontrados en la literatura.

*Tabla 8. Comparación con Trabajos Relacionados*

<b>Estudio</b>	<b>Simulador</b>	<b>PDR @ 300m</b>	<b>PDR @ 500m</b>
Este trabajo	OpenCV2X	≈ 97%	≈ 73-75%
Molina-Masegosa [53]	MATLAB	≈ 95-98%	≈ 70-80%
McCarthy et al. [22]	OpenCV2X	≈ 96%	≈ 72%
Todisco et al. [32]	WiLabV2Xsim	≈ 94-97%	≈ 68-75%

Como se observa en la tabla, los resultados obtenidos se encuentran dentro del rango reportado por investigaciones similares, lo cual valida tanto la metodología empleada como la correcta configuración del entorno de simulación. Las pequeñas variaciones observadas pueden atribuirse a diferencias en los parámetros específicos de simulación, modelos de canal utilizados, y densidades vehiculares consideradas.

## 4.6. Conclusiones del Capítulo

Los resultados presentados en este capítulo demuestran la viabilidad y efectividad del framework OpenCV2X para la evaluación de comunicaciones vehiculares C-V2X. Las principales conclusiones derivadas del análisis son:

- En situaciones de densidad vehicular baja a media, el sistema LTE-V2X Mode 4 garantiza un rendimiento fiable ( $PDR > 95\%$ ) para la comunicación entre vehículos en distancias que pueden llegar hasta los 400 metros.
- La densidad vehicular influye notablemente en el desempeño del sistema; se aprecia una disminución de hasta un 10 % en el PDR inicial cuando se transita de una situación de baja densidad a otra congestionada.

- Los resultados obtenidos son consistentes con la literatura científica existente, validando la correcta implementación del framework y la metodología de evaluación empleada.
- El framework OpenCV2X, integrado con OMNeT++ y SUMO, constituye una herramienta valiosa para la investigación y desarrollo de aplicaciones de comunicaciones vehiculares basadas en la tecnología C-V2X.
- Se identifica la necesidad de explorar mecanismos de optimización de recursos y algoritmos de scheduling mejorados para mitigar el impacto de la densidad vehicular en escenarios congestionados, lo cual constituye una línea de trabajo futuro relevante.

## BIBLIOGRAFÍA

- [1] W. Xu, W. Trappe, Y. Zhang, y T. Wood, “The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks”, 2005.
- [2] G. Mandal, A. Roy, y B. Palit, “Performance Analysis of Resource Allocation Algorithms for Vehicle Platoons over 5G eV2X Communication”, nov. 2024.
- [3] K. Jellid y T. Mazri, “Check for updates Comparative Analysis of ITS-G5 and C-V2X for Autonomous Vehicles with an Improved Algorithm of C-V2X”, en *Innovations in Smart Cities Applications Volume 7: The Proceedings of the 8th International Conference on Smart City Applications, Volume 1*, 2024, p. 168.
- [4] T. Bhargavi, A. Behara, L. Bommisetty, y V. T. Gopalakrishnan, “Performance analysis of the Radio Access Technologies (RAT) of Vehicular Communications”, en *2022 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, IEEE, dic. 2022, pp. 308–313. doi: 10.1109/ANTS56424.2022.10227718.
- [5] S. Adnan Yusuf, A. Khan, y R. Souissi, “Vehicle-to-everything (V2X) in the autonomous vehicles domain – A technical review of communication, sensor, and AI technologies for road user safety”, *Transp Res Interdiscip Perspect*, vol. 23, p. 100980, 2024, doi: <https://doi.org/10.1016/j.trip.2023.100980>.
- [6] I. Soto, M. Calderon, O. Amador, y M. Urueña, “A survey on road safety and traffic efficiency vehicular applications based on C-V2X technologies”, *Vehicular Communications*, vol. 33, p. 100428, 2022, doi: <https://doi.org/10.1016/j.vehcom.2021.100428>.
- [7] L. Sadeeq y Q. Ajao, “Research on Enhancing C-V2X Communication via Danger-Aware Vehicular Networking”, oct. 2024.
- [8] G. G. M. N. Ali, M. N. Sadat, M. S. Miah, S. A. Sharief, y Y. Wang, “A Comprehensive Study and Analysis of the Third Generation Partnership Project’s 5G New Radio for Vehicle-to-Everything Communication”, *Future Internet*, vol. 16, núm. 1, p. 21, 2024.
- [9] A. Nair y S. Tanwar, “Resource allocation in V2X communication: State-of-the-art and research challenges”, *Physical Communication*, vol. 64, p. 102351, 2024, doi: <https://doi.org/10.1016/j.phycom.2024.102351>.

- [10] F. Zhang y G. Wang, “Context-aware resource allocation for vehicle-to-vehicle communications in cellular-V2X networks”, *Ad Hoc Networks*, vol. 163, p. 103582, 2024, doi: <https://doi.org/10.1016/j.adhoc.2024.103582>.
- [11] Z. Ali, S. Lagén, L. Giupponi, y R. Rouil, “3GPP NR V2X Mode 2: Overview, Models and System-Level Evaluation”, *IEEE Access*, vol. 9, pp. 89554–89579, 2021, doi: 10.1109/ACCESS.2021.3090855.
- [12] S. K. Bhoi y P. M. Khilar, “Vehicular communication: a survey”, *IET Networks*, vol. 3, núm. 3, pp. 204–217, sep. 2014, doi: 10.1049/iet-net.2013.0065.
- [13] M. Iloski, “State of the art of Cellular Vehicular Communication in 5G networks”, 2019.
- [14] Allion, “C-V2X scenario simulations and validation”.
- [15] M. Chen, R. Chai, H. Hu, W. Jiang, y L. He, “Performance evaluation of C-V2X mode 4 communications”, en *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1–6.
- [16] W. Anwar, N. Franchi, y G. Fettweis, “Physical Layer Evaluation of V2X Communications Technologies: 5G NR-V2X, LTE-V2X, IEEE 802.11bd, and IEEE 802.11p”, en *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019, pp. 1–7. doi: 10.1109/VTCFall.2019.8891313.
- [17] K. Sehla, T. M. T. Nguyen, G. Pujolle, y P. B. Velloso, “Resource allocation modes in C-V2X: from LTE-V2X to 5G-V2X”, *IEEE Internet Things J*, vol. 9, núm. 11, pp. 8291–8314, 2022.
- [18] M. M. Saad, M. T. R. Khan, S. H. A. Shah, y D. Kim, “Advancements in Vehicular Communication Technologies: C-V2X and NR-V2X Comparison”, *IEEE Communications Magazine*, vol. 59, núm. 8, pp. 107–113, 2021, doi: 10.1109/MCOM.101.2100119.
- [19] M. H. C. Garcia *et al.*, “A Tutorial on 5G NR V2X Communications”, *IEEE Communications Surveys & Tutorials*, vol. 23, núm. 3, pp. 1972–2026, 2021, doi: 10.1109/COMST.2021.3057017.
- [20] M. Hasan, S. Mohan, T. Shimizu, y H. Lu, “Securing Vehicle-to-Everything (V2X) Communication Platforms”, *IEEE Transactions on Intelligent Vehicles*, vol. 5, núm. 4, pp. 693–713, 2020, doi: 10.1109/TIV.2020.2987430.
- [21] B. McCarthy, A. Burbano-Abril, V. R. Licea, y A. O’Driscoll, “OpenCV2X: Modelling of the V2X Cellular Sidelink and Performance Evaluation for Aperiodic Traffic”, 2021.

- [22] B. McCarthy, A. Burbano-Abril, y A. O’Driscoll, “MCS Adaptation and its role in improving MAC Scheduling for Variable Packet Sizes in NR-V2X”, en *6th International Conference on Advanced Communication Technologies and Networking, CommNet 2023, Rabat, Morocco, December 11-13, 2023*, IEEE, 2023, pp. 1–7. doi: 10.1109/COMMNET60167.2023.10365302.
- [23] A. Hegde y A. Festag, “Artery-C”, en *Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, New York, NY, USA: ACM, nov. 2020, pp. 47–51. doi: 10.1145/3416010.3423240.
- [24] C. Bernad, K. Gilly, S. Filiposka, N. Thomas, P. J. Roig, y S. Alcaraz, “Comparing the Use of Simu5G and 5G-Sim-V2I/N Modules When Analysing the Edge Computing Resource Management Efficiency”, *Modelling and Simulation in Engineering*, vol. 2025, núm. 1, ene. 2025, doi: 10.1155/mse/7283539.
- [25] G. Nardini, G. Stea, y A. Viridis, “Scalable Real-Time Emulation of 5G Networks With Simu5G”, *IEEE Access*, vol. 9, pp. 148504–148520, 2021, doi: 10.1109/ACCESS.2021.3123873.
- [26] C. Bernad, K. Gilly, S. Filiposka, N. Thomas, P. J. Roig, y S. Alcaraz, “Comparing the Use of Simu5G and 5G-Sim-V2I/N Modules When Analysing the Edge Computing Resource Management Efficiency”, *Modelling and Simulation in Engineering*, vol. 2025, núm. 1, ene. 2025, doi: 10.1155/mse/7283539.
- [27] G. Nardini, G. Stea, A. Viridis, y D. Sabella, “Simu5G: A System-level Simulator for 5G Networks”, en *Proceedings of the 10th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, SCITEPRESS - Science and Technology Publications, 2020, pp. 68–80. doi: 10.5220/0009826400680080.
- [28] G. Nardini, D. Sabella, G. Stea, P. Thakkar, y A. Viridis, “Simu5G—An OMNeT++ Library for End-to-End Performance Evaluation of 5G Networks”, *IEEE Access*, vol. 8, pp. 181176–181191, 2020, doi: 10.1109/ACCESS.2020.3028550.
- [29] G. Nardini, G. Stea, y A. Viridis, “Scalable Real-Time Emulation of 5G Networks With Simu5G”, *IEEE Access*, vol. 9, pp. 148504–148520, 2021, doi: 10.1109/ACCESS.2021.3123873.
- [30] G. Nardini, D. Sabella, G. Stea, P. Thakkar, y A. Viridis, “Simu5G—An OMNeT++ Library for End-to-End Performance Evaluation of 5G

- Networks”, *IEEE Access*, vol. 8, pp. 181176–181191, 2020, doi: 10.1109/ACCESS.2020.3028550.
- [31] V. Todisco, S. Bartoletti, C. Campolo, A. Molinaro, A. O. Berthet, y A. Bazzi, “Performance Analysis of Sidelink 5G-V2X Mode 2 Through an Open-Source Simulator”, *IEEE Access*, vol. 9, pp. 145648–145661, 2021, doi: 10.1109/ACCESS.2021.3121151.
- [32] C. Campolo, V. Todisco, A. Molinaro, A. Berthet, S. Bartoletti, y A. Bazzi, “Improving Resource Allocation for beyond 5G V2X Sidelink Connectivity”, en *2021 55th Asilomar Conference on Signals, Systems, and Computers*, 2021, pp. 55–60. doi: 10.1109/IEEECONF53345.2021.9723155.
- [33] A. Bazzi, C. Campolo, A. Molinaro, A. O. Berthet, B. M. Masini, y A. Zanella, “On Wireless Blind Spots in the C-V2X Sidelink”, *IEEE Trans Veh Technol*, vol. 69, núm. 8, pp. 9239–9243, 2020, doi: 10.1109/TVT.2020.3001074.
- [34] Annu y P. Rajalakshmi, “Towards 6G V2X Sidelink: Survey of Resource Allocation—Mathematical Formulations, Challenges, and Proposed Solutions”, *IEEE Open Journal of Vehicular Technology*, vol. 5, pp. 344–383, 2024, doi: 10.1109/OJVT.2024.3368240.
- [35] CTTC, “5G-LENA module”, junio de 2025. [En línea]. Disponible en: <https://5g-lena.cttc.es/>
- [36] K. Koutlia, B. Bojovic, Z. Ali, y S. Lagén, “Calibration of the 5G-LENA system level simulator in 3GPP reference scenarios”, *Simul Model Pract Theory*, vol. 119, p. 102580, 2022, doi: <https://doi.org/10.1016/j.simpat.2022.102580>.
- [37] N. Patriciello, S. Lagen, B. Bojovic, y L. Giupponi, “An E2E simulator for 5G NR networks”, *Simul Model Pract Theory*, vol. 96, p. 101933, 2019, doi: <https://doi.org/10.1016/j.simpat.2019.101933>.
- [38] C. Sommer *et al.*, “Veins: The Open Source Vehicular Network Simulation Framework”, 2019, pp. 215–252. doi: 10.1007/978-3-030-12842-5\_6.
- [39] X. Ma, X. Hu, T. Weber, y D. Schramm, “Evaluation of Accuracy of Traffic Flow Generation in SUMO”, *Applied Sciences*, vol. 11, núm. 6, p. 2584, mar. 2021, doi: 10.3390/app11062584.
- [40] D. Krajzewicz, “Traffic Simulation with SUMO – Simulation of Urban Mobility”, 2010, pp. 269–293. doi: 10.1007/978-1-4419-6142-6\_7.

- [41] P. A. Lopez *et al.*, “Microscopic Traffic Simulation using SUMO”, en *IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2018. [En línea]. Disponible en: <https://elib.dlr.de/124092/>
- [42] X. Ma, X. Hu, T. Weber, y D. Schramm, “Evaluation of Accuracy of Traffic Flow Generation in SUMO”, *Applied Sciences*, vol. 11, núm. 6, p. 2584, mar. 2021, doi: 10.3390/app11062584.
- [43] D. A. Guastella, E. Montero-Porras, A. Morales-Hernández, y G. Bontempi, “Traffic Modeling with SUMO: a Tutorial”, ene. 2025.
- [44] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, y V. Koltun, “CARLA: An Open Urban Driving Simulator”, en *Proceedings of the 1st Annual Conference on Robot Learning*, S. Levine, V. Vanhoucke, y K. Goldberg, Eds., en *Proceedings of Machine Learning Research*, vol. 78. PMLR, jun. 2017, pp. 1–16. [En línea]. Disponible en: <https://proceedings.mlr.press/v78/dosovitskiy17a.html>
- [45] S. Malik, M. A. Khan, y H. El-Sayed, “Carla: Car learning to act—an inside out”, *Procedia Comput Sci*, vol. 198, pp. 742–749, 2022.
- [46] I. U. Hassan, H. Zia, H. S. Fatima, S. A. Yusuf, y M. Khurram, “A Lightweight Convolutional Neural Network to Predict Steering Angle for Autonomous Driving Using CARLA Simulator”, *Modelling and Simulation in Engineering*, vol. 2022, pp. 1–11, ago. 2022, doi: 10.1155/2022/5716820.
- [47] R. Gutiérrez-Moreno, R. Barea, E. López-Guillén, J. Araluce, y L. M. Bergasa, “Reinforcement learning-based autonomous driving at intersections in carla simulator”, *Sensors*, vol. 22, núm. 21, p. 8373, 2022.
- [48] T. R. Riley George F. and Henderson, “The ns-3 Network Simulator”, en *Modeling and Tools for Network Simulation*, M. and G. J. Wehrle Klaus and Güneş, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–34. doi: 10.1007/978-3-642-12331-3\_2.
- [49] Opensim, “OMNET++ Discrete Event Simulator”, junio de 2024. [En línea]. Disponible en: <https://omnetpp.org/>
- [50] A. Varga, “OMNeT++”, en *Modeling and Tools for Network Simulation*, M. and G. J. Wehrle Klaus and Güneş, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 35–59. doi: 10.1007/978-3-642-12331-3\_3.
- [51] C. Batard, F. Poitiers, C. Millet, y N. Ginot, “Simulation of Power Converters Using Matlab-Simulink”, en *MATLAB*, V.

N. Katsikis, Ed., Rijeka: IntechOpen, 2012, 3. doi:  
10.5772/46419.