

**UNIVERSIDAD POLITÉCNICA SALESIANA**

**SEDE CUENCA**

**CARRERA: INGENIERÍA ELECTRÓNICA**

ANÁLISIS DE SIMULACIÓN DE VIDEO STREAMING SOBRE REDES  
VEHICULARES AD-HOC MEDIANTE EL USO DE SOFTWARE LIBRE

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO**  
**DE:**

**INGENIERO ELECTRÓNICO**

Autores:

Luis Christian Criollo Ordóñez

José Rafael Duchitanga González

Tutor:

Dr. Ing. Jack Bravo Torres.

**CUENCA, ABRIL DE 2016**

**© DERECHOS DE AUTOR**

Nosotros, Luis Christian Criollo Ordóñez y José Rafael Duchitanga González, por medio del presente documento certificamos que hemos leído la Política de Propiedad Intelectual de la Universidad Politécnica Salesiana y estamos de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo de investigación quedan sujetos a lo dispuesto en la Política.

Asimismo, autorizamos a la Universidad Politécnica Salesiana para que realice la digitalización y publicación de este trabajo de investigación en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

**Cuenca, abril del 2016**



**Luis Christian Criollo Ordóñez**

**AUTOR**



**José Rafael Duchitanga González**

**AUTOR**

## CERTIFICACIÓN

En calidad de DIRECTOR DEL TRABAJO DE FIN DE GRADO “*Análisis de simulación de video streaming sobre redes vehiculares Ad-Hoc mediante el uso de software libre.*”, elaborada por Luis Christian Criollo Ordóñez y José Rafael Duchitanga González, declaro y certifico la aprobación del presente trabajo de tesis basándose en la supervisión y revisión de su contenido.

Cuenca, abril del 2016



Dr. Ing. Jack Fernando Bravo Torres

**DIRECTOR DEL TRABAJO DE GRADO**

## DECLARATORIA DE RESPONSABILIDAD

Los autores son los únicos responsables por los contenidos, conceptos, ideas, análisis, resultados investigativos y manifestados en la presente tesis “*Análisis de simulación de video streaming sobre redes vehiculares Ad-Hoc mediante el uso de software libre*”,

Cuenca, abril del 2016



**Luis C. Criollo Ordóñez**

**AUTOR**



**José R. Duchitanga González**

**AUTOR**

# ÍNDICE

<b>Índice.....</b>	<b>I</b>
<b>Índice de Figuras.....</b>	<b>IV</b>
<b>Índice de Tablas .....</b>	<b>V</b>
<b>Abreviaturas Usadas.....</b>	<b>VI</b>
<b>Agradecimientos.....</b>	<b>VII</b>
<b>Dedicatoria .....</b>	<b>VIII</b>
<b>Capítulo I.....</b>	<b>1</b>
<b>1. INTRODUCCIÓN.....</b>	<b>1</b>
1.1. Herramientas para el análisis de red .....	2
1.1.1. Network Simulator 2.....	3
1.1.2. Network Simulator 3.....	3
1.1.3. OMNeT++.....	4
1.1.4. MATLAB.....	4
1.2. Herramientas para el análisis de movilidad vehicular .....	5
1.2.1. SUMO, Simulation of Urban Mobility .....	5
1.2.2. FreeSim .....	5
1.2.3. CityMob .....	6
1.3. Objetivos de este trabajo .....	6
<b>Capítulo II .....</b>	<b>8</b>
<b>2. Estado del arte.....</b>	<b>8</b>
2.1. Redes Vehiculares Ad-Hoc (VANETs).....	8
2.1.1. Tipos de Redes VANET .....	9
2.1.2. Aplicaciones de Redes VANET .....	10
2.2. Clasificación de los protocolos para VANET .....	11
2.2.1. Topology-based Routing.....	11
2.2.2. Geographical Routing (position-based).....	12
2.2.3. Hierarchical Routing (clustering).....	12
2.2.4. Geo Cast-based routing.....	13
2.2.5. Flooding Routing (broadcasting).....	13
2.3. Estándar IEEE 802.11p.....	14
2.4. Streaming .....	15
2.5. Video Streaming en Redes Vehiculares.....	15
2.6. Protocolos para Video Streaming .....	17
2.6.1. An Application-Centric Inter-Vehicle Routing Protocol for Video Streaming over Multi-Hop Urban VANETs.....	17
2.6.2. A Reliable Beaconless Routing Protocol for VANETs (RBRP).....	22
2.6.3. QoE-aware Routing for Video Streaming over VANETs .....	24

2.6.4.	Opportunistic Vehicular Routing: Opportunistic Multicast for Emergency Video Streaming .....	25
2.6.5.	Opportunistic Routing for Live Video Streaming in Vehicular Ad Hoc Networks.....	26
2.6.6.	ReViV: Selective Rebroadcast Mechanism for Video Streaming over VANET.....	28
2.6.7.	A Hybrid Video Dissemination Protocol for VANETs: HIVE.....	29
2.6.8.	Beacon-less Video Streaming Management for VANETs Based on QoE and Link-Quality, QOALITE .....	32
2.6.9.	A Reactive and Scalable Unicast Solution for Video Streaming over VANETs.....	35
2.6.10.	The K-hop Cooperative Video Streaming Protocol Using H.264/SVC Over the Hybrid Vehicular Networks.....	38
2.6.10.1.	Streaming Task Assignment Scheme .....	39
2.6.10.2.	Packet Forwarding Scheme .....	40
2.6.11.	CLWPR - A Novel Cross-Layer Optimized Position Based Routing Protocol for VANETs .....	40
<b>Capítulo III.....</b>		<b>42</b>
<b>3. ANÁLISIS Y DISEÑO DEL SISTEMA .....</b>		<b>42</b>
3.1.	Selección de los Protocolos.....	42
3.1.1.	Ad-Hoc On-Demand Distance Vector (AODV) .....	42
3.1.1.1.	Mantenimiento de Rutas .....	43
3.1.2.	Optimized Link State Routing (OLSR) .....	44
3.1.3.	CLWPR.....	46
3.1.3.1.	CLWPR (Cross-Layer,Weighted, Position-based Routing) .....	46
3.1.3.2.	Repositorios de CLWPR.....	47
3.1.3.3.	Mecanismo de descubrimiento del vecino.....	49
3.1.3.4.	El mecanismo de Reenvío.....	50
3.1.3.5.	Mejoras utilizadas en el módulo CLWPR .....	52
3.1.3.6.	Limitaciones .....	53
3.2.	Consideraciones de Simulación.....	54
3.3.	Selección de los Nodos Fuente y Destino.....	54
3.4.	Traza de Video .....	56
<b>Capítulo IV .....</b>		<b>59</b>
<b>4. ANÁLISIS DE RESULTADOS.....</b>		<b>59</b>
4.1.	Métricas para análisis de rendimiento .....	59
4.2.	Análisis de BER .....	60
4.3.	Análisis de Delay.....	63
4.4.	Análisis de Throughput.....	65
<b>Capítulo V.....</b>		<b>68</b>
<b>5. CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>68</b>
5.1.	Conclusiones .....	68
5.2.	Trabajos Futuros.....	70
<b>APÉNDICES .....</b>		<b>71</b>
APÉNDICE A .....		71

**REFERENCIAS BIBLIOGRAFICAS .....76**

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Arquitecturas VANETs: (a) V2V, (b) V2I, (c) Arquitectura híbrida.....	10
<b>Figura 2.</b> Ejemplo de cómo funciona el Hierarchical Routing (clustering) [45]. .....	12
<b>Figura 3.</b> Mapa de los diferentes protocolos de enrutamiento VANET [46]. .....	13
<b>Figura 4.</b> Banda y canales en el espectro DSRC para U.S. [44]. .....	14
<b>Figura 5.</b> Topología de red: RSU recibe vídeo desde un servidor VOD a través de conexión a Internet por cable. El RSU actúa como fuente de streaming de vídeo a través de la red inalámbrica. Los vehículos actúan con comunicación de salto para retransmitir los paquetes de video a los vehículos de destino [2]......	19
<b>Figura 6.</b> Escenario de Movilidad Vehicular [2]. .....	20
<b>Figura 7.</b> Dirección de la zona de reenvío [4]. .....	23
<b>Figura 8.</b> Escenario de un modelo de retransmisión [8]. .....	26
<b>Figura 9.</b> Transmisión de video desde el lugar del accidente hacia las pantallas de destino [9]. .....	27
<b>Figura 10.</b> Ejemplo de la selección de nodos de retransmisión [10]......	29
<b>Figura 11.</b> Reenvío, cancelación y programación de nodos en la ruta [13]. .....	36
<b>Figura 12.</b> Mecanismo de trabajo de AODV [50]. .....	43
<b>Figura 13.</b> Comparación del flujo de mensajes originado por un único mensaje de difusión. A la izquierda todos los nodos retransmiten el mensaje; a la derecha solo lo hacen los nodos seleccionados como MPR (nodos en azul) [57]......	45
<b>Figura 14.</b> Formato del mensaje “HELLO” de CLWPR [41]......	49
<b>Figura 15.</b> Escenario de Simulación: Cuenca, Centro Histórico.....	55
<b>Figura 16.</b> Simulación en NetAnim.....	56
<b>Figura 17.</b> BER, Tasa de transmisión: 6Mbps.....	60
<b>Figura 18.</b> BER, Tasa de transmisión: 12Mbps.....	61
<b>Figura 19.</b> BER, Tasa de transmisión: 24Mbps.....	61
<b>Figura 20.</b> Delay, Tasa de transmisión: 6Mbps.....	63
<b>Figura 21.</b> Delay, Tasa de transmisión: 12Mbps.....	63
<b>Figura 22.</b> Delay, Tasa de transmisión: 24Mbps.....	63
<b>Figura 23.</b> Throughput, Tasa de transmisión: 6Mbps.....	65
<b>Figura 24.</b> Throughput, Tasa de transmisión: 12Mbps.....	65
<b>Figura 25.</b> Throughput, Tasa de transmisión: 24Mbps.....	66

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Información contenida en el mensaje “HELLO” [41]. .....	41
<b>Tabla 2.</b> Entrada de un vecino para CLWPR [41]. .....	48
<b>Tabla 3.</b> Posición basada en la entrada para CLWPR [41]. .....	48
<b>Tabla 4.</b> Tabla de enrutamiento para CLWPR [41]. .....	50
<b>Tabla 5.</b> Parámetros de Simulación .....	57

## ABREVIATURAS USADAS

AODV	Ad-Hoc On Demand Distance Vector
AU	Application Unit
BER	Bit Error Rate
CLWPR	Cross-Layer,Weighted, Position-based Routing
DSRC	Dedicated Short-Range Communications
HNA	Host and Network Association
ITS	Intelligent Transport Systems
LS	Location Service
LVS	Live Video Streaming
MAC	Media Access Control
MANET	Mobile Ad-Hoc Network
MPR	Multipoint Relay
NS-3	Network Simulator 3
OBU	On Board Unit
OFDM	Orthogonal Frequency Division Multiplexing
OLSR	Optimized Link State Routing
PAS	Position Association Set
PHY	Physical layer
PSNR	Peak-Signal to-Noise Ratio
RSU	Road-Side-Unit
SNIR	Signal-to-Noise Interference ratio
SUMO	Simulation Of Urban Mobility
UDP	User Datagram Protocol
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
VANET	Vehicular Ad-Hoc Network
WAVE	Wireless Access Vehicular Enviroment

## **AGRADECIMIENTOS**

Doy gracias a mis padres, a mi familia, amigos y profesores quienes me han acompañado a lo largo de toda mi vida, a mi compañero José por su dedicación y a Jack Bravo por el apoyo brindado en el desarrollo de este proyecto.

*Christian Criollo Ordóñez.*

Agradezco a Dios y a todas las personas que me apoyaron a lo largo de mis estudios, en especial a mis padres; por lo que gracias a su esfuerzo puedo cumplir una meta más en mi vida, a mis hermanos que me motivaban para seguir adelante, a mi compañero del proyecto que trabajó arduamente conmigo y al Ing. Jack Bravo que nos guió para la consecución de este proyecto.

*José Duchitanga González.*

## **DEDICATORIA**

A Dios, por permitirme estar aquí, de manera especial a mis padres, familia y sobrinos, quienes a parte de darme consejos han sido mi inspiración y motivación para luchar cada día, y a cada una de las personas que a su manera han sido parte de este logro.

*Christian Criollo Ordóñez.*

Este trabajo de fin de grado va dedicado a Dios por permitirme conseguir una meta más en la vida y a mi familia que me apoyó a lo largo de mi carrera.

*José Duchitanga González.*

## RESUMEN.

La integración entre las comunicaciones inalámbricas y los vehículos comenzó hace algunos años, dando lugar a las denominadas redes vehiculares Ad-Hoc (VANETs, del inglés *Vehicular Ad-Hoc Network*), donde los vehículos se comportan como nodos de transmisión, restando importancia a la presencia o no de infraestructura fija. Estas redes VANET toman importancia ya que en el futuro los vehículos serán capaces de intercambiar información acerca del tráfico o de la seguridad en carretera, sin dejar de lado la información de entretenimiento para el usuario.

Un tipo de información que juega un papel importante, es el video. Este tipo de transmisión supone algunos desafíos; alta variabilidad de la densidad vehicular, topologías dinámicas, enlaces con periodos de vida corta. Por otro lado, el video *stream* normalmente implica una transmisión de gran cantidad de datos, los cuales en el receptor deben cumplir con ciertas características que determinen la calidad del video.

Este trabajo de fin grado analiza el rendimiento de protocolos de enrutamiento tales como AODV, OLSR y CLWPR, los dos primeros enfocados en redes MANET, y el tercero a redes VANET. Los tres protocolos fueron usados para transmitir una traza de video en un escenario urbano, cada protocolo fue puesto a prueba bajo las mismas condiciones iniciales y bajo el mismo escenario.

CLWPR está basado en OLSR y toma en cuenta parámetros de la capa PHY y la MAC, además de aprovechar ventajas del mecanismo de capa cruzada, todo esto para determinar mejores rutas y mejorar el rendimiento del protocolo. Finalmente, los resultados muestran que en general CLWPR mantiene un rendimiento mejor que OLSR y AODV; aunque en ocasiones la superioridad de CLWPR sobre OLSR no es

muy marcada. Por otro lado, tanto CLWPR como OLSR muestran mejor rendimiento que AODV sin importar el escenario de simulación.

La herramienta de simulación de la red VANET fue NS-3 ya que permite configurar los parámetros que simulen un ambiente vehicular. El escenario de simulación urbano es el centro histórico de Cuenca, creado mediante SUMO y con el cual se obtuvieron las trazas de movilidad.

## CAPÍTULO I

### 1. INTRODUCCIÓN

En el futuro las redes vehiculares serán tan comunes como el uso de un smartphone. En la actualidad, los vehículos están equipados con sensores, sistemas de radio, capacidad de cómputo que les permite comunicarse con otros vehículos o infraestructura inteligente existente al borde de carretera. Este ambiente hace posible la presencia de los Sistemas de Transporte Inteligente (ITS, del inglés *Intelligent Transport Systems*).

Las redes vehiculares Ad-Hoc (VANET) son un tipo especial de redes móviles Ad-Hoc (MANETs) y son parte de los ITSs, donde los nodos de comunicación son vehículos en movimiento, como carros particulares o autobuses con capacidad de comunicación. Se puede clasificar en tres categorías: V2I, V2V y una red híbrida. Una red V2I es la comunicación entre el vehículo y un RSU (del inglés, *Road Side Unit*), mientras que un V2V sólo existe entre los vehículos, finalmente en la red híbrida un vehículo puede comunicarse con la infraestructura haciendo uso de uno o varios vehículos más.

Un tipo de información que juega un papel importante en el confort y seguridad del usuario, es el video. Se debe superar algunos desafíos para realizar una transmisión de *video streaming* sobre VANETs, ya que una red de este tipo es propensa a tener topologías dinámicas, enlaces entre nodos con bajos tiempos de vida, velocidad, o dirección de los autos y en cuanto al video *stream* normalmente tiende a ser datos de gran peso y que al momento de ser receptados deben cumplir con ciertas características como pérdida de paquetes, *throughput*, *delay*, etc., todo esto determina la calidad del video.

La selección del protocolo también es de gran importancia ya que de él depende el desenvolvimiento de la comunicación en distintos escenarios. Existen muchos protocolos de enrutamiento adaptados para VANETs. Sin embargo, pocos de ellos son adecuados para la entrega de *streaming* de vídeo en tiempo real.

Como se ha dicho, parámetros tanto físicos como lógicos intervienen en el desarrollo de una transmisión de video sobre VANETs. Se debe establecer un simulador de red que permita analizar la interacción de los nodos, su escalabilidad, etc., y además modelar también la tasa de datos, protocolos de encaminamiento o de transmisión, etc., NS-3 es un simulador el cual tiene la capacidad de configurar dichos parámetros.

### **1.1. Herramientas para el análisis de red**

La investigación sobre VANETs, desde las etapas de diseño hasta el desarrollo, se hace una tarea difícil debido a las limitaciones físicas de despliegue a gran escala o debido a sus limitaciones financieras. Por lo tanto, es necesaria una etapa de pre-desarrollo que permita evaluar el rendimiento de servicios prestados sobre VANETs. Los simuladores proporcionan las herramientas y escenarios para llevar a cabo el desarrollo de la investigación [38].

Existen varios simuladores, entre los más populares se encuentran: Network Simulator 2 (NS-2), Network Simulator 3 (NS-3), *MATrix LABORatory* (MATLAB), *Objective Modular Network Testbed in C++* (OMNeT++). Lo más importante es que el investigador encuentre un simulador fácil de usar, flexible, con varias herramientas para modelar y para análisis de resultados, y todos estos métodos deben ser válidos y confiables [38].

### **1.1.1. Network Simulator 2**

Es un simulador de eventos discretos diseñado para investigación en redes de comunicaciones, fue lanzado en 1989. NS-2 se basa en código C++ y OTCL. Puede soportar módulos para redes cableadas e inalámbricas. Para el caso de VANETs, se pueden usar protocolos como extensiones o como módulos, además tiene una serie de protocolos relacionados a transporte, capas de red y de datos. En cuanto a su capa física inalámbrica soporta varios módulos para transmisión, interfaces wireless y modelos de canales de propagación [38].

Una desventaja es que el código NS-2 no es apto para ser usado en una implementación real; además, solo soporta antenas bidireccionales y omnidireccionales, y sus nodos deben ser programados manualmente. Su código puede lograr extenderse a cientos de líneas [38,43].

### **1.1.2. Network Simulator 3**

Es un simulador de eventos discretos, C++ es usado para escribir los elementos del núcleo del simulador, de esta manera se ha hecho más escalable y más fácil de usar comparado con otros simuladores. Es un software libre que fue lanzado en 2009 y ha crecido rápidamente debido a la ayuda de la comunidad desarrolladora, esto ha facilitado la creación de modelos de redes Ad-Hoc y modelos de desvanecimiento de propagación. Tiene mejores características en lo que se refiere a dispositivos, canales e incluso ya implementa un modelo de movilidad vehicular [38-39].

NS-3 aún no posee muchos modelos en comparación con NS-2, esto se debe a que recién está disponible por algo más de 5 años. NS-3 no es compatible con NS-2, aunque una ventaja es que su código si puede ser usado en una implementación real. Además, los resultados pueden ser obtenidos en formatos estándar. Otra ventaja es que es un

software que es actualizado frecuentemente gracias a la ayuda de los mismos usuarios-desarrolladores que usan el software [38-39].

### **1.1.3. OMNeT++**

Fue lanzado en 1997, es ampliable y modular basado en componentes C++, principalmente está desarrollado para simulación de redes. Su principal fuerte es que posee una interfaz gráfica, inspectores de objetos para mostrar el estado de cada componente durante la simulación. Tiene un alto grado de detalles en la implementación de módulos y protocolos [38].

Sin embargo, OMNeT++ es un simulador lento y consume bastantes recursos computacionales, y además es algo difícil de usar. Esto se compensa en que comparado a los otros es más fácil de aprender y finalmente implementar nuevos escenarios. En cuanto a VANETs, OMNeT++ tiene un apartado solo para dicho fin: *Veins - Vehicles in Network Simulator* [38].

### **1.1.4. MATLAB**

Es un software para cómputo numérico, y últimamente se ha convertido en una herramienta para simulación de redes. Matlab es fácil de usar y su ambiente muestra los problemas y soluciones de forma matemática [38].

Cada simulador tiene ventajas y desventajas, la elección correcta del simulador dará al investigador beneficios como ahorrar tiempo, exacta eficiencia de los algoritmos o escalabilidad. Los simuladores NS-3 y OMNeT++ pueden manejar simulaciones a gran escala de manera muy eficiente, pero para el desarrollo de nuestros experimentos se ha escogido a NS-3. De acuerdo a los resultados mostrados en [47], en cuanto a tiempo de cómputo y uso de memoria, NS-3 posee un rendimiento total mejor que

todos los demás y aunque aún posee pocos modelos debido a que ha pasado poco tiempo desde su creación esto no es problema ya que tiene todo lo necesario para el desarrollo de este trabajo de fin de grado, además NS-3 tiene herramientas que permiten el fácil análisis de datos.

## **1.2. Herramientas para el análisis de movilidad vehicular**

### **1.2.1. SUMO, Simulation of Urban Mobility**

Sumo (del inglés *Simulation of Urban MObility*) modela el comportamiento y la manera en que interactúan los vehículos dentro de las calles, el patrón de comportamiento del vehículo está ligado con las rutas, las intersecciones, el número de carriles, las señales de tráfico, etc. Su licencia es gratuita, y además es capaz de generar trazas de salida que pueden ser utilizadas en varios simuladores de red, como por ejemplo NS-3.

### **1.2.2. FreeSim**

“Programado en Java, este simulador permite la representación y carga de escenarios como estructuras gráficas de múltiples sistemas de carreteras. Acepta crear algoritmos para el tráfico y aplicarlos a toda la red, un nodo o a un vehículo. Los datos de entrada pueden ser creados por el usuario, o importados de alguna organización del transporte” [59].

### **1.2.3. CityMob**

“Es un simulador diseñado para explotar los diferentes modelos de movilidad en VANETs, con una clara vocación a utilizarse como herramienta para una posterior medida del impacto en el rendimiento de la comunicación intervehicular. Crea escenarios de movilidad urbana y simula coches accidentados, de tal forma que las trazas generadas pueden alimentar al simulador de redes NS-2” [59].

## **1.3. Objetivos de este trabajo**

### **Objetivo General**

Analizar el rendimiento de *video streaming* sobre VANETs a través de simulación con NS-3.

### **Objetivos Específicos**

- **O.E.1** Realizar el estudio del estado del arte
- **O.E.2** Comprender el funcionamiento de los simuladores NS-3 y SUMO
- **O.E.3** Implementar algún protocolo de *video streaming* sobre NS-3
- **O.E.4** Crear un Escenario de Simulación
- **O.E.5** Analizar el rendimiento de *video streaming* sobre VANETs en escenarios de simulación en la ciudad de Cuenca.

Este documento se estructura de la siguiente manera, en el capítulo 2 se describe el estado del arte que sustenta las simulaciones, en esta sección principalmente se detalla la infraestructura de las redes VANETs y sus protocolos de enrutamiento. En el

capítulo 3 se presenta la elección de los protocolos y el diseño del escenario de simulación propuesto, así como la configuración de los parámetros para emular un ambiente WAVE (del inglés *Wireless Accesss Vehicular Enviroment*), además se detalla todo el proceso de simulación. Finalmente, en los capítulos 4 y 5 mostramos el análisis de los resultados y las conclusiones de las simulaciones.

## CAPÍTULO II

### 2. ESTADO DEL ARTE

Con el avance tecnológico, y por la necesidad de estar comunicados en cualquier circunstancia, se han desarrollado las denominadas redes VANETs, las cuales conllevan la comunicación entre vehículos. Dichas redes tienen potenciales aplicaciones futuras, una de ellas es la transmisión de *video streaming*. Por tal motivo, su estudio y desarrollo es un reto dentro del área de las telecomunicaciones.

Para conocer el rendimiento en la transmisión de *video streaming* en redes vehiculares Ad-Hoc se debe realizar estudios y simulaciones antes de implementarlos físicamente, con el fin de emular el comportamiento de la red lo más cercano posible a la realidad, sacando conclusiones y así poder seguir desarrollando la tecnología. Por tal motivo, la fundamentación teórica presentada a continuación nos ayudará a comprender hasta donde se ha avanzado con la investigación del tema y aportar a su desarrollo.

Este capítulo está estructurado de la siguiente manera, En la sección 2.1 y 2.2 se analizan las VANETs y sus protocolos, mientras que en la sección 2.3 se analiza el estándar 802.11p. Finalmente en la sección 2.4, 2.5 y 2.6 se analiza el *video streaming* y protocolos para soportar estas transmisiones.

#### 2.1. Redes Vehiculares Ad-Hoc (VANETs)

Las redes inalámbricas pueden categorizarse en dos tipos, las que son basadas en infraestructura, y que básicamente dependen de *Access Points*, o las que no dependen de infraestructura (Ad-Hoc), las cuales son redes auto organizadas [40].

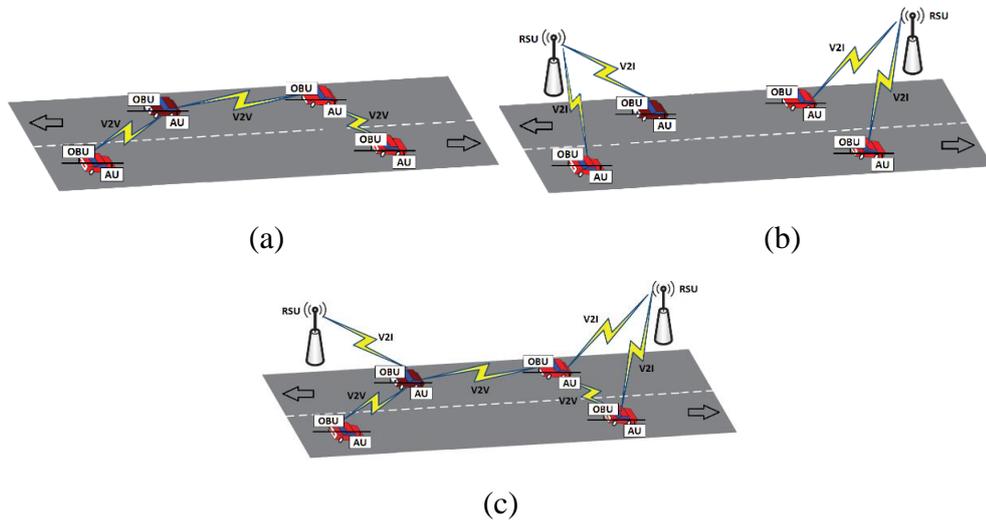
Cuando los nodos se mueven en diferentes direcciones y velocidades la red se la nombra Red Móvil Ad-Hoc (MANET). Cuando los autos se consideran nodos móviles la red pasa a llamarse red VANET la cual se considera un tipo especial de las MANETs. En las VANETs cada nodo actúa como un *router* y mantiene distinta información de la red de acuerdo al protocolo de enrutamiento [40].

Esta tecnología permite Sistemas de Transporte Inteligente los cuales están destinados a proveer aplicaciones de usuario ya sea *infotainment applications* o anuncios de negocios, escuelas, restaurantes, teatros, etc. También puede proveer aplicaciones de seguridad como movilidad segura, para manejo de tráfico o reducción de accidentes.

### **2.1.1. Tipos de Redes VANET**

Las redes VANETs pueden clasificarse según el tipo de comunicación, en: V2V (Vehículo a Vehículo del inglés *Vehicle to Vehicle*), V2I (Vehículo a Infraestructura, del inglés *Vehicle to Infrastructure*) y una arquitectura híbrida. [37]

La comunicación V2V se da entre dos vehículos equipados con dispositivos *wireless* llamados OBU (del inglés *On Board Unit*) lo que les permiten intercambiar mensajes. Las interfaces para los usuarios son las AU (del inglés *Application Unit*) las cuales residen en el OBU y sirve para el intercambio de mensajes, ver Figura 1(a). Mientras que una red V2I se forma entre un vehículo y una red inteligente preexistente, la cual se compone principalmente de RSUs (unidad ubicada a lo largo de la carretera, del inglés *Road-Side-Unit*), una red V2I permite un rango de transmisión mucho más largo que una red V2V, ver Figura 1 (b). Finalmente en una red híbrida, ver Figura 1(c), un vehículo puede comunicarse con un RSU usando uno o varios saltos dependiendo la distancia hacia el RSU [37, 42, 48].



**Figura 1.** Arquitecturas VANETs: (a) V2V, (b) V2I, (c) Arquitectura híbrida

Para realizar este tipo de comunicación entre vehículos y dispositivos ubicados en la vía se utiliza el concepto de Comunicaciones Dedicadas de Corto Alcance (DSRC, del inglés *Dedicated Short-Range Communications*). El formato DSRC especifica las frecuencias y anchos de banda de operación [35].

### 2.1.2. Aplicaciones de Redes VANET

Las redes VANETs se las puede utilizar para ofrecer varios tipos de aplicaciones, a través de las cuales se puede intercambiar información, de seguridad o de entretenimiento para proporcionar confort al usuario.

- **Aplicaciones de Seguridad.**

Las aplicaciones enfocadas en la seguridad consisten en notificar sobre accidentes o peligros en la carretera, la transmisión de datos debería ser confiable y su retardo debería ser muy bajo. También se puede intercambiar periódicamente mensajes de alerta, tales como la velocidad o locación del vehículo, por ser solo de alerta estos datos ya no estarían sujetos a tener un bajo retardo [37].

- **Aplicaciones para manejo de Tráfico.**

Para evitar congestiones y gestionar el viaje desde el inicio, lo que conlleva a reducir el tiempo de viaje o consumo de gasolina por medio del manejo de las rutas a seguir [37].

- **Aplicaciones de información y entretenimiento.**

Son las llamadas aplicaciones de valor agregado, por ejemplo, video juegos, películas, o email, las cuales pretenden hacer placentero el viaje [37].

## **2.2. Clasificación de los protocolos para VANET**

Una forma de categorizar los protocolos es de acuerdo a su diseño, tales como: *Topology-based Routing*, *Geographical Routing (position-based)*, *Hierarchical Routing (clustering)*, *Geo Cast-based routing* y *Flooding Routing (broadcasting)*.

### **2.2.1. Topology-based Routing**

Se divide en dos categorías: los protocolos proactivos y reactivos. Los protocolos proactivos introducen constantemente una sobrecarga en la red la cual se incrementa dependiendo del tamaño y topología de la red con el propósito de mantener actualizadas las tablas de enrutamiento. Los protocolos reactivos añaden un retardo en el inicio de la comunicación con el objetivo de descubrir la ruta inundando la red con esa petición [41].

También existen protocolos híbridos que aprovechan las mejores características del enrutamiento proactivo y reactivo. Estos protocolos descubren las rutas entre zonas para reducir la carga en la red. Parte del enrutamiento híbrido es el enrutamiento proactivo con el cual se trabaja en intra-zona, mientras que protocolos reactivos son usados para enrutamiento inter-zonal [46].

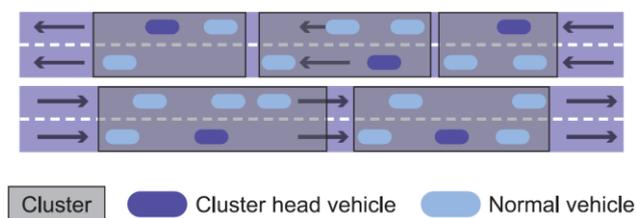
### 2.2.2. Geographical Routing (position-based)

Es de los que mejor se adapta a las necesidades de las VANETs. Se da por hecho que el nodo es capaz de conocer su posición, algo lógico debido al amplio uso de GPS en vehículos; además se da por hecho que cada nodo conoce o es capaz de conocer la posición del nodo de destino cuando sea necesario, esto se logra usando principalmente mapas de calles o servicios de GPS [41,46].

Estos tipos de protocolos se adaptan a las redes Ad-Hoc ya que puede funcionar mejor en redes de gran escala debido a que solo necesita conocer el siguiente salto en vez de conocer toda la ruta hacia el destino, por lo tanto, este protocolo no almacena rutas ni tablas de enrutamiento. Algo importante también es que la sobrecarga de la red se minimiza ya que un nodo solo hace un *broadcast* hacia los vecinos [41].

### 2.2.3. Hierarchical Routing (clustering)

Es principalmente usado en grandes redes. Este protocolo, virtualmente divide la red en *clusters*, como se ve en la Figura 2, los cuales comparten algunas características durante un periodo de tiempo. Cada *cluster* tiene un coordinador (*cluster head*) el cual es el responsable de coordinar la red tanto hacia dentro como hacia afuera del *cluster*. El *delay* y la sobrecarga necesaria para mantener el *cluster* es una desventaja [41, 45,46].



**Figura 2.** Ejemplo de cómo funciona el Hierarchical Routing (*clustering*) [45].

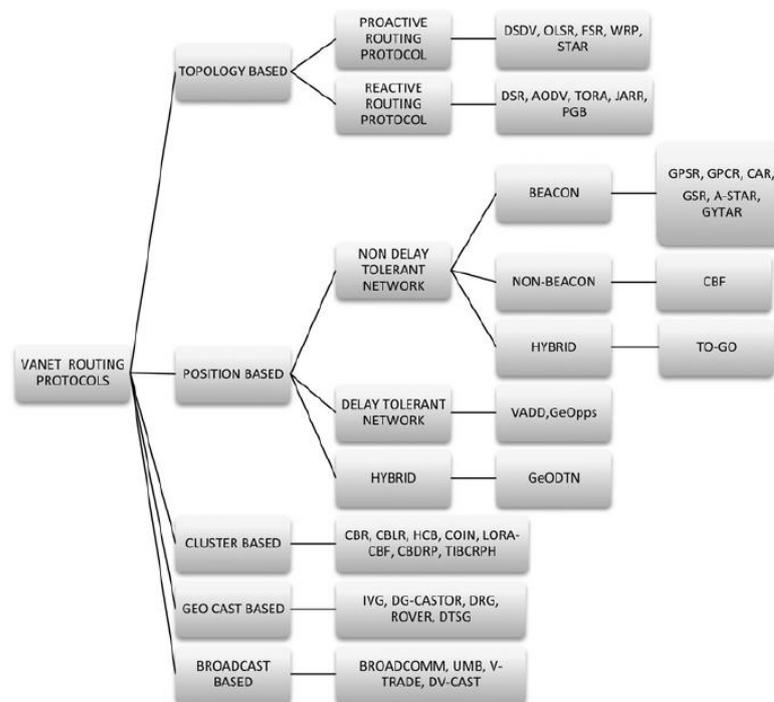
## 2.2.4. Geo Cast-based routing

El mensaje se transmite hacia una región usando una comunicación *multicast* para la difusión del mensaje. Se puede usar una técnica ya sea de inundación o no, de esta manera se reduce las colisiones y sobrecarga de los paquetes [46].

## 2.2.5. Flooding Routing (broadcasting)

El mensaje es difundido entre los autos y entre V2I/I2V. Es la manera más simple de difundir un mensaje en la red, la complejidad del protocolo se minimiza, pero la sobrecarga, colisión de mensajes y el ancho de banda se incrementa exponencialmente con el tamaño de la red y la carga de tráfico [41,46].

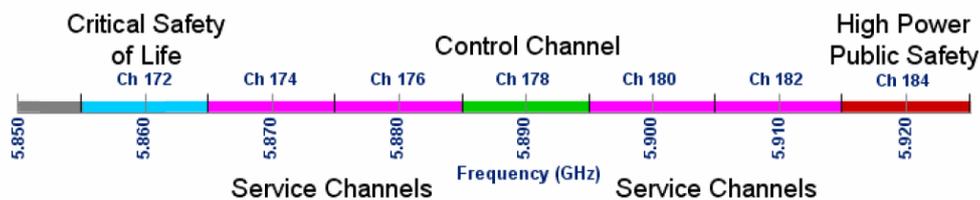
Un resumen de los tipos de protocolos se muestra en la Figura 3.



**Figura 3.** Mapa de los diferentes protocolos de enrutamiento VANET [46].

### 2.3. Estándar IEEE 802.11p

Es un estándar dentro de IEEE 802.11 para proporcionar Acceso Inalámbrico en Ambientes Vehiculares (WAVE, del inglés *Wireless Access in Vehicular Environments*). Este estándar define mejoras de 802.11 para soportar ITSs y define como es la comunicación entre vehículos y entre estructura fija. Esta tecnología trabaja con 75Mhz en el espectro de frecuencia, a lo largo de la banda de 5.9Ghz (5.850Ghz a 5.925Ghz). Como se puede ver en la Figura 4, el espectro está estructurado en 7 canales de 10Mhz de ancho, el canal 178 es el canal de control CCH y está restringido solo para para mensajes de seguridad, los dos canales externos se restringen para usos especiales mientras que el resto son los canales de servicio SCH y están disponibles para ser usados para cualquier uso [39,44]. La capa PHY está basada en OFDM, la misma de 802.11a.



**Figura 4.** Banda y canales en el espectro DSRC para U.S. [44].

WAVE estandarizado como 802.11p es una parte significativa de DSRC. WAVE define las técnicas de señalización y las funciones de interfaz que son controladas por la capa física y MAC, donde las propiedades de la capa física cambian rápidamente, y donde los intercambios de información son de corta duración. El propósito de este estándar es proporcionar especificaciones que garanticen la comunicación entre dispositivos inalámbricos en ambientes que cambian rápidamente [39,40].

DSRC/WAVE soporta un entorno en el que los vehículos pueden estar en movimiento a velocidades de hasta 200 kilómetros por hora, cubriendo un rango de comunicación

de 300 metros hasta 1000 metros con una velocidad de datos desde 3 hasta 27 Mbps [44].

## 2.4. Streaming

Mediante el *streaming* se puede transmitir información multimedia que puede ser tanto de voz, audio o video de manera continua, donde el contenido multimedia puede estar previamente almacenado o ser una transmisión en vivo. Esta información se la reproduce en tiempo real e inmediatamente se la descarta. Reduciendo de esta manera el tiempo de espera necesaria para comenzar la reproducción en el dispositivo del usuario, de manera la reproducción se puede realizar mientras el archivo se está descargando y además se ahorra espacio en disco ya que solo se necesita un *buffer*<sup>1</sup> pequeño [62].

Por lo tanto, *video streaming* es un método de transmisión de vídeo sobre una red que permite al usuario visualizar el vídeo en tiempo real sin necesidad de almacenar el archivo en su totalidad. Sin embargo, este tipo de tecnología es sensible al retardo, ya que si los datos no llegan a tiempo el proceso de visualización se pausará o detendrá, provocando molestias para el usuario receptor por la falta continuidad de flujo de vídeo. Por tal motivo se introduce un *buffer*, con el cual se almacena una cantidad de paquetes a la espera de completar los cuadros pendientes de visualizar. [62]

## 2.5. Video Streaming en Redes Vehiculares

Debido a la penetración de dispositivos de comunicación *Wireless* dentro de los autos, las investigaciones se han enfocado en darle un uso significativo mediante las VANETs [9]. La transmisión de video puede darse de dos maneras: video interactivo y *video streaming*. Video interactivo soporta comunicaciones como chat, video

---

<sup>1</sup> Buffer: espacio en la memoria en un disco reservado para el almacenamiento temporal de información digital.

conferencia, etc.; mientras que *video streaming* funciona en escenarios donde un nodo fuente realiza una transmisión en una sola vía hacia uno o varios receptores [42].

La comunicación por *video streaming* está concebida para prestar un gran beneficio para el manejo del tráfico. A largo plazo se podrá transmitir los datos tipo multimedia (por ejemplo, imágenes o clips cortos de inundaciones, incendios, daños de un terremoto, daño de ataque terrorista, etc.). El conductor al conocer la situación actual podrá tomar decisiones en base a sus prioridades. En situaciones de emergencia esto hará que el proceso de asistencia o procedimientos de evacuación sean más rápidos, así se podrá mejorar la seguridad de la navegación vehicular. *Video streaming* en redes vehiculares también puede proporcionar entretenimiento con valor agregado [8, 10, 13].

A pesar del potencial de las VANETs para apoyar alta conectividad, estas aplicaciones tratan con varios desafíos debido a la naturaleza de la transmisión, el área de cobertura, los requisitos de calidad del vídeo y topologías altamente dinámicas. Las decisiones de enrutamiento de los protocolos deben adaptarse a dichas condiciones y cumplir ciertos requerimientos específicos de QoS (del inglés *Quality of Service*) y de QoE (del inglés *Quality of Experience*). Deben recuperar o mantener el flujo de video con al menos un buen nivel de calidad para los seres humanos [12]. En general se debe tener en cuenta que como se indica en [61] el retardo en la transmisión no debe sobrepasar los 10 segundos, o que la pérdida de paquetes no debe exceder el 5% y a pesar de que la difusión de video implica un mayor ancho de banda, no se debe exagerar y transmitir cantidades abrumadoras de datos [13]. En cuanto al BER, en una transmisión inalámbrica los valores umbral<sup>2</sup> que indican desde una “muy pobre” hasta una “excelente” calidad de video están entre  $> 1 \times 10E - 3$  hasta  $< 1 \times 10E - 4$ , tal como se indica en [60].

---

<sup>2</sup> Valores Umbral de BER: Excelente ( $< 1 \times 10E - 4$ ); Buena ( $1 \times 10E - 4$  a  $4 \times 10E - 4$ ); Satisfactoria ( $4 \times 10E - 4$  a  $8 \times 10E - 4$ ); Pobre ( $8 \times 10E - 4$  a  $1 \times 10E - 3$ ); Muy pobre ( $> 1 \times 10E - 3$ )

## **2.6. Protocolos para Video Streaming**

La transmisión de *video streaming* sobre VANETs se ha vuelto un tema de gran interés ya que con ello se tiene la capacidad de proveer información más precisa. A pesar de que hay varios desafíos ya se han propuesto varios esquemas de enrutamiento. Sin embargo, la mayoría se han enfocado en minimizar el retardo en la entrega de los paquetes de datos, sin tomar en cuenta la distorsión que introduce usar varios canales inalámbricos, la cantidad de paquetes transmitidos, etc.

Las VANETs deben tener la capacidad de garantizar la entrega del *streaming* de video desde la fuente hasta el punto de destino y proporcionar una reproducción suave casi en tiempo real. Pero la alta velocidad de los vehículos, el cambio rápido de la densidad vehicular, los enlaces inalámbricos con pérdidas, y otros factores en las carreteras, hacen difícil encontrar un camino estable de extremo a extremo para la entrega de paquetes de vídeo [9].

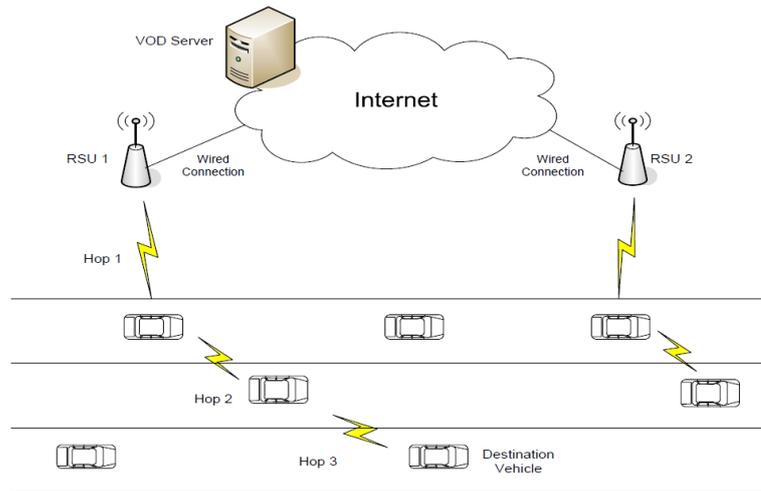
Algunos de los protocolos para *video streaming* estudiados se muestran a continuación.

### **2.6.1. An Application-Centric Inter-Vehicle Routing Protocol for Video Streaming over Multi-Hop Urban VANETs**

Este protocolo trata de transmitir por el camino que presente menos saltos. Aquí se propone el protocolo *An Application-Centric Inter-Vehicle Routing Protocol* para la transmisión de video en tiempo real a través de la red VANET en escenarios urbanos. Este protocolo de enrutamiento se basa en el intercambio de información entre la capa de aplicación y la capa de red con el fin de seleccionar la ruta que minimiza la función de distorsión de fotogramas de la capa de aplicación, mediante el uso de la probabilidad de error en la transmisión punto a punto [2].

Para el diseño del protocolo de enrutamiento se tiene en cuenta, la distribución espacial del tráfico y probabilidad de conectividad para los escenarios VANET, para esto se utiliza enrutamiento geográfico. La comunicación multi-salto mantiene la potencia de transmisión baja. El objetivo es maximizar la relación de señal a ruido de pico en el usuario final (PSNR del inglés *Peak Signal to Noise Ratio*) de la trama de vídeo recibido, lo que equivale a minimizar el error cuadrático medio de extremo a extremo (MSE del inglés *Mean Square Error*) de los fotogramas de vídeo entregados. El vehículo en cada salto utiliza un esquema de decodificación y reenvío para retransmitir los paquetes al siguiente salto sin amplificar los efectos del ruido. El vehículo que transporta los paquetes; primero selecciona un subconjunto de vehículos candidatos para reenviar sus paquetes. A continuación, se aplica la optimización centrado en las aplicaciones para entregar los fotogramas de vídeo con alta calidad [2].

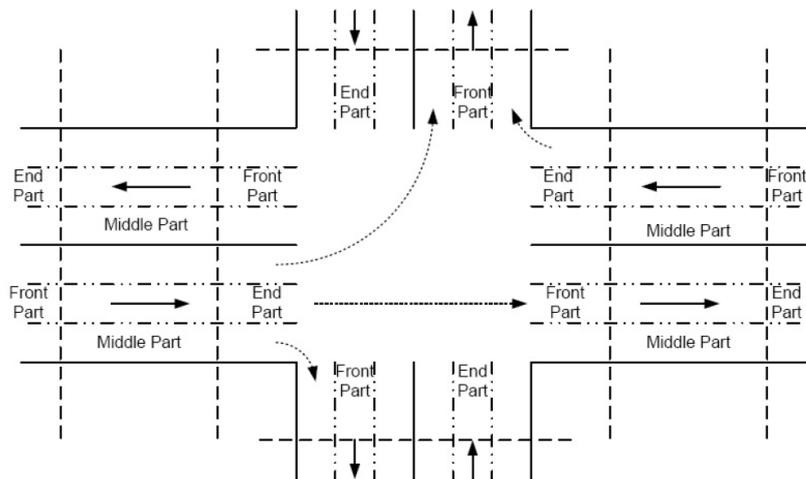
Este protocolo se basa en la topología que se aprecia en la Figura 5. Cada RSU transmite paquetes de vídeo al vehículo destino a través de la comunicación multi-salto, donde los vehículos intermedios actúan con una comunicación de salto para retransmitir los paquetes de vídeo al vehículo destino. La conmutación de la transmisión entre RSU durante una sesión de video se controla por medio un *router* de acceso (AR del inglés *Access Router*) que controla varios RSU. Cuando el vehículo sale del rango de cobertura de un AR, ocurre un traspaso y un nuevo AR controla la transmisión a través de un nuevo conjunto de RSUs que cubren la ubicación actual del vehículo destino. Existen enlaces inalámbricos entre RSU y vehículos (enlaces V2I) y vehículo a vehículo (enlaces V2V). La red vehicular se basa en el estándar de 5.9 GHz para las comunicaciones dedicadas de corto alcance (DSRC) y cada vehículo está transmitiendo a sus vecinos la información de la posición, la hora actual, dirección, velocidad, aceleración/deceleración, y eventos de tráfico. Por lo tanto, la RSU y vehículos pueden actualizar la información relativa a la ubicación y la dirección del movimiento de los nodos vecinos (vehículos) [2].



**Figura 5.** Topología de red: RSU recibe vídeo desde un servidor VOD a través de conexión a Internet por cable. El RSU actúa como fuente de *streaming* de vídeo a través de la red inalámbrica. Los vehículos actúan con comunicación de salto para retransmitir los paquetes de video a los vehículos de destino [2].

Para modelar la movilidad vehicular y simular un ambiente de tráfico, se sigue el modelo estocástico propuesto en [1], que es capaz de incluir el comportamiento de los vehículos por separado en escenarios dispersos y densos.

En este modelo, las diferentes categorías de velocidad se consideran en cada sección de la calle separados por líneas de trazos, (Ver Figura 6). En este sentido, la movilidad puede ser modelada por una red de colas de servicio (secciones de calles) en cascada y cada cola tiene clientes con varias clases (categorías de velocidad). En cada cola, la tasa de servicio es equivalente a la tasa de paso de vehículos de esa sección a la siguiente sección de la calle, es decir, un vehículo con una velocidad más alta se asocia con la clase de cliente con una mayor tasa de servicio. En un escenario disperso, los vehículos tienen patrones de movilidad independientes [2].



**Figura 6.** Escenario de Movilidad Vehicular [2].

Por lo tanto, cada cola es una  $M/G/\infty$  colas, es decir, la tasa de servicio no depende del número de clientes en la cola (número de vehículos en esa parte de la calle). Cuando todas las partes de la calle se encuentran en una situación dispersa, en realidad tenemos una red de colas que comprende del  $M/G/\infty$  colas con dos, tres, y dos clases de clientes, lo que corresponde a la parte delantera, media y partes de extremo, respectivamente. Una sección de la calle con tráfico denso puede ser modelado por  $M/G(n)/1$  colas [2].

El modelo de movilidad se utiliza para extraer la probabilidad de la conectividad entre dos vehículos que luego se utiliza para formular la QoS para ser optimizados en el *An Application-Centric Inter-Vehicle Routing Protocol* [2].

La probabilidad de conectividad se define como la probabilidad de que los vehículos a lo largo de las calles puedan comunicarse entre sí a través de un enlace de radio. En general, es posible tener varios enlaces de radio entre dos vehículos. Además, cada enlace de radio puede consistir en múltiples saltos [2].

Dos vehículos son capaces de comunicarse directamente si están dentro del rango de transmisión de radio del otro. Los límites de probabilidad conectividad inferior y superior para una calle típica se la puede ver en [29].

Este protocolo se centra en las características de los protocolos de enrutamiento geográficas, ya que son más adecuados para VANETs. El protocolo estándar de enrutamiento geográfico normalmente consta de los siguientes cuatro pasos [3]:

1. **La determinación de la coordenada de destino:** esta se determina por la fuente y estas coordenadas son incluidas en el encabezado del paquete para informar a los transmisores intermedios.
2. **Determinación de las coordenadas de los vecinos 1-salto:** esto se hace mediante el intercambio de mensajes *beacon*<sup>3</sup> que contienen el identificador del remitente y sus coordenadas.
3. **La determinación del siguiente transmisor:** el próximo transmisor se selecciona basándose en la ubicación de los remitentes, vecinos 1-salto y el nodo de destino.
4. **La entrega del mensaje:** los vecinos 1-salto recibe el mensaje, pero sólo uno de ellos contribuirá al de enrutamiento que contiene el identificador del nodo seleccionado.

La localización y dirección del nodo, velocidad de paquetes de vídeo en el enlace entre el nodo portador y el nodo vecino, probabilidad de error de paquetes en los enlaces de conexión y el tiempo total de transmisión del paquete son la información intercambiada entre nodos vecinos para encontrar el próximo vehículo (nodo) para la transmisión de paquetes de vídeo.

---

<sup>3</sup> Mensajes *Beacon*: Es el mecanismo que utilizan los *routers* para anunciar su presencia a clientes que podrían estar interesados en conectarse, es por medio del envío "abierto" o envío *broadcast*, estos *frames* contienen información relacionada con el dispositivo que se encuentra disponible y da algunas pautas para realizar la conexión con el mismo. Estos *Beacon frames* se transmiten con una periodicidad fija que permite a los clientes potenciales estar al tanto de posibles cambios que ocurren en el dispositivo [33].

### 2.6.2. A Reliable Beaconless Routing Protocol for VANETs (RBRP)

Este protocolo se utiliza para el reenvío *unicast* de paquetes sin la ayuda de *beacons*. En RBRP, se propone una zona de reenvío auto-adaptativa para evitar la difusión multipunto como se observa en la Figura 7.

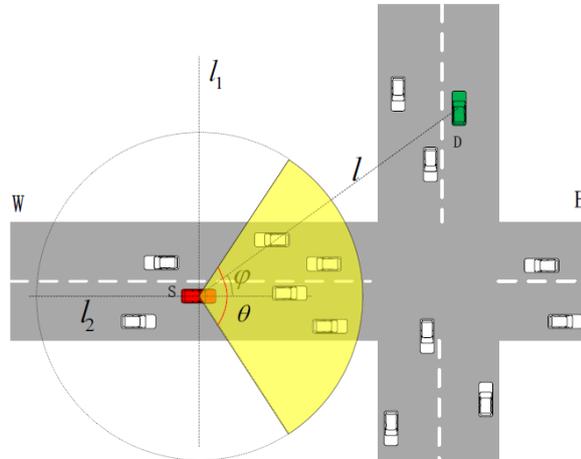
Los dos puntos clave en RBRP son la definición de la zona de reenvío y el algoritmo de tiempo de espera. La elección de la dirección de la zona de reenvío no sólo se basa en la ubicación del destino, sino también en la dirección de movimiento del emisor actual. El tamaño del ángulo de la zona de reenvío se puede ajustar de acuerdo con la densidad del tráfico. La densidad de tráfico se mide por el tiempo de transmisión del último salto al nodo actual [4].

Para el funcionamiento de este protocolo se asume las siguientes consideraciones [4]:

- Cada vehículo está equipado con GPS, y puede obtener su información de ubicación constantemente, además de la velocidad, la aceleración y la dirección de movimiento.
- Cada vehículo tiene un número de identificación exclusivo en las redes.
- El alcance de comunicación de cada nodo es igual.
- El ángulo de la zona de reenvío de la fuente es un valor constante ( $\theta = 60$  grados).

Se define la zona de reenvío en dos partes: la dirección y el tamaño del ángulo. Se considera la dirección de movimiento del nodo actual como la referencia de coordenadas de dirección (ver Figura 7), el ángulo entre la dirección de referencia y la línea que conecta emisor y el destino es  $\varphi$ . Si el valor absoluto del ángulo de  $\varphi$  es menor que 90 grados, la dirección de la zona de reenvío es la misma que la dirección de referencia, de lo contrario, la dirección de la zona de reenvío está en la dirección opuesta; en este caso  $\varphi$  es el ángulo entre  $l$  y  $l_2$ , donde  $l_2$  es la línea hacia la dirección

de movimiento, y  $l$  es una línea recta que conecta el emisor y el destino. En conclusión  $\theta$  es el ángulo de apertura de la zona de reenvío, es la zona pintada con amarillo de la Figura 7 [4].



**Figura 7.** Dirección de la zona de reenvío [4].

La prioridad de los nodos en la zona de reenvío se determina por el tiempo de espera; por lo tanto, el diseño de un algoritmo apropiado utiliza la ubicación de nodo, el tiempo de estancia del nodo de la zona de reenvío del último salto, la calidad del enlace entre el nodo y el último salto y la carga del nodo. Para conseguir esto se utiliza *Greedy Strategy*<sup>4</sup>, la calidad de servicio y la carga del nodo [4].

La idea principal de RBRP se describe de la siguiente manera:

1) El emisor determina si la distancia desde él hacia el destino es menor que su rango de comunicación  $R$ . Si es así, el emisor envía los paquetes y termina la entrega. De lo contrario, el emisor añade en los paquetes la información de su ubicación del destino, el ID del destino, el tiempo de emisión y el ángulo de la zona de reenvío del nodo [4].

<sup>4</sup> *Greedy Strategy*: Un algoritmo *greedy* elige, en cada paso, una solución local óptima [4].

2) Los receptores registran el tiempo de recepción de forma individual. Estos nodos se dividen en dos grupos: una parte de los nodos se encuentran en la zona de reenvío, y el resto están fuera de la misma [4].

3) Los nodos en la zona de reenvío calculan su tiempo de espera y compiten por el derecho de reenviar los paquetes según el esquema de contención<sup>5</sup>, mientras que los nodos fuera de la zona de reenvío descartan los paquetes [4].

4) El nodo en la zona de reenvío cuyo tiempo de espera expira primero consigue el derecho de reenviar los paquetes y es seleccionado como el nuevo emisor, entonces repite el proceso desde el paso 1.

### 2.6.3. QoE-aware Routing for Video Streaming over VANETs

Aquí se propone un protocolo de enrutamiento basado en la calidad de la experiencia para el *streaming* de vídeo a través VANETs [5]. Los protocolos de enrutamiento más conocidos son AODV (del inglés *Ad-Hoc On-Demand Distance Vector Routing*) [6] y OLSR (del inglés *Optimized Link State Routing*) [7].

El OLSR supera a la AODV en VANETs porque maneja bien el problema de la desconexión [15]. Para soportar trayectos múltiples en OLSR se propone MPOLSR, mediante el algoritmo de Dijkstra [30], con este algoritmo se puede obtener rutas sin bucles. MPOLSR muestra un mejor desempeño en el equilibrio de carga y la red de alta movilidad. Mediante QoE se controla el flujo de datos de la red, QoE, para evaluar la percepción de video en tiempo real se usa el *Pseudo-Subjective Quality Assessment*. (PSQA).

PSQA se basa en el aprendizaje estadístico utilizando la red neuronal aleatoria (RNN) y sus principios se describen en [34]. Las herramientas de PSQA toman los parámetros

---

<sup>5</sup> Esquema de contención: es un algoritmo general para calcular el tiempo de espera que considera la estrategia *Greedy* [4].

de las redes orientadas como ingresos dando el correspondiente MOS<sup>6</sup> como salida. El enrutamiento *QoE-aware* se ha propuesto para las redes en malla en donde cada nodo en la red usa aprendizaje por refuerzo para seleccionar el trayecto. El QOV es la combinación de QoE y MPOLSR, llamado QoE basado en el protocolo de enrutamiento para la transmisión de vídeo sobre VANETs (QoV). QoV adopta los esquemas de recuperación de la ruta y de detección de bucle de MPOLSR.

#### **2.6.4. Opportunistic Vehicular Routing: Opportunistic Multicast for Emergency Video Streaming**

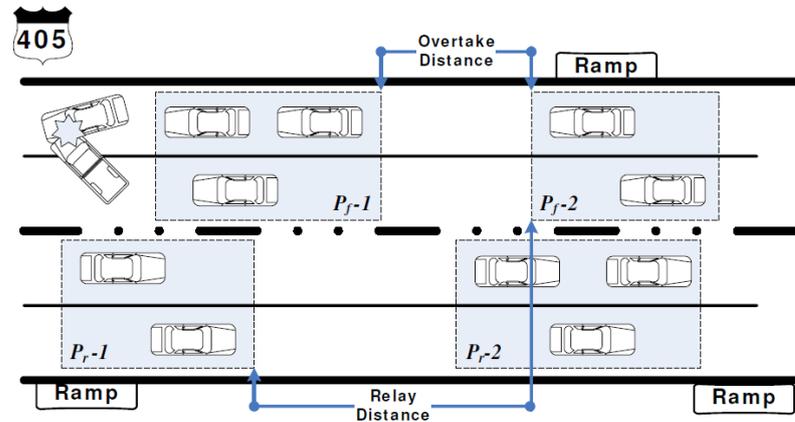
La difusión de información (por ejemplo, imágenes o clips cortos de inundaciones, incendios, daños de un terremoto, daño por ataque terrorista, etc) en situaciones de emergencia debe tener lugar en cuestión de minutos desde que se produce el accidente, mucho antes de que la policía y helicópteros de noticias puedan intervenir en la escena. La emisión debe ser eficiente y ligera para no congestionar la red. Además, la información debe ser entregada de forma fiable a todos los vehículos involucrados para que se pueda coordinar acciones. Por último, la entrega de vídeo sobre situaciones de emergencia tras grandes desastres no debe basarse en infraestructura fija, ya que, por ejemplo, en un terremoto las estaciones base celulares pueden verse afectadas [8].

En la difusión con múltiples saltos, los paquetes pueden corromperse y perderse, debido por ejemplo al desvanecimiento, interferencia del entorno, y la movilidad intrínseca de la red. En [8], se propone combinar el uso de diversidad de caminos junto con una estrategia de codificación de red. La estrategia tiene en cuenta escenarios (ver Figura 8) donde los autos pueden formar pelotones, por ejemplo, los pelotones Pf1, Pf2, Pr1, y Pr2, por lo tanto la información tendrá un retardo hasta llegar desde el primer pelotón  $P_{f1}$  hasta el segundo  $P_{f2}$  que llegue por detrás. Además, estos pelotones

---

<sup>6</sup> MOS: El *Mean Opinion Score* (MOS) recomendados por la Unión Internacional de Telecomunicaciones (UIT) es la escala común para la medición de la calidad de vídeo. Se puede dividir en cinco niveles que corresponden a la percepción de los usuarios de la siguiente manera: 5 (excelente), 4 (bueno), 3 (aceptable), 2 (pobre), 1 (malo).

estarán desconectados. Para mitigar este desafío se usa en enfoque de “mulas de datos” (del inglés *Muling Data*), es decir, aprovecharse de los vehículos que vienen en la dirección opuesta, en la figura los pelotones  $P_{r1}$ , y  $P_{r2}$  se comportan como mulas de datos, suponiendo que la carretera tiene varios carriles en ambas direcciones, los vehículos recogerán, transportarán y retransmitirán los paquetes [8].

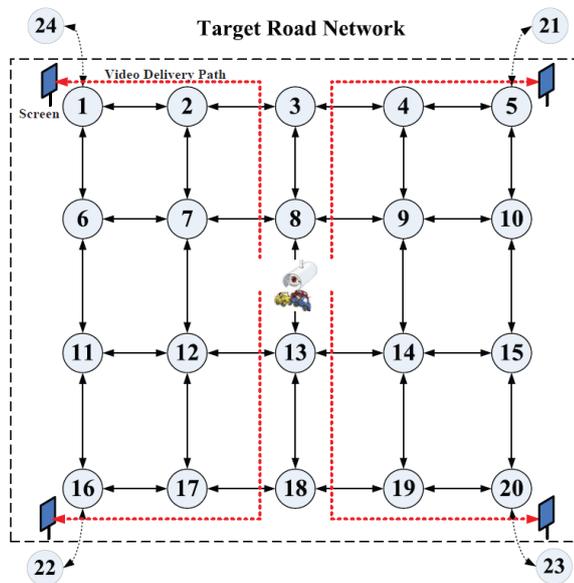


**Figura 8.** Escenario de un modelo de retransmisión [8].

### 2.6.5. Opportunistic Routing for Live Video Streaming in Vehicular Ad Hoc Networks

El protocolo propuesto toma en cuenta la interferencia y formula un modelo para la distorsión en *LVS* (del inglés *Live Video Streaming*) entregado sobre VANETs. Sobre la base de este modelo de distorsión del video, se diseña un método para la selección óptima del nodo repetidor, a través de este método se puede calcular la predicción de la tasa de distorsión mínima de transmisión de video a lo largo de cada segmento de carretera. Además, de acuerdo con éstas estimaciones de distorsión se podría seleccionar un segmento de ruta óptima. La parte central del protocolo es tomar en cuenta el retardo de *video streaming*, para lograr aquello se adapta un nuevo algoritmo que hace que los vehículos cambien su modo de entrega de forma adaptativa, y por lo tanto hace del reenvío de paquete de vídeo una estrategia de entrega llamada estrategia oportunista [9].

En la Figura 9 se muestra el escenario, donde cada círculo representa una intersección, se plantea desplegar pantallas a las afueras de la zona de objetivo, cuando existe congestión de tráfico la información de video se transmite en vivo hacia las pantallas para advertir a los conductores. Para evitar el problema de bucles de enrutamiento se deberá garantizar que los paquetes de video con el mismo destino se entreguen unidireccionalmente, para ello se usa un algoritmo propuesto en [9], se asume que los vehículos en la zona de destino están equipados con sistema de posicionamiento y navegación. Cada vehículo recoge continuamente su posición, velocidad, dirección, interferencia, y transmitir periódicamente a sus vecinos [9].



**Figura 9.** Transmisión de video desde el lugar del accidente hacia las pantallas de destino [9].

El protocolo tendrá en cuenta la interferencia y el tiempo de entrega de los paquetes, relacionando el error y el retardo del paquete. Por ejemplo, si el vehículo tiene la oportunidad de entregar el paquete, éste decidirá si entregarlo o no basados en el cálculo de la distorsión al entregar el paquete, y si no se entrega se incrementa el retardo. Por lo tanto, ese acarreo conlleva a que haya una distorsión de acarreo, así, si la distorsión de reenvío es menor a la distorsión de acarreo, se transmite el paquete,

caso contrario se espera hasta la siguiente oportunidad para entregar el paquete. Esta estrategia cae dentro del concepto de enrutamiento oportunista [9].

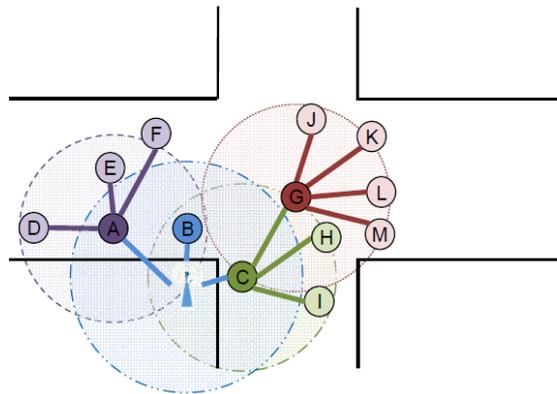
#### **2.6.6. ReViV: Selective Rebroadcast Mechanism for Video Streaming over VANET**

En un ambiente urbano la disponibilidad de espectro se hace cada vez menor, a su vez, en VANETs el uso de la transmisión tradicional incita a recibir un paquete y retransmitirlo, provocando más canales de transmisión ocupados, interferencia, pérdida de paquetes, por tanto, se produce degradación de la calidad de video [10].

En [10] se plantea el Protocolo ReViV, diseñado para ambientes urbanos donde los canales están sobrecargados y la red sufre un alto grado de interferencia. ReViV plantea un mecanismo que selecciona un mínimo subconjunto de vehículos de retransmisión con el objetivo de minimizar la interferencia y lograr una alta calidad en el video transmitido. Los vehículos se clasifican según su posición estratégica en la red y su capacidad para alcanzar a otros vehículos. Los nodos se clasifican basados en una nueva métrica de centralidad llamada “*Dissemination Capacity, DC*”, además la retransmisión se realiza sin ningún tipo de apoyo de los RSU. Los nodos deberían estar lo más centrales posibles con el fin de retransmitir a la mayor cantidad de vecinos sin adicionar muchas retransmisiones. Consecuentemente se logra minimizar el retardo extremo a extremo. Estas mejoras no implican que se sobrecargará la red ya que el cálculo lo hace cada vehículo y el resultado se lo comunica periódicamente dentro de la señal *beacon* [10].

La selección del nodo central se basa en el método *Social Network Analysis (SNA)*. Un ejemplo se puede ver en la Figura 10, en la esquina se está llevando a cabo una transmisión, solo los vehículos A, B, C están en el rango de alcance, solo A y C son escogidos para la retransmisión hacia todos los demás nodos usando Ad-Hoc. Después,

solo G es escogida para retransmitir el contenido, con este método se han eliminado tres nodos redundantes B, H e I [10].



**Figura 10.** Ejemplo de la selección de nodos de retransmisión [10].

Al aumentar la velocidad del *streaming* con el protocolo ReViV se hace más notorio la superioridad sobre otros protocolos como IEEE 1609.4 y AID, ya sea que consigue menor porcentaje de pérdida de paquetes, mayor PSNR y menor retraso en la entrega de paquetes. El desempeño de Reviv es tangible en escenarios con gran tráfico donde existe alta densidad de transmisión [10].

### **2.6.7. A Hybrid Video Dissemination Protocol for VANETs: HIVE**

Un protocolo robusto y eficiente debe garantizar la calidad de los vídeos transmitidos a través de una red en términos de calidad de servicio (QoS) y la calidad de la experiencia del usuario (QoE). Muchos protocolos se enfocan en mejorar el QoS sin considerar el QoE, ya que a pesar de que en algunos casos una pequeña pérdida de paquetes lleva a proporcionar un servicio de mala calidad [11].

El Protocolo híbrido de Difusión de vídeo (HIVE) implementa una técnica basada en el receptor<sup>7</sup> (del inglés *receiver-based*) para la selección eficiente del nodo de reenvío, además HIVE implementa un mecanismo de control de congestión en la subcapa *Media Access Control (MAC)*; así evita la colisión de paquetes y la latencia en la red. Una combinación de estas dos técnicas de forma integrada proporciona altas velocidades de entrega de paquetes. HIVE es capaz de recuperar en el receptor casi todos los paquetes de vídeo, aplicando la técnica de Codificación *Erasure*<sup>8</sup> (CE). Todas estas técnicas se combinan de una manera eficiente para garantizar una baja tasa de retardo y una sobrecarga de transmisión razonable mientras que los usuarios pueden experimentar una buena calidad de vídeo, este protocolo cumple con los requisitos de escalabilidad y de retardo para la transmisión de vídeo [11].

La cantidad de la pérdida de los paquetes es un indicador de QoS, en una transmisión se debería entregar el 95% de paquetes con un retardo de 4 a 5 segundos [17]. Por otro lado, el valor de PSNR puede usarse como un parámetro de medida de QoE al comparar la calidad del video transmitido versus el video receptado. Los datos multimedia son intrínsecamente, grandes y las colisiones de paquetes, tormentas de *broadcast* son muy comunes y suceden fácilmente cuando un gran número de vehículos retransmite los paquetes al mismo tiempo [19]. Debido a esto sobre este protocolo se han hecho aún más mejoras para permitir difusión sobre VANETs [11].

De acuerdo a los resultados mostrados en [20], el esquema de reenvío *receiver-based* supera al esquema *sender-based*<sup>9</sup> en términos de colisiones y las cargas generales; por lo tanto; este esquema es más capaz de proporcionar servicios de difusión de vídeo a través de redes vehiculares [11].

---

<sup>7</sup> *Receiver-Based*: Un protocolo de enrutamiento se clasifica como *receiver-based* cuando un nodo vecino recibe el paquete, y éste decide si debería o no ser retransmitido [28].

<sup>8</sup> Codificación *Erasure*: Consiste en dividir el mensaje en  $m$  bloques de datos, y luego los reconvierte en un gran conjunto de  $n$  bloques (mensajes codificados), de tal manera que el mensaje original puede ser construido usando cualquier subconjunto de bloques  $m$ . Usualmente  $n$  mayor a  $m$ ,  $R=m/n$  menor a 1, y  $R$  es el factor de codificación erasure [30].

<sup>9</sup> *Sender-Based*: Un nodo recibe el paquete, luego selecciona un subconjunto de vecinos a quienes reenviar el paquete, después el nodo reenvía hacia los nodos seleccionados, el proceso se repite en cada nodo [28].

*Reactive, Density-aware and Timely Dissemination Protocol (REACT-DIS)* [21] es una solución *receiver-based* que se implementa en la capa de red. Este protocolo previene el retardo de paquetes, que es un problema típico de los mecanismos de reenvío *receiver-based*, este logro se lo alcanza al mantener los mismos nodos de retransmisión durante cantidades específicas de tiempo. De igual manera, con considerar un número de copias de los mismos paquetes para determinar cuáles paquetes pueden reenviarse se intenta aliviar la sobrecarga total de transmisión [11].

Al tener menos fuentes de retransmisión e incluir la alta probabilidad de pérdida de paquetes en VANETs, se hace difícil la detección de bits del *stream* de vídeo en el nodo receptor, por lo tanto, la codificación de vídeo en la capa de aplicación es una solución para ello [22].

Técnicas de codificación como *Scalable Video Coding (SVC)*, o *Multi Description Coding (MDC)*, además, la redundancia de paquetes, la codificación *erasure* y la codificación de red también pueden ser implementadas en la capa de aplicación para maximizar la capacidad de recuperación de errores. También el enfoque de codificación de red basado en la Divulgación de Datos (CNDD del inglés *Network Coding based Data Dissemination*) [23] trabaja sobre la capa de aplicación, entonces, se aprovecha tanto la codificación de red como codificación en la fuente y nodos intermedios. Este mecanismo usa una técnica de recuperación de errores mediante el envío de mensajes de ayuda a los nodos vecinos solo cuando los paquetes adicionales son necesarios para cualquier nodo intermediario, así se logra mejorar la velocidad de la entrega de paquetes al transmitir un número menor de paquetes de vídeo [11].

El protocolo HIVE se basó en: WAVE-AOS<sup>10</sup>, REACT-DIS<sup>11</sup> y CNDD<sup>12</sup>; además implementó el concepto de paquetes redundantes y la codificación en la fuente (CE),

---

<sup>10</sup> WAVE-AOS: Este protocolo proporciona un mecanismo de control de congestión para evitar colisión de paquetes en áreas con una cantidad de nodos densa [11].

<sup>11</sup> REACT-DIS: Previene el retraso en los paquetes, la cual es una característica propia de los mecanismos *receiver-based* [11].

<sup>12</sup> CNDD: Técnica de recuperación de errores [11].

así como un esquema de filtrado no determinista<sup>13</sup>, estos métodos promueven la capacidad de recuperación de errores. Por lo tanto, con el fin de evitar la entrega de un alto porcentaje de paquetes duplicados y para evitar causar una alta sobrecarga en la red, un porcentaje de redundancia se utiliza para garantizar la entrega de los datos transmitidos a los receptores. Además, HIVE usa la Codificación *Erasure* debido a su bajo retardo y el potencial para manejar la pérdida de paquetes en las VANETs. Para evitar la difusión en tormenta, se lleva a cabo el método de la selección de un subconjunto de nodos de retransmisión, el acto mejora la fiabilidad del protocolo. La combinación de las técnicas mencionadas satisface los requisitos de calidad de la transmisión de vídeo en carretera [11].

#### **2.6.8. Beacon-less Video Streaming Management for VANETs Based on QoE and Link-Quality, QOALITE**

Al hablar de LVS en VANETs se debe sortear retos como la calidad del video o lidiar con las topologías altamente dinámicas propias de las redes VANETs, para ello se han propuesto enfoques *beacon-less* o *geographic receiver-based*. En general, las decisiones de enrutamiento se realizan sólo en base a las características del enlace y la ubicación del vehículo de la red, en resumen, tomando en cuenta solo las características de la red, los enlaces o nodos. Sin embargo, en situaciones reales estas decisiones de enrutamiento no son aprovechables para seleccionar los mejores nodos de retransmisión. El protocolo *QoE-Driven and Link-Quality Receiver-based (QOALITE)* permite LVS con la garantía de la calidad del video y de QoE en escenarios vehículo-a-vehículo (V2V) [12].

Un requisito fundamental en LVS es disminuir el tiempo empleado en el proceso de selección de nodos de retransmisión. Existen una variedad de soluciones, por un lado, los enfoques basados en el remitente (*sender-based*) que dependen de las rutas extremo a extremo, las cuales sufren de frecuentes interrupciones en VANETs. Por otro lado,

---

<sup>13</sup> Filtrado no determinista: También llamado filtrado estocástico, es el proceso de usar observaciones parciales y métodos estocásticos para hacer inferencias acerca de un sistema [31].

las estrategias no estáticas *receiver-based* (RB) mejoran la transmisión de paquetes a través de una decisión de enrutamiento *hop-by-hop* [24]. El concepto RB implica la no necesidad de conocer a los vecinos, por lo tanto, se elimina la transmisión de señal *beacon*, ahorrando ancho de banda, además permite la persistencia de los flujos en caso de errores de nodo, la movilidad, y las variaciones del canal inalámbrico [12].

“Los nodos de reenvío transmiten el paquete a sus vecinos a un salto, los cuales inmediatamente después de recibirlos activan un conteo regresivo, el primer nodo de expirar su temporizador de cuenta atrás, envía los paquetes recibidos más allá. Es así que los enfoques RB *beacon-less* no exploran múltiples criterios para construir caminos de enrutamiento. Parámetros, como el tipo de trama y la estimación de la distorsión deben ser utilizados como indicadores de QoE para apoyar la toma de decisión en cuanto al reenvío a nivel de enrutamiento. Por lo contrario, algunos protocolos consideran sólo información geográfica para las decisiones de enrutamiento. Sin embargo, debido a la imprevisibilidad de los entornos inalámbricos, el nodo más lejano podría sufrir de mala calidad del enlace, o no puede recibir los mejores *frames* de una secuencia de vídeo desde el punto de vista humano. Por lo tanto, es necesaria una propuesta que considere nodo, enlace, red, aplicación y percepción humana” [12].

QOALITE combina la información de la ubicación, las condiciones actuales de calidad del enlace, indicadores de QoE para así establecer redes troncales *multi-hop* para la difusión de *live video streaming* sobre VANETs, esto también implica que reduce los retardos *Hop-by-Hop* y *packet-by-packet*. El protocolo crea y controla rutas V2V para transmisiones de video en vivo, reaccionando bien a las caídas de los nodos, y mejora la experiencia del usuario o en el peor de los casos mantiene el nivel de calidad de la experiencia de los videos transmitidos en comparación con los esquemas impulsados por mecanismos no QoE. Por lo tanto, QOALITE mejora la transmisión de videos en la carretera para distancias medias (por ejemplo, 1 a 5 Km) [12].

QOALITE considera un enfoque RB sin señal *beacon* para seleccionar nodos y construir redes troncales, además incluye calidad del enlace, información de movilidad e indicadores QoE [12].

El mecanismo considera dos fases; *Distribution Contention-Based Forwarding (DCF)* y *Multi-Hop Backbone Forwarding (MBF)*. En DCF los candidatos a nodos de retransmisión empiezan la transmisión y compiten para participar en la fase MBF desde la fuente hacia el destino. Para la clasificación de los nodos se usan tres criterios: información de localización, calidad actual del enlace e indicadores QoE. La siguiente fase es MBF, la cual ofrece el reenvío usando el *backbone* que se construye en la anterior fase, y permite cambios dinámicos a otros caminos en caso de fallos de enlace y la pérdida de calidad. Esta fase mitiga retrasos adicionales, interferencia y paquetes duplicados entregados por la fase DCF. Durante la transmisión, el contenido de vídeo debe ser entregado incluso en presencia de errores de nodo o variaciones de canal. Estos problemas causan interrupciones, siendo indeseable para la QoE. QOALITE detecta fallos de ruta, proporcionando una gestión de rutas más suave [12].

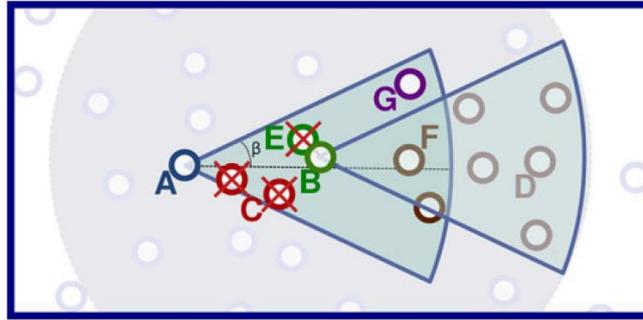
Cualquier *backbone* construido regresa a la fase de DCF cuando se detecta que la calidad del enlace o la distorsión caen por debajo de un umbral predefinido. Además, cualquier nodo que compone un *backbone* de transmisión considera que la ruta ya no es válida, siempre y cuando ya no reciba ningún mensaje de respuesta de su *hop* anterior dentro de un cierto período de tiempo, por lo tanto, vuelve a la fase de DCF para establecer un nuevo *backbone*. QOALITE determina el mejor vehículo para la retransmisión en cada salto [12].

### 2.6.9. A Reactive and Scalable Unicast Solution for Video Streaming over VANETs

El protocolo *Video Reactive Tracking-based Unicast (VIRTUS)* adopta una estrategia *receiver-based* donde la selección del nodo de retransmisión se realiza dentro de una ventana de tiempo y no en una única transmisión. Además, para mejorar esa selección se usa la información de locación para determinar la actual y futura posición [13].

Un nodo de reenvío en una solución *receiver-based* transmite el paquete a sus vecinos y a partir de su recepción los nodos vecinos se programan a sí mismos para retransmitir el paquete más allá. El primer nodo en expirar su tiempo de espera reenvía el paquete recibido mientras que cancela los demás nodos previamente programados con el fin de evitar transmisiones innecesarias. Los siguientes nodos realizan el mismo proceso hasta llegar al extremo. El tiempo de espera está relacionada con la idoneidad del nodo para ser un nodo de transmisión, los nodos considerados como más adecuados para retransmitir un paquete deben tener un tiempo de espera más corto [13].

Un concepto fundamental de VIRTUS es su zona de reenvío que limita la selección de los nodos de retransmisión a un sector de la región limitada por el alcance de comunicación de nodos. Este sector se define como la región dentro de un ángulo  $\beta$  de la línea que conecta tanto el nodo de reenvío y el nodo de destino (como se ve en la Figura 11). Este mecanismo evita la formación de bucles y evita que los nodos desconectados el uno del otro decidan convertirse en nodos de retransmisión.  $\beta = 30$  grados es el ángulo máximo que puede formar una zona de reenvío [13].



**Figura 11.** Reenvío, cancelación y programación de nodos en la ruta [13].

Un problema del concepto *receiver-based* para *vídeo streaming* es su alto retardo de extremo-a-extremo causado por el tiempo de espera necesario para retransmitir los paquetes entrantes. VIRTUS disminuye este problema al mantener el proceso de elección del nodo de retransmisión durante una ventana de tiempo. En VIRTUS, cuando un nodo receptor decide que es un nodo de retransmisión adecuado (es decir, su tiempo de espera expira y no se observó ninguna otra retransmisión del mismo paquete), se considera de esta manera durante una ventana de tiempo de  $\lambda$  segundos. Durante esta ventana, cualquier paquete recién recibido se reenvía inmediatamente. La duración de esta ventana de tiempo se denomina tiempo de reserva ( $\lambda$ ) y este es el período de tiempo que tarda en estimarse el nuevo nodo de retransmisión. La estimación del tiempo de un nodo se basa en la información de localización recibida que se almacena en las cabeceras de los paquetes entrantes [13].

La decisión de VIRTUS sobre la idoneidad de los nodos para transmitir más paquetes entrantes es un enfoque híbrido que toma en consideración el avance geográfico hacia el destino y la estabilidad del enlace. La estabilidad del enlace se calcula en base al tiempo de reserva  $\lambda$  (máximo valor máximo pre-definido de 5s). La idoneidad de un nodo para retransmitir un paquete se define por el tiempo  $\gamma$  que espera para decidir si un nodo se convierte en un nodo de retransmisión. Se calculan los dos tiempos de espera sobre la base de sólo los avances geográficos  $\gamma_{geo}$  y  $\gamma_{stab}$ ; y, el valor de  $\gamma$  viene a ser una combinación de estos dos elementos. Usando los valores descritos anteriormente también se determina el máximo valor de espera  $\Gamma$  [13].

“El seguimiento de ubicación (del inglés *Location Tracking*) es la capacidad para detectar y realizar el seguimiento del estado de un objetivo, o un conjunto de objetivos. El seguimiento se define como un problema de estimación en base a una serie de medidas: el objetivo es estimar el estado de destino y actualizar la estimación con las mediciones” [13].

Un sistema de seguimiento de objetivos se puede realizar de dos maneras: (i) de predicción, que utiliza un modelo de movimiento para determinar el estado de destino después de un tiempo, y (ii) de corrección, que utiliza las últimas mediciones para actualizar la función de densidad de probabilidad del objetivo con el paso de tiempo. VIRTUS usa el seguimiento mediante predicción con el fin de estimar el tiempo de reserva  $\lambda$ . El rendimiento de las soluciones *receiver-based* baja al aumentar las tasas de transmisiones. Las altas tasas de transmisiones incurren en colisiones más frecuentes que impiden la transmisión por el nodo de retransmisión más adecuado, estas retransmisiones innecesarias congestionan aún más el medio compartido [13].

Para solucionar el problema VIRTUS desacopla el proceso de transmisión de vídeo del proceso de selección de nodos de retransmisión. Con este mecanismo un paquete de control se transmite periódicamente desde la fuente al destino con el objetivo de desencadenar el proceso de selección de nodo de retransmisión. El tiempo entre los paquetes de control enviados consecutivamente por el nodo de origen se refiere como  $\rho$ . Por otra parte, cuando se transmiten paquetes que contienen el vídeo, los nodos que reciben comprueban sus estados previamente asignados y reenvían los paquetes si se establecen como nodos de retransmisión. En este proceso los nodos VIRTUS pueden caer dentro de uno de los tres posibles estados: inactivo, programado o retransmisor [13].

Otra de las causas de la congestión de comunicación cada vez más común es la transmisión a través de las regiones de alta densidad. VIRTUS tiene que equilibrar la desventaja de menor retardo de extremo a extremo y transmisiones simultáneas en

función de la densidad de vehículos. Este proceso de selección disociada del nodo repetidor aplica una política que tiene como objetivo adaptarse dinámicamente dependiendo de la densidad local. Los valores pequeños en el tiempo máximo de espera ( $\Gamma$ ) en regiones densas podría llevar a un mayor número de colisiones debido a los tiempos de espera de los diferentes nodos demasiado cerca uno del otro. Los valores grandes de  $\Gamma$  no conducen a altos retrasos en regiones densas, esto se debe a que hay una mayor probabilidad de tener nodos en mejores situaciones para retransmitir los paquetes; de este modo, se consiguen tiempos de espera inferiores. Por lo tanto, VIRTUS tiene un mecanismo que detecta la ocurrencia de múltiples nodos reenviando paquetes en la misma región, lo que entonces aumenta el valor de  $\Gamma$  [13].

#### **2.6.10. The K-hop Cooperative Video Streaming Protocol Using H.264/SVC Over the Hybrid Vehicular Networks**

El *Cooperative video streaming* permite pedir a otros miembros del mismo grupo descargar un video de forma cooperativa, es decir los miembros que se unen ayudan a descargar partes del video desde internet, y después envían los datos salto por salto a través de una red Ad-Hoc hacia el nodo que inicialmente hizo la petición. El protocolo *K-hop cooperative video streaming protocol using H.264/SVC* se despliega sobre redes híbridas como una red celular y otra red DSRC [14].

Con el fin de obtener una reproducción fluida del video sobre la red Ad-Hoc el protocolo propone: (1) un esquema de asignamiento de tareas de *streaming* que maneje las tareas salto-por-salto dentro de la red vehicular, y (2) estrategias de reenvío de paquetes que decidan la secuencia de reenvío de los datos de video [14].

### 2.6.10.1. Streaming Task Assignment Scheme

Los ayudantes, es decir los nodos que ayudan a realizar las tareas, se seleccionan utilizando el enfoque propuesto en [27]. En esta fase el solicitante elige algunos miembros y pide que proporcionen su ancho de banda 3G/3.5G para descargar los datos de vídeo. Sin embargo, el solicitante tiene que decidir (1) la forma de dividir el vídeo en múltiples “tareas de *streaming*” para los ayudantes y el reenvío de los datos de vídeo *hop-by-hop* hacia el solicitante y (2) decidir la carga de trabajo adecuada para cada ayudante ya que cada uno tiene capacidad diferente [14].

Inicialmente, el solicitante remitirá el perfil de los datos de vídeo para empaquetar todos esos datos de vídeo en múltiples Unidades de Asignación (UA del inglés *Assignment Units*). Cada unidad de asignación es la unidad básica de una tarea *streaming*, así, la tarea de *streaming* asignada a un ayudante se compone de varias UA's. Después de la partición de los datos de vídeo en múltiples UA, el solicitante puede iniciar el procedimiento de asignación de tareas de *streaming*. En primer lugar, el solicitante hará la programación inicial y determinará un intervalo de asignación, este intervalo se utiliza para decidir cuándo el siguiente proceso de asignación de tareas de transmisión se activa de nuevo para asignar a los ayudantes tareas de *streaming*. Después, el solicitante envía las solicitudes de datos de vídeo hacia los ayudantes a través de la red 3G/3.5G y espera la transmisión de los datos de vídeo de los ayudantes durante el intervalo de asignación. Cuando el temporizador del intervalo de asignación expira, el solicitante estima la cantidad apropiada de las AU como la carga de trabajo de las tareas de *streaming* para cada ayudante en el siguiente intervalo de asignación. A partir de entonces, el solicitante sigue un procedimiento para asignar las AU a cada ayudante hasta llegar a la capacidad estimada de UA. Después de determinar la tarea de *streaming* para cada ayudante, el solicitante envía la solicitud a cada ayudante de nuevo a través de redes 3G / 3.5G y espera a que el intervalo de asignación repita el procedimiento de estimación de nuevo. Se repetirá este procedimiento hasta que todas las UA de los datos de vídeo se hayan asignado [14].

### 2.6.10.2. Packet Forwarding Scheme

Dado que los datos de video son enviados hacia atrás por medio de varios ayudantes la calidad de transmisión de extremo a extremo se ve muy afectada. Para observar los datos almacenados, el video debe ser recibido simultáneamente desde el servidor (ayudante). Un ayudante con menor salto almacena más datos que uno que tiene mayor número de saltos [14].

En una DSRC el ancho de banda es limitado, en los saltos existe adición de retardo y además debido al entorno existe degradación en la señal transmitida. Debido a esta problemática el reenvío puede tomar cierta estrategia, por ejemplo, estrategias *First In First Out* (FIFO), *Playback Priority First* (PPF), o *Bandwidth AWare* (BAW) [14].

### 2.6.11. CLWPR - A Novel Cross-Layer Optimized Position Based Routing Protocol for VANETs

El protocolo es diseñado para ser un protocolo oportunista, no hay una selección de la ruta antes de la diseminación de datos, solo se selecciona el siguiente salto de acuerdo al mínimo peso de los vecinos. CLWPR es un protocolo basado en la posición, que usa la distancia de la vía como métrica, en vez de usar la distancia geográfica, y para ello usa *greedy forwarding*<sup>14</sup>, éste mecanismo no calcula la distancia geográfica, en vez de eso calcula la distancia que el vehículo tendrá que recorrer hacia el destino, la diferencia es que en este caso la distancia se basa en los autos y sus movimientos. La distancia se restringe a seguir caminos y esquinas hasta llegar al siguiente salto, para lograr esto se deberá tener disponible mapas digitales [41].

---

<sup>14</sup> *Greedy Forwarding*: es un algoritmo de enrutamiento geográfico eficiente y escalable para redes inalámbricas. Las ventajas de este algoritmo incluyen: (1) baja complejidad computacional en el nodo, (2) poco espacio en la cabecera de un paquete, y (3) la capacidad para conseguir longitudes de trayectorias cortas en menores números de saltos [36].

El algoritmo usa la predicción de la posición de los nodos e información de navegación con el fin de mejorar la eficiencia del protocolo dentro de la VANET. Además, usa información de la calidad del enlace en términos de SINR y tasa de error de *frame* en las capas PHY y MAC, con el propósito de disminuir el retardo extremo a extremo y estimar la calidad del enlace. También se considera la información de las colas en términos de utilización de los nodos. Toda esta información es combinada para calcular el peso de cada nodo vecino [41].

Cuando el nodo tiene que enviar un paquete se calcula un peso para cada destino dentro de su lista, con este método solo se calcula el nodo de reenvío y no se necesita conocer toda la información de la red. En el caso que el nodo no tenga ningún paquete que enviar, no se necesita calcular nada y así se reducen los cálculos a realizar. El cálculo del peso se basa en la información de un salto obtenida en base de los mensajes “HELLO” (también llamados mensajes *beacon*). Dicha información es difundida periódicamente y con gran frecuencia por cada nodo, sin embargo, esto incrementa la carga de la red. El mensaje “HELLO” contiene información como velocidad, posición, encabezado, etc. [41].

La información de las colas también se añade al mensaje “HELLO”, esto con el fin de balancear el tráfico en la red. La tabla 1 resume los valores dentro del mensaje “HELLO”.

**Tabla 1.** Información contenida en el mensaje “HELLO” [41].

<b><i>Information Carried</i></b>	<b><i>Value Range</i></b>
<i>Node Position (x,y)</i>	<i>(Double, Double)</i>
<i>Node Velocity (x,y)</i>	<i>(Double, Double)</i>
<i>Node Heading</i>	<i>Integer (0° – 360°)</i>
<i>Road ID</i>	<i>Integer</i>
<i>Node Utilization</i>	<i>Integer</i>
<i>MAC Frame Error rate</i>	<i>Double</i>
<i>Number of Cached Packets</i>	<i>Integer</i>

## **CAPÍTULO III**

### **3. ANÁLISIS Y DISEÑO DEL SISTEMA**

En este capítulo se estudian los protocolos escogidos para la transmisión de *video streaming* sobre VANETs. En este experimento de simulación se escogieron tres tipos de protocolos de encaminamiento, los cuales son AODV, OLSR y CLWPR. Los dos primeros son de tipo reactivo y proactivo en MANETs, y el tercero geográfico para VANETs.

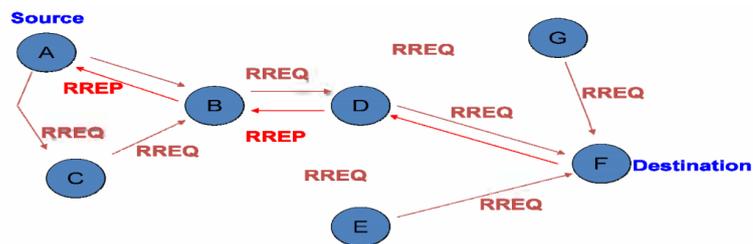
Aquí también se presenta el proceso y las consideraciones tomadas en cuanto al escenario y a los parámetros para la simulación en un ambiente vehicular urbano. Se indica la selección de los nodos de comunicación de acuerdo a su posición dentro del escenario; esto con el objetivo de realizar una simulación lo más cercana a la realidad. También se detalla el uso de SUMO en esta simulación, herramienta que sirvió para construir los patrones de movilidad de los vehículos. Finalmente se describe una tabla con todos los parámetros de simulación que serán configurados en NS-3.

#### **3.1. Selección de los Protocolos**

##### **3.1.1. Ad-Hoc On-Demand Distance Vector (AODV)**

El protocolo AODV es uno de los protocolos de enrutamiento reactivos más usados para redes móviles Ad-Hoc. AODV mantiene una tabla de rutas para almacenar la información de enrutamiento del siguiente salto para llegar el nodo destino. Cada tabla de enrutamiento puede ser usada por un periodo de tiempo, si la ruta no es requerida dentro del periodo, ésta expira y una nueva ruta necesita ser encontrada cuando sea necesario; es decir, simplemente crea una ruta cuando se necesita [49].

Cuando un nodo fuente tiene datos para enviar, éste busca la ruta en la tabla de enrutamiento; si existe una ruta dentro de la misma, la utiliza para transmitir el paquete de datos, caso contrario éste inicia el proceso de descubrimiento. Para ello, difunde un mensaje *Route Request (RREQ)* a todos sus vecinos. Si uno de los vecinos conoce la ruta hacia el destino, responde con un mensaje *Route Reply (RREP)*, si no es así, cada nodo sigue transmitiendo el mensaje hasta que llega a su destino, quien responderá con el paquete RREP, ver Figura 12. Cada nodo intermedio registra esta ruta para que pueda utilizarlo en futuras comunicaciones. Para el proceso de mantenimiento, cada nodo envía periódicamente un mensaje “*HELLO*” para detectar si hay un fallo en el enlace [51].



**Figura 12.** Mecanismo de trabajo de AODV [50].

El protocolo de enrutamiento AODV puede manejar tasas de movilidad bajas, moderadas y relativamente altas, así como una variedad de niveles de tráfico de datos. AODV se ha diseñado para reducir la diseminación del tráfico de control y eliminar la sobrecarga en el tráfico de datos, con el fin de mejorar la escalabilidad y rendimiento [49].

### 3.1.1.1. Mantenimiento de Rutas

Durante una comunicación entre el nodo fuente y el destino puede darse que algún nodo modifique su posición, esto conlleva a que se rompa el enlace y que la ruta quede

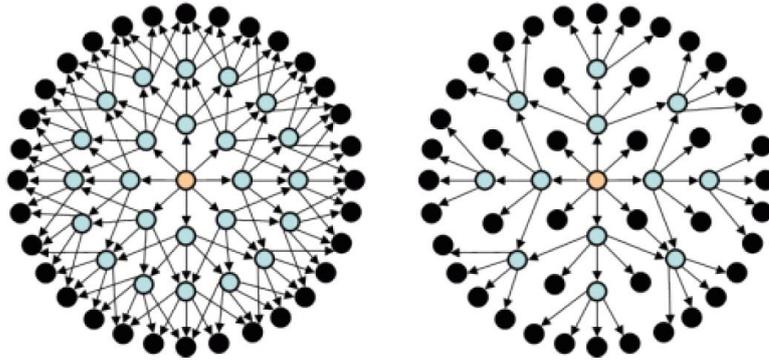
inutilizable. El nodo vecino que descubre el enlace roto es el encargado de informar al resto mediante el envío del mensaje RERR (del inglés *Route Error*) [49].

El mensaje RERR es similar al RREP pero con un número de salto (*hop count*) igual a infinito, lo que hace que cualquier otra ruta sea mejor y deban reencaminarse los paquetes por otro camino. De esta manera, el nodo fuente decide si la comunicación con el nodo destino ha terminado o si se debe iniciar un nuevo proceso de descubrimiento de ruta [49].

### **3.1.2. Optimized Link State Routing (OLSR)**

OLSR es un protocolo de enrutamiento proactivo para redes móviles Ad-Hoc. Es un protocolo de enrutamiento dinámico donde las rutas están construidas con anterioridad. Se caracteriza por intentar optimizar el número de mensajes de control que se producen en la red [49].

OLSR minimiza la sobrecarga del tráfico de control usando solo nodos seleccionados llamados MPRs (del inglés *Multipoint Relays*). Cada MPR elige un pequeño subconjunto de nodos, suficiente para cubrir una distancia de hasta dos saltos (ver Figura 13), los cuales reducen la cantidad de sobrecarga de intercambio de datos de enrutamiento. Esta técnica reduce significativamente el número de retransmisiones requeridas para difundir un mensaje a todos los nodos en la red. OLSR requiere conocer solo el estado parcial del enlace, para difundir con el fin de proporcionar rutas cortas de trayectoria. Los nodos seleccionados para que sean MPR toman la responsabilidad para reenviar los mensajes de enrutamiento [49].



**Figura 13.** Comparación del flujo de mensajes originado por un único mensaje de difusión. A la izquierda todos los nodos retransmiten el mensaje; a la derecha solo lo hacen los nodos seleccionados como MPR (nodos en azul) [57].

Los nodos designados como MPR se les permiten generar actualizaciones de estado de enlaces entre nodos MPR y sus nodos selectores con el propósito de mantener pequeño el tamaño de la tabla de actualizaciones. Por consiguiente, se pone a disposición a cada nodo solo información parcial de la topología; sin embargo, esta información es suficiente para calcular localmente el camino más corto para todos los demás nodos que están conectados a los nodos MPR. OLSR solamente utiliza actualizaciones periódicas para la disseminación del estado del enlace.

Cada nodo debe mantener la tabla de enrutamiento que contiene las rutas para cada nodo en la red. Esta tabla se actualiza periódicamente por si algún cambio ha ocurrido. Cada nodo envía un mensaje “HELLO” a todos sus nodos vecinos que contendrán información sobre la totalidad de sus vecinos. Por lo tanto, cada nodo ahora tiene información acerca de los nodos a dos saltos (*2-hop*), esta información se usa para seleccionar el MPR en base al cual vecino proporciona la mejor ruta a vecinos a dos saltos [51].

La información de la topología se utiliza para producir la tabla de enrutamiento que luego será distribuida con el mensaje de control de topología (TC). MPR y TC se utilizan para distribuir la información a cada vecino a través de la red [51].

### 3.1.3. CLWPR

En la actualidad las investigaciones en redes VANET se centran en técnicas de *cross-layer* con el fin de aumentar la fiabilidad del protocolo de enrutamiento. La idea principal detrás de esto es utilizar la información sobre la calidad del enlace y el estado del canal para aumentar la fiabilidad de ese enlace y actuar en consecuencia en el protocolo de enrutamiento.

Los primeros protocolos basados en la posición se introdujeron en la década de 1980. Uno de los protocolos de enrutamiento basado en la posición es GPSR, diseñado para MANET, que cuenta con el modo *Greedy* y *Perimeter*. El primero selecciona el siguiente salto en base a la distancia euclidiana mínima desde el destino, mientras que el segundo se selecciona cuando el nodo de transmisión se enfrenta a máximos locales<sup>15</sup>, y se reenvía el paquete a lo largo del perímetro nulo.

CLWPR fue basado en el protocolo OLSR que envía mensajes “HELLO” periódicamente además también proporciona facilidades de HNA<sup>16</sup> [41].

#### 3.1.3.1. CLWPR (Cross-Layer, Weighted, Position-based Routing)

CLWPR es un protocolo de enrutamiento basado en la posición, diseñado para redes VANET. Este protocolo utiliza la información de movilidad de los nodos y la

---

<sup>15</sup> Máximo local: Problema que enfrentan cuando nodo actual se pone más cerca del destino que cualquier otro nodo vecino.

<sup>16</sup> HNA: (*Host and Network Association*) Un mensaje HNA es básicamente una lista de direcciones y máscaras de la red.

información de *cross-layer* de PHY y la capa MAC con el objetivo de aumentar su eficiencia, fiabilidad y disminuir el retardo extremo a extremo, soportando además el mecanismo *carry-n-forward* [41].

El protocolo es diseñado para ser un protocolo oportunista, no hay una selección de la ruta antes de la disseminación de datos, solo se selecciona el siguiente salto de acuerdo al mínimo peso de los vecinos. Este peso se basa en la información *1-hop* obtenida a base de los mensajes “*HELLO*”, dicha información es difundida periódicamente por cada nodo.

### **3.1.3.2.Repositorios de CLWPR**

CLWPR cuenta con repositorios básicos, *Neighbor Set* y *Position Association Set*.

*Neighbor Set* es la modificación de *NeighborSet* de OLSR con la finalidad de facilitar información adicional como la posición, velocidad, rumbo etc., este repositorio se utiliza para conocer la lista de nodos (*1-hop*), directamente accesibles. Cada vez que se recibe un mensaje “*HELLO*” se mira dentro de este repositorio y o bien se actualiza o se crea una entrada con los elementos de la Tabla 2 [41].

**Tabla 2.** Entrada de un vecino para CLWPR [41].

<b>Main Address</b>
Interface
Status
Position
Velocity
Heading
RoadID
Utilization
MAC Info
SNIR info
CnF info
Times-tamp
Expire-Time

Cada entrada será automáticamente eliminada después de un periodo de  $2.5 * HelloInterval\ time$ , con el fin de no mantener a vecinos que no estén dentro de la región, pero asimismo mantiene la información en caso de que el mensaje “HELLO” se pierda [41].

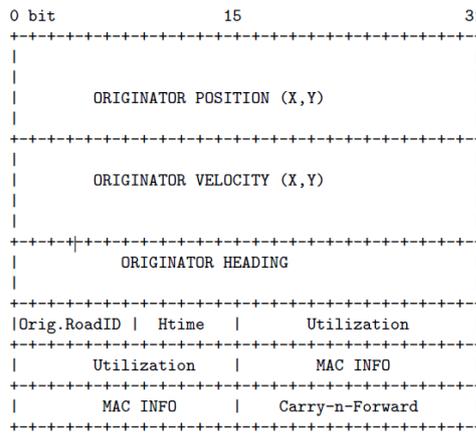
**Position Association Set** es la lista de nodos de destino. Incluye información relacionada con la movilidad del nodo de destino. La información almacenada en este grupo es mostrada en la Tabla 3.

**Tabla 3.** Posición basada en la entrada para CLWPR [41].

<i>Main Address</i>	<i>Position</i>	<i>Velocity</i>	<i>Heading</i>	<i>RoadID</i>	<i>Time-stamp</i>
---------------------	-----------------	-----------------	----------------	---------------	-------------------

### 3.1.3.3.Mecanismo de descubrimiento del vecino.

El mecanismo de descubrimiento de vecino se basa en la difusión periódica de mensajes “*HELLO*” de un salto, con el fin de proporcionar información de posición, información de navegación, así como, la información en diversas capas, la carga útil de estos mensajes “*HELLO*” incluye la siguiente información de los nodos: posición, velocidad, rumbo, identificación del camino, *Hello interval time*, la información de la capa MAC y el número de paquetes almacenados en el caché del mecanismo *carry-n-forward*. El formato del mensaje “*HELLO*” se lo puede ver en la Figura 14 [41].



**Figura 14.** Formato del mensaje “*HELLO*” de CLWPR [41].

El mensaje *HELLO* contiene información como velocidad, posición, encabezado, etc. el cual pasa por la capa MAC que es la responsable de su correcta difusión. El protocolo selecciona el siguiente salto entre los vecinos utilizando una métrica basada en la posición de ese nodo. Utiliza información adicional, como información del mapa y de navegación, para minimizar el efecto de máximo local. Utiliza una estrategia de recuperación para hacer frente a este problema [41].

### 3.1.3.4.El mecanismo de Reenvío

CLWPR usa la distancia de la vía como métrica, en vez de usar la distancia geográfica. Además, realiza un seguimiento a parámetros como SNIR y la tasa de error de *frame* MAC en las capas PHY y MAC con el propósito de estimar la calidad del enlace. Además, es tomada en consideración la información de las colas en términos de utilización de los nodos. Toda esta información es combinada para calcular el peso de cada nodo vecino.

El mecanismo de reenvío de CLWPR se basa en el mínimo peso, cada nodo calcula la distancia y el peso a partir de cada entrada que tiene en su *Neighbor Set (1-hop nodes)*. Para cada entrada dentro del *Position Association Set (destination nodes)*. Por lo tanto, si un nodo no tiene ninguna entrada en su *Position Association set*, no tendría que calcular ninguna tabla de enrutamiento. La tabla de enrutamiento de cada nodo se la aprecia en la Tabla 4 [41].

**Tabla 4.** Tabla de enrutamiento para CLWPR [41].

<i>Destination Address</i>	<i>Next Hop Address</i>	<i>Interface</i>	<i>Weight</i>
<i>Destination1</i>	<i>Neighbor1</i>	<i>if1</i>	<i>w11</i>
<i>Destination1</i>	<i>Neighbor2</i>	<i>if1</i>	<i>w12</i>
<i>Destination2</i>	<i>Neighbor1</i>	<i>if1</i>	<i>w21</i>
<i>Destination2</i>	<i>Neighbor2</i>	<i>if1</i>	<i>w22</i>

Además, cada vez que se cambia uno de los dos conjuntos, la tabla de enrutamiento se recalcula con el fin de comprobar si hay nuevas rutas a los paquetes en cola desde el mecanismo de CnF.

La métrica de reenvío en CLWPR es el resultado de la función de peso conjunta:

$$\begin{aligned} \text{Weight} = & f_1 * \text{Distance} + f_2 * \text{NormAngle} + f_3 * \text{NormRoad} + f_4 \\ & * \text{Utilization} + f_5 * \text{MAC info} + f_6 * \text{CnF info} + f_7 \\ & * \text{SNIR info} \end{aligned}$$

Donde:

*$f_i$* : es el factor de peso para cada parámetro.

***Distance***: es la distancia al destino, medida en la carretera.

***NormAngle***: es el peso normalizado para el parámetro de ángulo.

***NormRoad***: es el peso normalizado para el parámetro de la ruta.

***Utilization***: es el número de paquetes en la cola del nodo.

***MACinfo***: es el MAC Frame Error Rate.

***CnF info***: es el número de paquetes alcanzados a partir del mecanismo carry  
– n – forward

***SNIR info***: es el peso del valor de SNIR de los paquetes recibidos.

El nodo con el menor peso será seleccionado como el siguiente salto, si el nodo actual tiene menos peso (problema de máximo local), entonces el paquete es almacenado en el caché. En el caso que el nodo no tenga ningún paquete que enviar ya no se necesita calcular la tabla de enrutamiento y así se reducen los cálculos a realizar.

### 3.1.3.5. Mejoras utilizadas en el módulo CLWPR

Aparte de la información de la posición básica que se utiliza por el *greedy forwarding*, cuenta con grandes mejoras: información de navegación, información de *cross-layer* y el mecanismo *carry-n-forward* [41].

#### ***Información de navegación***

Usando la posición, velocidad y dirección se puede predecir la posición del nodo vecino y el nodo destino, ya que la movilidad es restringida a la topología del camino; de esta forma se reduce la frecuencia de los mensajes “*HELLO*”, aun así CLWR puede proporcionar alta tasa de entrega de paquetes cuando se utiliza la predicción. Los nodos están equipados con mapas digitales de la topología de la carretera y son capaces de encontrar el camino [41].

Fue desarrollada la clase NS3::gridmap la cual proporciona información relacionada con la navegación, tales como identificación del camino y la distancia “*courvetric*<sup>17</sup>”.

#### ***Cross-Layering***

La investigación se ha centrado en optimizaciones de capa cruzada para aumentar la fiabilidad de los enlaces de comunicación, con este fin, se utilizan dos indicadores claves de la calidad del enlace. Primero, los valores de SNIR de un paquete recibido y segundo el *Frame Error Rate* calculado en la capa MAC. Además, con el fin de equilibrar la carga de la red a través de diferentes nodos, también se utiliza el tamaño de la cola de la MAC de un nodo como un indicador de la utilización [41].

---

<sup>17</sup> *Courvetric*: distancia entre dos nodos siguiendo la topología de la carretera.

SNIR está disponible en *ns3::WiFiPhy* y específicamente el *ns3::YansWifiPhy*. Cuando se recibe un paquete (denotado por el método *YansWifiPhy::EndReceive ()*) los valores de la señal y el ruido (más interferencia) son calculados, con el fin de proporcionar esta información a las capas superiores (enrutamiento) se ha desarrollado una etiqueta (*struct SnrTag:public Tag*) que se las añade a cada paquete recibido, esta etiqueta es leída después en el protocolo de enrutamiento cuando se recibe un mensaje "HELLO" y el valor es almacenado junto con el resto de información en el *neighboring set*. En cuanto a la información MAC, se utilizó la tasa de tramas erróneas (*Frame Error Rate*) que está disponible en la clase *NS3::WifiRemoteStation Info*. Por lo tanto, cuando se prepara un mensaje "HELLO", se accede a la *NS3::WifiRemote StationManager* de ese nodo el cual proporciona la información de la estación. Por otra parte, el acceso a la ruta MAC (*NS3::WifiMac Queue*) nos proporciona el tamaño de cola que de nuevo se inserta en el mensaje "HELLO" como un indicador de la utilización de ese nodo.

### ***Carry-n-Forward***

Cuando un nodo se encuentra dentro de un máximo local se emplea el mecanismo CnF para almacenar el paquete localmente en el caché hasta que se encuentre una ruta más conveniente. La cola del caché tiene un tamaño limitado (número de paquetes almacenados) y un tiempo de caducidad de los paquetes almacenados en caché. Utiliza etiquetas en los paquetes para obtener valores de SNIR de PHY para los mensajes "HELLO". *Carry-n-Forward* utiliza *DeferredRouteTag* como una implementación de AODV para almacenar los paquetes en el caché producido por los máximos locales. CnF se basa en *ns3::aodv::RequestQueue*, cada vez que la tabla de enrutamiento se actualiza, la cola del caché es buscado con el fin de comprobar cuál es el mejor nodo para el reenvío, desencolando el paquete [41].

### **3.1.3.6. Limitaciones**

Dentro de las limitaciones que presenta CLWPR se tiene:

- La clase *GridMap* está limitada a solo 15 calles de dimensión.

- Sólo compatible con IPv4
- Funciona con una sola interfaz.

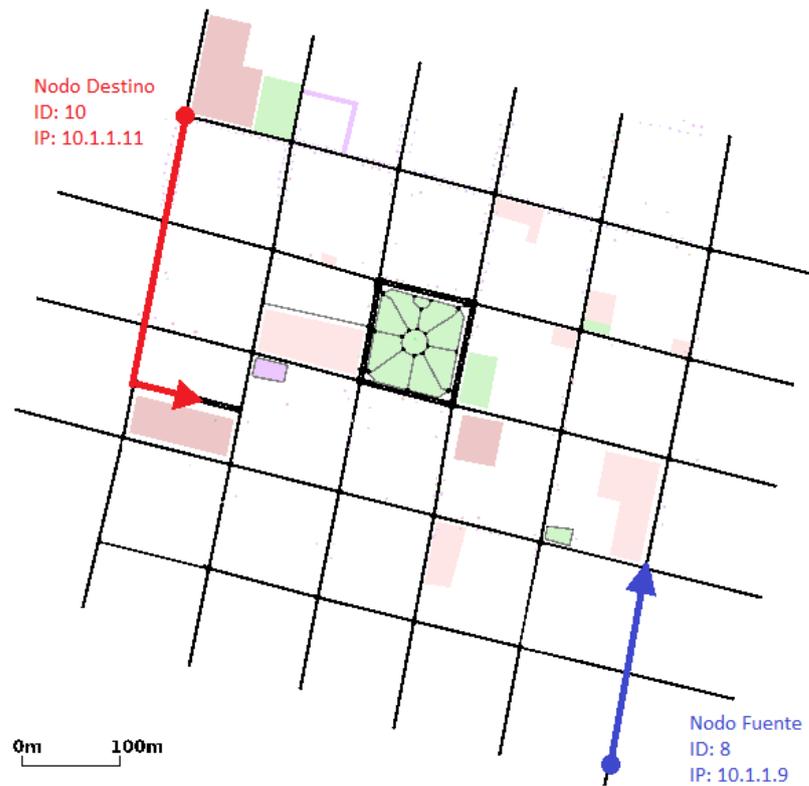
### **3.2. Consideraciones de Simulación**

Tanto MANETs como VANETs comparten ciertas características como la movilidad o su auto-organización de los nodos, por lo tanto, se evaluará el rendimiento de protocolos MANET (AODV, OLSR) vs un protocolo VANET (CLWPR), esto se desarrolla dentro de un escenario VANET.

El área de simulación escogido es el centro histórico de Cuenca, de esta zona urbana se ha tomado un área de 700m x 600m para realizar la simulación, ver Figura 15. Los experimentos se desarrollan en tres ocasiones, para cada escenario se varía el número de nodos, en total habrá 25, 50 y 100 autos. Además, para cada escenario se harán experimentos donde se cambiará el protocolo de enrutamiento (AODV, OLSR y CLWPR) y la tasa de datos (6Mbps, 12Mbps y 24Mbps). Además, para cada simulación el rango de alcance fue configurado en 200, 400 y 600m, todo esto de acuerdo al estándar IEEE802.11p.

### **3.3. Selección de los Nodos Fuente y Destino**

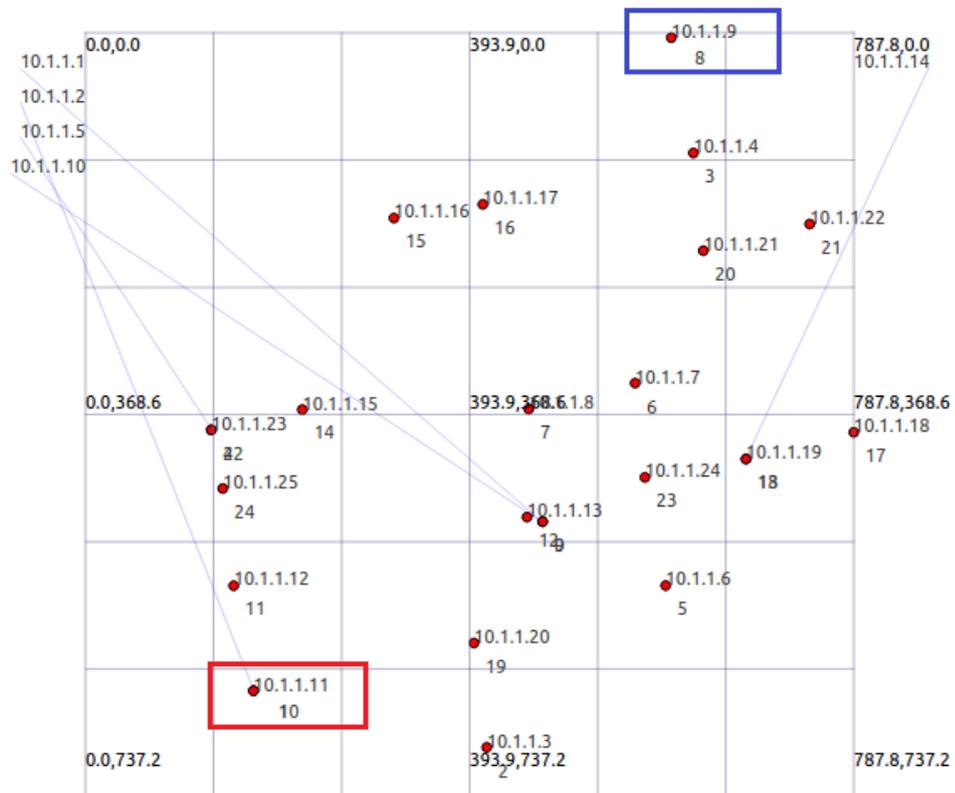
El tipo de comunicación es V2V donde los nodos con ID 8 y 10 fueron seleccionados para ser los nodos de envío y recepción respectivamente. La selección de dichos nodos se debió a que se ubican en los extremos opuestos del mapa seleccionado, además el mismo par de nodos fue usado en cada uno de los tres escenarios con diferente densidad de nodos. La ubicación y el recorrido que realiza cada nodo se puede ver en la Figura 15.



**Figura 15.** Escenario de Simulación: Cuenca, Centro Histórico

SUMO 0.23 es el software que se usó para generar tráfico en el escenario escogido, los archivos generados que contienen los patrones de movimiento de los autos fueron enlazados con NS-3 (versión 3.23) mediante el uso de NS2 Mobility Helper. Estos archivos pueden ser creados manualmente, o se puede importar los datos de mapas de diferentes formatos y trabajar con ellos. En esta investigación se usó un mapa que fue descargado de OpenStreetMap (OSM). En la simulación fueron tomadas en cuenta las diferentes estructuras como edificaciones y otros elementos que hacen más realista el ambiente de simulación.

En la simulación se asume que inicialmente todos los nodos ya están dentro del escenario y cada nodo realiza su recorrido generado en SUMO, para cada escenario el recorrido de los autos es el mismo. En la Figura 16 se puede ver la simulación en NS-3, la figura fue realizada con NetAnim y muestra el escenario con 25 autos, por cuestiones del proceso de simulación a la imagen con respecto al escenario original se le ha realizado un flip vertical que no altera el comportamiento de los autos.



**Figura 16.** Simulación en NetAnim

Algo importante es que para cada escenario se mantienen los mismos parámetros de simulación para asegurar una correcta comparación de los resultados.

### 3.4. Traza de Video

Una muestra de video en formato MPEG-4 fue usada para simular el *streaming* de video. El video a transmitir en esta investigación es un tramo de una película (Die Hard III, disponible en: <http://trace.eas.asu.edu/TRACE/ltvt.html>). El video está codificado con un *bit rate* de 256Kbps, siendo su resolución de 176x144p. Esta traza de video contiene todos los paquetes que representan el *stream* de video. La traza es cargada y transmitida por el nodo fuente hacia el nodo destino.

Durante la simulación los paquetes se envían sobre UDP. Cada paquete tiene tamaño variable y son enviados a una tasa de transmisión de 6, 12 y 24 Mbps.

Las simulaciones fueron realizadas en Ubuntu 15.04. La tecnología de transmisión está basada en 802.11p, debido a esto los siguientes parámetros basados en ese estándar fueron configurados al iniciar la simulación y así simular un ambiente wave, el modelo de propagación fue TwoRayGroundPropagationLossModel, otros parámetros de simulación se muestran en la Tabla 5:

**Tabla 5.** Parámetros de Simulación

<b>Parámetro</b>	<b>Valor</b>
Simulador	NS-3
Protocolo de enrutamiento	AODV, OLSR, CLWPR
Escenario simulación	Cuenca, Centro Histórico
Área de simulación	700m x 600m
Modelo de Propagación	Two-Ray Ground
Multiplexación	OFDM
Data rate	6, 12, 24 Mbps
Tiempo de simulación	100 s
Numero de nodos	25, 50, 100
Nodo fuente	ID: 8, IP:10.1.1.9
Nodo destino	ID: 10, IP:10.1.1.11
Protocolo de transporte	UDP
MAC	IEEE802.11p
Rango de transmisión	200, 400, 600m
Tamaño de los paquetes	Variable
Video Bit Rate	MPEG-4 256Kbps (176x144p)

En la siguiente sección se muestran los resultados y la discusión sobre el rendimiento de los protocolos.

## CAPÍTULO IV

### 4. ANÁLISIS DE RESULTADOS

Aquí se presentan los resultados del rendimiento de los protocolos AODV, OLSR y CLWPR dentro de un escenario VANET. Las métricas para evaluar el rendimiento son *BER*, *delay* y *throughput*.

El objetivo principal de los experimentos fue medir el rendimiento en la transmisión de *video streaming* usando diferentes protocolos de enrutamiento dentro de un escenario vehicular. Para medir esta habilidad del protocolo usamos algunas condiciones de transmisión como densidad de nodos, alcance y velocidad de transmisión. Con el propósito de obtener comparaciones justas y directas entre los protocolos las condiciones del ambiente fueron mantenidas para todos los experimentos.

#### 4.1. Métricas para análisis de rendimiento

En una transmisión de *video streaming* es importante que ciertos valores de los datos transmitidos se mantengan dentro de un umbral aceptable, esto determina el nivel de calidad de la comunicación. En esta investigación las métricas utilizadas para medir el rendimiento de cada uno de los protocolos son: *BER*, *average delay* y el *throughput*.

- *BER, Bit Error Rate*

Para una señal digital binaria, relación entre el número de bits erróneos recibidos y el número total de bits recibidos durante un intervalo de tiempo determinado [58].

- **Average Delay**

“Es el tiempo promedio que a un paquete le toma viajar por la red. Es el tiempo tomados desde la generación del paquete en el nodo emisor hasta que es recibido en el destino. El menor promedio de la transmisión extremo a extremo muestra el mejor rendimiento del protocolo de enrutamiento [49].

- **Throughput**

Es la máxima tasa de transferencia de datos entre dos nodos finales dentro de una red. El *throughput* más alto indica un mejor rendimiento del protocolo de enrutamiento [49].

Con los parámetros configurados en el capítulo 3 se obtuvieron las gráficas del rendimiento de cada protocolo. Para cada métrica se obtuvieron distintos valores dependiendo de la tasa de transmisión, el rango de alcance y la densidad de nodos intermedios.

## 4.2. Análisis de BER

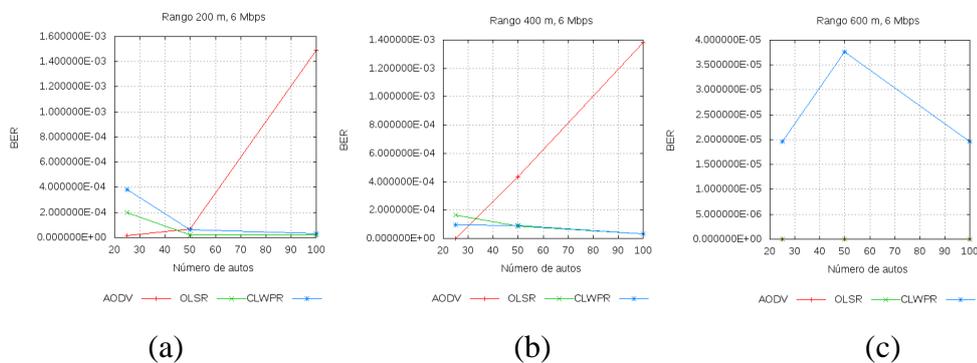
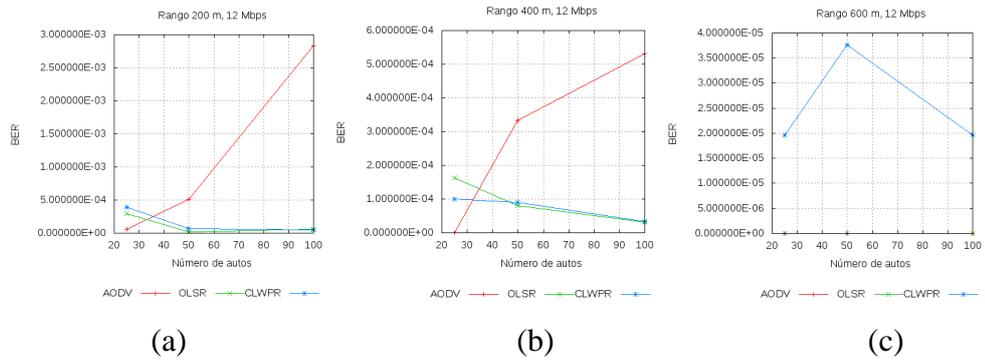
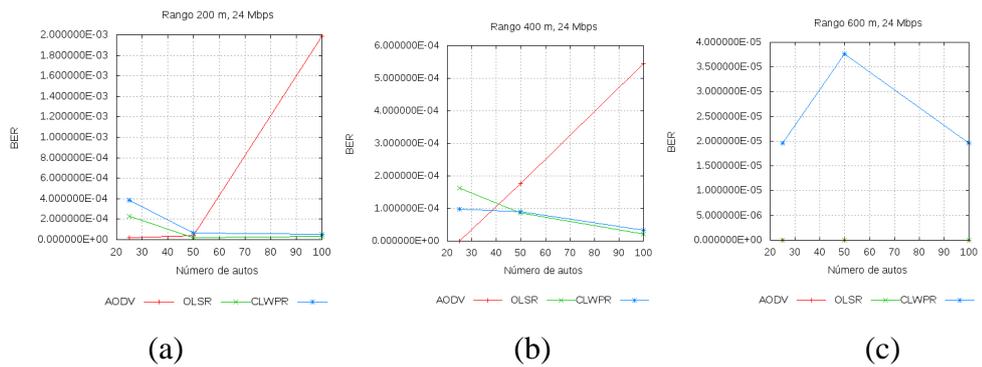


Figura 17. BER, Tasa de transmisión: 6Mbps



**Figura 18.** BER, Tasa de transmisión: 12Mbps



**Figura 19.** BER, Tasa de transmisión: 24Mbps

En la Figura 17 (a) se observa que con un rango de cobertura de 200m y con la tasa de transmisión de 6Mbps el rendimiento de AODV disminuye mientras la densidad de autos se incrementa; esta tendencia se repite con el rango de alcance a 400m como se ve en la Figura 17 (b); para ambos casos el rendimiento del BER baja desde “excelente” hasta “muy pobre” al aumentar la densidad de autos. Por otro lado, el rendimiento de CLWPR y OLSR va aumentando desde “bueno” hasta “excelente”, esto se da cuando la densidad de autos aumenta y sin importar si el rango es de 200m o 400m como se puede ver en la Figura 17 (a), (b). Mientras tanto, al subir el rango de cobertura a 600m todos los protocolos tienen un rendimiento “excelente” sin importar el número de autos. Este comportamiento se debe a que al aumentar el rango de cobertura se necesita menos saltos para llegar al destino.

Al aumentar la tasa de transmisión a 12Mbps se repite el mismo patrón de comportamiento que al usar 6Mbps. En este caso, al aumentar el rango desde 200m a 400m AODV pasa de un rendimiento “muy pobre” a un rendimiento “satisfactorio” como se ve en las figuras: Figura 18 (a), (b), mientras que CLWPR y OLSR siguen manteniendo rendimientos “buenos” a “excelentes”. Por otro lado, con un rango de cobertura de 600m los tres protocolos presentan un rendimiento “excelente”.

Como se ve en las figuras: Figura 19 (a), (b) y (c); con la tasa de transmisión de 24Mbps el rendimiento de los tres protocolos es similar a usar una tasa de transmisión de 12Mbps, esto se debe a que la tasa de transmisión de 24Mbps es más de la necesaria para la transmisión de video.

En términos generales los resultados en la Figura 17 muestran que el valor de BER usando CLWPR se mantiene constante mientras se varía el rango de alcance de la transmisión, el valor promedio es  $1.9623E-5$  (rendimiento “excelente”). Mientras que usando AODV y OLSR el BER mejora y cae a valores cercanos a 0 mientras el rango de comunicación aumenta a 600m. En este caso el orden de rendimiento es OLSR, CLWPR y AODV. También se puede ver que indistintamente de la densidad de autos y del rango de alcance, para CLWPR y OLSR el valor de BER se mantiene constante, siendo CLWPR superior a OLSR. Mientras que el rendimiento de AODV mejora al aumentar la cantidad de autos, aunque su rendimiento sigue siendo inferior a los demás protocolos. El rendimiento de los protocolos se mantiene igual al variar la tasa de transmisión desde 12Mbps hasta 24Mbps, esto se puede ver en las figuras: Figura 18 y Figura 19.

### 4.3. Análisis de Delay

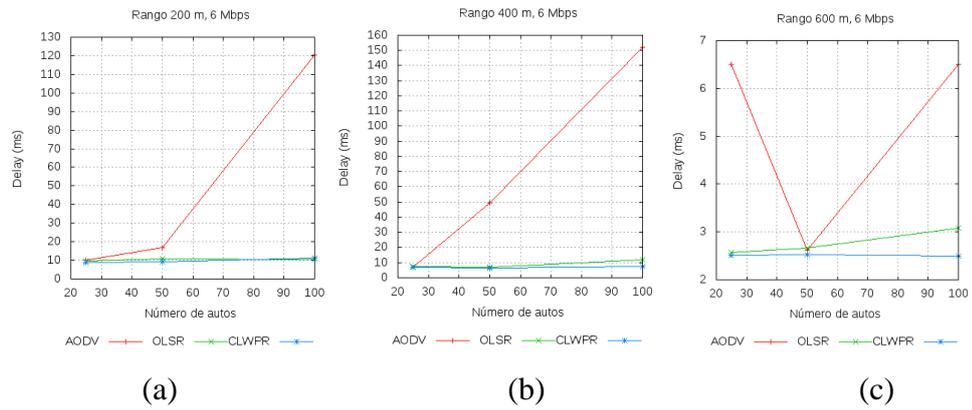


Figura 20. Delay, Tasa de transmisión: 6Mbps

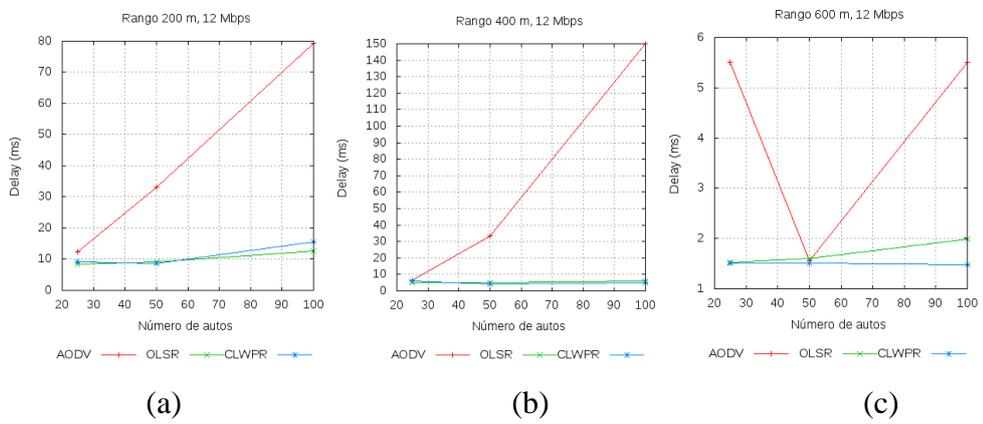


Figura 21. Delay, Tasa de transmisión: 12Mbps

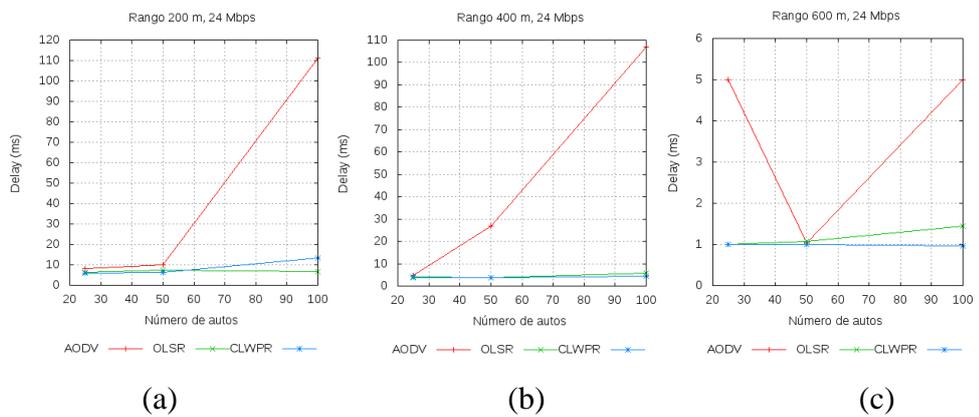


Figura 22. Delay, Tasa de transmisión: 24Mbps

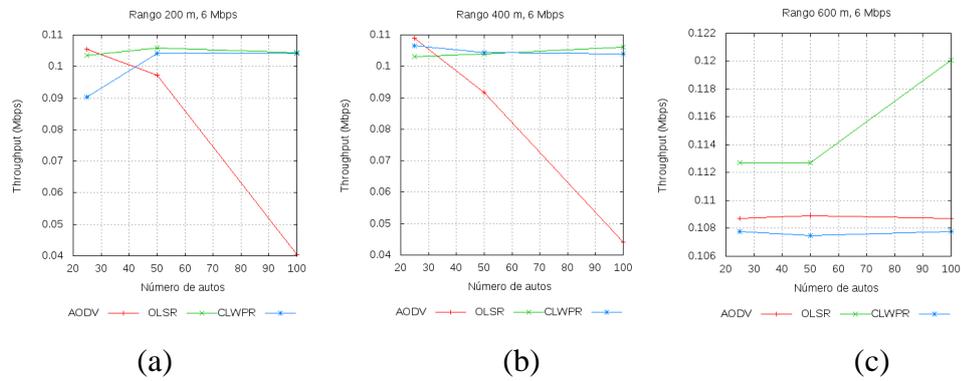
Como se esperaba CLWPR y OLSR tienen mejor rendimiento que AODV, en las figuras: Figura 20 (a), (b) y (c), se muestra que indistintamente del rango de transmisión cuando se aumenta la densidad de autos los dos primeros protocolos mantienen un comportamiento estable, mientras que con el tercer protocolo los valores van cambiando, esto se debe a que CLWPR y OLSR manejan bien el problema de la desconexión gracias a sus características proactivas, es decir que al añadir nodos entre la fuente y destino dichos protocolos descubren rápidamente la ruta, mientras que AODV tiene que calcular todo ante cualquier cambio. Al variar la tasa de transmisión a 12Mbps y 24Mbps el comportamiento de los protocolos ante los cambios de rango de transmisión o de densidad de autos es el mismo que cuando se usa 6Mbps, esto se puede ver en las figuras: Figura 21 (a), (b) y (c) y Figura 22 (a), (b) y (c).

En las figuras: Figura 20, Figura 21 y Figura 22, se puede ver que para CLWPR y OLSR el valor de *delay* mejora al variar el rango de alcance, esto se debe a que se necesitan menos saltos para llegar al destino, además la ruta es descubierta previamente, por lo tanto con OLSR se añade menos retardo extremo a extremo. Por otro lado, CLWPR es la mejora de OLSR, la información adicional que usa como la calidad del enlace, la tasa de error de *frame* y la predicción de posición del nodo si cumple con el propósito de disminuir el retardo extremo a extremo, aunque no presenta gran diferencia a OLSR.

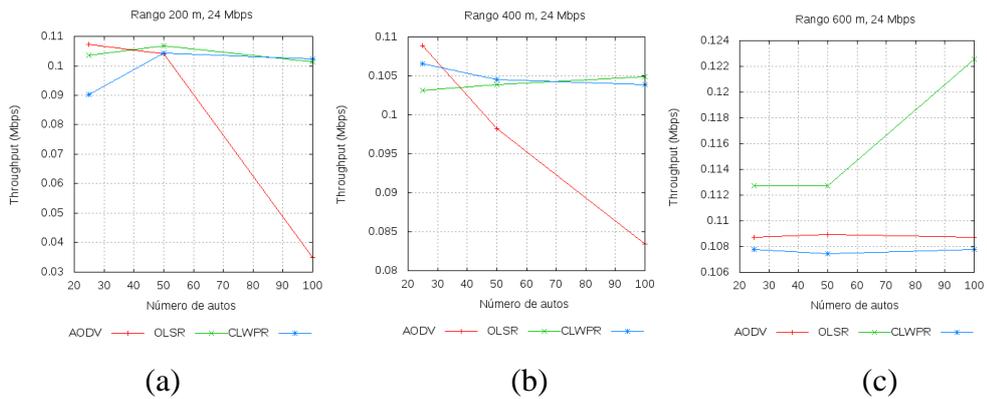
En general el rendimiento de todos los protocolos mejora significativamente al variar la tasa de transmisión desde 6Mbps, 12Mbps hasta 24Mbps. Dentro de cada experimento ya sea con el rango de alcance de 200, 400 o 600m, y al variar la densidad de autos se repite que usando CLWPR y OLSR los valores de *delay* no varían significativamente, aunque para OLSR si hay un cambio pero no tan grande y para CLWPR el valor se mantiene casi constante, en conclusión la cantidad de autos no altera el valor del *delay* total. Por lo contrario, para AODV el *delay* si varía bruscamente al aumentar la cantidad de autos dentro del escenario. Todo esto se puede analizar en las figuras: Figura 20, Figura 21 y Figura 22. También se puede ver que CLWPR es el que mejores resultados muestra ya que las mejoras introducidas hacen

más rápida la comunicación. Para todos los casos el valor del *delay* está dentro del umbral permitido para las transmisiones de *video streaming*, es decir menor a 10s.

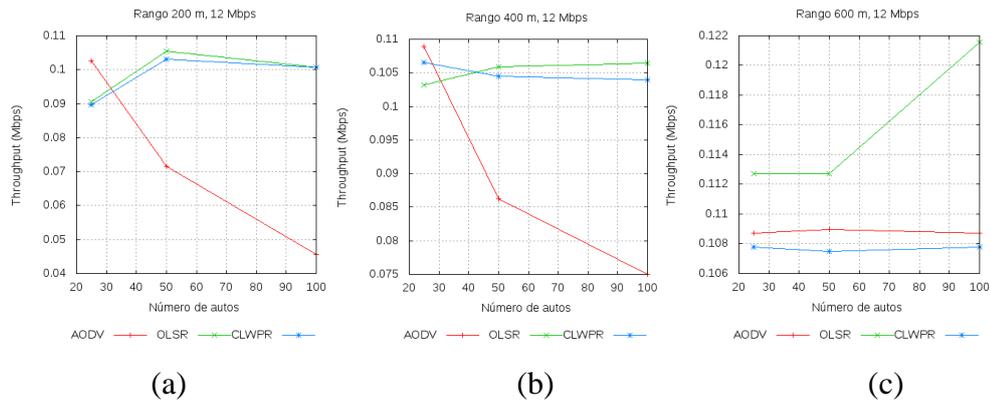
#### 4.4. Análisis de Throughput



**Figura 23.** Throughput, Tasa de transmisión: 6Mbps



**Figura 24.** Throughput, Tasa de transmisión: 12Mbps



**Figura 25.** *Throughput*, Tasa de transmisión: 24Mbps

En las figuras: Figura 23 (a), (b), Figura 24 (a), (b) y Figura 25 (a), (b) se puede ver que AODV solo es eficiente al tener una baja cantidad de autos ya que en cada experimento, solo cuando hay 25 autos AODV mantiene un rendimiento similar a CLWPR y OLSR. Al aumentar la cantidad de autos, el protocolo se hace ineficiente, excepto cuando el rango de alcance es 600m ya que en este caso el valor de *throughput* se mantiene constante al aumentar la densidad de nodos. Este comportamiento se repite sin importar el valor de la tasa de transmisión. Esto coincide con la literatura, ya que en [63] se indica que el valor de *throughput* debería disminuir debido a la interferencia que provocan los autos alrededor.

CLWPR, por su parte, mantiene el valor de *throughput* para casi todos los experimentos, como se ve en las figuras: Figura 23 (a), (b), (c), Figura 24 (a), (b) (c) y Figura 25 (a), (b), (c). Este patrón de comportamiento se da sin importar si se aumenta la cantidad de autos desde 25 a 100 autos, o si se varía la tasa de transmisión desde 6Mbps hasta 24Mbps. Sin embargo, el rendimiento de CLWPR es inferior a OLSR y también a AODV, en el caso de los experimentos con 600 metros, aunque la diferencia entre ambos protocolos no es considerable.

Tanto para OLSR y CLWPR, cuando los nodos de comunicación están más cercanos uno del otro, el emisor encuentra una ruta válida más fácilmente, por lo tanto el *throughput* es más alto, esto se puede observar en las figuras: Figura 23, Figura 24 y

Figura 25 donde los escenarios con mayor rango de cobertura dan como resultado un mayor valor de *throughput*.

En resumen, teniendo en cuenta los valores de BER, delay y throughput, en todos los casos cuando se incrementa el rango de cobertura a 600m, todos los protocolos tienen un excelente rendimiento, esto se debe a que con dicho rango se puede llegar a nodos que se encuentren más lejos y así disminuir el número de saltos.

## CAPÍTULO V

### 5. CONCLUSIONES Y RECOMENDACIONES

#### 5.1. Conclusiones

El área de las redes Ad-Hoc ha recibido gran atención, en especial en un ambiente de movilidad vehicular y para transmitir un particular tipo de datos como lo es el *video streaming* ya que de esta forma se puede transmitir información más precisa y sin la necesidad de tener infraestructura fija. Varios protocolos para VANETs y *video streaming* han sido propuestos; pero pocos de ellos muestran su comportamiento en ambientes lo más cercanos a la realidad. Este trabajo de fin de grado ha logrado crear un ambiente de simulación que toma en cuenta trazas reales de movilidad y también simula el ambiente real de la ciudad al tomar en cuenta edificaciones y elementos.

En este trabajo se propuso analizar el rendimiento de *video streaming* sobre VANETs en escenarios de simulación en la ciudad de Cuenca. Esto se logró mediante la utilización de software libre como NS-3 y SUMO. Mediante SUMO se consiguió crear una traza del centro histórico de la ciudad de Cuenca y en ella implementar patrones de movilidad para el tránsito vehicular. Posteriormente, este escenario fue usado en NS-3 donde los diferentes protocolos fueron implementados. Ambos programas son ampliamente usados por la comunidad académica e investigadora para el estudio de las redes Ad-Hoc.

Dentro de estos protocolos escogimos AODV y OLSR, por estar dentro de los más utilizados y de referencia para comparaciones de rendimiento en muchos artículos en la literatura [49,50,51]. Adicionalmente se implementó CLWPR el cual es enfocado para VANETs. Entre los tres protocolos se realizó una comparación del rendimiento para *video streaming* en el escenario urbano.

Tanto los protocolos CLWPR, OLSR y AODV tienen una escalabilidad restringida debido a sus características, los dos primeros al ser proactivos y el tercero reactivo. En el protocolo AODV la escalabilidad causa mayor inundación de mensajes en las redes con elevada movilidad. En el protocolo OLSR afecta en el tamaño de la tabla de enrutamiento y en la cantidad de mensajes de actualización de la topología. CLWPR está bajo las condiciones de OLSR ya que está basado en dicho protocolo, pero las mejoras como son *carry-n-forward* y *cross-layering* que han sido incrementadas en este protocolo han mejorado la eficiencia en cuanto al *delay*, el cual es un parámetro importante para la transmisión de *video streaming*.

En cuanto al BER se observa en los resultados que el rendimiento de los protocolos mejora al aumentar el rango de cobertura y la tasa de transmisión; para el caso de 200 y 400m para AODV el rendimiento se mantiene en los rangos entre “pobre” y “satisfactorio”, mientras que para OLSR y CLWPR el rendimiento mejora desde “bueno” a “excelente”. En el caso de 600m el rendimiento de AODV y OLSR aumentan considerablemente al incrementar los parámetros antes mencionados se tiene valores cercanos a 0; mientras que el rendimiento de CLWPR también aumenta pero no tanto como los protocolos antes mencionados, el mejor rendimiento en CLWPR es 1,9623E-05 y se da cuando existe un tráfico denso (100 autos) independiente de la tasa de transmisión utilizada. Para los tres protocolos con 600m el rendimiento es “excelente”.

Con respecto al *delay*, el rendimiento CLWPR y OLSR se mantienen estables en los rangos de 200m y 400m, con un promedio de retardo de 10ms. En la mayoría de casos el *delay* de CLWPR es el mejor, esto se debe a las mejoras implementadas como lo son *carry-n-forward* y *cross-layering*; sin importar la tasa de transmisión o el rango de alcance para todos los protocolos los valores de *delay* se mantienen dentro del umbral aceptable. Para AODV el rendimiento decae al aumentar la densidad de autos en transmisiones con rango de cobertura de 200m y 400. Para los tres protocolos con el rango de cobertura incrementado a 600m su rendimiento mejora obteniendo valores

de *delay* menores a 7ms, independiente de la densidad de autos y de la tasa de transmisión.

En cuanto al *throughput*, el valor en CLWPR se mantiene estable sin importar la cantidad de autos, la tasa de transferencia o el rango de alcance de la comunicación, manteniendo un valor promedio de 106 Kbps. De igual manera OLSR tiene similar rendimiento a CLWPR pero se destaca que solo cuando el rango de alcance es 600m y con la cantidad de nodos incrementada a 100 el valor promedio se incrementa de 115 Kbps hasta 124 Kbps. En cuanto a AODV el protocolo no tiene buen rendimiento con rangos de alcance menores a 400m ya que solo con 600 m su rendimiento se asemeja a los demás protocolos. El *throughput* como se esperaba disminuyó con respecto al *bit rate* debido a las interferencias que presenta la red vehicular, en este caso en *throughput* en el mejor de los casos tiene un valor de 124 Kbps que corresponde al 48% del bit rate.

Sobre la base de los diferentes resultados obtenidos en cada caso, como se ha indicado anteriormente, se puede concluir que los protocolos VANET dependen de un conjunto de variables que conforman el entorno de simulación tales como la movilidad, la densidad, el tamaño de la red, configuración del dispositivo OBU, etc.

## **5.2. Trabajos Futuros**

Este trabajo de fin de grado se centra en analizar el rendimiento de diferentes protocolos de enrutamiento enfocados en la transmisión de *video streaming* dentro de un escenario lo más cercano a la realidad. En base a estos resultados se puede tomar decisiones y generar nuevas líneas de investigación sobre transmisión de video y además de identificar aspectos que se deberían mejorar en cada protocolo para mejorar el rendimiento general. Nuestro trabajo servirá para dar sustento para actuales y futuras investigaciones del grupo de investigación GITEL.

## APÉNDICES

### APÉNDICE A

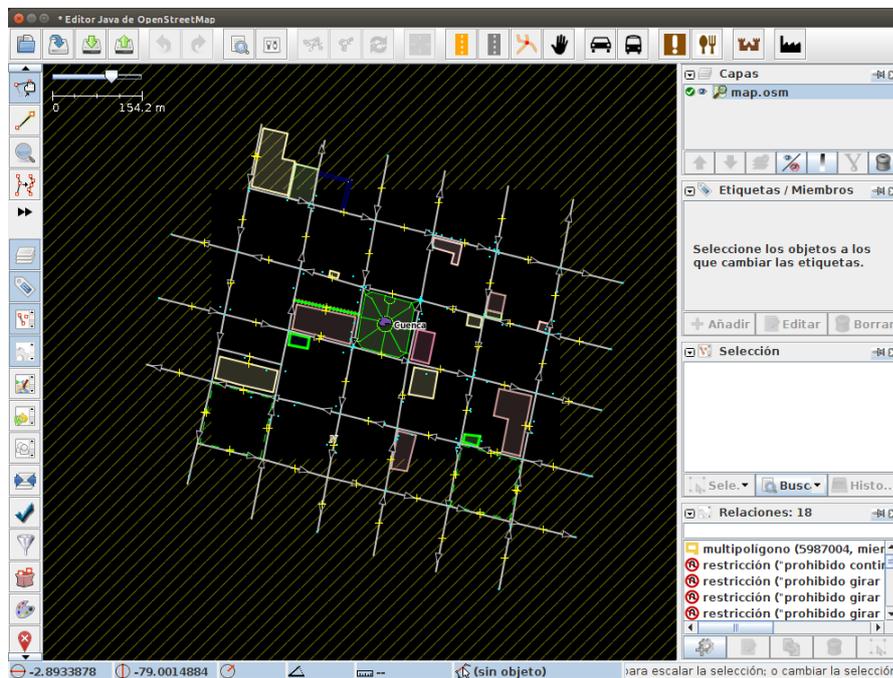
#### A. PASOS PARA CREAR EL ESCENARIO DE SIMULACIÓN

Todos los siguientes pasos se deben realizar dentro del terminal de ubuntu, y con la dirección de la carpeta donde están los archivos iniciales, los cuales son: *map.osm*, *typemap.xml*, *script.tcl* y *script\_rou.tcl*

El archivo *map.osm* contiene la red vial y se lo descarga desde:

<https://www.openstreetmap.org/>

Para editar o restringir el acceso de los autos a ciertas calles o lugares se utiliza el editor de mapas OSM llamado *JOSM – Editor Java de OpenStreetMap*. Ver figura A1



**Figura A1.** Edición del mapa usando JOSM

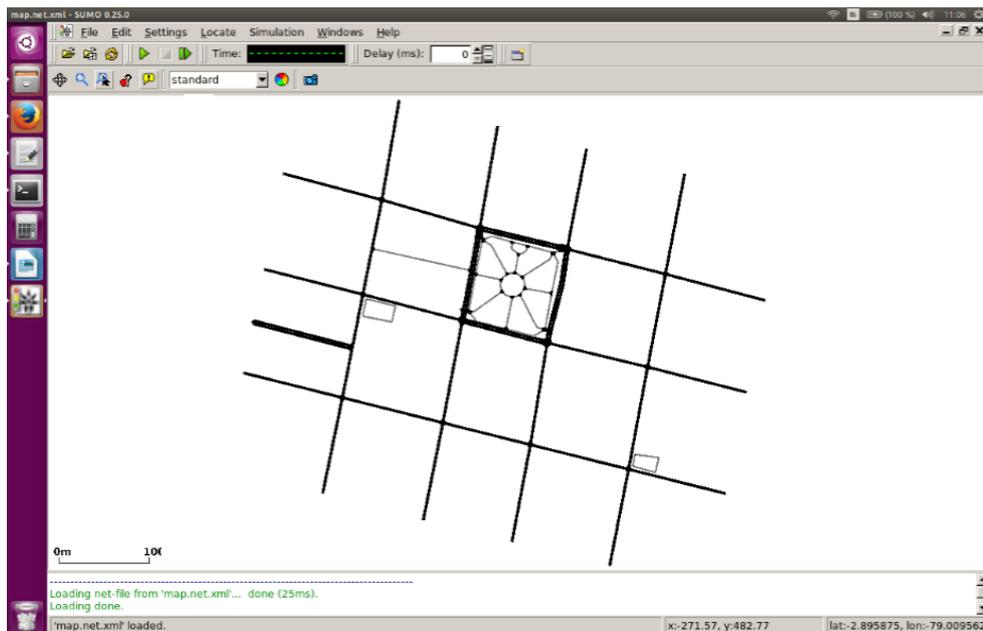
Los siguientes pasos servirán para convertir el archivo de formato *.osm* en un formato que SUMO pueda leer y posteriormente cambiar a un formato que pueda ser usado en NS-3.

## A.1. Generar Rutas

1) Netconvert es una herramienta que importa información digital de redes viales y la almacena en otros formatos que pueden ser usados por otras herramientas, en este caso dicha información está contenida en el archivo “.osm”. El siguiente comando genera una red vial, al ejecutarlo se genera el archivo *map.net.xml*.

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

Ejecutando el comando *sumo-gui -n example.net.xml* se puede ver la red vial como se observa en la figura A2.



**Figura A2.** Red Vial vista en la Interfaz GUI de SUMO

De la siguiente página se debe copiar la estructura y se debe guardar en un archivo llamado *typemap.xml*

```
http://sumo.dlr.de/wiki/Networks/Import/OpenStreetMap
```

2) Los mapas “.osm” no solo contienen información de las calles sino información de ríos, edificaciones y muchos elementos más, el archivo *typemap.xml* ayuda a interpretar esta información. Con el siguiente comando se genera el archivo *map.poly.xml*

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

3) “randomTrips.py” genera rutas aleatorias en una red dada (-n), lo hace usando los nodos y edges, la distribución de las rutas se la hace tomando en cuenta la longitud del mapa (-l). Hasta ahora solo se crea el inicio y final de la ruta, al ejecutar la opción -r (con el nombre del archivo resultante que contendrá las rutas) se llama directamente a DUAROUTER la cual es una herramienta que crea el flujo de cada vehículo para cada ruta creada con “randomTrips.py”. Para poner fin en la creación de rutas se usa (-e), el valor se escribirá en segundos.

```
python /home/user/sumo-0.25.0/tools/randomTrips.py -n map.net.xml -e 400 -l
python /home/user/sumo-0.25.0/tools/randomTrips.py -n map.net.xml -r map.rou.xml
-e 400 -l
```

Los anteriores comandos generan los archivos “trips.trips.xml”, “map.rou.xml” y “map.rou.alt.xml”. El primer archivo contiene solo el inicio y final de las rutas, y el segundo y tercero contienen el flujo para cada vehículo, a diferencia de que el tercer archivo contiene más parámetros como por ejemplo si el auto volverá a correr una vez que termine su recorrido u otras opciones más, en este caso todas esas opciones están configuradas como default por lo tanto con ambos archivos se obtiene los mismos resultados al momento de simular.

4) El tercer paso es la compilación e integración de los archivos generados, con este archivo ya se puede ejecutar y visualizar la simulación. A continuación, hay que buscar en la carpeta de instalación de Sumo el archivo *sumo.test.cfg*, se deberá copiar a la carpeta donde se hayan realizado los anteriores pasos, para facilidad el archivo es renombrado como *map.sumo.cfg*. Dentro de ese archivo se debe modificar los nombres y llamar a los archivos creados en los pasos anteriores.

O se puede también crear un archivo con el nombre indicado y como contenido pondremos:

```
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/sumoConfiguration.xsd">
    <input>
```

```
<net-file value="example.net.xml"/>
<route-files value="example.rou.xml"/>
<additional-files value="map.poly.xml"/>
</input>

<time>
  <begin value="0"/>
  <end value="1000"/>
</time>
</configuration>
```

Este archivo llama a los archivos anteriormente creados, además se añade el archivo *map.poly.xml* el cual contiene la información de edificios y el entorno. Para iniciar la visualización se usa el siguiente comando:

```
sumo-gui map.sumo.cfg
```

5) Para reajustar las rutas se usa la siguiente línea la cual configura los valores de “depart” a 0, es decir todos los autos empezaran su recorrido en el tiempo 0. Con este comando se genera el archivo *map\_process.rou.xml*

```
/home/user/SUMOOK/script_rou.tcl
```

## A.2. Generar Trace Files

La creación de trazas se realizará por medio de los pasos 6 y 7. A pesar de que estemos trabajando en NS-3, las trazas creadas serán en formato NS-2 ya que en NS-3 la herramienta NS2MobilityHelper trabaja fácilmente con estas trazas.

6) Usando la simulación hasta ahora creada obtenemos un “fcd output”, el cual es un archivo en formato SUMO, El archivo FCD (Floating Car Data) contiene ubicación y la velocidad junto con otra información para cada vehículo en la red. La salida se comporta un poco como un dispositivo GPS super-preciso para cada vehículo. Con este comando genero el *sumoTrace.xml*.

```
sumo -c /home/user/SUMOOK/map.sumo.cfg --fcd-output  
/home/user/SUMOOK/sumoTrace.xml
```

7) *traceExporter.py* es un comando que convierte un archivo “SUMO fcd output” a un diferente formato de “trace file” en este caso el formato es NS-2. Con el siguiente comando se genera el archivo de trazas de NS-2, el archivo generado es ns2mobility.tcl

```
/home/user/sumo-0.25.0/tools/traceExporter.py -fcd-input  
/home/user/SUMOOK/sumoTrace.xml -ns2mobility-output  
/home/user/SUMOOK/ns2mobility.tcl
```

8) Posteriormente se deberá reajustar las trazas, así se genera el ns2mobility\_process.tcl

```
/home/user/SUMOOK/script.tcl
```

## B. ENLACE DE NS-3 Y SUMO

Para poder usar las trazas de formato NS-2 en el simulador NS-3 se debe usar la herramienta Ns2MobilityHelper, para ello se deberán colocar algunos códigos dentro del archivo de NS-3

En el archivo de NS-3 se deberá incluir las siguientes librerías:

```
#include "ns3/netanim-module.h"  
#include "ns3/ns2-mobility-helper.h"
```

Posteriormente dentro del código (archivo con extensión .cc) después de crear los nodos se debe instalar la movilidad usando el siguiente código:

```
Ns2MobilityHelper ns2 = Ns2MobilityHelper ("scratch/ns2mobility_process.tcl");  
ns2.Install ();
```

Finalmente antes de la línea “*Simulator::Stop (Seconds (duration));*” se debe colocar:

```
AnimationInterface anim ("animation.xml");
```

## REFERENCIAS BIBLIOGRAFICAS

- [1] G. Mohimani, F. Ashtiani, A. Javanmard, and M. Hamdi, "Mobility Modeling, Spatial Traffic Distribution, and Probability of Connectivity for Sparse and Dense Vehicular Ad Hoc Networks," *IEEE Trans. on Vehicular Technology*, vol. 58, pp. 1998–2007, May 2009.
- [2] Asefi, M.; Mark, J. W. & Shen, X. An Application-Centric Inter-Vehicle Routing Protocol for Video Streaming over Multi-Hop Urban VANETs Communications (ICC), 2011 IEEE International Conference on, 2011, 1-5
- [3] S. Olariue and M. C. Weigle, *Vehicular Networks, From Theory to Practice*. Chapman and Hall/CRC, 2009.
- [4] Chen, Y.; Li, C.; Han, X.; Gao, M. & Zhu, L. A Reliable Beaconless Routing Protocol for VANETs Computer and Information Technology (CIT), 2014 IEEE International Conference on, 2014, 94-99
- [5] Pham, T. A. Q.; Piamrat, K. & Viho, C. QoE-Aware Routing for Video Streaming over VANETs Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th, 2014, 1-5
- [6] C. Perkins, E. Belding-Royer, and S. Das, *Ad hoc On-Demand Distance Vector (AODV) Routing*, IETF Std. RFC 3561, July 2003.
- [7] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *IEEE INMIC*, 2001, pp. 62–68.
- [8] Lee, K.C.; Gerla, M., "Opportunistic vehicular routing," in *Wireless Conference (EW)*, 2010 European , vol., no., pp.873-880, 12-15 April 2010
- [9] Honghai Wu; Huadong Ma, "Opportunistic routing for live video streaming in vehicular ad hoc networks," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2014 IEEE 15th International Symposium on a , vol., no., pp.1-3, 19-19 June 2014

- [10] Bradai, A.; Ahmed, T., "ReViV: Selective Rebroadcast Mechanism for Video Streaming over VANET," in Vehicular Technology Conference (VTC Spring), 2014 IEEE 79th , vol., no., pp.1-6, 18-21 May 2014
- [11] Naeimipoor, F.; Boukerche, A., "A Hybrid Video Dissemination Protocol for VANETs," in Communications (ICC), 2014 IEEE International Conference on , vol., no., pp.112-117, 10-14 June 2014
- [12] Quadros, C.; Cerqueira, E.; Santos, A.; Lim, J.; Gerla, M., "Beacon-less video streaming management for VANETs based on QoE and link-quality," in Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on , vol., no., pp.191-198, 11-15 May 2015
- [13] Rezende, C.; Boukerche, A.; Ramos, H.S.; Loureiro, A.A.F., "A Reactive and Scalable Unicast Solution for Video Streaming over VANETs," in Computers, IEEE Transactions on , vol.64, no.3, pp.614-626, March 2015
- [14] Chao-Hsien Lee; Chung-Ming Huang; Chia-Ching Yang; Hsiao-Yu Lin, "The K-hop Cooperative Video Streaming Protocol Using H.264/SVC Over the Hybrid Vehicular Networks," in Mobile Computing, IEEE Transactions on , vol.13, no.6, pp.1338-1351, June 2014
- [15] I. Khan and A. Qayyum, "Performance evaluation of AODV and OLSR in highly fading vehicular ad hoc network environments," in IEEE INMIC, Dec 2009, pp. 1–5.
- [16] Honda, T.; Ikeda, M.; Spaho, E.; Hiyama, M. & Barolli, L. Effect of Buildings in VANETs Communication: Performance of OLSR Protocol for Video Streaming Application Broadband and Wireless Computing, Communication and Applications (BWCCA), 2013 Eighth International Conference on, 2013, 323-327
- [17] T. Szigeti and C. Hattingh. End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs (Networking Technology). Cisco Press, 2004.
- [18] A. Ibanez, C. Flores, P. Reyes, A. Barba, and A. Reyes. A performance study of the 802.11p standard for vehicular applications. In Intelligent Environments (IE), 2011 7th International Conference on, pages 165 –170, July 2011.

- [19] V. Gau, C.-W. Huang, and J.-N. Hwang. Reliable multimedia broadcasting over dense wireless Ad-Hoc networks (invited paper). *Journal of Communications*, 4(9), 2009.
- [20] F. Xie, K. Hua, W. Wang, and Y. Ho. Performance study of live video streaming over highway vehicular ad hoc networks. In *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pages 2121–2125, 30 2007-oct. 3 2007.
- [21] C. Rezende, A. Boukerche, and R. W. Pazzi. A reactive solution with a redundancy-based error correction mechanism for video dissemination over vehicular ad hoc networks. In *The 15th ACM International Conference on Modeling, Analysis and Simulation on Wireless and Mobile Systems, Paphos, Cyprus, 2012*. (to appear)
- [22] M. Lindeberg, S. Kristiansen, T. Plagemann, and V. Goebel. Challenges and techniques for video streaming over mobile ad hoc networks. *Multimedia Systems*, 17:51–82, 2011. 10.1007/s00530-010-0187-8.
- [23] J.-S. Park, U. Lee, S. Y. Oh, M. Gerla, and D. S. Lun. Emergency related video streaming in vanet using network coding. In *Proceedings of the 3rd international workshop on Vehicular ad hoc networks, VANET '06*, pages 102–103, New York, NY, USA, 2006. ACM.
- [24] C.-J. Hsu, H.-I. Liu, and W. K. Seah, “Opportunistic routing—a review and the challenges ahead,” *Computer Networks*, vol. 55, no. 15, pp. 3592–3603, 2011.
- [25] D. Rosário, Z. Zhao, A. Santos, T. Braun, and E. Cerqueira, “A beaconless opportunistic routing based on a cross-layer approach for efficient video dissemination in mobile multimedia iot applications,” *Computer Communications*, vol. 45, pp. 21–31, 2014.
- [26] M. Di Felice, L. Bedogni, and L. Bononi, “Group communication on highways: An evaluation study of geocast protocols and applications,” *Ad Hoc Networks*, vol. 11, no. 3, pp. 818–832, 2013.

- [27] C.-M. Huang, C.-C. Yang, and H.-Y. Lin, "A K-hop bandwidth aggregation scheme for member-based cooperative transmission over vehicular networks," in Proc. 17th IEEE ICPADS, Tainan, Taiwan, 2011, pp. 436–443.
- [28] G. Novy, M. Machado, F. Duarte-Figueiredo, and R. Mini, "A hybrid sender and receiver-based routing protocol for Wireless Sensor Networks," Journal of Applied Computing Research, vol. 4, no. 1, Apr. 2015.
- [29] G. Mohimani, F. Ashtiani, A. Javanmard, and M. Hamdi, "Mobility Modeling, Spatial Traffic Distribution, and Probability of Connectivity for Sparse and Dense Vehicular Ad Hoc Networks," IEEE Trans. on Vehicular Technology, vol. 58, pp. 1998–2007, May 2009.
- [30] H. Weatherspoon and J. D. Kubiatowicz, "Erasure coding vs. replication: A quantitative comparison," in Peer-to-Peer Systems, Springer, 2002, pp. 328–337.
- [31] A. Bain and D. Crisan, Fundamentals of Stochastic Filtering, vol. 60. New York, NY: Springer New York, 2009.
- [32] Torrubia, G. Sá. & Terrazas, V. L. Algoritmo de Dijkstra. Un tutorial interactivo VII Jornadas de Enseñanza Universitaria de la Informática (JENUI 2001), 2012
- [33] Seguridad en Sistemas y Técnicas de Hacking. TheHackerWay (THW) [Online]. Available: <http://thehackerway.com/2012/03/21/wireless-hacking-conceptos-basicos-sobre-seguridad-en-redes-wireless-parte-ii/>
- [34] S. Mohamed and G. Rubino, "A study of real-time packet video quality using random neural networks," IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, no. 12, pp. 1071–1083, 2002.
- [35] DSRC. [Online]. Available: <http://www.etsi.org/technologies-clusters/technologies/intelligent-transport/dsrc>
- [36] Weisheng Si; Scholz, B.; Gudmundsson, J.; Guoqiang Mao; Boreli, R.; Zomaya, A., "On graphs supporting greedy forwarding for directional wireless networks," in Communications (ICC), 2012 IEEE International Conference on , vol., no., pp.751-756, 10-15 June 2012

- [37] L. Katsikas, K. Chatzikokolakis, and N. Alonistioti, “Implementing clustering for vehicular ad-hoc networks in ns-3,” 2015, pp. 25–31.
- [38] O. T. T. Kim, V. Nguyen, and C. S. Hong, “Which network simulation tool is better for simulating Vehicular Ad-hoc network?,” 2014 Corea Ciencias de la Información Sociedad de la 41 Reunión General Anual y Conferencia de Invierno, pp. 930–932, 2014.
- [39] E. Spaho, M. Ikeda, L. Barolli, F. Xhafa, V. Kolicic, and M. Takizawa, “Performance Evaluation of OLSR Protocol in a Grid Manhattan VANET Scenario for Different Applications,” 2013, pp. 47–52.
- [40] N. Vineeth and H. S. Guruprasad, “Performance analysis of network coded video streams in VANETs based on mobility models,” in Advance Computing Conference (IACC), 2015 IEEE International, 2015, pp. 170–175.
- [41] K. Katsaros, M. Dianati, R. Tafazolli, and R. Kernchen, “CLWPR—a novel cross-layer optimized position based routing protocol for VANETs,” in Vehicular Networking Conference (VNC), 2011 IEEE, 2011, pp. 139–146.
- [42] N. Vineeth, and H. S. Guruprasad, “Delay Analysis of Network Coded Video Streams in VANETs,” International Journal of Information Engineering and Electronic Business, vol. 7, no. 4, pp. 16–23, Jul. 2015.
- [43] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, “A comprehensive survey on vehicular Ad Hoc network,” Journal of Network and Computer Applications, vol. 37, pp. 380–392, Jan. 2014.
- [44] D. Jiang and L. Delgrossi, “IEEE 802.11 p: Towards an international standard for wireless access in vehicular environments,” in Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE, 2008, pp. 2036–2040.
- [45] M. Srotyr, Z. Lokaj, and T. Zelinka, “Advanced VANET routing design,” in Smart Cities Symposium Prague (SCSP), 2015, 2015, pp. 1–5.
- [46] P. M. Khilar and S. K. Bhoi, “Vehicular communication: a survey,” IET Networks, vol. 3, no. 3, pp. 204–217, Sep. 2014.

- [47] E. Weingärtner, H. Vom Lehn, and K. Wehrle, “A performance comparison of recent network simulators,” in Communications, 2009. ICC’09. IEEE International Conference on, 2009, pp. 1–5.
- [48] F. D. Da Cunha, A. Boukerche, L. Villas, A. C. Viana, and A. A. Loureiro, “Data communication in VANETs: a survey, challenges and applications,” INRIA Saclay, 2014.
- [49] Hamid, B. & Mokhtar, E. N. E. Performance analysis of the Vehicular Ad hoc Networks (VANET) routing protocols AODV, DSDV and OLSR 2015 5th International Conference on Information Communication Technology and Accessibility (ICTA), 2015, 1-6
- [50] A Modak, S. Saha, Dr. U. Roy, Dr. D.O. Sinah, AODV ROUTING PROTOCOL MODIFICATION WITH STACK FOR VANET IN CITY SCENARIOS: IJETCAS, ISSN (Print): 2279-0047 ISSN (Online), Issue 8, Volume 2, pp 128-133, 2014.
- [51] Sallam, G. & Mahmoud, A. Performance Evaluation of OLSR and AODV in VANET Cloud Computing Using Fading Model with SUMO and NS3 Cloud Computing (ICCC), 2015 International Conference on, 2015, 1-5
- [52] JOSM – Java OpenStreetMap editor, Available: <http://josm.openstreetmap.de>
- [53] OpenStreetMap, Available: <http://www.openstreetmap.org>
- [54] SUMO – Simulation of Urban MObility, Available: [http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931\\_read-41000/](http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/)
- [55] NS-3, Network Simulator, Available: <https://www.nsnam.org/>
- [56] Video Trace Library, Available: <http://trace.eas.asu.edu/TRACE/ltvt.html>
- [57] O. Delgado, Nuevos Protocolos y Esquemas de Seguridad para Redes Ad-Hoc Móviles Inalámbricas. Universidad Carlos II de Madrid, 2010
- [58] Términos Y Definiciones Rec. UIT-R V.662-2 1, recomendación UIT-R V.662-2.

- [59] P. Gonzalez, Estudio del Simulador de Redes Vehiculares Veins, Universidad Carlos III de Madrid, 2012.
- [60] A. A. Atayero, O. I. Sheluhin, and Y. A. Ivanov, "Modeling, Simulation and Analysis of Video Streaming Errors in Wireless Wideband Access Networks," in IAENG Transactions on Engineering Technologies, vol. 170, H. K. Kim, S.-I. Ao, and B. B. Rieger, Eds. Dordrecht: Springer Netherlands, 2013, pp. 15–28.
- [61] Performance and quality of service requirements for International Mobile Telecommunications-2000 (IMT-2000) access networks R-REC-M.1079-2, 2003-06, IMSW-E
- [62] Luciano Iglesias. Selección de medidores de streaming. Universidad Nacional de la Plata. Octubre 2013.
- [63] I. W.-H. Ho, K. K. Leung, and J. W. Polak, "A methodology for studying vanet performance with practical vehicle distribution in urban environment," 2012.