



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
LABORATORIO DE REDES MÓVILES, INALAMBRICAS Y DISTRIBUIDAS - ICARO

Análisis de Protocolos de Enrutamiento en Redes Vehiculares Ad-hoc

Trabajo Especial de Grado
Presentado ante la Ilustre
Universidad Central de Venezuela
Por el Bachiller
Christian Argelio Puello Conde
Para optar al título de
Licenciado en Computación

Tutora: Profa. María Elena Villapol

Caracas, Noviembre / 2014



*A mis padres...
Que con mucho esfuerzo
han hecho de mi lo que soy.*

*“Lo que habéis heredado de vuestros padres, volvedlo a ganar a
pulso o no será vuestro.”
Goethe (1749-1832)*

*Tengo mucho que aprender de ustedes y
no tengo como pagarles.*



Agradecimientos

Dios en primer lugar, esa fuerza, ese motor que con gran sabiduría rige todo lo que hacemos, gracias por darme las fuerzas y oportunidad de cerrar este tan importante ciclo.

A mis padres, siempre preocupados porque sea un hombre de bien, estudiado y con el futuro que ellos no pudieron tener. A ustedes me debo, por ustedes estoy aquí y sé a dónde voy.

Mis hermanos, mis primeros amigos, mis primeras risas, mis primeras peleas. Ahora son mis más grandes apoyos. Sé que ustedes han esperado esto tanto como yo.

A Lorena, mi amiga, cómplice y compañera por elección, gracias por tus grandes energías, sonrisas interminables, positivismo y sobre todo por estar siempre aquí incluso cuando no estabas.

Mis amigos que ven mi logro como un logro familiar. Porque eso somos.

Profesores y académicos que me han formado, gracias especiales a mi tutora, su conocimiento y su paciencia para conmigo.

¡Gracias a todos!...



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN

TRABAJO ESPECIAL DE GRADO

Título: ANÁLISIS DE LOS PROTOCOLOS DE ENRUTAMIENTO EN REDES VEHICULARES AD-HOC

Autor: Christian Argelio Puello Conde

Unidad de Investigación: Laboratorio De Redes Móviles, Inalámbricas Y Distribuidas – ICARO

Palabras Clave: Simulación, NS-2, VANET, Protocolos, Enrutamiento, Tráfico, Vehicular

Tutora: María Elena Villapol

Fecha de Presentación: Noviembre, 03 de 2014

RESUMEN

Las Redes Vehiculares Inalámbricas Ad-hoc cobran auge entre la infinidad de aplicaciones que se pueden implementar para solucionar problemas de tráfico en las ciudades. Estas redes se componen de los propios nodos móviles autónomos comunicándose entre sí, desapareciendo el control y enrutamiento centralizado. Los nodos deben asumir responsabilidades de enrutamiento y mantenimiento de la red.

Sin duda el desafío a superar al implementar una VANET se refiere al enrutamiento de los paquetes. Este se verá afectado por los cambios de topología extremadamente rápidos, debido a la constante movilidad de los nodos de la red. Se requiere de un estudio preliminar que permita, basado en la simulación y sus resultados, determinar que protocolo de enrutamiento ofrece mejor rendimiento en términos de parámetros de red.

Para el desarrollo se utilizaron herramientas de simulación de red (NS-2) y de tráfico (SUMO) engranadas en una plataforma gracias al Generador de Movimiento VANET (MOVE), que permitiese realizar pruebas con protocolos en escenarios base y variaciones de parámetros sobre el mapa de un sector del Municipio Chacao de Caracas, Venezuela. Arrojando resultados enmarcados en métricas de rendimiento para así verificar diferencias entre estos.

Al finalizar, una vez estudiados los protocolos seleccionados, su comportamiento y ventajas propias, se determinó que la mejor opción para las condiciones presentadas fue el protocolo reactivo AODV.



ÍNDICE

Glosario de Acrónimos	5
1. Introducción	6
1.1. Planteamiento del Problema	7
1.2. Objetivo General	8
1.3. Objetivos Específicos	8
1.4. Justificación e Importancia	8
1.5. Distribución del Documento	9
2. Marco Teórico	11
2.1. Redes MANET	11
2.1.1. Definición	12
2.1.2. Características de una red MANET	15
2.1.3. Consideraciones de enrutamiento	17
2.2. Redes VANET	19
2.2.1. Definición	19
2.2.2. Características de las redes VANET	20
2.2.3. Tecnologías en redes VANET	21
2.2.4. Desafíos en redes VANET	25
2.3. Protocolos de Enrutamiento VANET	26
2.3.1. Clasificación de protocolos de enrutamiento VANET	26
2.3.2. Protocolos de enrutamiento proactivos	27



2.3.2.1.	DSDV	28
2.3.2.2.	OSLR	29
2.3.2.3.	FSR	29
2.3.3.	Protocolos de enrutamiento reactivos	30
2.3.3.1.	AODV	31
2.3.3.2.	DSR	31
3.	Metodologías y Herramientas	34
3.1.	Fases para el desarrollo	34
3.2.	Herramientas para el desarrollo	35
3.2.1.	Hardware y Sistemas Operativos	35
3.2.2.	Simuladores y otros Softwares	36
3.2.2.1.	NS-2	37
3.2.2.2.	SUMO	42
3.2.2.3.	MOVE	43
4.	Escenarios, Métricas y Plataforma de Pruebas de Simulación	45
4.1.	Escenarios	45
4.2.	Métricas utilizadas en el desarrollo	48
4.2.1.	Rendimiento de tráfico (<i>throughput</i>)	48
4.2.2.	Porcentaje de entrega	48
4.2.3.	Retraso extremo a extremo	48
4.2.4.	Sobrecarga (<i>overhead</i>)	49
4.3.	Procesos de Simulación	49
4.4.	Instalación y ejecución de la Plataforma	51



5. Análisis de Resultados	69
5.1. Análisis de Métricas en Circuito Urbano con UDP	71
5.2. Análisis de Métricas en Circuito Urbano con TCP	76
5.3. Análisis de Métricas en Circuito Autopista con UDP	83
5.4. Análisis de Métricas en Circuito Autopista con TCP	88
5.5. Análisis Final	94
6. Conclusiones	95
6.1. Logros	95
6.2. Futuros Trabajos	96
Referencias	98



GLOSARIO DE ACRÓNIMOS

AODV: Ad-hoc On-Demand Distance-Vector Routing
CBR: Constant Bit Rate
DSDV: Destination Sequenced Distance Vector
DSR: Dynamic Source Routing
FSR: Fisheye State Routing
FTP: File Transfer Protocol
GPS: Global Positioning System
IEEE: Institute of Electrical and Electronics Engineers
IP: Internet Protocol
MAC: Media Access Control
MANET: Mobile Ad-hoc Network
MOVE: Mobility Model Generator for Vehicular Networks
MPR: Multi Point Relay
NS-2: Network Simulator 2
NAM: Network Animator
OLSR: Optimized Link State Routing
OSI: Open System Interconnection
OTCL: Objet Tcl, version extendida de Tcl
PDFR: Packet Delivery Fraction
QoS: Quality of Service
RIP: Routing Information Protocol
RREQ: Route Request
RREP: Route Reply
RTS: Request To Send
SUMO: Simulation for Urban Mobility
TCL: Tool Command Language
TCP: Transmission Control protocol
UDP: User Datagram Protocol
UMTS: Universal Mobile Telecommunications System
VANET: Vehicular Ad-hoc Network
WAVE: Wireless Access in the Vehicular Environment
WIMAX: Worldwide Interoperability for Microwave Access
WPAN: Wireless Personal Area Network



1. INTRODUCCIÓN

En la actualidad las comunicaciones a través de redes inalámbricas dinámicas y espontáneas con independencia de infraestructura alguna, como son las redes ad-hoc, es un hecho. Debido a de su infinidad de posibles nuevas aplicaciones, estas se encuentran en pleno desarrollo e investigación evolucionado los intercambios de datos y definiendo el nuevo paradigma del “*Always On - Always Connected*”. Esto nos lleva a pensar que las comunicaciones en entornos vehiculares abren un abanico de una infinidad de beneficios a la sociedad, gracias a este nuevo campo de investigación en la comunidad científica.

Una VANET o Vehicular Ad-Hoc Network, consiste en una red ad-hoc donde sus nodos se corresponden con vehículos, en concreto se está haciendo referencia a una red MANET (*Mobile Ad-Hoc Network*), es decir, una red Ad-hoc móvil dónde sus nodos formarán dicha red de forma dinámica con las dificultades que esto conlleva. Este tipo de redes se compone de los propios nodos móviles autónomos comunicándose entre sí por enlaces inalámbricos. En este entorno desaparece el control centralizado de la red que proporcionaba la BS. Los nodos deben asumir responsabilidades de enrutamiento y mantenimiento de la red. El control de red está distribuido entre los mismos nodos.

Pero sin duda alguna el más importante desafío a superar se refiere al enrutamiento de los paquetes en una VANET, este se verá afectado por los cambios de topología extremadamente rápidos, debido a la constante movilidad de los nodos de la red. Esto implica una necesidad de cambiar las tablas de enrutamiento de dichos nodos constantemente. Para llevar a cabo una implementación eficiente se necesitará realizar un estudio previo que permita determinar cuál de los protocolos de enrutamiento propuestos para redes inalámbricas posee mejores prestaciones para este nuevo entorno aleatorio.



1.1. Planteamiento del Problema

Muy ligado al crecimiento de las grandes ciudades, tales como Caracas, está el crecimiento del transporte urbano y las necesidades de transporte público. El transporte público superficial del Distrito Metropolitano de Caracas está constituido por autobuses, minibuses y rústicos operados en su mayoría por organizaciones privadas tales como asociaciones civiles, cooperativas y compañías, operando en decenas de rutas. Actualmente, el sistema de transporte superficial trabaja de forma desarticulada, carece de control y fiscalización, consume tiempos de recorrido muy largos debido al gran volumen de tráfico vehicular, además de ser inseguro. Las unidades de transporte no cuentan ni con dispositivos de comunicación y ni con una red de comunicación articulada que permitan intercambiar información, tal como ubicación del vehículo, condiciones de tráfico, condiciones viales, condiciones del tiempo, incidentes de seguridad, entre otros, entre ellos y con sus supervisores. Algunas unidades de transporte como las de taxis y escasos autobuses, agrupados en líneas, han adquirido unidades de radio que permiten el intercambio de información de voz entre miembros de dicha línea. En su mayoría los choferes de unidades de autobuses, minibuses y taxis cuentan con teléfonos celulares de uso personal, que le permiten reportar algún incidente con el vehículo. Adicionalmente, el sistema de transporte público del Distrito Metropolitano de Caracas, así como en general del país, no utiliza ningún sistema de localización tales como GPS, que son importantes para mantener la localización de estos vehículos y permitir su rastreo y monitoreo constante.

La creación de una red de comunicación inalámbrica que permita comunicar las unidades de transporte público del Distrito Capital sería un paso inicial para lograr una mejor articulación entre las unidades de este sistema a la vez que permitiría atacar ciertos problemas relacionados al monitoreo de estas unidades, rastreos en caso de incidentes de seguridad, recepción de alertas, entre otros. Actualmente, algunos sistemas de transporte públicos tales como alguna redes de taxis utilizan sus propios sistemas de comunicación vía radio, que le permiten comunicarse entre miembros de sus propia línea. Igualmente, algunos dueños de unidades han integrado en sus vehículos sistemas de localización basados en GPS. Sin embargo, estas soluciones son aisladas y no contribuyen a aliviar el problema del sistema de transporte de forma global. Además, estas soluciones implican un gasto en equipos y en servicios que en muchos casos es trasladado a los usuarios del servicio.



Un primer paso en el desarrollo de una red como la planteada anteriormente es la realización de estudios de simulación de una red de transporte público de Caracas en función de los protocolos de enrutamiento para VANET. Esto con el fin de responder a la interrogante de ¿cuál sería el diseño de red VANET que ofrezca mejor rendimiento en términos de parámetros de red?

1.2. Objetivo General

Analizar los protocolos de enrutamiento a ser implementados en una red vehicular Ad-hoc para la red de transporte público de la ciudad de Caracas de acuerdo a modelos de movilidad de vehículos y tráfico.

1.3. Objetivos Específicos

- Determinar los protocolos de enrutamiento a ser estudiados.
- Definir escenarios y casos de estudio para los protocolos de enrutamiento.
- Diseñar la red vehicular para transporte público.
- Estudiar el comportamiento de la red vehicular en función de los protocolos de enrutamiento usando métodos de simulación.
- Capturar y analizar aspectos cuantitativos para realizar comparaciones entre protocolos.
- Proponer la red vehicular para el transporte público en función de los resultados obtenidos.

1.4. Justificación

Con este estudio se pretenden alcanzar varios objetivos en distintos planos: Por un lado, se quiere consolidar la investigación sobre redes vehiculares Ad-hoc, para su desarrollo e implementación en la ciudad de Caracas. En cuanto a la investigación en sí, y teniendo en cuenta que las redes VANET son un tema que surge recientemente y que presenta una diversidad de posibles aplicaciones que todavía están por desarrollar, se quiere centrar atención en la factibilidad del desarrollo de un sistema prototipo automático de control y prevención de accidentes de tránsito urbano.



Una de las fallas encontradas en muchos de los proyectos que involucran el desarrollo de redes de comunicación es la falta de un diseño preliminar que involucre los componentes de software (protocolos de comunicación) que regirán la comunicación eficiente entre los dispositivos. Los resultados de este proyecto permitirán el desarrollo e implementación de un sistema de comunicación inter vehicular para unidades de transporte público urbano del país que funcione eficientemente bajo las condiciones de tráfico y viales de las regiones urbanas del país.

1.5. Distribución del Documento

En este apartado se detallarán los procesos para los estudios de simulación que se quieren realizar. Debido a que desplegar y evaluar redes VANET supone un gran costo y una tarea ardua, la metodología basada en la simulación se muestra como una alternativa a la implementación real. Con el fin de evaluar la propuesta de este estudio en entornos realistas, en las simulaciones se tendrán muy en cuenta tanto la movilidad de los vehículos, como la transmisión de radio en entornos urbanos, especialmente cuando los edificios interfieren en la propagación de la señal de radio, así como también la ruta en el mapa que seguirán estos vehículos. Con este propósito, se pretenden utilizar herramientas para la simulación de VANET más precisas y aproximadas a la realidad, mejorando tanto la modelización de la propagación de radio, como la movilidad de los vehículos, obteniendo una solución que permite integrar mapas reales en el entorno de simulación.

Finalmente, se evaluarán y analizarán los resultados y prestaciones de los protocolos haciendo uso de una plataforma de simulación mejorada, que será el resultado del trabajo en conjunto de una herramienta de simulación de redes inalámbricas, una herramienta de creación de mapas reales de rutas y simulación de tráfico como una herramienta de generación de la movilidad de los vehículos a través de las rutas, evidenciando la importancia del uso de un entorno de simulación adecuado para conseguir datos más realistas y poder obtener conclusiones más relevantes.

El contenido de lo antes descrito está constituido de los siguientes capítulos dentro de este documento:



- *Capítulo 1 - Introducción:*

Introducción, planteamiento y explicación del problema a solventar. Objetivos a alcanzar junto con la justificación de los mismos. Distribución del documento.

- *Capítulo 2 - Marco Teórico:*

Marco teórico, donde se definen y explican términos claves para el desarrollo de la investigación, como son redes móviles inalámbricas y su caso para redes vehiculares, protocolos de enrutamiento para estas, y las tecnologías utilizadas para las mismas.

- *Capítulo 3 - Metodologías, Herramientas y Métricas:*

Metodologías de desarrollo en la investigación, fases para el desarrollo, herramientas de hardware y software, descripción de los simuladores utilizados y métricas obtenidas para el estudio de la simulación.

- *Capítulo 4 - Implementación de los Escenarios y Ejecución de las Pruebas:*

Explicación y selección de los escenarios a evaluar. Desarrollo e implementación de los escenarios seleccionados dentro de la plataforma de simulación. Ejecución de pruebas previas a la ejecución final. Ejecución de la plataforma de simulación y extracción de datos.

- *Capítulo 5 - Análisis de Resultados:*

Resultados finales en formato de gráficas y cuadros comparativos, contrastando los diferentes resultados en diferentes escenarios, realizando tanto el análisis de estos como un análisis final general.

- *Capítulo 6 - Conclusiones:*

Expresa las conclusiones luego de analizada la información final de la investigación, junto a las limitaciones presentadas y recomendaciones para posibles investigaciones futuras.



2. MARCO TEÓRICO

2.1. Redes MANET

Actualmente las tecnologías inalámbricas están jugando un papel muy significativo en la vida cotidiana con el auge de dispositivos portátiles, tales como: computadores inalámbricos, asistentes digitales personales, teléfonos móviles, dispositivos de audio, entre otros. Este crecimiento en la popularidad de la computación móvil es indiscutible. Los progresos de la microelectrónica para mejorar las capacidades de procesamiento y favorecer la miniaturización de los dispositivos, han ido de la mano con el desarrollo de aplicaciones software dirigidas a satisfacer las necesidades de diferentes segmentos de mercado que van desde el consumo masivo hasta los ambientes empresariales, e incluso militares.

La tendencia hacia el futuro no muestra precisamente un estancamiento de este comportamiento, sino por el contrario, una adopción más marcada de las tecnologías móviles como soporte fundamental a las tareas de la vida diaria: se estima que para el año 2017 siete trillones de dispositivos móviles servirán a 7 billones de personas alrededor del mundo (WWRF, *Technologies for the Wireless Future: Wireless World Research Forum*, 2008).

Esta tendencia ha generado nuevos paradigmas en la evolución de este tipo de tecnologías con el ánimo de satisfacer las necesidades del mundo actual, en el cual la movilidad, la flexibilidad y la capacidad de autoconfiguración de las redes que soportan los servicios, toman cada vez más relevancia. Escenarios dinámicos donde los dispositivos son capaces de detectar las características del ambiente circundante y responder a cambios en el mismo para adaptar el brillo de la pantalla, detectar el idioma que se esté utilizando o entregar el contenido adecuado por ejemplo, son una muestra clara de este tipo de requerimientos. Pero, al desplegar una red en un área densamente poblada, donde es necesario recurrir al uso de transmisiones con tecnología inalámbricas, surgen problemas en consecuencia de la interferencia resultante de usar el espectro libre, estos dispositivos requieren de algún tipo de infraestructura preestablecida como son los puntos de acceso para WiFi o WiMAX y las estaciones base para la telefonía celular, además la necesidad de que todas las estaciones de usuario tengan línea de vista (LOS) con la estación base, sin mencionar problemas de tráfico y enrutamiento de paquetes.



Recientemente, una nueva tecnología se abre paso y se establece para dar el soporte de red adecuado a los exigentes ambientes móviles modernos y los requerimientos planteados por escenarios similares a los descritos anteriormente: las redes móviles ad-hoc o MANET (*Movil Ad-hoc Network*).

2.1.1. Definición

Con el propósito de construir una definición para el concepto de redes MANET, es necesario referirse en primera instancia al concepto de redes Ad-hoc. En términos simples, una red Ad-hoc es aquella que no requiere una infraestructura preestablecida y puede conformarse sobre demanda, aún si los nodos que la conforman son estacionarios (Charles E. Perkins, 2001). Esto significa que existen dos perspectivas que pueden ser abordadas desde el punto de vista de configuración de la red: la primera, considera que cada uno de los nodos está dentro del rango de cobertura inalámbrica de los otros; en este sentido, la red Ad-hoc está completamente conectada. Sin embargo, una conectividad total puede ser inviable en la práctica, no sólo por la dificultad para lograr una cobertura plena, sino por el desperdicio de potencia para los terminales, que en la mayoría de los casos son operados por batería. La segunda perspectiva, un poco más interesante, considera la distribución de nodos en la red sin mantenerse necesariamente en el área de cobertura de otros nodos inalámbricos; ante la ausencia de una infraestructura de enrutadores disponible, es evidente la necesidad de un enrutamiento multi-salto entre los nodos. Esta característica evoluciona el concepto de redes Ad-hoc hacia el concepto de redes Mesh, también conocidas como redes de infraestructura instantánea (Richard Draves, Jitendra Padhye, and Brian Zill, 2004).

En las redes Mesh, los nodos no sólo operan como host sino también como enrutadores, reenviando los paquetes en representación de otros nodos que no se encuentran directamente en el rango de transmisión inalámbrica de sus destinos (Ian F. Akyildiz and Xudong Wang, 2005). Esto significa que cada nodo puede estar conectado con otro en la red de forma directa o indirecta, mejorando así la confiabilidad, ya que pueden existir muchos caminos posibles para conectar dos nodos específicos (John S. Bommer, 2007).

La figura 1 ilustra brevemente el funcionamiento de una red Mesh simple, compuesta por ocho nodos y sus respectivos enlaces. Los nodos tienen la capacidad de moverse relativamente entre ellos; mientras esto sucede, pueden romperse varios enlaces y establecerse otros nuevos. Por ejemplo, la figura muestra cómo el nodo N1 se aleja de N2 y establece nuevos vínculos con N6 y N8. Igualmente, la mayoría de algoritmos soporta la aparición de nuevos nodos móviles y la desaparición de nodos previamente disponibles.

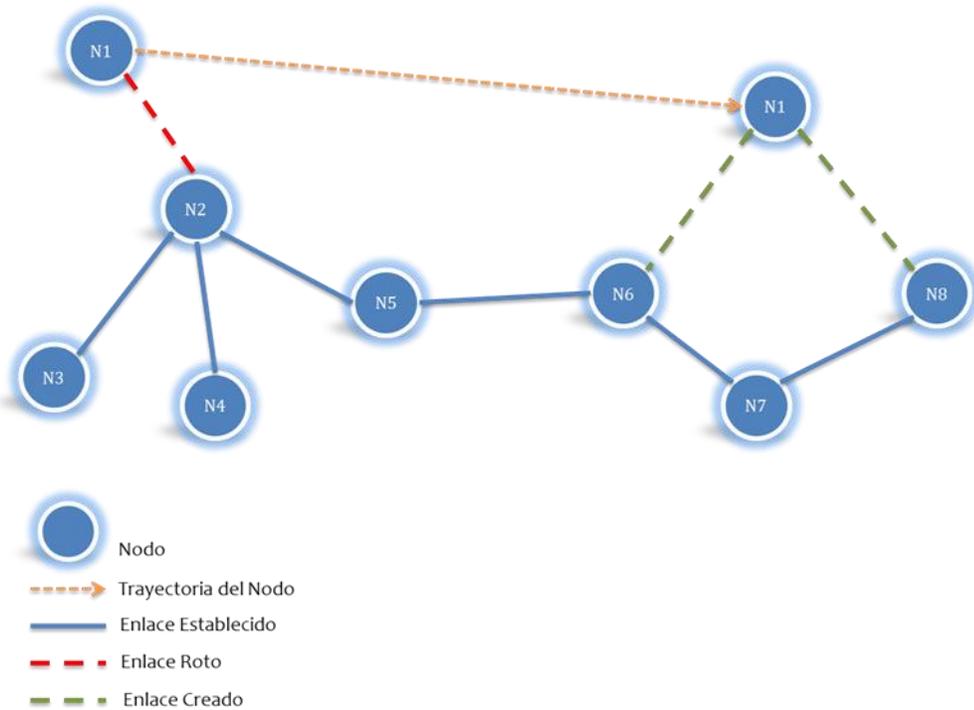


Figura 1. Funcionamiento General de Redes Mesh.

Tomando como base el contexto anterior, una Red Móvil Ad-hoc puede ser definida como una red auto-configurable de enrutadores móviles y sus hosts asociados, conectados a través de enlaces inalámbricos formando una topología arbitraria (NIST, 2008).

Las Redes Móviles Ad-hoc, denominadas también **MANET** como acrónimo de **Mobile Ad-hoc Network**, en algunas ocasiones también llamadas red malla de nodos móviles (*mobile mesh network*), son sistemas autónomos constituidos por nodos móviles que se comunican a través de enlaces inalámbricos o *wireless* de múltiples saltos y que poseen propiedades de auto-configuración. En términos más sencillos esto quiere decir que una red móvil Ad-hoc permite que una red se pueda establecer sin la necesidad de una administración central o de infraestructura preestablecida, ya que la red se conforma sólo de usuarios móviles capaces de transmitir y recibir información entre sí.

Durante los últimos años, se han realizado grandes esfuerzos en investigación en el campo de las MANET. Como ya se ha mencionado, estas, no poseen infraestructura ni puntos de acceso, cada nodo ejerce simultáneamente de nodo fuente, nodo final y de *router*. Es lo que se llama una red inalámbrica multi-salto (*multi-hop*) como se ilustra en la figura 2. El

término Ad-hoc viene del latín y se refiere en realidad a algo improvisado. Sin embargo, en el campo de las comunicaciones inalámbricas el propósito de las redes Ad-hoc es el de proporcionar flexibilidad y autonomía aprovechando los principios de la auto-organización.

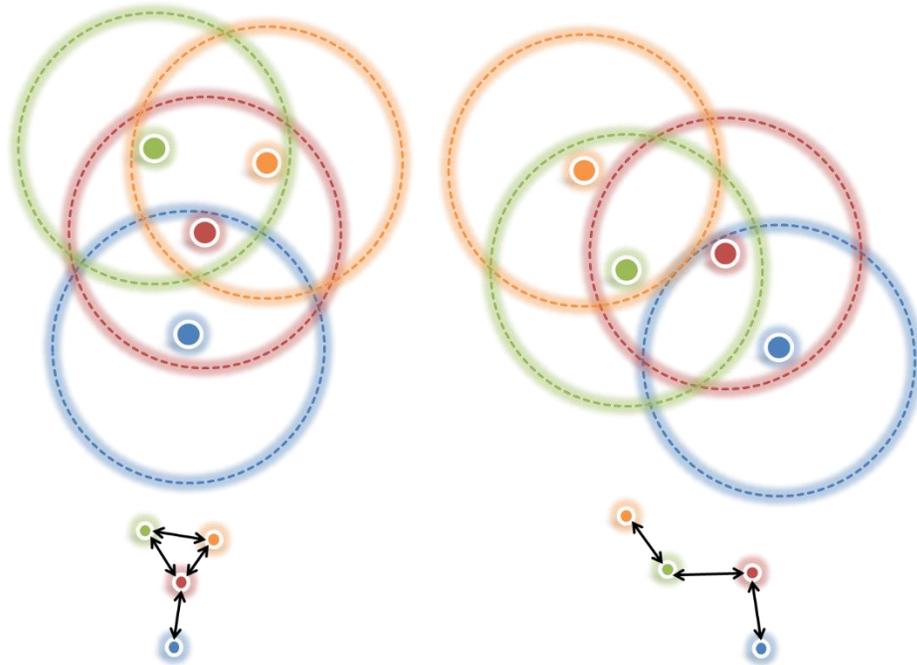


Figura 2. Funcionamiento multi-salto en redes ad-hoc.

Este último concepto es de gran importancia en el desarrollo de los sistemas de comunicaciones dinámicos actuales. Los sistemas que presentan auto-organización se establecen sin la necesidad de una entidad externa que los controle, ya que los entes individuales que conforman estos sistemas interactúan directamente entre sí de una manera localizada. Esto resulta ventajoso ya que una sola entidad no es la encargada de la organización general del sistema, sino que cada una de ellas aporta al comportamiento colectivo. La noción de sistemas auto-organizados apenas está empezando a aplicarse en el campo de las telecomunicaciones ya que se busca desarrollar nuevos sistemas completamente descentralizados.

Las redes móviles Ad-hoc o las redes sensoriales son ejemplos de dichos sistemas. La auto-organización es un fenómeno que se puede apreciar claramente en muchos procesos de la naturaleza. En un cardumen de peces, por ejemplo, cada pez individual establece su comportamiento basado en la observación de la posición y la velocidad de sus vecinos más próximos. No existe una entidad central que dirija todo el cardumen. El mismo caso se presenta en un grupo de aves. Todas tratan de viajar a una misma velocidad sin colisionar



entre sí y siguen un movimiento coordinado sin la presencia de un control central. De esta manera, se puede observar que en los sistemas que presentan auto-organización, el comportamiento sencillo de las entidades individuales conlleva a una organización sofisticada del sistema general.

Las MANET se basan en este comportamiento y por esta razón se constituyen en una tecnología ideal para establecer comunicación en aplicaciones donde los usuarios son móviles. No obstante, las redes móviles Ad-hoc presentan grandes retos para la comunidad científica porque las topologías en este tipo de redes son aleatorias, de múltiples saltos y dinámicas. Además presentan limitaciones en cuanto al uso del ancho de banda y al consumo de la potencia. Como se espera que las MANET puedan soportar cientos o miles de usuarios móviles, la escalabilidad se debe asegurar con el fin de obtener un buen desempeño.

De esta manera, los nodos deben poseer la habilidad para moverse libremente y organizarse aleatoriamente; así, una red MANET puede operar de forma autónoma o estar conectada a una red más grande. De acuerdo con esta definición, no se advierte en primera instancia una diferencia relevante con respecto a las redes Mesh; por esta razón, es importante aclarar la relación que existe entre estos dos conceptos que frecuentemente son tratados como sinónimos. Tanto las redes Mesh como las redes MANET hacen uso de las capacidades de *networking* Ad-hoc y las redes MANET utilizan la técnica multi-salto definida por las redes Mesh. En este orden de ideas, las MANET se pueden considerar como una especialización de las redes Mesh para entornos móviles (Ian F. Akyildiz and Xudong Wang, 2005).

2.1.2. Características de una red MANET

Las redes MANET poseen características que las hacen atractivas para desempeñarse en entornos que requieren una alta movilidad pero también una gran flexibilidad en la configuración de la red. En primera instancia, la capacidad multi-salto es esencial para que las redes MANET logren dos objetivos importantes: por un lado, extender la cobertura sin sacrificar la capacidad del canal y por otro, proporcionar conectividad sin línea de vista (Vivek Srivastava, Amr B. Hilal, Michael S. Thompson, Jawwad N. Chattha, Allen B. MacKenzie, & Luiz A. DaSilva, 2008). Como consecuencia, se logran tasas de transferencia más altas a través de enlaces cortos sin sacrificar cobertura, causando menos interferencia entre los nodos y haciendo un reuso de frecuencia más eficiente.

En este sentido, la topología y conectividad de la red dependen de la movilidad de los usuarios, gracias a las capacidades que tienen los nodos finales. Esta característica impone retos importantes en los protocolos de enrutamiento, así como también en la configuración y despliegue de la red que aún son materia de investigación (Jain *Shweta* & R. Das *Samir*, 2008) (L. *Mottola*, G. *Cugola*, & G. P. *Picco*, 2008).

Por otro lado, gracias al dinamismo de las redes MANET, éstas pueden pasar por diferentes estados de operación que pueden clasificarse de acuerdo con la tasa de conexiones y desconexiones, como se muestra en la Figura 3. De esta manera, una red MANET se considera estable si maneja una tasa baja de conexiones y desconexiones; de alta conexión, si existe una tasa alta de conexiones y una tasa baja de desconexiones; de alta desconexión, si la tasa de desconexiones es alta y la tasa de conexiones es baja; altamente dinámica si existe una tasa elevada de conexiones y desconexiones. Así, una red de este tipo puede operar en diferentes estados en instantes distintos, o un segmento de la red puede funcionar en modos diferentes al mismo tiempo (*Radhakrishnan, Racherla, Sekharan, Rao, & Batsell*, 2003).



Figura 3. Tasas de Conexiones y Desconexiones



A manera de síntesis, algunas de las principales características de las redes MANET, que han sido ya mencionadas, se pueden describir con más detalle:

- **Topología dinámica:** Los nodos son libres para moverse libremente a diferentes velocidades, por tanto, la red debe ser capaz de cambiar aleatoriamente y de manera impredecible.
- **Multi-salto:** El camino que debe transitar cada paquete entre origen y destino atraviesa múltiples nodos.
- **Escalabilidad:** La red ad-hoc puede crecer y tener varios miles de nodos (sensores, despliegue del campo de batalla, muros de vehículos...).
- **Limitaciones energéticas:** La mayoría de los nodos dentro de la red funcionan con baterías que tienen una vida limitada y por tanto la optimización de protocolos para que hagan un uso eficiente de ella se antoja crucial dentro de las redes MANET.
- **Ancho de Banda limitado:** Los enlaces inalámbricos siguen teniendo una capacidad menor que aquellos con una estructura fija.
- **Seguridad:** La capacidad de procesamiento de los nodos y la seguridad física limitada hace que este tipo de redes sean más vulnerables a posibles ataques que las redes cableadas.

2.1.3. Consideraciones de Enrutamiento

Como consideraciones generales el momento de pensar en el enrutamiento en una MANET se debe tener un objetivo o meta simple: lograr la transferencia de mensajes de un nodo a otro. Pero no únicamente poder conectar y transferir entre nodos, sino también de una manera eficiente.

Para ello, antes de preguntarse ¿cuál es el mejor camino?, se debe tener en cuenta ¿qué es el “mejor” camino?, por lo general los protocolos de enrutamiento tratan de optimizar una de las siguientes opciones:

- Ruta más corta (menor número de saltos).
- El tiempo más corto (menor latencia).
- El camino más corto ponderado (utilizar ancho de banda disponible, la batería).



Pero como bien se ha explicado antes, las MANET no aplican el enfoque centralizado, lo que hace revelar una importante cuestión: ¿Quién decide esto, Fuente o Nodos Intermedios?

De hecho puede ser de las dos formas antes mencionadas. La decisión de elegir la ruta o el próximo salto puede ser responsabilidad del nodo fuente o simplemente los nodos intermedios se encargarán de determinar el mejor camino a seguir en su próximo salto.

- *Source ("path") Routing*: Se asemeja al proceso en las aerolíneas de viaje, la fuente especifica toda la ruta y los nodos intermedios solo reenvían al siguiente salto.
- *Destination ("hop-by-hop") Routing*: Es más bien como servicio postal donde la fuente especifica el destino en la cabecera de mensaje y los nodos intermedios lo examinan al pasar por ellos, consultan sus tablas internas de enrutamiento para determinar el siguiente salto apropiado.

Ya sabiendo todo esto, aun se debe tener en cuenta que los protocolos de enrutamiento Ad-hoc poseen ciertas características o cualidades propias que obligan a tomar consideraciones especiales a la hora del enrutamiento. A continuación se mencionarán algunas de las propiedades de Enrutamiento MANET:

- No existe configuración externa de red: "auto-configuración".
- Eficiente cuando la topología de red es dinámica.
- Se auto inicia.
- Adaptable a las condiciones de red.
- Propiedades Cualitativas
 - Operación Distribuida.
 - Libertad de Bucle.
 - Operación Basada en Demanda.
 - Seguridad.
 - Operación en período inactivo.
 - Soporte de enlace unidireccional.
- Propiedades Cuantitativas
 - Procesamiento de Datos *End-to-End*.
 - Retrasos.
 - Tiempo de adquisición de Ruta.
 - Entregas fuera de orden (en porcentaje).
 - Eficiencia.
- La movilidad del Host.
- Tasa de enlace falla/repación pueden ser altos cuando los nodos se mueven rápido.



- Nuevos criterios de rendimiento se utilizan
 - Estabilidad de la ruta a pesar de la movilidad.
 - Consumo de energía.
 - Posición del Host.
- Soluciones dinámicas son mucho más difícil de desarrollar.

2.2. Redes VANET

En este apartado se dará una visión general de los desafíos que conllevan el planteamiento de redes vehiculares Ad-hoc (VANET). Basados en la cantidad y envergadura de la investigación actual sobre estas redes y la finalidad de la investigación, se limitará a realizar una exposición superficial en esta sección, quedando así muchos temas por abordar en cuanto a seguridad y calidad de servicio en particular. Aunque se detallará de forma minuciosa, por estar directamente vinculado con los objetivos del proyecto, los diferentes protocolos de enrutamiento en redes VANET y tecnologías que permiten la comunicación en este tipo de redes.

2.2.1. Definición

Introducidos ya los conceptos más importantes sobre las redes ad-hoc móviles, se enfatizará en un tipo en concreto, el cual se encuentra totalmente relacionado a los objetivos de este estudio.

Las redes VANET son un caso particular de las MANET, redes ad-hoc móviles, enfocadas a entornos vehiculares (autos, camiones, autobuses, motocicletas, etc.). Consiste en un conjunto de nodos que se comunican entre sí mediante vínculos inalámbricos sin la necesidad de una infraestructura de red fija. Cada nodo actúa como *router* y tiene capacidades de enrutamiento para redirigir paquetes hacia su destino.

En una red VANET cabría la posibilidad de que estos nodos formen la red en pleno movimiento de manera automática, por ejemplo mientras se circula por una autopista o avenida, por lo tanto, los nodos podrían moverse de forma arbitraria y comunicarse entre ellos (*vehicle-to-vehicle*), pudiendo contemplarse también que exista algún equipo fijo cercano que formará parte de la red y que también dotará a dicha red de una conexión hacia Internet (*vehicle-to-environment*), de la misma manera en la que acceden a los terminales móviles con acceso a Internet a través de *General Packet Radio Service* (GPRS) o *Universal Mobile Telecommunications System* (UMTS).

En la figura 4 se observa un típico escenario VANET en donde las comunicaciones con el resto de los vehículos proveen un sistema cooperativo mediante enlaces inalámbricos, permitiéndonos desarrollar gran cantidad de aplicaciones.

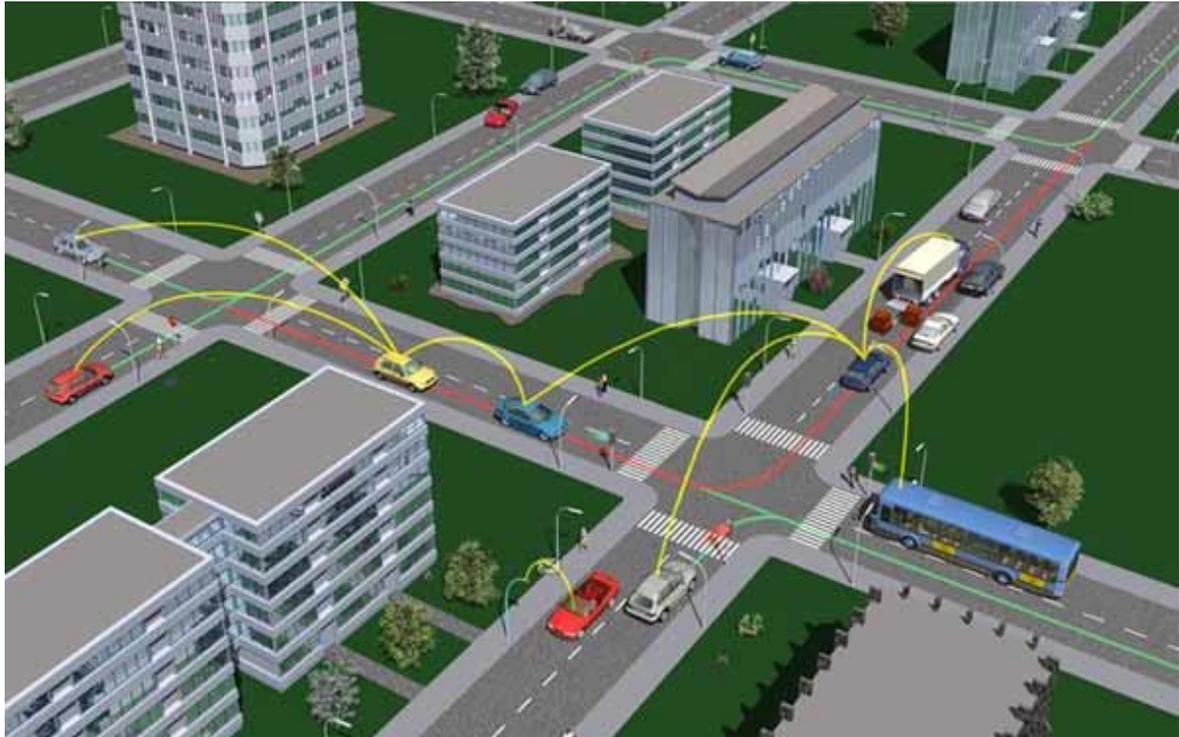


Figura 4. Escenario VANET – Fuente: Revista Digital 20minutos.es (25.09.2007).

2.2.2. Características de las Redes VANET

A continuación se verán un conjunto de propiedades que caracterizan a una red VANET:

- **Nodos Limitados:** Los nodos de este tipo de redes, embarcados en los vehículos, en la mayoría de los casos serán terminales con capacidades limitadas de procesamiento, almacenamiento y comunicación. Por lo que es crucial optimizar estos recursos con algoritmos refinados.
- **Autonomía:** Cada uno de sus nodos posee autonomía con capacidad de procesar la información que se intercambia en la red, por lo tanto, el control del mismo no



depende de una entidad externa ni centralizada, proporcionando así una tolerancia mayor a fallos al distribuir el control entre todos los nodos.

- **Enrutamiento Distribuido:** Estos nodos deben ser capaces de enrutar información como usualmente lo haría un *Router*, esto debido a su propiedad de autonomía. Los protocolos a implementar deben soportar esta característica.
- **Topología Variable:** En una red MANET los nodos se pueden mover de manera arbitraria y de igual manera sucederá en las VANET, aunque en estas últimas los nodos podrán seguir patrones de movimiento, tales como un circuito urbano de transporte. Sin embargo los vehículos se desplazan a mayores velocidades que un nodo de red móvil clásico. Debido a este posicionamiento tan variable pueden surgir pérdidas de paquetes significativos. Es necesario definir mecanismos que permitan detectar estas situaciones y minimizando sus efectos.
- **Capacidad Variable de Enlaces:** Cada nodo actuará como un *Router* y la información viajará por varios enlaces, teniendo estos, en algunos casos, más capacidad que otros. Esta característica es intrínseca al medio de transmisión de las redes inalámbricas y se intensifica más aun en este tipo de redes vehiculares.

2.2.3. Tecnologías en las Redes VANET

Para la transferencia de información en una red VANET se ha recurrido a un gran número de tecnologías que permiten, de forma inalámbrica, la transmisión de datos. A continuación se detallan las diferentes tecnologías más utilizadas en comunicaciones VANET:

- **IEEE 802.11:** Basada en el estándar 802.11, es popularmente conocida como WiFi. El hecho de operar en bandas libres y que sus versiones b, g y n se hayan extendido tanto, ha traído como consecuencias que la mayoría de equipos móviles y PDAs la traigan incorporado de fábrica. Posee unos cuantos centenares de metros de alcance con un ancho de banda en teoría de hasta 600 Mbps, dependiendo de la versión. En la revisión 802.11i fue mejorada la seguridad del estándar que siempre ha existido en sus versiones.
- **802.11p:** Este estándar opera en el espectro de frecuencias de 5.9 GHz y de 6.2 GHz, especialmente indicado para automóviles. Es la base de las comunicaciones dedicadas de corto alcance DSRC (*Dedicated Short Range Communication*). La tecnología DSRC permitirá el intercambio de datos entre vehículos y entre automóviles e infraestructuras en carretera. También es conocida como WAVE por *Wireless Access for the Vehicular Environment*. Por estar esta tecnología



destinada a soportar las comunicaciones vehiculares se verá con un detalle mayor más adelante. Es una evolución del estándar IEEE 802.11a con modificaciones a nivel físico y MAC para mejorar su comportamiento en el entorno vehicular y dar soporte a sistemas de transporte inteligente (*Intelligent Transportation Systems*). Hoy día existen muchos proyectos en desarrollo basados en 802.11p.

- **Bluetooth:** Es una especificación industrial para Redes Inalámbricas de Área Personal (WPANs). También conocido como 802.15.1. Es la tecnología más extendida en cuanto a comunicaciones inalámbricas personales. Existen diferentes clases o versiones y cada una como diferencias de alcances, consumo de potencia y tasas de transferencia.
- **UWB:** Es una tecnología en el rango de las PAN (*Personal Area Network*) basada en 802.15.3. Permite enviar paquetes de información a tasas elevadas (480 a 500 Mbps) pero su alcance es muy limitado (<10 ms.). Funciona emitiendo a muy baja potencia en un espectro enorme.
- **ZigBee:** Es la tecnología más utilizada en redes de sensores ad-hoc. Se basa en el estándar 802.15.4. Presenta anchos de banda muy pequeños y cobertura reducidas (250 Kbps hasta 75 mts.). Es de gran utilidad para enviar poca información en pequeñas distancias. La gran ventaja es que su consumo es extremadamente reducido.

Una vez presentadas las tecnologías más usadas en redes VANET se puede ahora sintetizarlas en el cuadro 1:

<i>Tecnología</i>	<i>Cobertura*</i>	<i>Banda de Frecuencia</i>	<i>Tasas</i>	<i>Consumo</i>
802.11b	~460 mts	2,4 GHz.	~11 Mbps	Alto
802.11g	~350 mts	2,4 GHz.	~54 Mbps	Alto
802.11n	~400 mts	2,4 GHz y 5 GHz	~600 Mbps	Medio
802.11p	~1000 mts	5,9 GHz y 6,2 GHz	~27 Mbps	
WiMAX (802.16e)	~3,5 Kms	2,3 y 2,5 Ghz	30 Mbps	Alto
Bluetooth v4 (802.15.1)	~80 mts	2,4 GHz.	24 Mbps	Bajo
UWB (802.15.3)	<10 mts	3.1 GHz hasta 10.6 GHz.	480 Mbps	Bajo
ZigBee (802.15.4)	~100 mts	2,4 GHz	250 Kbps	Muy Bajo

Cuadro 1. Características aproximadas de tecnologías inalámbricas consideradas.

* En Exteriores

También puede detallar su alcance y ciertas características observando la figura 5.

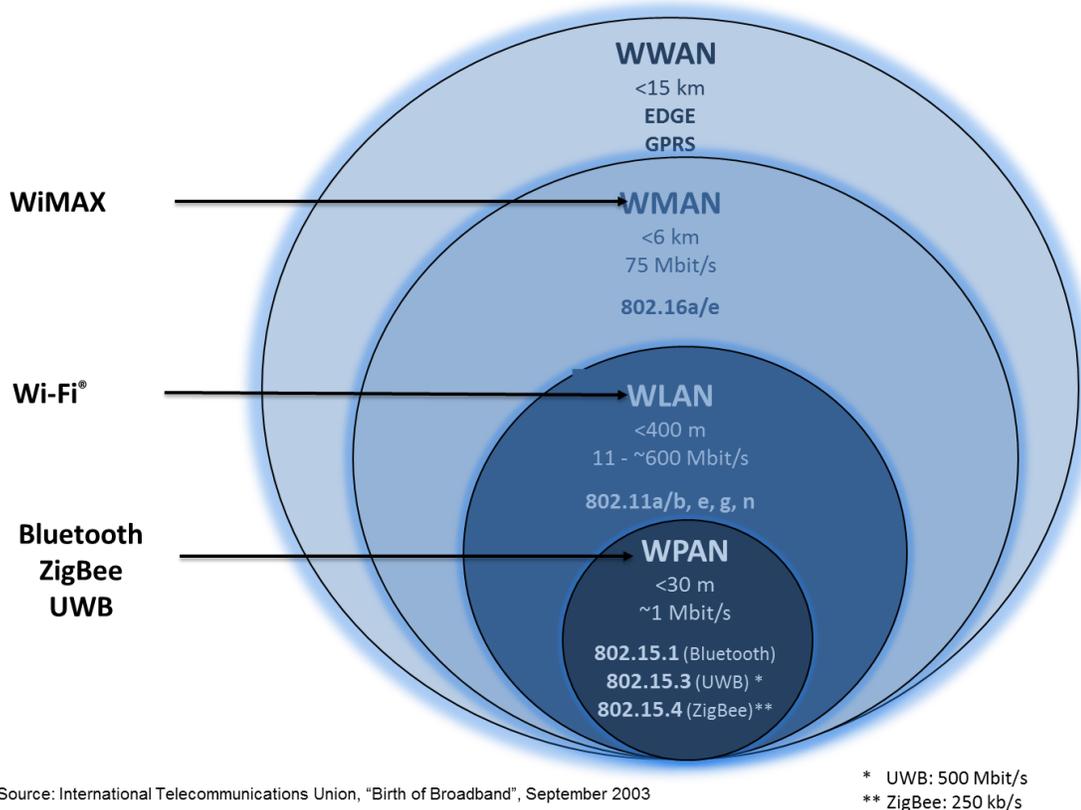


Figura 5. Fuente: International Telecommunications Union, "Birth of Broadband", September 2003.

Cuando se habla de redes inalámbricas de área extensa WWAN, se consideran las de mayor y más amplio alcance de todas las redes inalámbricas. Por ejemplo, todos los teléfonos móviles están conectados a una red inalámbrica de área extensa. Las tecnologías principales son:

- GSM (*Global System for Mobile Communication*)
- GPRS (*General Packet Radio Service*)
- EDGE (*Enhanced Data Rates for GSM Evolution*)
- UMTS (*Universal Mobile Telecommunication System*)

En cuanto a **IEEE 802.11p**, también conocida como *Wireless Access for the Vehicular Environment (WAVE)*, como se mencionó anteriormente, es una evolución del estándar IEEE 802.11a con modificaciones a nivel físico y MAC para mejorar su comportamiento en el entorno vehicular. Al igual que IEEE 802.11a, WAVE utiliza OFDM, pero con tasas de transmisión de 3, 4.5, 6, 9, 12, 18, 24, y 27 Mbps en canales de 10 MHz. Utiliza 52 sub-



portadoras moduladas utilizando BPSK, QPSK, 16-QAM o 64-QAM así como codificaciones de ratios 1/2, 2/3, o 3/4, que indican la cantidad de bits que se ponen para proteger la señal del ruido. El valor más pequeño es el que protege más la señal, normalmente para los datos de alta prioridad se utiliza 1/2 o 2/3. Cuan más protegida esté la señal, menor será el “*bit rate*” útil en el receptor.

En cuanto a la canalización, la norma define 7 canales no solapados de 10MHz en la banda de los 5.9GHz, teniendo así 6 canales de servicio (SCH) y uno de control (CCH). El CCH es utilizado como canal de referencia para realizar una primera detección de los vehículos cercanos como paso previo al establecimiento de las comunicaciones. Al mismo tiempo, dicho canal es utilizado para anunciar los servicios disponibles en los canales SCH (acceso a Internet, descarga de contenidos, etc.). Teniendo en cuenta que el canal CCH es el canal de referencia, WAVE lo emplea para la transmisión en modo *broadcast* de mensajes de seguridad vial. Por la importancia de los mensajes que se transmiten y la necesidad de garantizar su correcta recepción, el canal CCH transmite con una tasa de datos de 6Mbps, correspondiente a una modulación QPSK con un ratio de codificación de 1/2.

En la capa MAC, WAVE también utiliza el método de acceso de IEEE 802.11, basado en CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*). Para mitigar el problema del *hidden* terminal, WAVE mantiene el intercambio de mensajes RTS/CTS (*Request-To-Send/Clear-To-Send*), aunque se encuentra deshabilitado en el canal CCH por transmitir siempre en modo *broadcast*. Como consecuencia, todos los nodos que utilizan el canal de control están sujetos a dicho problema, incrementando el riesgo de pérdidas de paquetes y de congestión de canal.

En el Cuadro 2, como bien se ha mencionado, se realiza una breve comparación a nivel físico entre el estándar IEEE 802.11a y su derivado IEEE 802.11p.

Parámetros	IEEE 802.11a	IEEE 802.11p	Cambio
Tasa de Bit (Mbit/s)	6, 9, 12, 18, 24, 36, 48, 54	3, 4.5, 6, 9, 12, 18, 24, 27	Mitad
Modulación	BPSK, QPSK, 16QAM, 64QAM	BPSK, QPSK, 16QAM, 64QAM	No
Ratio de Codificación	1/2, 2/3, 3/4	1/2, 2/3, 3/4	No
# de Sub Portadoras	52	52	No
Duración de Símbolo	4µs	8µs	Doble
Tiempo de Guarda	0.8µs	1.6µs	Doble
Período FFT	3.2µs	6.4µs	Doble
Preámbulo	16µs	32µs	Bajo
Sub Portadora	0.3125 MHz	0.15625 MHz	Mitad

Cuadro 2: – Comparación entre las capas físicas usadas en IEEE 802.11a y IEEE 802.11p



2.2.4. Desafíos

Gracias a muchas de las limitaciones y problemas que presentan este tipo de redes y debido a las características propias de las mismas, se abren nuevos campos de investigación para la comunidad científica. A continuación se verán una serie de tópicos que quedan abiertos a investigación:

Para violar la seguridad de una red cableada es necesario tener acceso físico al cable, sin embargo en una red inalámbrica las consideraciones de seguridad no son tan robustas, esto debido a que las comunicaciones se realizan a través del aire que actúa como canal de difusión. Aún más en las VANET este problema se ve agravado, ya que no existe una infraestructura que pueda centralizar los servicios de seguridad como cifrado de paquetes o autenticación. Aun hoy en día, se siguen realizando investigaciones para proporcionar mecanismos de seguridad a estas redes.

Ahora bien, para proporcionar Calidad de Servicio (Quality of Service) se necesitan mecanismos de reserva de recursos. En una red VANET la reserva de recursos se complica debido a los cambios drásticos que sufre la topología de estas redes. Además es imprescindible contar con calidad de servicio en cualquier red, ya que aplicaciones de tiempo real exigen un alto nivel. Las soluciones propuestas para QoS solo son teóricas, simuladas o simplemente implementadas para pocos nodos, sin proporcionar una respuesta definitiva.

Anteriormente se ha destacado, como característica de una VANET, que los nodos embarcados en los vehículos serán terminales de carácter limitados, por lo cual es importante optimizar los procesos de comunicación y procesamiento para garantizar un bajo consumo de energía. Este recurso depende mucho de la tecnología usada.

En cuanto al enrutamiento de los paquetes en una VANET, este se verá afectado por los cambios de topología extremadamente rápidos, debido a la constante movilidad de los nodos de la red. Esto implica una necesidad de cambiar las tablas de enrutamiento de dichos nodos constantemente. Para resolver esta problemática se han propuesto soluciones de enrutamiento, las cuales en este estudio, se exponen en detalle en el siguiente capítulo.



2.3. Protocolos de Enrutamiento

La inestabilidad de las topologías, la continua movilidad de los nodos y la falta de infraestructura centralizada, entre otras, son algunas de las características que se tienen que considerar a la hora de implementar el enrutamiento en las Redes Ad-hoc Móviles, ya que son, estas mismas características, las que hacen obsoletos los distintos protocolos que se utilizan en las redes fijas.

En este capítulo se describirán y clasificarán los diferentes protocolos de enrutamiento para redes VANET. En la actualidad existen más de 80 protocolos de enrutamiento. La explicación a este gran número es que todavía están en fase de investigación. En capítulos anteriores se indicaron ciertas consideraciones para la implementación de los protocolos de enrutamiento de una Red Ad-hoc Móvil, estos deben ser capaces de superar el desafío de trabajar de una manera distribuida, improvisada y automática.

2.3.1. Clasificación de los Protocolos de Enrutamiento

Se pueden considerar ciertos criterios al momento de clasificar los protocolos de enrutamiento VANET, esto para facilitar la descripción de las características comunes entre ellos.

El gran abanico de criterios a tomar en cuenta serán con respecto a:

- El alcance: *unicast*, *multicast*, *broadcast*, etc.
- La ubicación geográfica: geográficos, no geográficos.
- Niveles de jerarquía: jerárquicos, planos.
- El modo de descubrimiento de las rutas: proactivo, reactivo, híbrido.
- El responsable de enrutamiento: enrutamiento en el origen, salto a salto.
- La cantidad de rutas hacia el destino: *singlepath*, *multipath*.
- El algoritmo implementado: vector distancias, estado de enlace.



Pero para este caso de estudio se escogió el modo de descubrimiento de las rutas ya que podrá proporcionar información relevante para la investigación.

En las VANET y Redes Móviles Ad-hoc en general se usan dos esquemas distintos para descubrir las rutas hacia los nodos destino de la red. Estos son el esquema reactivo y el proactivo.

2.3.2. Protocolos de Enrutamiento Proactivos

Los protocolos proactivos, independientemente de las necesidades de enrutamiento, intentan tener en cada momento una visión precisa y previa del estado de la red, para que al momento de necesitar una ruta tenerla de antemano. Consisten en mantener las tablas actualizadas de enrutamiento en todo momento a través del envío de mensajes de control de forma periódica. La mejor ruta a tomar se calcula previamente, mediante un algoritmo y la información de la red.

Por lo general, este tipo de protocolos utilizan algoritmos de tipo Dijkstra o Bellman-Ford para conseguir dicha ruta. Esta característica suele ofrecer un buen comportamiento en los escenarios de movilidad alta, esto debido a que procura una respuesta rápida ante solicitudes de ruta. Estos algoritmos reducen considerablemente el tiempo para encontrar una ruta, en comparación de los protocolos reactivos. Sin embargo, para mantener una actualización permanente de las rutas, surge la desventaja de los protocolos proactivos, estos requieren una carga adicional en la red inalámbrica debido a la transmisión periódica de mensajes de control e indudablemente si la red llegase a tener gran tamaño, la cantidad de recursos necesarios para almacenar cada tabla de enrutamiento actualizada en cada uno de los nodos de la red sería mucho mayor cada vez más, limitando así su escalabilidad.

Entre los proactivos más destacados se encuentran DSDV "*Destination-Sequenced Distance Vector*", OLSR "*Optimized Link State Routing Protocol*" y FSR "*Fisheye State Protocol*".



2.3.2.1. DSDV (*Destination-Sequenced Distance Vector*)

Destination-sequenced Distance-Vector Routing (DSDV) (Daniel Lang. 2003) es un protocolo proactivo adaptado del tradicional RIP (*Routing Information Protocol*). Uno de sus principales objetivos es evitar que en la actualización de las tablas de enrutamiento existan los problemas de bucle. Para lograr esto se añade un nuevo campo en las tablas RIP. Este campo contiene un número de secuencia que permite distinguir entre una tabla antigua y una más reciente.

El algoritmo que se implementa en este protocolo está basado en el vector de distancias. Esto quiere decir que almacena tablas con todos sus destinos accesibles junto con el siguiente salto, la métrica, y un número de secuencia de la entrada en la tabla generado por el nodo destino. Las tablas se mandan en modo *broadcast* de forma periódica o cuando ocurre un cambio significativo de la topología de red. Una ruta es considerada mejor que otra si tiene un número de secuencia mayor o, en caso de empate, si la distancia al destino es menor.

Cuando un nodo N1 detecta que la ruta hacia cierto destino N2 se ha roto, inunda la red con una actualización de esa entrada en la que se ha incrementado el número de secuencia en uno y la distancia se marca como infinita. Cuando un nodo N3 recibe este mensaje incorpora a su tabla la actualización de la entrada hacia N2 a través de N1 siempre y cuando N3 no tuviera una entrada mejor para alcanzar N2.

Para conseguir una cierta consistencia en las tablas de enrutamiento de cada nodo al cambiar la topología de la red, las actualizaciones deben ser frecuentes y suficientemente rápidas para que cada nodo pueda tener una visión realista de la red en un momento dado. El problema fundamental de DSDV es la elevada sobrecarga de control que genera. Al no haber una especificación estándar, no hay productos comerciales basados en este protocolo. Sin embargo, es la base sobre cual se han desarrollado otros protocolos como por ejemplo AODV.



2.3.2.2. OLSR (*Optimized Link State Routing Protocol*)

Optimized Link State Routing (OLSR) (Clausen – Jacquet 2003) es un protocolo proactivo basado en el estado de enlace. OLSR es una optimización directa del algoritmo de Estado de Enlace adaptado a los requisitos específicos de una WLAN con alta movilidad. La optimización consiste principalmente en la reducción del tamaño de las tablas de enrutamiento intercambiadas, así como del número de retransmisiones de cada nodo necesarias durante los períodos de inundación. La clave del algoritmo reside en el uso de retransmisiones multipunto con nodos MPR (*Multi Point Relay*).

Los nodos intercambian periódicamente mensajes HELLO con sus vecinos que permiten detectar la presencia de un nodo vecino así como recoger información referente al estado del enlace con ese vecino. En los mensajes HELLO se pueden incluir información indicando si el nodo es un nodo MPR. Usando esa información cada nodo elige dentro de su conjunto de vecinos un subconjunto que declara subconjunto MPR. Así, cada nodo tiene conocimiento de un subconjunto de nodos MPR que le permite tener conectividad con todos los nodos distantes de uno o dos saltos. De este modo, sólo los nodos MPR se encargarán de retransmitir los mensajes *broadcast*.

Para descubrir la topología de la red, los nodos intercambian información acerca del estado de enlace que los conectan con los nodos MPR. Los intercambios son periódicos o generados por eventos relativos a ruptura de enlace.

Incluir en las tablas sólo los enlaces a los nodos MPR reduce el tamaño de las mismas, lo que permite reducir el tráfico durante su intercambio. Al mismo tiempo permite que las rutas que se vayan creando a posteriori sean óptimas en cuanto a números de saltos ya que sólo usan nodos MPR. OLSR se adapta bien a redes con gran número de nodos y alta movilidad.

2.3.2.3. FSR (*Fisheye State Protocol*)

Fisheye State Protocol (FSR) (Gerla – Hong 2002) es un protocolo proactivo basado en el concepto de estado de enlaces. El "Ojo de un pez" es un mecanismo mediante el cual se captura con detalle los píxeles que se encuentran cerca del punto de focal. El detalle disminuye a medida que se aumenta la distancia al punto focal. FSR se basa en una analogía de ese algoritmo. Mantiene distancias exactas y alta calidad de la información relativa a los nodos los más cercanos y pierde progresivamente detalles a medida que la distancia al nodo aumenta.



FSR, de manera similar al algoritmo de estado de enlace manda mensajes de información de forma periódica o seguido a un evento de ruptura de enlace. Sin embargo, los mensajes no inundan la red sino que se intercambian únicamente entre vecinos locales. En la implementación de este protocolo, cada nodo almacena:

- Lista de vecinos
- Tabla con la topología (TT)
- Tabla con el próximo salto de la ruta
- Tabla de distancia al destino

En redes grandes, el tamaño de los mensajes intercambiados puede ser muy grande y el consumo de ancho de banda importante. Por esas razones FSR utiliza diferentes frecuencias para el envío de esos mensajes. Las entradas correspondientes a nodos más cercanos son propagadas con una frecuencia más alta. Mediante esa técnica, FSR funciona bien en redes de gran tamaño y mantiene baja la sobrecarga u *overhead* sin comprometer la exactitud de la computación de rutas cuando el destino está cerca. Cuando la movilidad de los nodos aumenta, las rutas hacia los destinos remotos se hacen menos exactas. Sin embargo, cuando un paquete se acerca a su destino, encuentra información de encaminamiento en redes VANET de encaminamiento más exacta. Como resultado, FSR es más útil para redes de gran tamaño donde la movilidad es alta y el ancho de banda bajo.

2.3.3. Protocolos de Enrutamiento Reactivos

Por otra parte, en los protocolos reactivos o por demanda, la sobrecarga y el uso de recursos es mucho menor, ya que crean entradas en las tablas de rutas solo cuando estas son requeridas. Sin embargo, al tener que buscar las rutas recurrentemente los retrasos en las transmisiones son mayores por el cálculo del enrutamiento sobre la marcha, pero este retraso suele ocurrir al principio de cada transmisión mientras se resuelve el enrutamiento.

Se puede nombrar entre estos al protocolo AODV “*Ad hoc On-Demand Distance Vector*” y DSR “*Dynamic Source Routing*” como ejemplos.



2.3.3.1. AODV (*Ad hoc On-Demand Distance Vector*)

El *Ad-hoc On-Demand Distance-Vector Routing* (Daniel Lang. 2003), basado en el algoritmo de vector distancias, es un protocolo reactivo que está construido sobre el protocolo DSDV, anteriormente detallado, minimizando el número de paquetes *broadcast* requeridos para crear rutas, y al ser bajo demanda, los nodos que están en el camino no tienen que participar en el intercambio de tablas ni que mantener la ruta.

A pesar de ser un protocolo reactivo, AODV tiene la peculiaridad de transmitir mensajes alertando sobre su presencia, de forma periódica, mediante una técnica llamada *Link Layer Feedback*. Esto permite que los nodos tengan información de sus vecinos más cercanos y mantengan sus tablas actualizadas reflejando los cambios en la topología cercana. Estas tablas se mantienen actualizadas a lo largo del tiempo, eliminando las entradas que ya no se necesiten.

Si un nodo origen N1 quiere enviar un mensaje a un nodo destino N2 y no posee una ruta válida hacia este nodo, comienza el mecanismo de descubrimiento (*Path Discovery*). Primero se difunde un mensaje *broadcast* con una petición de ruta, *ROUTE REQUEST* (RREQ) a todos sus vecinos. En este mensaje va incluida su dirección, la dirección del nodo destino N2, y el último número de secuencia recibido de N2, esto solo si se hubiera recibido algún dato anteriormente. Este mensaje inunda la red y los nodos por los que pasa guardan una ruta inversa hacia N1, lo que implica que AODV sólo soporta enlaces bidireccionales. Cuando llega a un nodo que dispone de la ruta hacia N2, se comprueba el número de secuencia para el destino N2. Si este es mayor que el incluido en el mensaje, se ha encontrado una ruta válida hacia N2 y el nodo que dispone de la entrada hacia N2 manda un mensaje de respuesta *ROUTE REPLY* (RREP) de vuelta hacia N1 siguiendo la ruta creada durante el *broadcast*. En este mensaje se incluye el último número de secuencia recibido por el emisor del mensaje RREP. Los nodos que reciben el RREP guardan una entrada con la información de la ruta hacia N2 que recibieron del nodo que ha respondido el mensaje, aunque solo se guarda en la tabla el siguiente salto y no la ruta entera. Si pasado un cierto tiempo y que no se ha recibido ningún RREP, N1 considera que no hay ruta válida hacia N2 en ese momento.

2.3.3.2. DSR (*Dynamic Source Routing*)

Dynamic Source Routing (Daniel Lang. 2003) es un protocolo reactivo que opera bajo demanda. Dos mecanismos son los que componen a DSR: el descubrimiento y el mantenimiento de rutas que permiten a un nodo origen descubrir y mantener las rutas hacia un nodo destino cuando se necesita mandar tráfico en la red Ad-hoc. Se basa en una técnica llamada "*Source Routing*". La función de dicha técnica es definir e identificar la mejor ruta completa hacia un nodo destino. El nodo origen envía una trama de exploración inundando



la red. Al recibir una réplica de la trama exploradora, en la cabecera de la trama cada nodo se agrega explícitamente, y actualiza sus tablas con la información obtenida en la cabecera de dicha trama. El descubrimiento de rutas es el mecanismo que permite a un nodo origen N1 que desea mandar tráfico a un nodo destino N2, obtener la ruta hacia N2. Si N1 no dispone de ninguna ruta hacia N2, comienza el mecanismo de descubrimiento de rutas mediante un *broadcast* del *Route Request Packet* (RREQ). La dirección destino, la dirección del nodo fuente y un número único de identificación, forman el contenido de este paquete. Cada nodo que recibe un paquete RREQ, revisa si conoce la ruta hacia el destino. Si no la conoce, se reenvía el paquete.

Si la tiene la ruta hacia el nodo destino contesta en sentido inverso con un *Route Reply Packet* (RREP). Todos los nodos que participan al reenvío del RREP añaden su dirección en la cabecera del paquete, creando de ese modo la ruta completa hasta el destino.

En cambio, el mantenimiento de rutas consiste en la capacidad de detectar que una ruta almacenada en una tabla ya no se puede usar debido a un cambio de topología. El mantenimiento de rutas detecta que un enlace en la ruta hacia un nodo N2 ha desaparecido. Se producen paquetes de error del tipo *Route Error Packets* en un nodo cuando la capa de enlace encuentra un problema grave de transmisión. Este paquete de error contiene las direcciones de los dos nodos que están unidos por el enlace que falló. En este caso, si N1 conoce otra ruta hacia N2 se puede usar, o bien se vuelve a invocar el mecanismo de descubrimiento de rutas para remplazar la ruta caída hacia N2. El mantenimiento de rutas sólo tiene cabida cuando N1 está mandando tráfico a N2.

DSR es puramente reactivo, lo que implica que no existe ningún tipo de mensaje que se transmita periódicamente, lo que permite reducir en gran medida el tráfico de control en la red y aprovechar más los recursos de red para paquetes útiles. Además cada vez que se lleva a cabo el descubrimiento de rutas, los nodos implicados pueden extraer y almacenar información sobre la topología de red, lo que ahorra muchos mensajes de control.

Cuando un nodo recibe un RREQ se puede producir el problema de múltiples respuestas simultáneas. Para solucionar esto, se introduce un pequeño retardo variable en la respuesta de cada nodo con una ruta en su caché. Antes de responder con una de las rutas conocidas, cada nodo debe efectuar las siguientes acciones:

- Elegir un retardo $d = H \times (h - 1 + r)$, donde h es la métrica (en saltos) de la ruta que se debe enviar, r es un número flotante aleatorio entre 0 y 1, y H es un pequeño retardo constante: $H \geq 2 \times R$, (R es el retardo máximo de propagación en el enlace).
- Retardar la transmisión de un RREP para este nodo por un período igual a d .

- Durante ese período si el nodo recibe otra respuesta mejor a dicho RREQ proveniente de otro nodo, se cancela la respuesta y se retransmite la nueva.

Se puede observar un ejemplo del proceso de descubrimiento y mantenimiento de rutas realizado en el protocolo DSR ilustrado en las figuras 6 y 7 respectivamente.

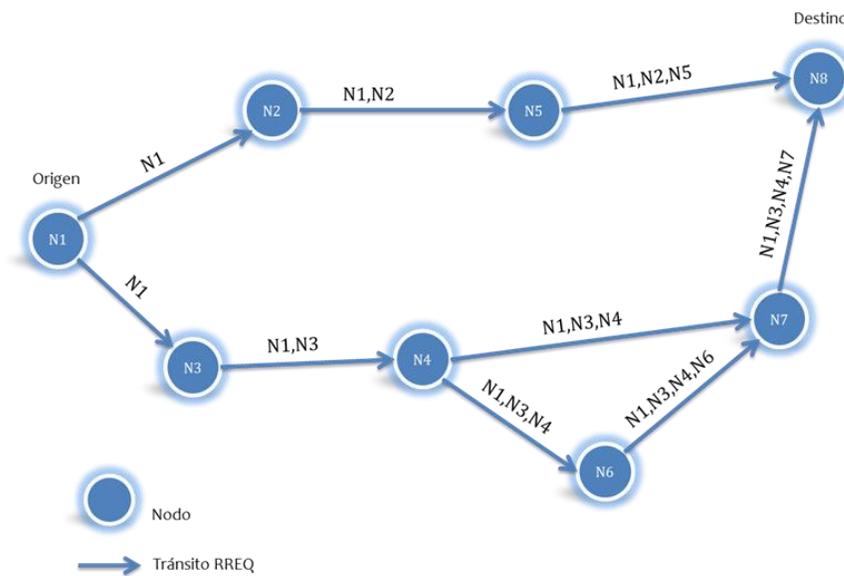


Figura 6. Tránsito RREQ, Mecanismo de Descubrimiento de Rutas DSR.

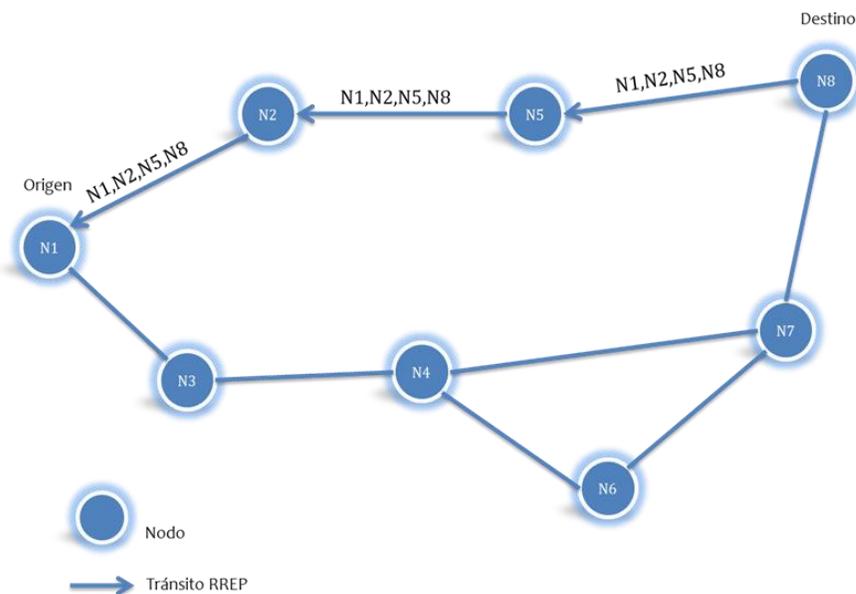


Figura 7. Tránsito RREP, Mecanismo de Descubrimiento de Rutas DSR.



3. METODOLOGÍA Y HERRAMIENTAS

Con el fin de estudiar los protocolos de red seleccionados, se necesita crear un ambiente de prueba donde se puedan ejecutar estos protocolos de manera controlada. Para esto, se ha elegido utilizar una simulación como banco de pruebas (o *testbed*), con el que se podrán configurar los parámetros y situaciones necesarias en la búsqueda de resultados para ser estudiados, sin requerir todos los equipos disponibles físicamente.

3.1. Fases para el Desarrollo

En este estudio para llevar a cabo un buen desarrollo basado en simulaciones, es necesario seguir una serie de pasos que lleven la investigación hasta la obtención de los resultados más relevantes, veraces y fiables.

A continuación se detallarán cada uno de los pasos:

- *Escogencia y descripción de los escenarios:*

Se dice que en un escenario se colocan los elementos necesarios para representar un ambiente donde se llevará a cabo algún acto. En el caso de las simulaciones, luego de definido los parámetros esenciales que se mantendrán fijos durante las ejecuciones, se nombran los parámetros a variar para obtener distintos casos. Cada una de estas variaciones se toma como un escenario realista de prueba a ser estudiado.

- *Selección de métricas de importancia a utilizar:*

Una vez elegidos los distintos escenarios, el siguiente paso es decidir que métricas son representativas y relevantes para el posterior análisis de los resultados del proyecto. Entre muchas métricas disponibles, se deben seleccionar aquellas que representen los cambios más importantes, de donde se puedan identificar ventajas o desventajas en el rendimiento de cada uno de los protocolos a estudiar.



- *Implementación de los escenarios:*

Se implementará una plataforma de simulación engranada para emular comportamientos en los escenarios ya definidos. Es aquí donde se configurarán los escenarios utilizando la sintaxis que provea esta plataforma. Además, en esta fase se realizarán las pruebas necesarias previas a las ejecuciones finales para verificar el buen funcionamiento de los escenarios.

- *Ejecución de pruebas y generación de resultados:*

Una vez diseñados los escenarios y configurados, se procede a ejecutar las simulaciones con los parámetros finales. Estas simulaciones generarán resultados que serán verificados, ordenados y graficados, para así tener información suficiente y una visión más práctica que permita ver el comportamiento de los protocolos en los ambientes seleccionados.

- *Análisis de resultados:*

Una vez generados todos los resultados, cuadros comparativos y gráficas, se deben interpretar el significado de estos. Se compararan los resultados entre ellos, determinando luego del análisis la información crítica para llegar a la conclusión definitiva del estudio.

3.2. Herramientas para el Desarrollo

Para desarrollar este estudio se necesitaron el siguiente conjunto de herramientas:

3.2.1. Hardware y Sistemas Operativos

En su totalidad la parte práctica e implementación de plataforma de este estudio se basó en software libre, esto para facilitar la reproducción de las simulaciones para futuros estudios por parte de otros investigadores. Por lo que todo software se implementó sobre el sistema operativo Linux, distribución Ubuntu 12.04.5 LTS (Precise Pangolin) sobre un equipo físico que se detallará en el cuadro 3:



Procesador	Pentium® Dual-Core CPU E5300 @ 2,60Ghz
RAM	4,00 GB DDR3
Almacenamiento	3x450 GB HDD en RAID5
Sistema Operativo	Ubuntu 12.04.5 LTS (Precise Pangolin)
Tipo del Sistema	64 bits

Cuadro 3. Características del Equipo físico a utilizar.

3.2.2. Simuladores y Otros Softwares

Hoy en día existe una amplia gama de software de simulación para redes inalámbricas, tales como OPNET, NCTUns, OMNeT++, Shunra, QualNet, etc. Aunque para el estudio se decidió utilizar aquellos de código abierto, software libre o libre distribución (al menos para los fines académicos del proyecto), teniendo una especial atención a uno que se ajusta a los propósitos del estudio, como lo es NS-2 (*Network Simulator*), ya que posee licencia de software libre y académicas.

Entre otros objetivos que se buscan alcanzar con estas herramientas o plataformas de simulación se tiene:

- El soporte para sistemas operativos libre, en especial GNU/Linux.
- La opción de poseer nodos multi-interfaz debe estar soportada por el simulador.
- Permitir la creación de redes con topologías complejas.
- Diversidad de Protocolos Inalámbricos.
- Posibilidad de un nivel físico configurable.
- Posibilidad de uso, modificación o implementación desde cero de protocolos.
- Modo de Comando y modo GUI.

Por estas razones se ha inclinado a la aceptación a NS-2 como la base de la plataforma de simulación. Además, el hecho de que un programa sea muy utilizado es una ventaja porque dispondrá de numerosas implementaciones de protocolos, documentación y foros de usuarios.



3.2.2.1. NS-2

NS-2 es un simulador de redes de comunicación de eventos discretos extremadamente popular en el ámbito académico por su extensibilidad. Su desarrollo comenzó en 1989 a partir del simulador *REAL Network Simulator*, y ha contado con el soporte y financiación de diferentes entidades públicas y privadas desde su comienzo hasta el día de hoy (DARPA, Xerox, CONSER, Sun, UCB, etc.). Con su filosofía de desarrollo abierta, NS-2 se ha convertido, debido a su decisivo crecimiento, en una de las herramientas más completas tanto en el estudio de redes cableadas como inalámbricas. Aunque se sigue trabajando en NS-2 y siguen apareciendo nuevas versiones, desde 2006 está en desarrollo paralelo una versión (no compatible) del simulador, ns-3.

En cuanto al funcionamiento NS-2 es capaz de simular las diferentes capas OSI (física, MAC, enlace, IP, transporte, aplicación) pudiendo modificar las características de cada una de ellas. NS-2 se apoya en dos lenguajes de programación para su correcto funcionamiento. Por un lado, el usuario introduce las especificaciones del escenario a simular a través del lenguaje OTcl (*Object Tcl*), versión extendida de Tcl (*Tool Command Language*). Por otro lado, la implementación de los protocolos se escribe en C++.

NS-2 registra cada paquete de datos que atraviesa la red junto a sus características principales como por ejemplo, el instante de recepción, el número de secuencia, el tipo de paquete, etc. Estos parámetros forman la traza como resultado del procedimiento de NS-2, dicha traza que facilitará el posterior análisis de las simulaciones.

Secuencia de Ejecución: A la hora de abordar la simulación de un protocolo en NS-2, es necesario seguir los siguientes pasos:

- En primer lugar se podría implementar el protocolo a analizar mediante la incorporación de código C++ y OTcl dentro del núcleo de NS-2. Este paso no es necesario del todo si se desea usar protocolos ya propuesto por NS-2. NS-2 incluye muchos protocolos ya implementados en su versión descargable (AODV, TORA, DSDV, DSR, por ejemplo). Pero puede ser necesario añadir la implementación de nuevos protocolos o modificación de los antes mencionados.
- Descripción de la simulación mediante OTcl. Se trata de definir el escenario a simular.
- Ejecución de la simulación. Se inicia el simulador proporcionándole el archivo de descripción de simulación previamente definido.



- Existen múltiples formas de analizar los resultados de la simulación. En primer lugar, para visualizar la ejecución se puede usar el animador gráfico de NS-2, nam (Network Animator. 2013. – Recuperado de <http://www.isi.edu/nsnam/nam/>) . Pero no proporciona los datos cuantitativos necesarios para un estudio riguroso.

Análisis: NS-2 cuenta con una licencia GNU General Public License (GPL) versión 2 (software libre) y está diseñado para sistemas operativos POSIX como FreeBSD, GNU/Linux, SunOS, Solaris o Cygwin. El código con núcleo en C++ (437k líneas) e interfaz de usuario en OTcl (230k líneas). OTcl (*Object Tcl*) es una versión extendida de Tcl (*Tool Command Language*). Como capa de interfaz entre C++ y OTcl se usa TclCL.

Protocolos inalámbricos:

NS-2 posee diversidad de protocolos para conexiones inalámbricas para sus simulaciones, entre ellos se tiene:

WiFi: Además del módulo CMU (implementación tradicional de 802.11 en NS-2), en la versión 2.33 incorpora dos nuevos módulos, que no solo un complemento, sino que lo sustituyen por completo y el usuario debe decidirse por cuál de los tres utilizar:

CMU: Único existente hasta la versión NS-2.33, el modelo desarrollado por la *Carnegie Mellon University* implementa DCF (*Distributed Coordination Function*) y usa el esquema RTS / CTS / DATA / ACK para paquetes *unicast* y DATA para paquetes *multicast*. Según la comunidad se trata de una implementación no del todo robusta, a decir verdad muy pobre, ya que obvia muchos aspectos del estándar y los simplifica hasta el punto que compromete la fidelidad de los resultados. Una de las carencias más notorias es la falta de presencia de *beacons*, paquetes de control de potencia de transmisión o asignación dinámica de canales. Así como la no implementación de exploración de canales *scanning*, asociación ni autenticación. Estas dos últimas carencias se han resuelto por Ilango Purushothaman para NS-2.33 y ahora se da un soporte mucho más completo para el modo infraestructura.

802.11Ext: Desarrollado en la Universidad de Karlsruhe por un equipo de Mercedes-Benz, implementa un nuevo 802.11 MAC y capa física (Mac802_11Ext y WirelessPhyExt). Entre otras características el módulo presenta un diseño estructurado del MAC, realiza el cálculo de la SINR (*Signal-to-Interference Noise Ratio*), posee soporte para múltiples modulaciones, control de pérdida de paquetes en capa física e implementación del modelo Nakagami de atenuación que resulta adecuado para largas distancias.



dei802mr: Derivada del módulo MCU, la implementación del DEI (*Department of Information Engineering* de la Universidad de Padua) incorpora modelos físicos con diferentes tasas de transmisión para 802.11b/g. Además lleva a cabo un cálculo más preciso del SINR, del PER (*Packet-Error-Rate*) a partir de las curvas PER vs SINR vs tamaño paquete proporcionadas para 802.11b/g y modificables por el usuario, la potencia de ruido es configurable en los scripts Tcl. Asimismo, la afectación entre nodos se puede limitar según la distancia entre los mismos. En las 3 implementaciones el valor *SlotTime* es un parámetro configurable desde los scripts Tcl, mientras que *ACKTimeout* es un valor fijo en el código C++. Sería necesario algún ajuste.

En los 3 diferentes módulos se emplean distintas implementaciones de estándares 802.11:

802.11a: Soportado por el módulo 802.11Ext.

802.11b/g: Soportados por el módulo dei802mr con velocidad variable (para b: 1/2/5.5/11Mb, para g: 6/9/12/18/24/36/48/54Mb).

802.11e/n: Los módulos oficiales de NS-2 no soportan los estándares. Se han publicado diferentes implementaciones externas, aunque ninguna para la última versión del NS-2.

WIMAX 802.16-2004: Existen diferentes módulos externos que le dan un soporte, si se puede decir, completo.

NS-2mesh80216: Es un módulo desarrollado por la Universidad de Pisa y el *Georgia Institute of Technology* para NS-2.33 que da soporte 802.16-2004 para redes *mesh* pero no para PMP. En la capa MAC esta implementada completamente la transmisión de datos, el *handshake* de 3 vías descrito en el estándar y un planificador FEBA (*Fair End-to-end Bandwidth Access*). Es las especificaciones se advierte que la capa MAC de este módulo no puede interoperar con los algoritmos de enrutamiento de NS-2 ni con los módulos de interferencia a nivel físico, así que el módulo contiene unas clases propias de enrutado (OSPF) y modelado del canal inalámbrico.

Existe un módulo desarrollado en la Universidad Chang Gung en Taiwan, CSIE, que implementa 802.16 PMP para NS-2.29. Consta de la funcionalidad básica de la CS (*Sublayer Converge*), el CPS (*MAC Common Part Sublayer*) y la capa física.

NIST: Este módulo provee soporte para 802.16d y 802.16e. En capa física implementa OFDM y en capa MAC TDD.



En cuanto a nodos Multi-interfaz, esta característica no está soportada para interfaces inalámbricas, lo que implica una de las deficiencias de diseño más graves de NS-2. A pesar de esto se han desarrollado, a lo largo del tiempo, diversos correcciones o parches para resolver el problema, entre los que se puede nombrar MITF, TENS (parche para NS-2.1b9a), y Hyacinth (parche para NS-2.1b9a).

Sobre el enrutamiento, existen en NS-2 tres tipos de algoritmos de enrutamiento para comunicación *unicast*. Entre ellos se tiene el algoritmo de caminos mínimos de Dijkstra Estático, en donde las rutas se computan sólo al inicio de la comunicación, su contra parte Dinámica, en este caso las rutas se computan cuando hay cambios en la topología durante la simulación y por último enrutamiento DV (*Distributed Bellmand-Ford*, también conocido como *Distant-Vector*). Si lo desea, el usuario puede no usar protocolo alguno de enrutamiento y crear manualmente las rutas. Para redes *mesh* con nodos móviles están implementados los protocolos DSDV (*Destination Sequence Distance Vector*), DSR (*Dynamic Source Routing*), TORA (*Temporally-Ordered Routing Algorithm*) y AODV (*Ad-hoc On-demand Distance Vector*).

En cuanto a la posibilidad de configurar a nivel físico, cada módulo visto anteriormente tiene sus particularidades, pero en general los parámetros del canal físico son muy configurables.

En la versión tradicional de NS-2 se implementa un sistema sencillo y simple para la pérdida de paquetes: si la potencia recibida de un paquete supera el umbral CS (*Carrier Sense*) de recepción, los paquetes pasan a la siguiente capa; en caso contrario, se descartan. No tiene un mecanismo de simulación de errores bit por bit que diera lugar a un PER. No obstante, y como ya se ha visto, la librería *dei80211mr* ofrece mecanismos (curvas PER vs SNIR) que resuelven estas limitaciones.

La interfaz con NS-2 se hace a través de comandos. Existen aplicaciones gráficas como *Network Editor* (que forma parte de *nam*), *Extended Name Editor*, *nsbench* o *nsg*. La opción mayoritaria en cuanto a la visualización de los resultados, es el interfaz gráfico oficial *nam* (*Network Animator*), figura 8, y aplicaciones desarrolladas por terceros como *tracegraph* (Installing TraceGraph on Ubuntu. 2014 – Recuperado de <http://getch.wordpress.com/2011/04/20/installing-tracegraph2-02-application-in-ubuntu/>).

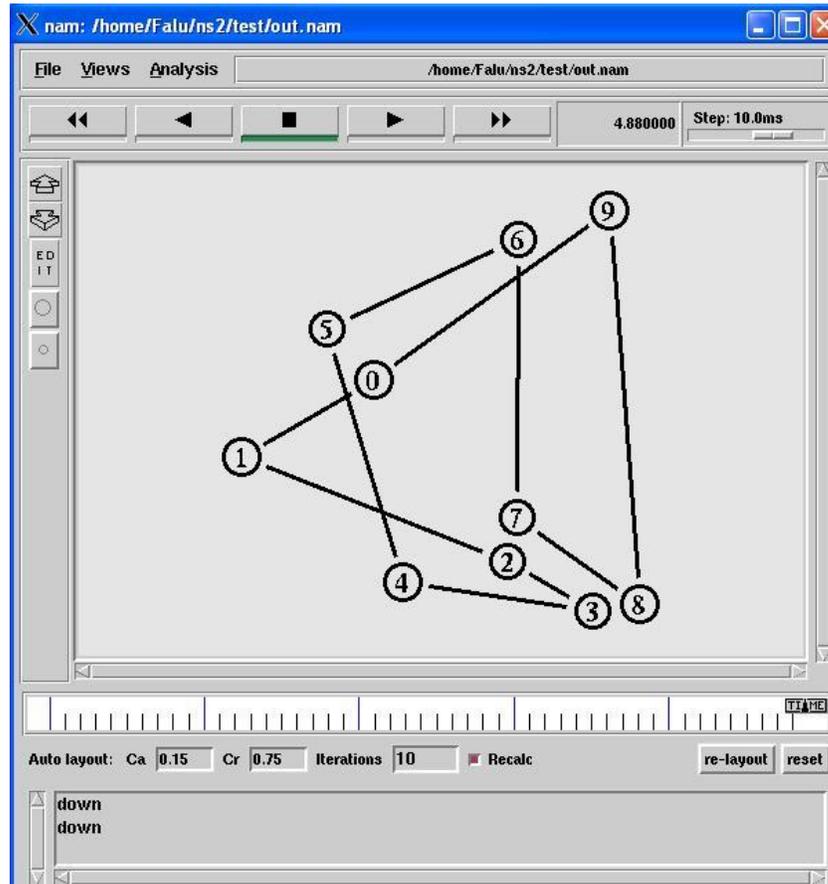


Figura 8. GUI de nam

En cuanto a la comunidad y estado del proyecto la última versión estable: NS-2.35 del 04 de noviembre de 2011 (Network Animator. 2013. – Recuperado de <http://www.isi.edu/nsnam/nam/>). No se sabe con exactitud si se sigue trabajando en nuevas versiones. Para esa misma fecha se lanzó nam-1.15, xgraph-12.1, tcl-debug-1.7. La última versión de OTcl-1.14 fue el 30 de octubre de 2011 junto con TclCL, un paquete formalmente conocido como libTcl.

Sin duda el simulador de red con la comunidad más amplia. Sirva como ejemplo la extensa página oficial de nsnam (Network Animator. 2013. – Recuperado de <http://www.isi.edu/nsnam/nam/>) de contribuciones de código, aunque en realidad se trata sólo de una pequeña parte del código disponible. Existen listas de desarrolladores, usuarios, de anuncios de nuevas versiones, y una especializada en usos educativos del simulador.

3.2.2.2. SUMO

En este apartado se describirá el simulador de tráfico de entornos VANET más conveniente para este estudio. Se ha escogido SUMO (Eichler, S., Schroth, C., Eberspächer, J., 2006).

Una de las grandes ventajas de SUMO o *Simulation of Urban Mobility* (SUMO - Simulation of Urban Mobility. 2013 – Recuperado de <http://sumo.sourceforge.net/>) es que se trata de un simulador de VANET de código abierto y gratuito desarrollado por la DLR o *Deutsche Gesellschaft für Luft und Raumfahrt*, es decir, el Centro Aeroespacial Nacional Alemán.

Se trata de un simulador que permite definir entornos de movilidad reales gracias a la utilización de mapas digitales, ofrece la posibilidad de utilizar diferentes tipos de vehículos, carreteras que cambian en cuanto a su composición (carriles, velocidad, prioridades, etc.) y un amplio abanico de posibilidades e integración con otros simuladores debido a que es código abierto (véase figura 9).

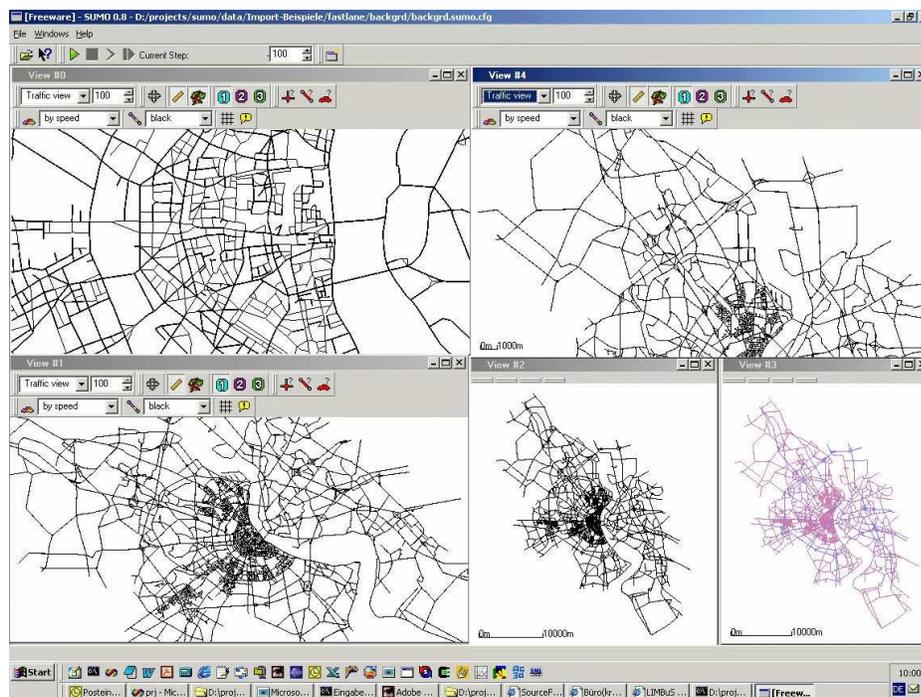


Figura 9. Captura del simulador SUMO

Una de las limitaciones de SUMO, es el hecho de que éste calcula las rutas de los vehículos antes de realizar la simulación, cosa que dificulta la evaluación de la comunicación *vehicle-to-vehicle* afectada por variaciones en el comportamiento de los mismos durante la propia simulación, es decir, que será más difícil evaluar a tiempo real la comunicación entre vehículos frente a una variación provocada de su comportamiento de una forma repentina (aceleración, cambio de ruta, frenada, etc.) ya que con SUMO se tiene que prever todo esto antes de hacer la propia simulación.

3.2.2.3. MOVE (*Mobility Model Generator for Vehicular Networks*)

Esta herramienta (MObility model generator for VEhicular networks (MOVE). 2013 – Recuperado de <http://www.csie.ncku.edu.tw/~klan/move/index.htm>) permite crear mapas de carreteras urbanas o interurbanas (incluso se puede importar mapas reales). Permite crear también flujos de tráfico controlados, implementar semáforos, etc. Ha sido desarrollada por la *School of Computer Science and Engineering* de la *Universidad New South Wales* de Australia en Java.

La primera tarea a realizar es crear un mapa, definiendo nodos y carreteras. Se hace de manera muy intuitiva a través de la interfaz gráfica de usuario como se muestran en las figura 10 y 11.

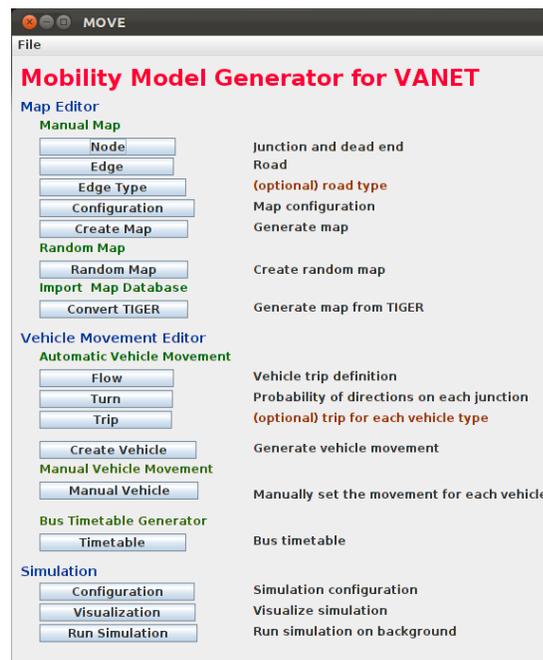


Figura 10. Interfaz del Menú de MOVE

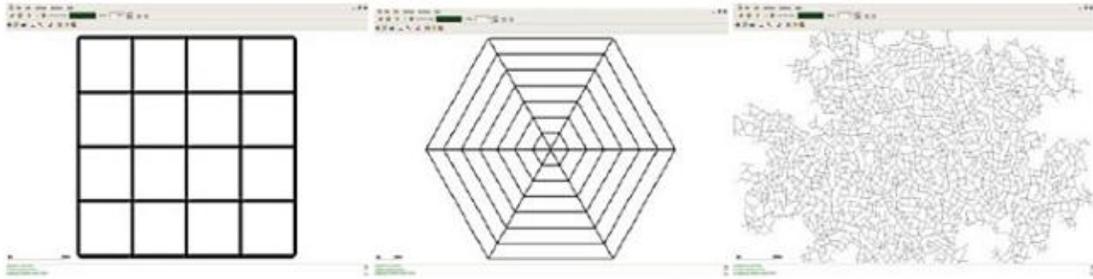


Figura 11. Tipos de Mapas generados en MOVE



4. ESCENARIOS, MÉTRICAS Y PLATAFORMA DE PRUEBAS DE SIMULACIÓN

En este apartado se detallan e implementan los estudios de simulación que se quieren realizar. Se entiende por estudio de simulación los escenarios aplicados a las topologías de red descritas anteriormente. Para cada escenario definido se obtendrán datos para ambas topologías (circuito urbano y autopista) con los diferentes perfiles de tráfico para cada una de las topologías (alto, medio y bajo). Además, para realizar la simulación se diseñó un mapa comprendido entre sectores de La Castellana, pasando por Altamira, hasta Los Palos Grandes en la ciudad de Caracas, Venezuela. Urbanizaciones que contienen rutas de la flota de transporte público TransChacao del Municipio Chacao.

Para cada caso se harán 2 tipos de pruebas: una usando conexiones TCP y otra usando conexiones UDP, que son los dos protocolos de comunicación más usados por los servicios corrientes en la actualidad. Las aplicaciones usando esos protocolos deben ser fáciles de simular. Se ha elegido FTP sobre TCP y CBR sobre UDP.

4.1. Escenarios

Escenario 1: VANET Pura

Este escenario debe permitir simular una situación clásica en redes VANET, es decir una comunicación inalámbrica entre dos vehículos en modo ad-hoc, sin infraestructura ni red de respaldo. Dado que la comunicación es directa y no implica nodos intermedios, el rendimiento de esta comunicación sólo va a depender del esquema de movilidad y de la tecnología inalámbrica considerada. En este caso, y en espera del estándar 802.11p, se usará la tecnología inalámbrica 802.11b para las comunicaciones vehiculares. Por lo cual, con la tecnología fijada, este escenario permite tener una comparación de los efectos de los esquemas de movilidad en las comunicaciones, viendo cómo influyen en los rendimientos de red.

El escenario es el siguiente: se establecerá una comunicación desde un nodo origen cuando el destino se encuentra a su alcance, mientras que si la distancia de

separación es mayor que dicho radio, la comunicación se cortará. A continuación véase la figura 12.

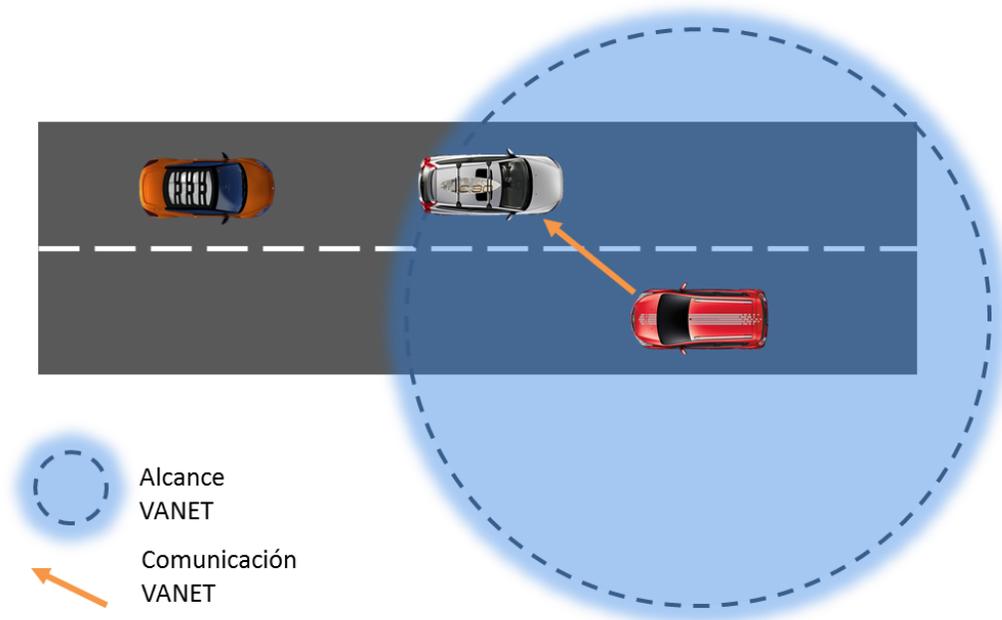


Figura 12. Escenario de comunicación VANET Pura

Escenario 2: Comunicación VANET pura a través de nodos intermedios

Este escenario permitirá obtener una comparación de diferentes protocolos de encaminamiento Unicast. Existen muchos estudios que se proponen comparar protocolos de encaminamiento cuando el número de conexiones entre los vehículos se multiplica. En este trabajo se centró en la influencia de la topología de red para cada uno de los protocolos considerados. Por lo cual, se eligieron dos nodos para las comunicaciones. Estos nodos deben ser elegidos con cuidado ya que deben ser representativos de las diferentes densidades de cada topología y densidades de tráfico. Se deben elegir nodos que en las diferentes topologías mantengan la misma trayectoria en el tiempo y lo único que cambie de una densidad a otra es el número de nodos que se encuentra entre ambos vehículos a lo largo de la simulación.

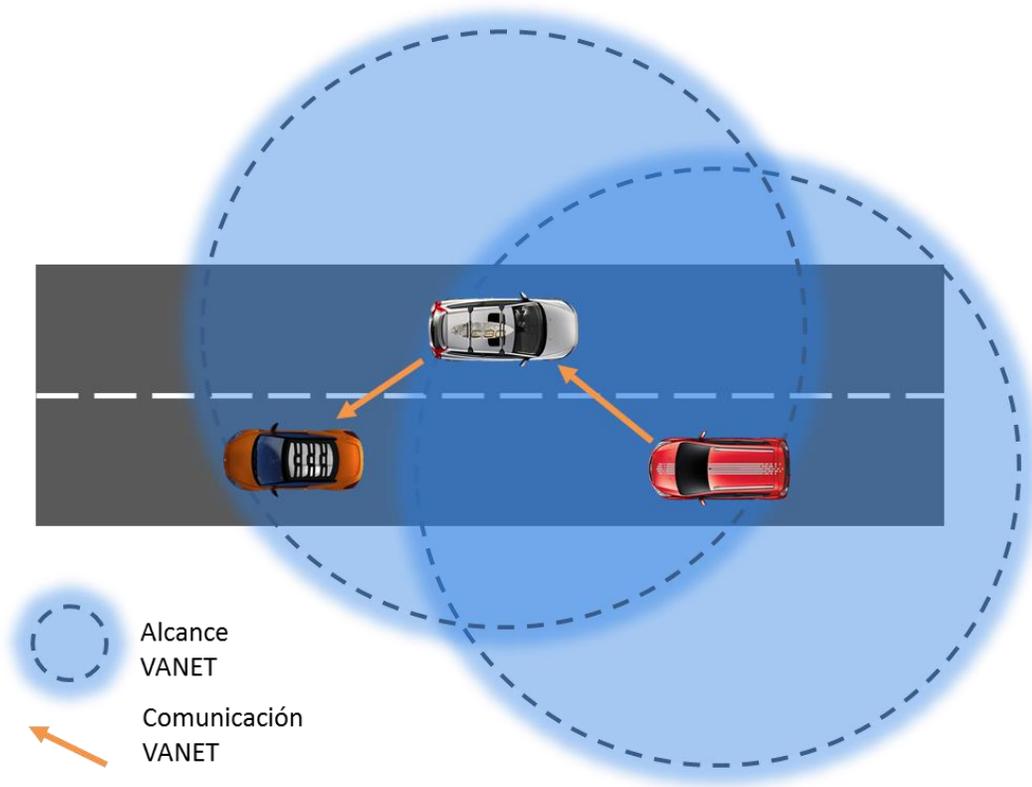


Figura 13. Escenario de comunicación VANET Pura con nodos intermedios

Se pueden tomar estos dos escenarios como escenarios base, sobre los cuales actúan ciertos parámetros que se pueden variar para abarcar un mayor rango de posibilidades. Estas variaciones se puede realizar en cuanto a la densidad de tráfico (Alto, Medio, Bajo), el tipo de circuito (Urbano, Autopista), el protocolo de comunicación (UDP, TCP) y la velocidad de los nodos de acuerdo a la densidad y circuito.



4.2. Métricas Utilizadas en el Desarrollo

Las medidas utilizadas para estimar el desempeño de la red son las siguientes:

4.2.1. Rendimiento de Tráfico (*throughput*)

Se llama *throughput* o rendimiento al volumen de trabajo o de información neto que fluye a través de un sistema en un tiempo determinado y que llega correctamente a su destino. En redes, es usual utilizar bits transmitidos en un segundo (bps) o bytes transmitidos por segundo (Bps), o alguna unidad múltiplo en cuanto a la cantidad de datos (kilobits por segundo, megabytes por segundo, etc.). Este dato tiene sentido únicamente en conexiones TCP ya que el protocolo de comunicación UDP no se preocupa por saber si un paquete ha llegado o no al destino.

4.2.2. Porcentaje de Éxito (*Packet Delivery Fraction – pdfr*)

Cuantitativamente medir la cantidad de paquetes que llegan a su destino en contraste a todos los paquetes que fueron enviados, diría mucho de la calidad del servicio de red. El *pdfr* es el porcentaje de paquetes enviados que llegan al destino. Un paquete enviado se puede perder bien porque el retardo de transmisión es tal que no llega a tiempo y se considera como perdido, o bien porque se ha dañado y los mecanismos de control del destino lo rechazan. El *pdfr* es un buen indicador para obtener una comparativa relativa a las tasas de pérdidas para cada protocolo. Un protocolo con tasas de pérdida altas no es satisfactorio ya que los rendimientos de red empeoran.

4.2.3. Retardo Extremo a Extremo

Es imprescindible medir el retardo de transmisión, es decir el lapso necesario para que un paquete viaje de la fuente al destino. Es importante considerar el retardo por varios motivos. Por un lado, el retardo define la velocidad global de una red por lo cual da indicaciones de rendimientos importantes. Por otro lado, algunos servicios no pueden soportar retardos muy elevados, como los servicios de tiempo real por ejemplo, que necesitan calidad de servicio. Por lo cual, la medida del retardo extremo a extremo para cada protocolo de encaminamiento es una medida importante para la comparativa.



4.2.4. Sobrecarga (*Overhead*)

Es la información adicional de control generada por cada protocolo extra a los paquetes de datos netos. Este indicador es extremadamente importante, sobre todo en este caso donde se trata de medir el rendimiento de los protocolos de encaminamiento. Es obvio que un protocolo de encaminamiento introduce *overhead* al ser un mecanismo de control. Sin embargo, un protocolo que introduce una sobrecarga excesiva para su funcionamiento influye de forma muy negativa sobre el rendimiento de la red, ya que para enviar un paquete de datos útiles, se tendrá que enviar más paquetes de control, consumiendo precioso ancho de banda.

4.3. Procesos de Simulación

1. Generación del mapa de carretera con MOVE. Inicialmente hay que definir las carreteras proporcionando las coordenadas de origen, destino, número de carriles, velocidad máxima de la vía, prioridad y extensión. También existen las opciones de importar mapas en el formato TIGER o crearlos de forma aleatoria, en cuadrícula o grid.
2. Creación de los flujos de vehículos con MOVE. Se definen los movimientos de tráfico, carretera inicial, carretera final, número de vehículos y tiempo durante el cual transcurren los movimientos. Existen opciones para especificar manualmente el movimiento de los vehículos proporcionando las carreteras por las que viaja, el tiempo, aceleraciones, deceleraciones, velocidades máximas, etc. También se pueden definir recorridos de autobuses y sus horarios habituales.
3. Simulación de tráfico con SUMO. Consiste en ejecutar la simulación de tráfico vial con el mapa y las condiciones de tráfico definidas anteriormente. El simulador SUMO mostrará de forma gráfica el comportamiento de los vehículos de modo que se podrá visualizar si corresponde con lo esperado. A la vez que se ejecuta la simulación se crea un archivo con las trazas de movimiento de los vehículos que sería utilizado por el simulador NS-2.
4. Simulación de red con NS-2. Usando el patrón de movimiento creado en el paso anterior se define conexiones entre los diferentes vehículos. Es necesario crear un script de NS-2 que defina las tecnologías radio a utilizar, el tipo de antena, el modelo de propagación, algoritmo de encaminamiento a simular, protocolo de transporte, etc. Con el script creado se ejecuta la simulación obteniendo dos trazas; una interpretable por el animador gráfico nam y otro con las trazas *wireless* que produce el simulador.

5. Análisis de los resultados. Gracias a *Tracegraph* que procesa las trazas y extrae datos significativos de la simulación se podrán comparar y analizar los resultados de las diferentes simulaciones. Al cabo de ese paso se pretende concluir sobre el carácter óptimo de un protocolo en cada escenario.



Figura 14. Proceso de Simulación



4.4. Instalación y ejecución de la Plataforma

Para poder poner en marcha y engranar todos los componentes de la plataforma de simulación, se instala el siguiente software en Linux Ubuntu 12.04.5:

- Java SDK 1.6. (Java SDK 1.6. 2014 – Recuperado de <http://java.sun.com>)
- Sumo versión: 0.12.3 (SUMO - Simulation of Urban Mobility. 2013 – Recuperado de <http://sumo.sourceforge.net/>)
 - Xerces (XML-parser) (Xerces (XML-parser). 2013 – Recuperado de <http://xerces.apache.org/xerces-c/index.html>)
 - FOX-Toolkit (GUI Toolkit) (FOX-Toolkit (GUI Toolkit). 2013 – Recuperado de <http://www.fox-toolkit.org/>)
 - PROJ (Cartographic Projections Library) (PROJ (Cartographic Projections Library). 2013 – Recuperado de <http://www.remotesensing.org/proj/>)
 - GDAL (Geospatial Data Abstraction Library) (GDAL (Geospatial Data Abstraction Library). 2013 – Recuperado de <http://www.remotesensing.org/gdal/>)
- NS-2 Versión: 2.34 (all-in-one) o posterior (The Network Simulator: Building Ns. 2013 – Recuperado de <http://www.isi.edu/nsnam/ns/ns-build.html>)

Nota: Refiérase a la website de cada aplicación para detalles de la instalación en cuanto a la versión su sistema operativo. Este documento solo muestra cómo instalar los programas para una específica versión. De cualquier modo, de acuerdo a las versiones de Xerces, FOX-Toolkt, PROJ y GDAL por favor revisar los requerimientos de SUMO.

En Ubuntu, se necesitan instalar una cantidad de paquetes extras para que SUMO y NS-2 puedan correr con normalidad.

Para Sumo

```
# sudo apt-get update
```

Para más detalles visite la wiki de SUMO

<http://sourceforge.net/apps/mediawiki/sumo/index.php?title=LinuxBuild>



Para NS-2

```
# sudo apt-get install build-essential
# sudo apt-get install tcl8.4 tcl8.4-dev tk8.4 tk8.4-dev
# sudo apt-get install libxmu-dev libxmu-headers
```

Instalación Xerces

Enlace de referencia: <http://xerces.apache.org/xerces-c/build-winunix.html>

Por ejemplo, si esta es su ruta de instalación = /home/temp/xerces-c-src_2_7_0

Lo que sigue es configurar su ruta raíz Xerces-C ++ de la siguiente manera:

```
export XERCESCROOT=/home/temp/xerces-c-src_2_7_0
```

Una de las formas más comunes para construir Xerces-C ++ es la siguiente:

```
cd src / xercesc
./runConfigure -plinux -cgcc -xg ++ -minmem -nsocket -tnative -rpthread
```

Ahora se construye xercesc

```
gmake
make install
```

Instalación FOX-Toolkit

FOX-Toolkit se puede configurar simplemente como:

```
./configure --with-opengl=yes --prefix=$HOME
make
make install
```



Instalación de PROJ y GDAL

Se configura de esta manera:

```
./configure --prefix=$HOME  
make  
make install
```

Instalación de SUMO

Enlace de referencia: <http://sumo.sourceforge.net/wiki/index.php/LinuxBuild>

Se configura simplemente como:

```
./configure --with-fox=$HOME --with-proj-gdal=$HOME --with-xerces=$HOME --  
prefix=$HOME  
make  
make install
```

Nota: Si tiene algún problema la versión gcc, se sugiere ingresar el comando:

```
CC=gcc34  
CXXFLAGS=-fpermissive ./configure --with-fox=$HOME -con-proy-gdal=$HOME -with-  
xerces=$H  
OME -prefix=$HOME
```

Puede utilizar la GUI SUMO para comprobar si la instalación fue correcta o no. Por ejemplo, en una ruta de instalación = /home/temp/sumo-0.9.8

```
/home/temp/sumo-0.9.8/src cd  
./sumo-guisim
```



Si la instalación es exitosa, verán una ventana como la figura 15.

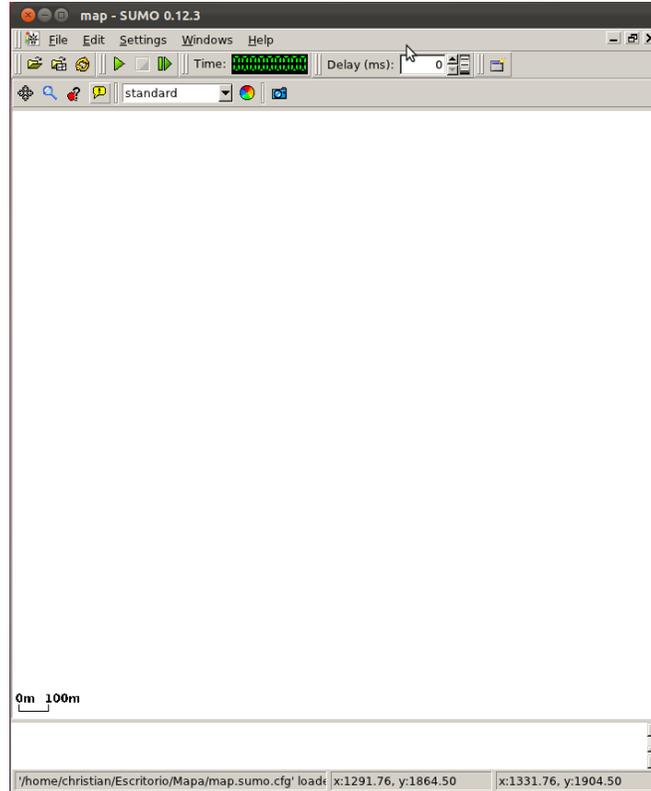


Figura 15. GUI de SUMO



Instalación de NS-2

Enlace de referencia: <http://www.isi.edu/nsnam/ns/ns-build.html#allinone>

Una vez descargado [ns-allinone-2.31.tar.gz](#)

```
tar -zxvf ns-allinone-2.31.tar.gz
cd ns-allinone-2.31
./install
```

Ahora se configuran las variables de entorno y editar el directorio de usuario en el archivo `.bashrc`. Además se adicionará una variable de entorno con `PATH`.

`LD.LIBRARY_PATH`, and `TCL_LIBRARY`

```
export NS_HOME='pwd'/ns-allinone-2.31
export PATH= $NS_HOME /tc18.4.14/unix: $NS_HOME /bin: $NS_HOME
/tk8.4.14/unix:$PATH
export LD_LIBRARY_PATH=$NS_HOME /otcl-1.13: $NS_HOME /lib:$LD_LIBRARY_PATH
export TCL_LIBRARY=$NS_HOME /tc18.4.14/library
```

También se sugiere editar el archivo `~/bash_profile` y evitar reestablecer las variables cada vez que se inicie el sistema.

Esto sería suficiente para empezar a utilizar el simulador y su animador, para comprobar que la instalación se ha realizado de forma correcta existe una serie de validaciones y demostradores accesible desde el repertorio principal a través del comando:

```
./validate
```



MOVE

Como MOVE es una aplicación Java se necesitará construir las fuentes simplemente compilando con "javac *.java". Para ejecutarlo "vanetsim java".

También, como en este caso, puede ejecutar MOVE archivo jar sin la construcción de las fuentes. Se ejecuta "java -jar MOVE.jar" como muestra la figura 16.

```
christian@puello: ~/Escritorio/Tesis
christian@puello:~/Escritorio/Tesis$ java -jar MOVE.jar
```

Figura 16. Sentencia inicio MOVE

Ejecutando la Plataforma

Generación de Movilidad: En esta parte MOVE generará el modelo de movilidad creado para SUMO. En primer lugar se selecciona “*Mobility Model*” en el menú principal y aparecerá la ventana de generador de modelos de movilidad para VANET (figura 17)

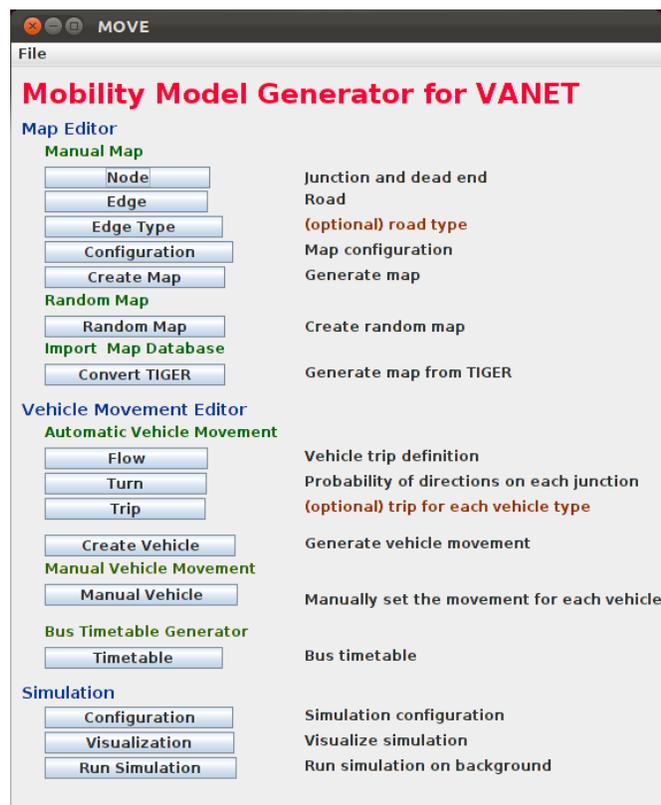


Figura 17. Generador de modelos de movilidad para VANET

Se necesita generar mapas que contengan esquinas e intersecciones (nodes) y calles o avenidas (edges) con sus respectivos canales. En la siguiente figura se muestra un ejemplo de la estructura de un mapa sencillo:

Creación de Nodos



Figura 19. Opción de creación de Nodos

Manualmente se crean los nodos basados en un mapa realista que comprende sectores de La Castellana, Altamira y Los Palos Grandes en la ciudad de Caracas, Venezuela. Urbanizaciones que contienen Rutas de la flota de transporte público TransChacao (TransChacao. 2013 – Recuperado de <http://es.wikipedia.org/wiki/TransChacao>) del Municipio Chacao.

Este mapa (figura 20) fue generado con la ayuda de herramientas como GoogleMaps (Google Maps. 2014 – Recuperado de <https://maps.google.com/>) y uso de coordenadas correlativas:

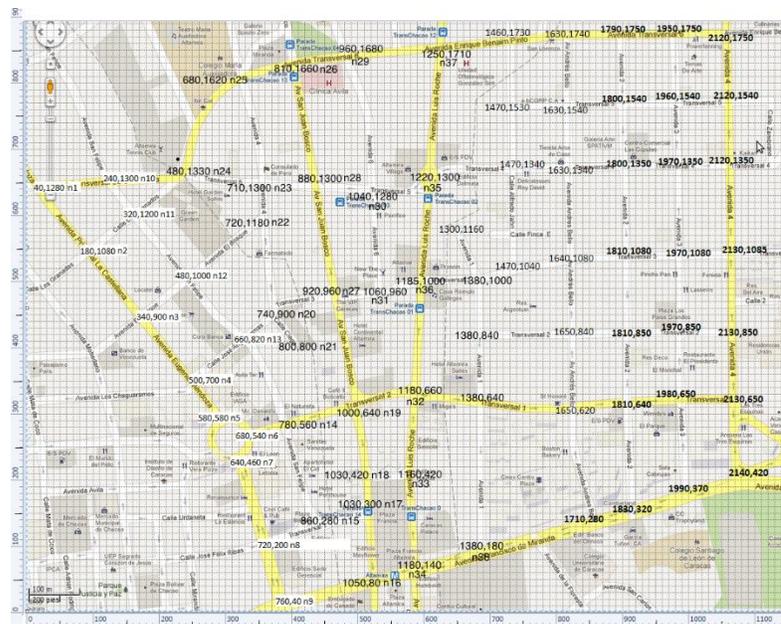


Figura 20. Mapa utilizado en las Simulaciones

Se insertan manualmente cada uno de los parámetros para cada nodo como un Identificador, posiciones (x,y) correlativas y si hay presencia de semáforo o no en ese nodo como se presenta en la figura 21:

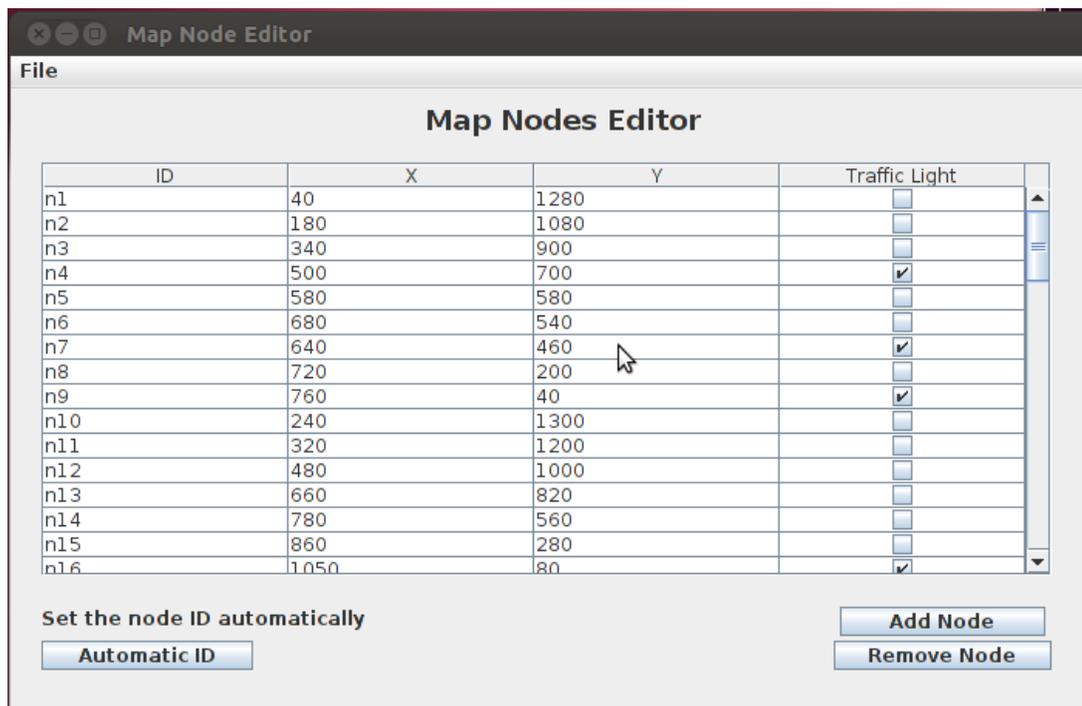


Figura 21. Creación de Nodos

Se selecciona la opción "file" y esto guardará un archivo llamado <name>.nod.xml. Y este archivo tendrá un aspecto como este ejemplo:

```
<nodes>
<node id="node1" x="100.0" y="100.0" type="traffic_light"/>
<node id="node2" x="400.0" y="100.0" type="priority"/>
<node id="node3" x="700.0" y="100.0" type="traffic_light"/>
<node id="node4" x="100.0" y="300.0" type="traffic_light"/>
<node id="node5" x="400.0" y="300.0" type="traffic_light"/>
<node id="node6" x="700.0" y="300.0" type="traffic_light"/>
<node id="node7" x="100.0" y="600.0" type="traffic_light"/>
<node id="node8" x="400.0" y="600.0" type="traffic_light"/>
<node id="node9" x="700.0" y="600.0" type="traffic_light"/>
<node id="node10" x="100.0" y="800.0" type="traffic_light"/>
<node id="node11" x="400.0" y="800.0" type="traffic_light"/>
<node id="node12" x="700.0" y="800.0" type="priority"/>
</nodes>
```

Creación de Vías

Al igual que los nodos, las carreteras, avenidas o vías en general se realiza manualmente en un editor con aspecto muy similar al de los nodos.

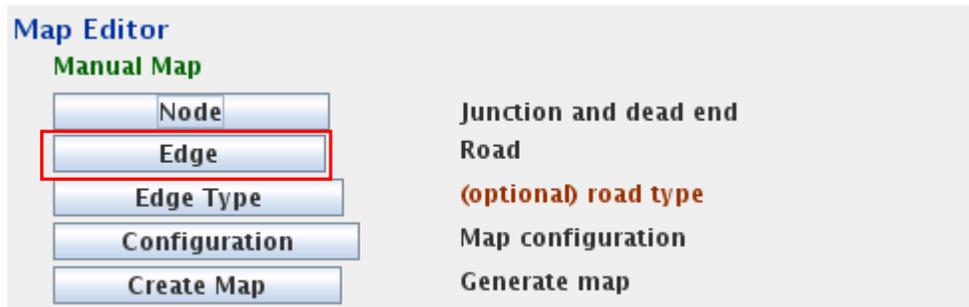


Figura 22. Opción Creación de Vías

Las vías crearán conexiones entre 2 nodos previamente definidos. Es aquí donde se especifican todos los parámetros de las vías tales como el identificador, nodo origen, nodo destino, Tipo, Nro. de Carriles, Velocidad y Prioridad (Véase figura 23)

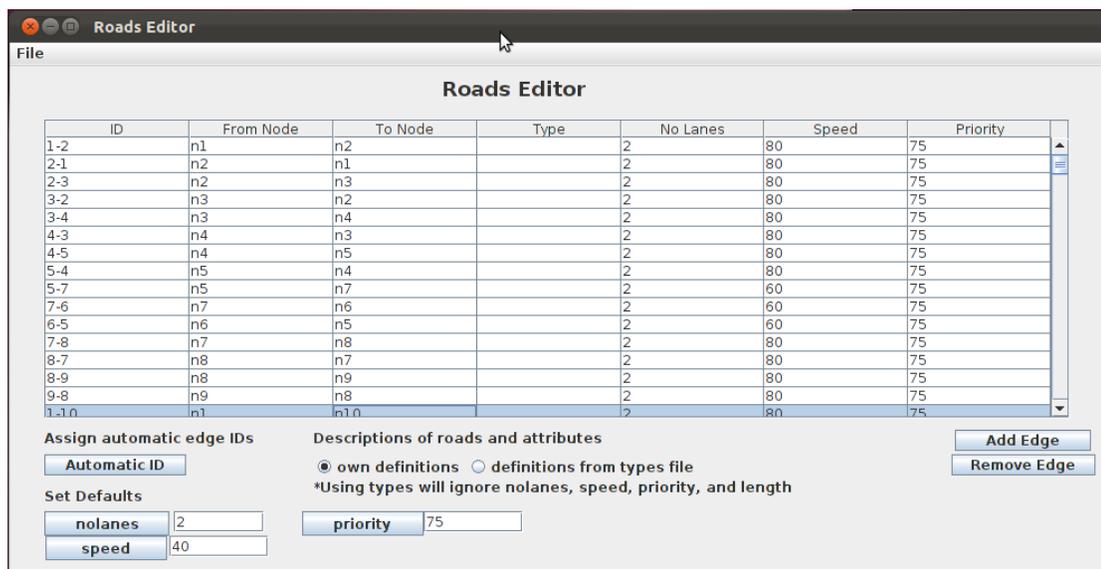


Figura 23. Creación de Vías



Al igual que con los nodos, al guardar las definiciones se generará un archivo .xml con información de las vías.

```
<edges>
<edge id="edgeR-0-0" fromnode="node1" tonode="node2" priority="75"
nolanes="2" speed="40" />
<edge id="edgeR-0-1" fromnode="node2" tonode="node3" priority="75"
nolanes="2" speed="40" />
<edge id="edgeL-0-0" fromnode="node2" tonode="node1" priority="75"
nolanes="2" speed="40" />
<edge id="edgeL-0-1" fromnode="node3" tonode="node2" priority="75"
nolanes="2" speed="40" />
.
.
.
<edge id="edgeR-3-0" fromnode="node10" tonode="node11" priority="75"
nolanes="2" speed="40" />
<edge id="edgeR-3-1" fromnode="node11" tonode="node10" priority="75"
nolanes="2" speed="40" />
<edge id="edgeL-3-0" fromnode="node11" tonode="node12" priority="75"
nolanes="2" speed="40" />
<edge id="edgeL-3-1" fromnode="node12" tonode="node11" priority="75"
nolanes="2" speed="40" />
</edges>
```

Configuración y Creación del Mapa

Una vez generados los archivos <name>.nod.xml y <name>.edg.xml se dispone a la configuración del mapa introduciendo estos archivos como parámetros.

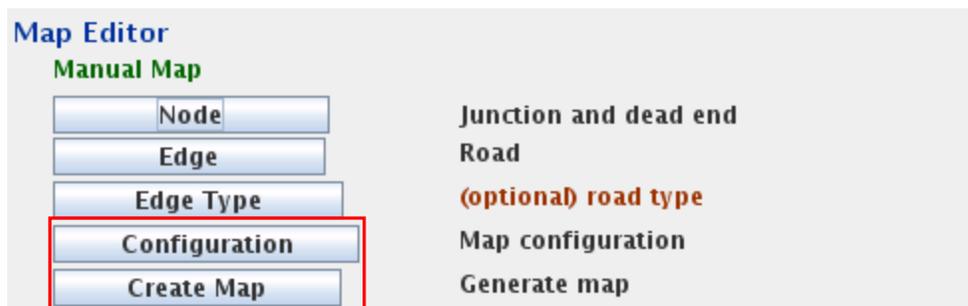


Figura 24. Opciones para generar el Mapa

Una vez configurado se introduce el archivo de configuración y el mapa está listo. En este caso, el mapa que se utiliza para la simulación posee 74 nodos y 182 vías con sus respectivas características realistas como semáforos, nro. de carriles, sentido de la vía, etc. Como se muestra en la figura 25.

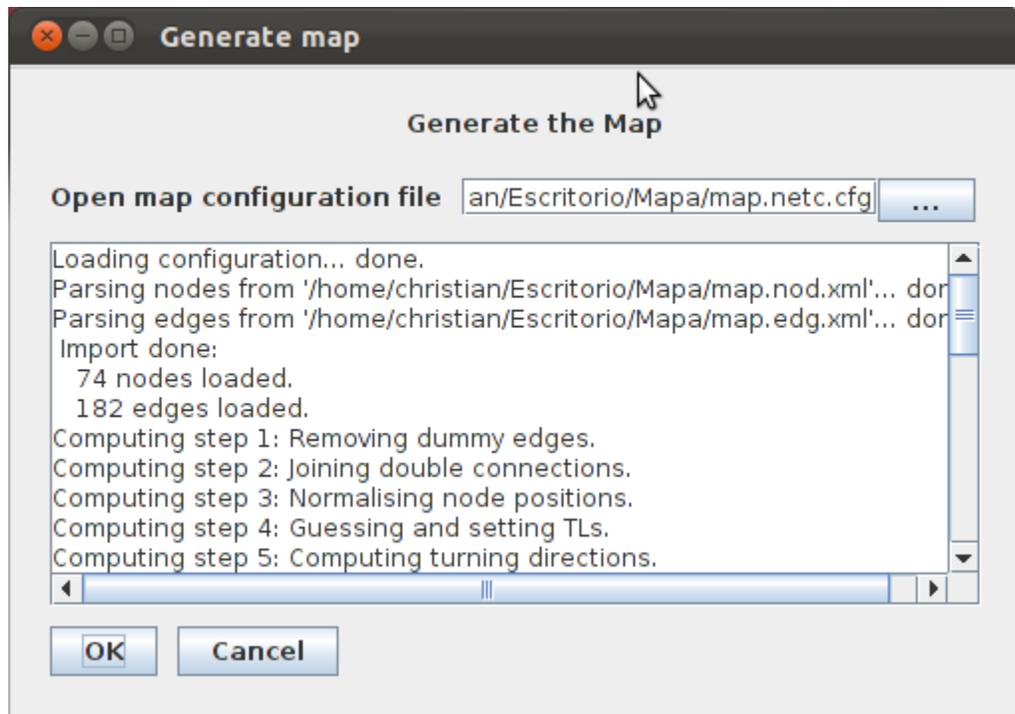


Figura 25. Generación del Mapa

Definición de Flujos

Los flujos indican al modelo de movilidad de cuáles nodos se originarán el o los flujos vehiculares que transitarán de manera aleatoria a través del mapa. Esto permitirá que luego de cierto instante de tiempo todo el mapa este inundado por vehículos. En este caso se colocan 3 generadores de flujo, uno desde lo que corresponde a La Castellana, el segundo Altamira y el último Los Palos Grandes como se muestra en la figura 27.

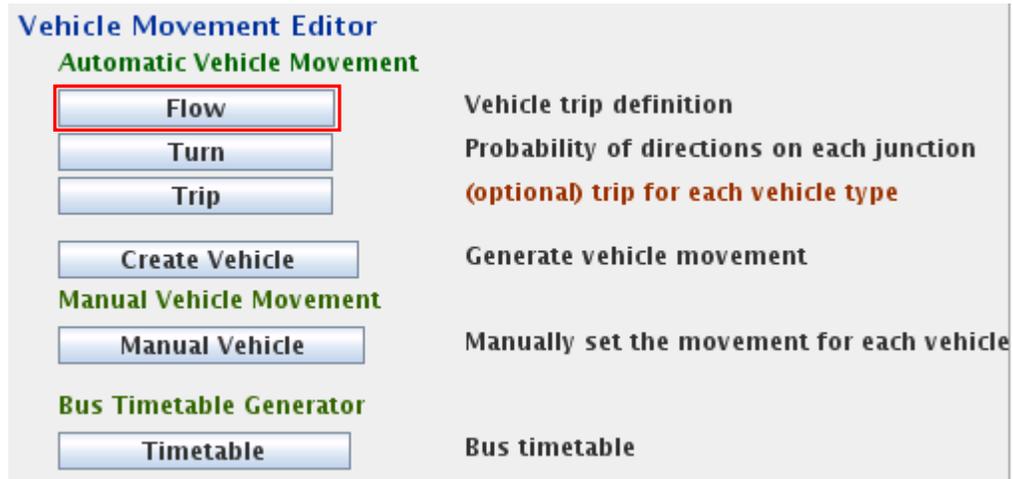


Figura 26. Opción para definir flujos vehiculares

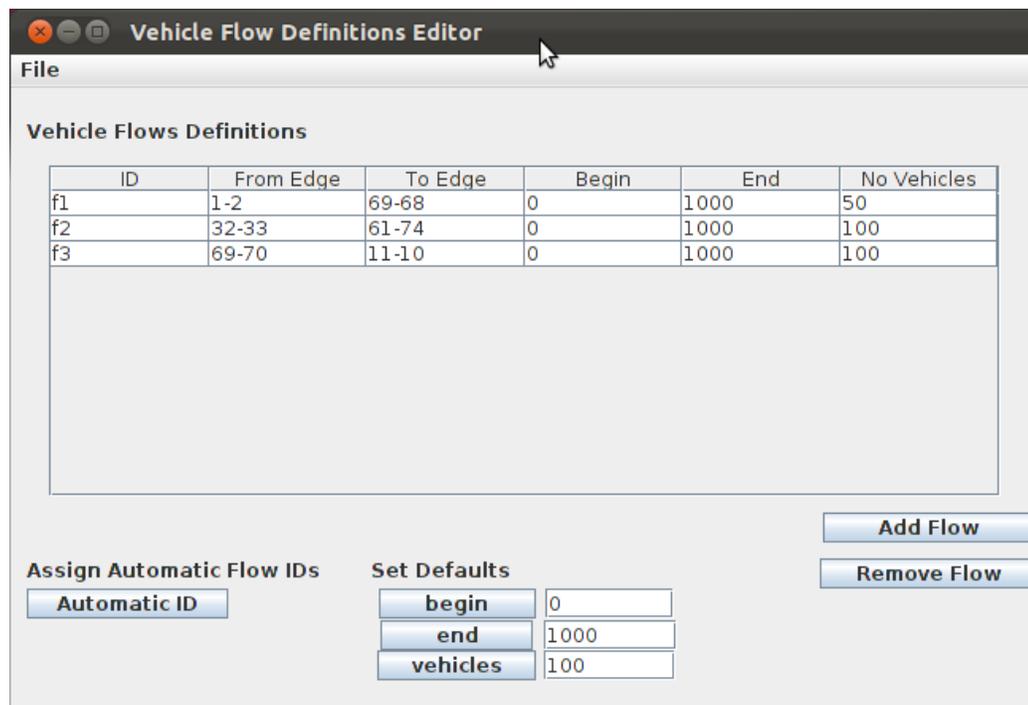


Figura 27. Configuración de flujos vehiculares

Una vez definidos todos los flujos se crean los vehículos sobre el mapa en la opción "Create Vehicle".



Configuración de la Simulación

Luego que el mapa y las configuraciones de movimiento están listas, se dispone a configurar en sí la simulación seleccionando “*Configuration*” en el botón de menú de MOVE.

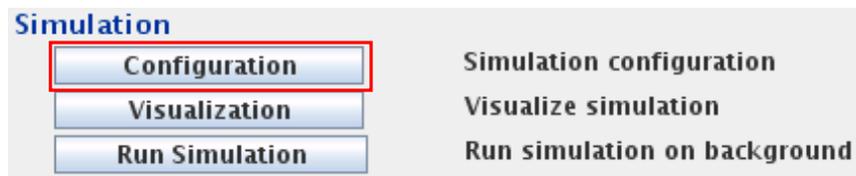


Figura 28. Opción para definir flujos vehiculares

Y si se desea visualizar el movimiento vehicular en el mapa se puede hacer en SUMO para detallar el modelo antes de comenzar a simular, esto permitirá realizar ajustes antes de migrar la información a NS-2. Véase las figuras 30 y 31.

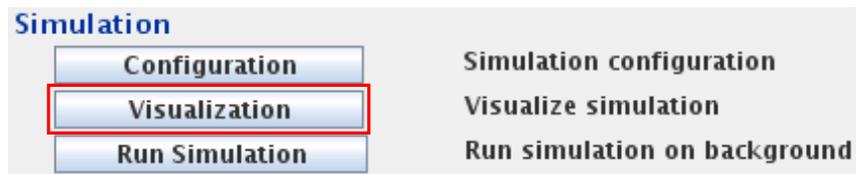


Figura 29. Opción para definir flujos vehiculares

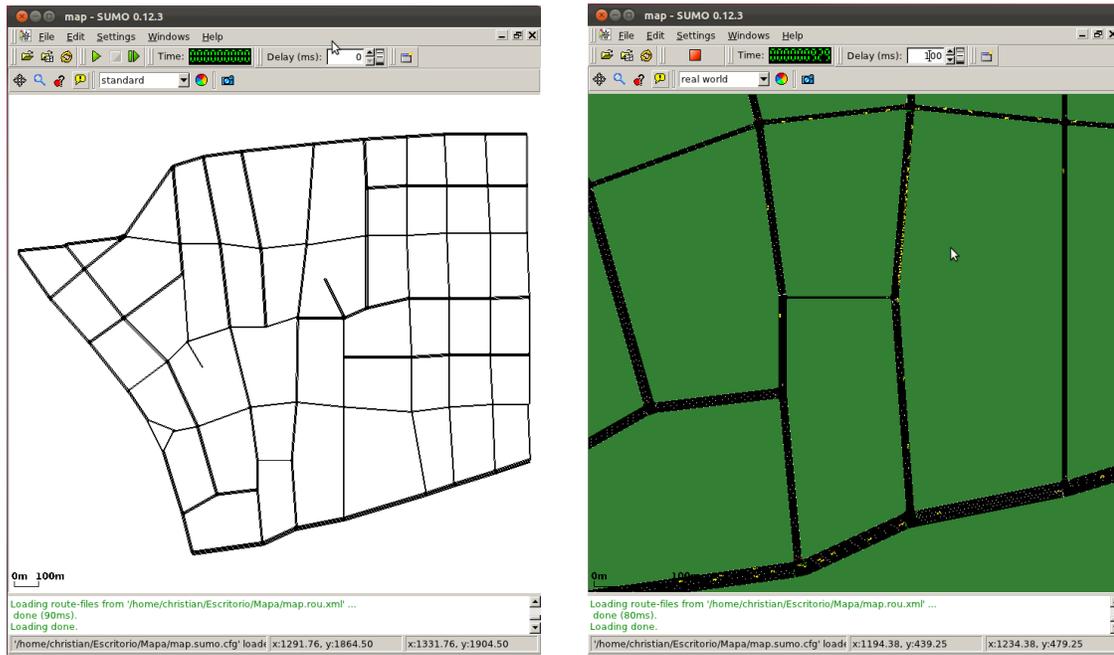


Figura 30. Visualización Mapa en SUMO



Figura 31. Visualización Mapa en SUMO



Ejecutando en NS-2

Para comenzar a correr la simulación se elige la opción “Traffic Model” en el menú principal de MOVE, trasladándonos este al *Traffic Model Generator for VANET*.

El generador de modelo de tráfico se compone de dos secciones principales: por NS-2 y QualNet. Pero como ya se definió para este estudio se usará NS-2.

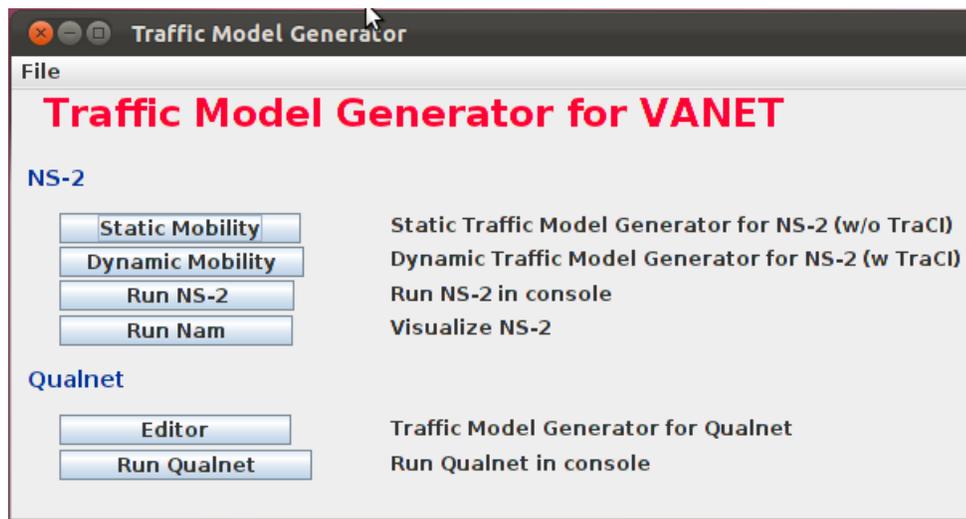


Figura 32. Menú del Generador de Modelo de Tráfico

Se selecciona la opción “Static Mobility” como se muestra en la figura 33, para poder configurar los aspectos de comunicación inalámbrica importantes que se requieren para la simulación, tales como el tipo de protocolo de enrutamiento, tipo de comunicación, tamaños de los paquetes, tiempo de conexión, etc. En este punto se debe cargar la traza generada para SUMO que contiene todos los aspectos de movilidad vehicular y mapa. Después que la carga se realice con éxito, se especifican las opciones del archivo *tcl* para la simulación en NS-2, como se muestra en la figura 33 a continuación.

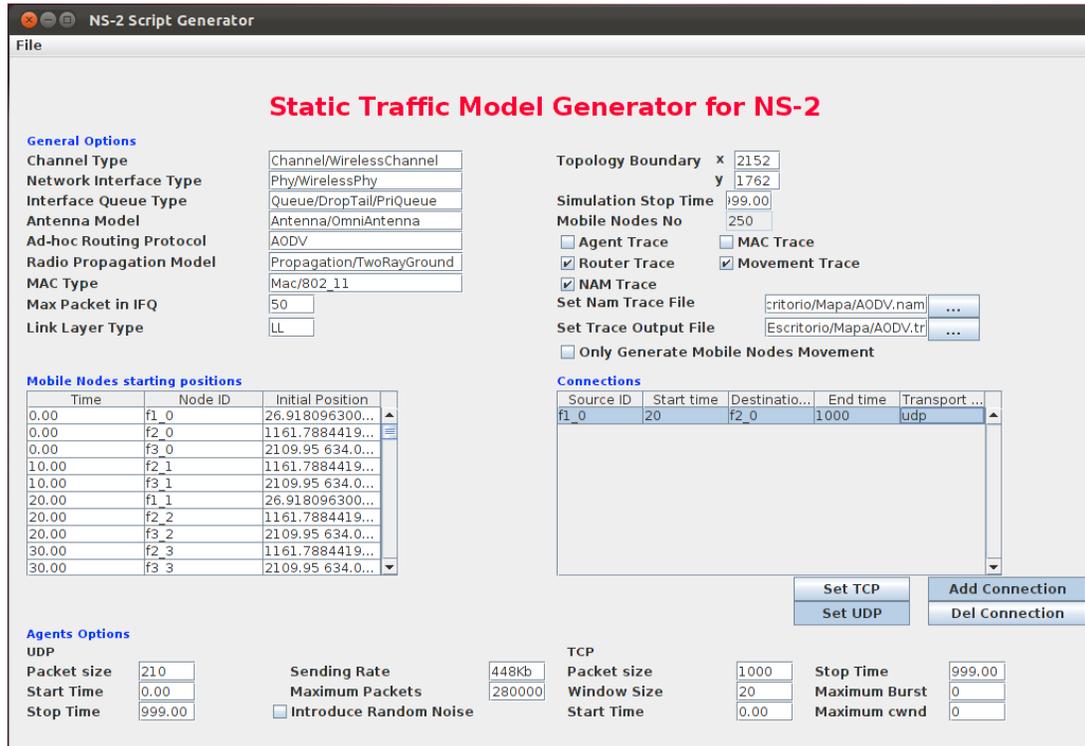


Figura 33. Configuración de Tcl para NS-2

Es aquí donde también se especificará la creación de las trazas tanto para visualizar la simulación final en *nam*, como para el posterior análisis en *Tracegraph*. Véase la figura 34.

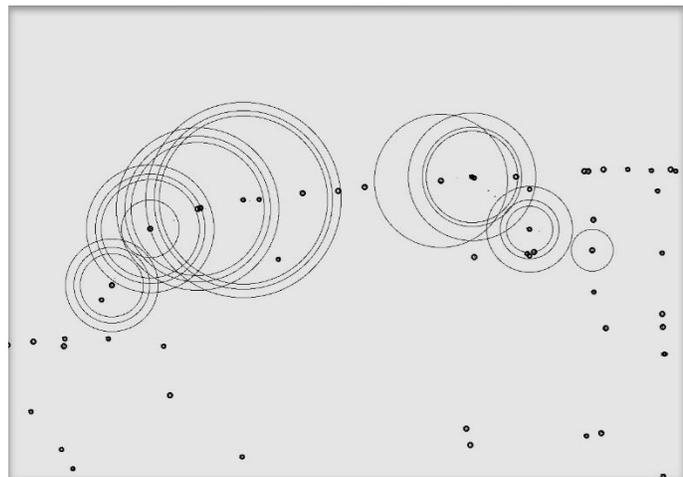
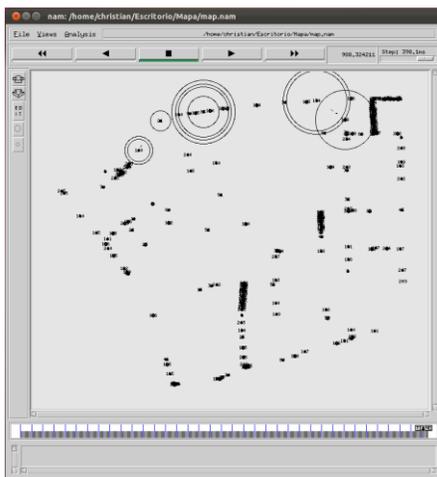


Figura 34. Nam mostrando las simulaciones en NS-2



5. ANÁLISIS DE RESULTADOS

Todas las simulaciones se han realizado según los escenarios y locaciones descritas anteriormente. Se observaron los diferentes protocolos *unicast* y sus comportamientos para cada uno de los escenarios con variaciones de densidades de tráfico (alta, media y baja) con los protocolos de comunicación TCP y UDP.

En las simulaciones se definieron dos escenarios bases: comunicación VANET pura y comunicaciones VANET a través de nodos intermedios.

En este estudio se simularán diferentes protocolos ad-hoc *unicast*:

- AODV
- DSR
- DSDV
- FSR
- OLSR

Las características de cada protocolo ya han sido descritas, pero es conveniente recordar en este punto a que grupo pertenece cada uno de ellos. AODV y DSR son protocolos reactivos mientras que los demás son proactivos.

La simulación consiste en que para cada topología se eligen dos nodos dentro de todos los nodos presentes y se creará una conexión entre ellos, primero UDP y luego TCP. Para así poder observar el comportamiento de dichos protocolos a la hora de transmitir.

Se ha elegido una aplicación UDP CBR (*Constant Bit Rate*) ya que se aproxima bien a las aplicaciones UDP que envían tráfico a una velocidad constante y sin preocuparse de si los paquetes han sido entregados o no. Para simular una conexión TCP se ha elegido una aplicación FTP.

Las características de las conexiones a definir en el archivo *tcl* del simulador de red NS-2 son las siguientes:

- Para FTP sobre TCP: Se envían paquetes de 1000 bytes. El algoritmo de control de congestión es el TAHOE TCP con una ventana de 20 paquetes.
- Para CBR sobre UDP: Se envían paquetes de 210 bytes a una velocidad de 448 kbps a un intervalo de 3.75 ms. Se podrá enviar un máximo límite de 218453 paquetes.



En las simulaciones de estas conexiones todos los protocolos serán comparados entre sí, en cuanto al rendimiento de cada uno de ellos en términos de porcentajes de éxito (*pdfr*), retardos extremo a extremo medido en segundos y la sobrecarga u *overhead* de paquetes introducidos por el protocolo en la comunicación. Además para las conexiones TCP se medirá en *throughput*. Se presentarán gráficas del promedio de los resultados obtenidos para una aproximación visual del rendimiento de cada protocolo en sus diferentes métricas.

Los resultados de estas simulaciones son un promedio de repetidas simulaciones por cada escenario, variación de tráfico y cada protocolo de enrutamiento. Es sumamente importante asegurarse de que los todos nodos que se eligieron para la comunicación extremo a extremo tengan las mismas posiciones relativas en los distintos escenarios con diferentes flujos de tráfico y diferentes protocolos. En efecto, hay que estudiar con detenimiento el escenario antes de llevar a cabo el trabajo. Esta condición es imprescindible para que los resultados de la simulación sean viables. Si se escogiesen los nodos al azar el resultado también será aleatorio: por ejemplo puede que los vehículos estén siempre tan distantes que no se vean y no pueden llegar a transmitir ningún paquete efectivo de datos, o que al revés, los nodos estén tan cerca que la conexión sea siempre directa y que el *pdfr* sea de 100%.

Se realizan pruebas previas a las simulaciones para determinar los nodos a considerar. Gracias al animador *nam* se puede observar los esquemas de movilidad. Se planificó que los nodos 0 y 1 sean los primeros en llegar por cada carril y que al mantener el mismo esquema de velocidades y ruta en todos los escenarios, mantienen sus posiciones relativamente constante siendo el número de vehículos entre ellos la única variación.



5.1. Resultado y Análisis de Métricas Circuito Urbano UDP

Indiferentemente de las comunicaciones TCP o UDP, las conexiones se planificaron para producirse desde el instante 20 hasta el final de la simulación en el instante 1000 entre el nodo 0 y el nodo 1.

Nivel de Tráfico Alto

Protocolo	Paq. Enviados	% Entrega	Retardo	Overhead (Paq.)
DSDV	211334	19,89	0,2787	805
AODV	211334	39,14	0,5093	418
DSR	211334	39,36	0,3818	1047
OLSR	211334	24,65	0,2833	4731
FSR	211334	13,44	0,0538	1170

Cuadro 4: UDP Urbano tráfico Alto

Nivel de Tráfico Medio

Protocolo	Paq. Enviados	% Entrega	Retardo	Overhead (Paq.)
DSDV	211334	22,48	0,1612	1738
AODV	211334	43,84	0,3269	396
DSR	211334	42,62	0,36	124
OLSR	211334	26,97	0,3158	9103
FSR	211334	12,74	0,0786	2244

Cuadro 5: UDP Urbano tráfico Medio

Nivel de Tráfico Bajo

Protocolo	Paq. Enviados	% Entrega	Retardo	Overhead (Paq.)
DSDV	211334	13	0,0042	1078
AODV	211334	30,41	0,2128	199
DSR	211334	32,29	0,2706	223
OLSR	211334	21,33	0,1664	6132
FSR	211334	12,89	0,0295	1716

Cuadro 6: UDP Urbano tráfico Bajo

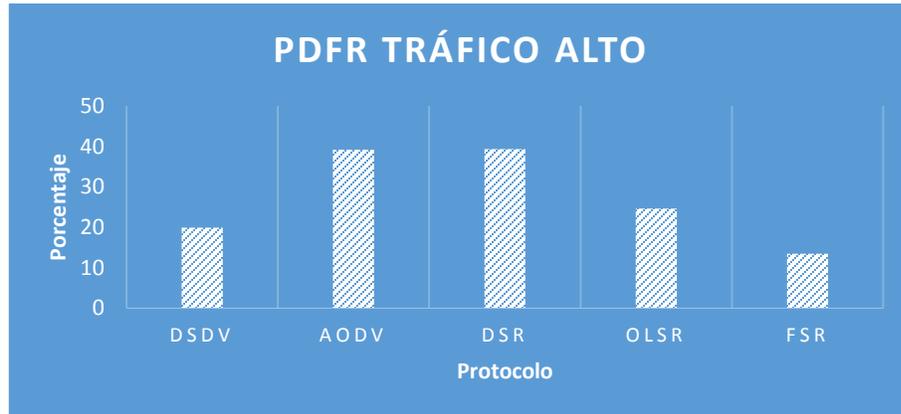


Gráfico 1.1: Porcentaje de éxito UDP en tráfico Alto Urbano

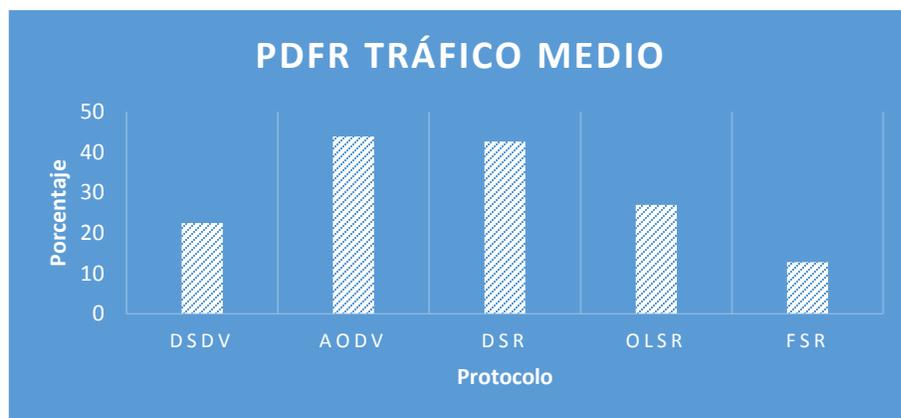


Gráfico 1.2: Porcentaje de éxito UDP en tráfico Medio Urbano

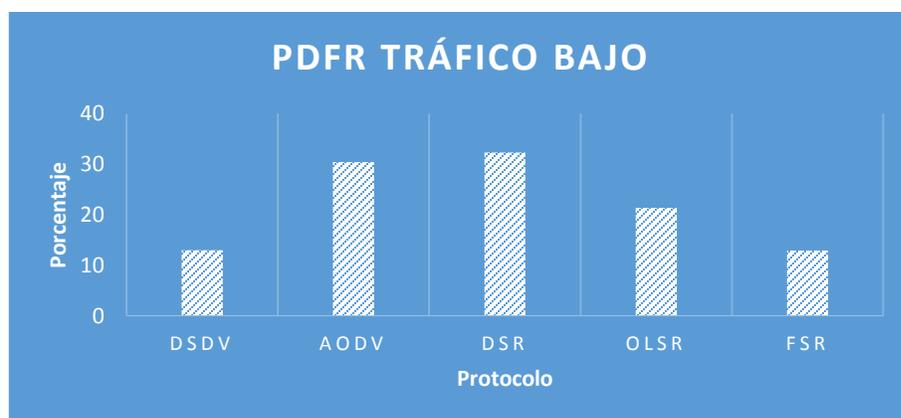


Gráfico 1.3: Porcentaje de éxito UDP en tráfico Bajo Urbano



Gráfico 2.1: Retardo UDP en tráfico Alto Urbano

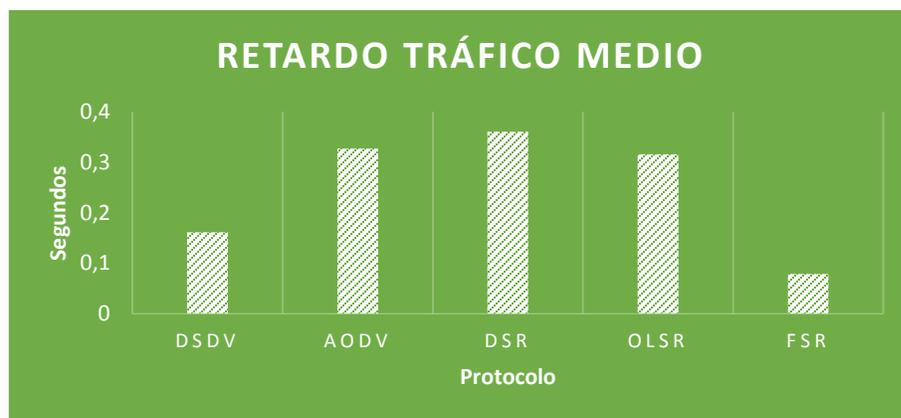


Gráfico 2.2: Retardo UDP en tráfico Medio Urbano



Gráfico 2.3: Retardo UDP en tráfico Bajo Urbano

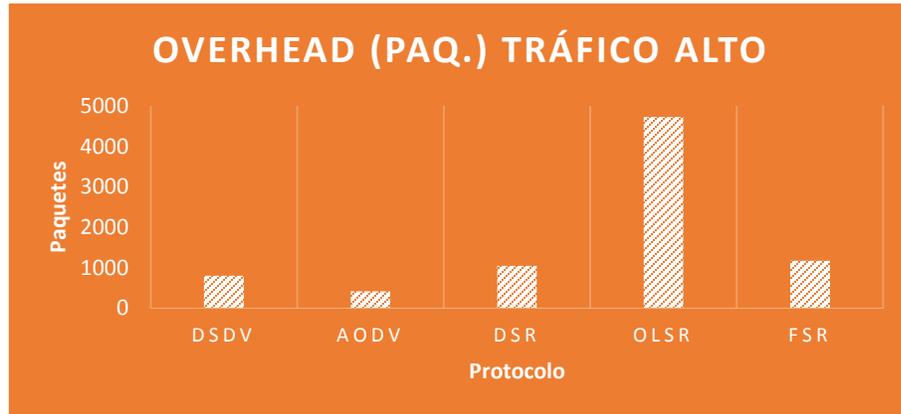


Gráfico 3.1: Sobrecarga UDP en tráfico Alto Urbano

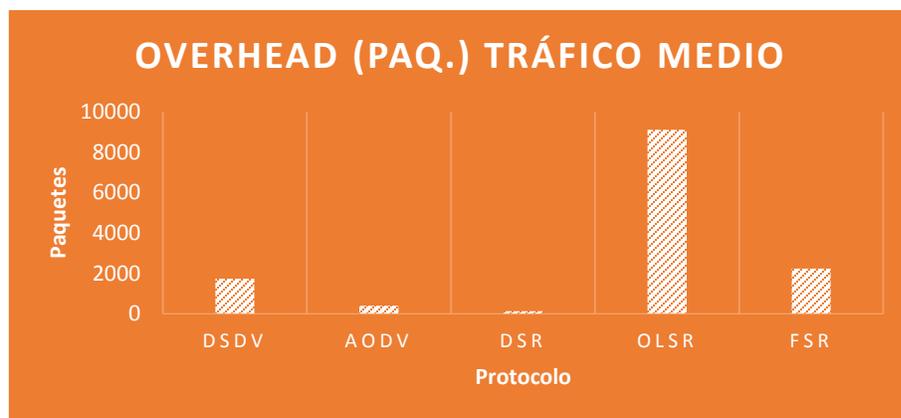


Gráfico 3.2: Sobrecarga UDP en tráfico Medio Urbano

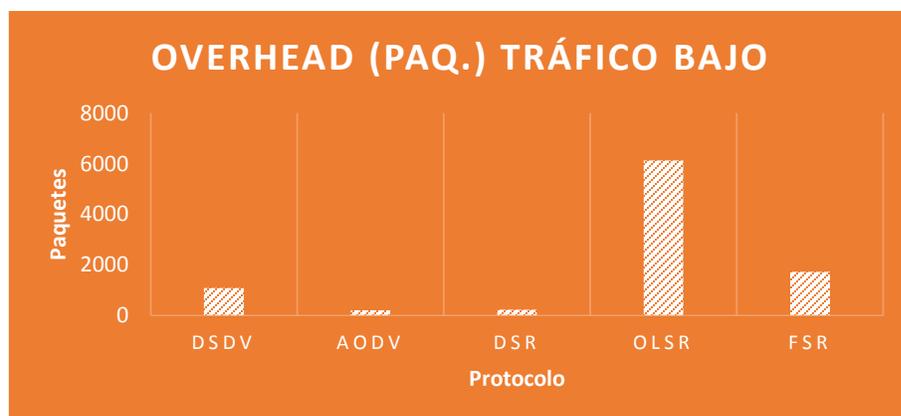


Gráfico 3.3: Sobrecarga UDP en tráfico Bajo Urbano



En este primer escenario considerado se puede observar que se obtiene un óptimo comportamiento general en la densidad de tráfico medio. Y en efecto, es dentro de este escenario que el *pdfr* es mayor para todos los protocolos. También se observa que los protocolos reactivos presentan mayor porcentaje de éxito, menores retardos y *overhead*. Esto coincide con lo previsto ya que en las VANET, donde los cambios de topologías son muy frecuentes, los protocolos proactivos se quedan rápidamente obsoletos. Véase más en detalle las conclusiones de cada conjunto de gráficas:

- **Porcentaje de Entregas (*pdfr*):** Se nota una muy clara ventaja de los protocolos reactivos (DSR seguido muy de cerca por AODV) contra los protocolos proactivos. Los protocolos reactivos presentan mejores tasas de entregas de paquetes. Se puede notar que para el protocolo FSR las tasas de entregas son casi constantes, no aumentan al aumentar la densidad de coches. Eso sugiere que FSR no reacciona bien ante los cambios de topología en cuanto al número de nodos y no es capaz de optimizar las rutas con los nuevos nodos que aparecen entre el emisor y el receptor. Por otro lado, en situaciones de poco tráfico el *pdfr* disminuye para todos los protocolos en general. Eso es así porque al haber menos cantidad de nodos entre emisor y receptor se pierde la comunicación cuando los nodos se encuentran a una distancia considerable entre ellos.
- **Retardo:** Lo primero que se puede constatar es que con densidad de tráfico bajo, los retardos disminuyen de forma significativa para todos los protocolos. También es cierto que menos paquetes llegan al destino con densidad de tráfico bajo, pero los que lo hacen presentan menos retardos ya que pasan por menos nodos intermedios, que retrasan recalculando de ruta antes de alcanzar el destino. Está característica afecta a los protocolos reactivos, que aunque presentan *pdfr* mejores, introducen retardos mayores. Eso se explica por el hecho de que consiguen rutas actualizadas y más frescas pero que el mecanismo de descubrimiento de rutas que se lleva a cabo cada vez que se tiene que mandar un paquete introduce retardos de transmisiones. Dentro de los protocolos reactivos, AODV tiene retardos menores que DSR salvo en la densidad de tráfico alto.
- **Overhead:** En las gráficas de sobrecarga, llama la atención el comportamiento de OLSR que presenta un *overhead* muy por encima de los demás protocolos. OLSR es un protocolo proactivo cuyo algoritmo de enrutamiento siempre procura optimizar y reducir los tamaños de sus tablas de enrutamiento. Esta supuesta optimización no es aplicable en topologías rápidamente cambiantes como las VANET, explicando esto sus altos tiempos de retardo. Por lo tanto se puede concluir que su uso no es recomendable en esos entornos. Los protocolos reactivos cargan menos la red.

5.2. Resultado y Análisis de Métricas Circuito Urbano TCP

En TCP, una métrica importante que surge es el *throughput*, que mide el número de paquetes que ha sido capaz de enviar el nodo origen. Este indicador es quizás el más relevante ya que indica que el protocolo ha sido capaz de enviar mayor cantidad de información y el resto de los indicadores deberán ser estudiados teniendo muy en cuenta este dato.

Nivel de Tráfico Alto

Protocolo	Paq. Enviados	% Entrega	Retardo	Overhaed (Paq.)
DSDV	903	98,11	0,1395	796
AODV	2704	97,56	0,21051	1239
DSR	912	98,79	0,122	41
OLSR	1713	96,15	0,1801	4666
FSR	860	98,95	0,1196	1170

Cuadro 7: TCP Urbano tráfico Alto

Nivel de Tráfico Medio

Protocolo	Paq. Enviados	% Entrega	Retardo	Overhaed (Paq.)
DSDV	863	97,91	0,1308	1785
AODV	2824	97,38	0,2381	570
DSR	2181	98,71	0,2818	299
OLSR	1547	93,73	0,1371	9353
FSR	850	96,71	0,0832	2244

Cuadro 8: TCP Urbano tráfico Medio

Nivel de Tráfico Bajo

Protocolo	Paq. Enviados	% Entrega	Retardo	Overhaed (Paq.)
DSDV	856	98,83	0,1885	1117
AODV	1865	98,18	0,1753	398
DSR	1262	99,84	0,2305	630
OLSR	1301	96,77	0,1519	6084
FSR	847	98,35	0,0961	1716

Cuadro 9: TCP Urbano tráfico Bajo

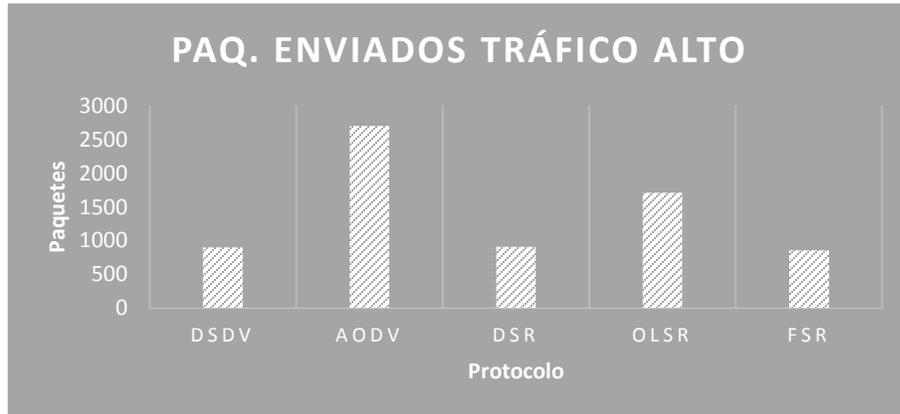


Gráfico 4.1: *throughput* TCP en tráfico Alto Urbano

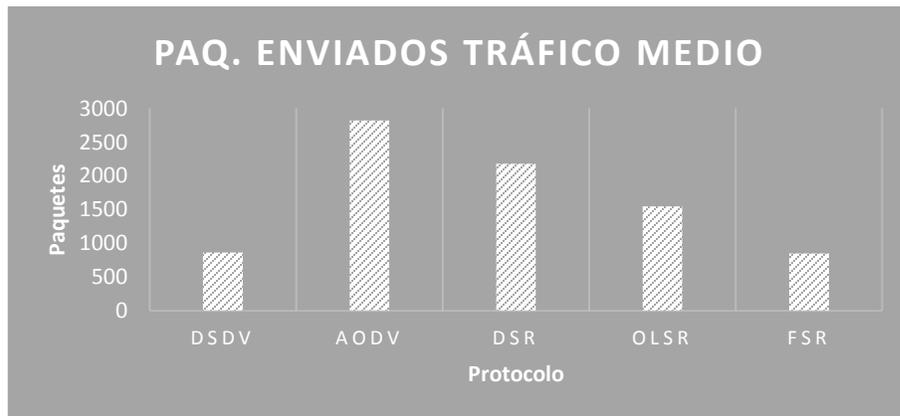


Gráfico 4.2: *throughput* TCP en tráfico Medio Urbano

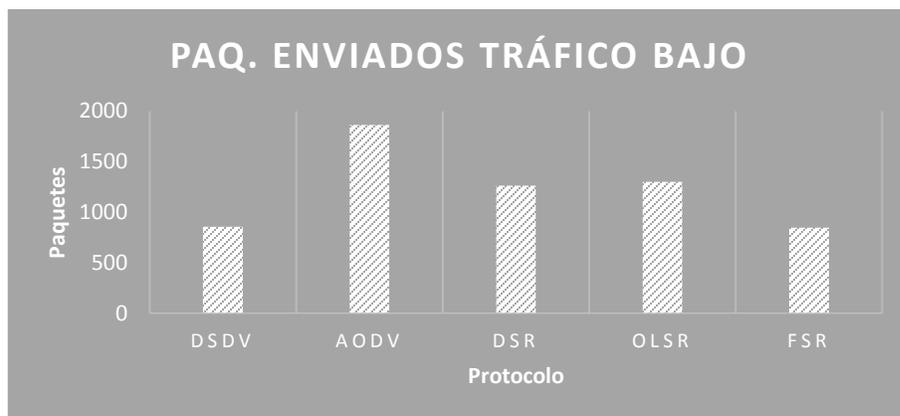


Gráfico 4.3: *throughput* TCP en tráfico Bajo Urbano

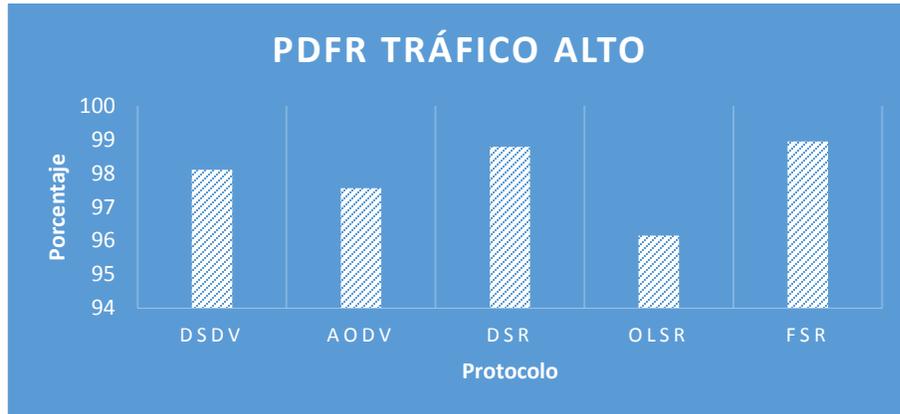


Gráfico 5.1: Porcentaje de entrega TCP en tráfico Alto Urbano

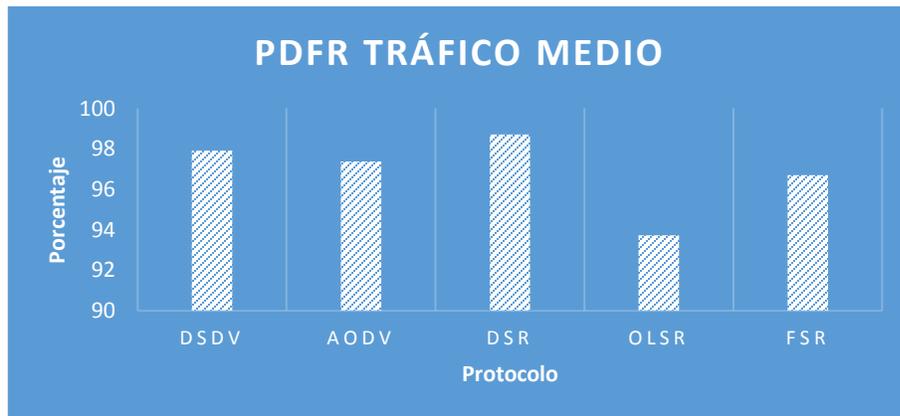


Gráfico 5.2: Porcentaje de entrega TCP en tráfico Medio Urbano

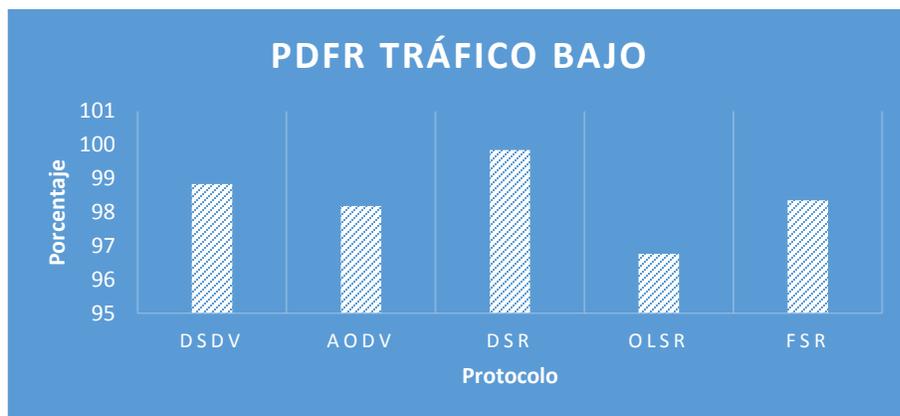


Gráfico 5.3: Porcentaje de entrega TCP en tráfico Bajo Urbano



Gráfico 6.1: Retardo TCP en tráfico Alto Urbano

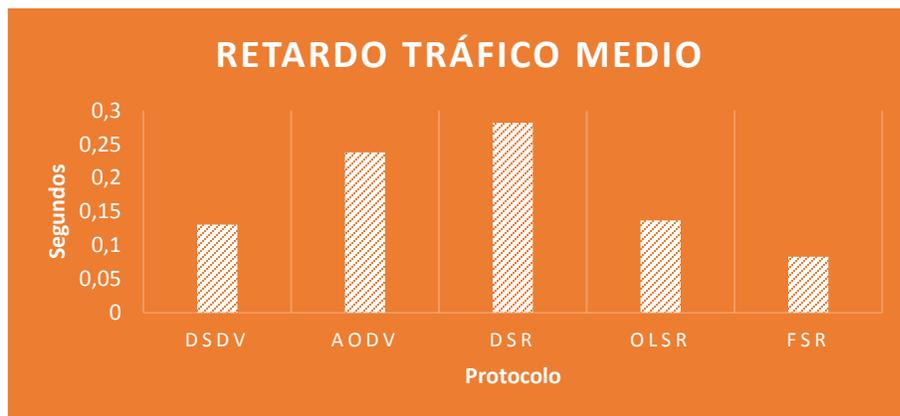


Gráfico 6.2: Retardo TCP en tráfico Medio Urbano

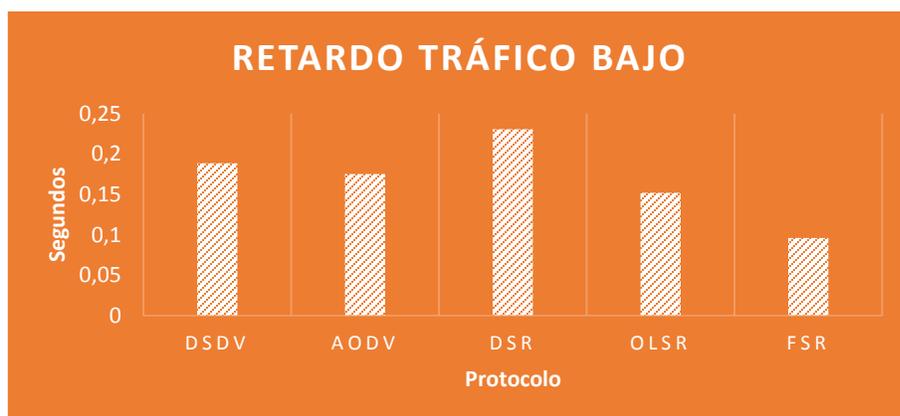


Gráfico 6.3: Retardo TCP en tráfico Bajo Urbano

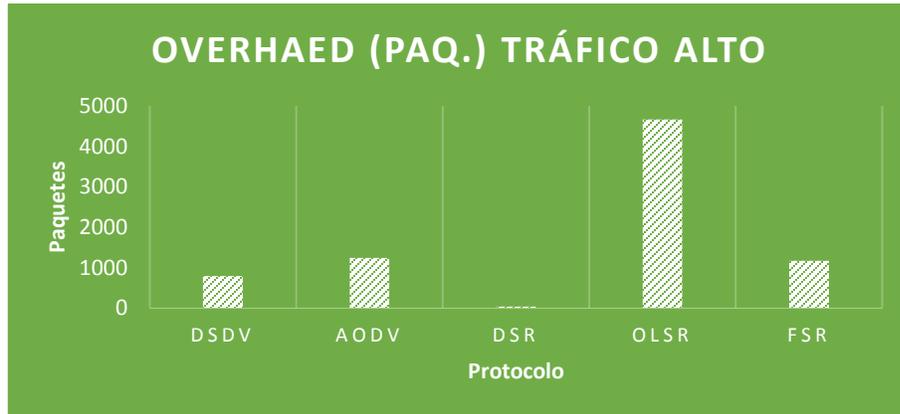


Gráfico 7.1: Sobrecarga TCP en tráfico Alto Urbano

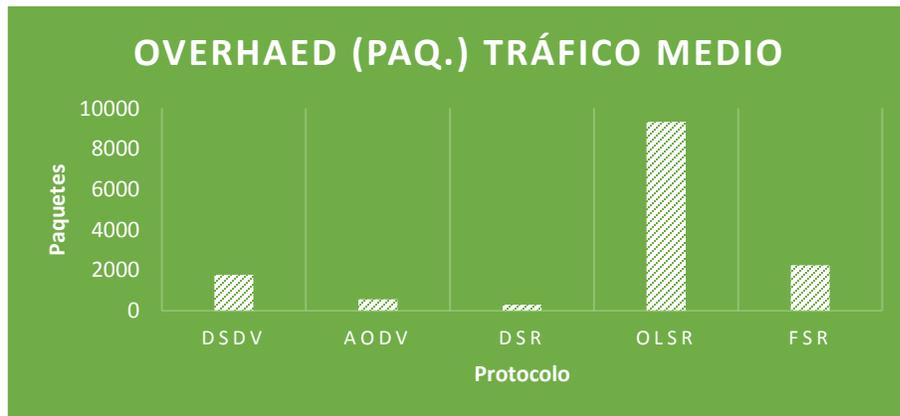


Gráfico 7.2: Sobrecarga TCP en tráfico Medio Urbano

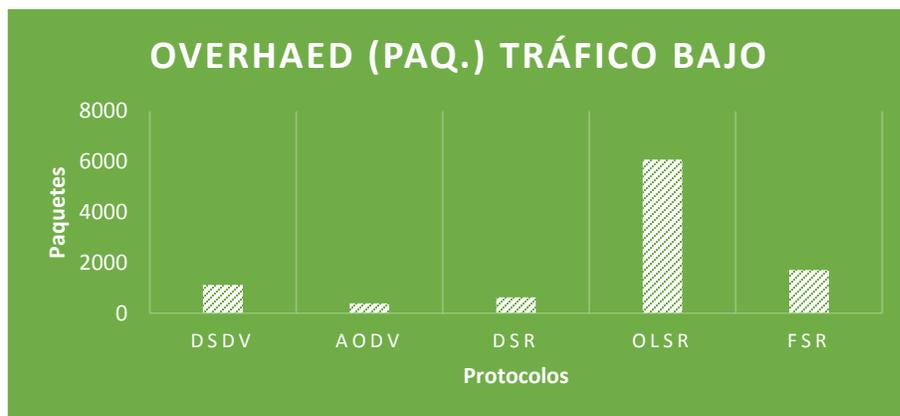


Gráfico 7.3: Sobrecarga TCP en tráfico Bajo Urbano



Ahora bien, para este caso de comunicación TCP para el circuito urbano definido aparece un máximo de rendimiento general con la densidad de tráfico medio, al igual que en UDP, con el que mayor densidad de información se consigue mandar. Una vez más y de forma general, los protocolos reactivos presentan mejores prestaciones, con la excepción de FSR con densidad alta de nodos que presenta un comportamiento un poco parecido al de los reactivos. Se detallan cada conjunto de gráficas:

- **Throughput:** Se puede observar unos números interesantes de AODV que consigue mandar mucha más información que DSR, sobre todo con densidades altas y por supuesto que los protocolos proactivos. Se vuelve a constatar lo concluido en el caso de comunicaciones con UDP y que nuevamente concuerda con lo que se esperaba con la teoría: los protocolos proactivos reaccionan mal frente a cambios de topología de la VANET ya que no son capaces de actualizar las rutas lo suficientemente rápido para que al querer mandar la información dicha ruta aun esté vigente.
- **Porcentaje de Entregas (*pdf*):** Se puede decir que el protocolo más óptimo en este rubro resulta ser DSR, seguido de DSDV, AODV, FSR y OLSR respectivamente. Eso significa que aunque AODV consiga mantener un muy elevado *throughput* en comparación, su porcentaje de éxito en entrega de paquetes es menor, pierde más paquetes que el protocolo proactivo DSDV. Por otra parte OLSR vuelve a presentar prestaciones muy inferiores al resto de los protocolos.
- **Retardos:** Los protocolos que consiguen mandar mayor cantidad de información son los que presentan retardos mayores, salvo en el caso de AODV cuyos retardos extremo a extremo son menores que los de DSR en promedio. Los protocolos proactivos aunque consiguen un *throughput* significativamente bajo en comparación a los reactivos, sus tiempos de retardos son en promedio la mitad de los tiempos de los protocolos reactivos. Pero se observa en AODV valores entre 0,2 y 0,25 segundos de media, un retardo aceptable dado la cantidad de información que este manda a través de la VANET.
- **Overhead:** OLSR vuelve a destacar por su excesiva sobrecarga. Los protocolos reactivos siguen presentando *overhead* menores en promedio y en particular en el caso de DSR en densidades de tráfico alto. Es el indiscutible ganador. Pero también se nota algo interesante al observar un comportamiento proporcional a la densidad del tráfico por parte de AODV, que con tráfico alto presenta sus valores de sobrecarga más elevados. En tráfico medio reduce su sobrecarga de paquetes casi a la mitad y en tráfico bajo nuevamente reduce significativamente sus números teniendo inclusive menor sobrecarga que DSR. Y que viceversa es el caso de DSR, cuyo comportamiento en cuanto a la sobrecarga es inversamente proporcional a la densidad de tráfico. Esto puede ocurrir debido a que las tramas de exploración que envía DSR para generar las rutas, no consiguen respuesta dado la baja densidad de



nodos, generando a su vez un aumento en el retardo y es por esta razón que AODV es el vencedor frente a DSR cuando en tiempo de retardo se habla.

5.3. Resultado y Análisis de Métricas Circuito Autopista UDP

En esta sección del estudio se pretende comparar los protocolos ad-hoc con diferentes densidades de tráfico en la autopista variando la velocidad y número de nodos para cada caso, para ello se vuelven a establecer las comparativas de conexión FTP sobre TCP y otra CBR sobre UDP respectivamente.

Nivel de Tráfico Alto

Protocolo	Paq. Enviados	% Entrega	Retardo	Overhead (Paq.)
DSDV	211334	24,16	0,0251	2876
AODV	211334	32,83	0,1183	1751
DSR	211334	29,02	0,1064	2281
OLSR	211334	22,77	0,1995	10318
FSR	211334	22,07	0,113	2604

Cuadro 10: UDP Autopista tráfico Alto

Nivel de Tráfico Medio

Protocolo	Paq. Enviados	% Entrega	Retardo	Overhead (Paq.)
DSDV	211334	18,46	0,1166	3309
AODV	211334	25,27	0,17	3324
DSR	211334	28,38	0,4753	5794
OLSR	211334	16,03	0,2277	10926
FSR	211334	15,32	0,1243	2604

Cuadro 11: UDP Autopista tráfico Medio

Nivel de Tráfico Bajo

Protocolo	Paq. Enviados	% Entrega	Retardo	Overhead (Paq.)
DSDV	211334	23,91	0,0692	2096
AODV	211334	48,12	0,3033	1438
DSR	211334	44,98	0,3772	3712
OLSR	211334	31,75	0,2277	9054
FSR	211334	21,5	0,0957	1860

Cuadro 12: UDP Autopista tráfico Bajo

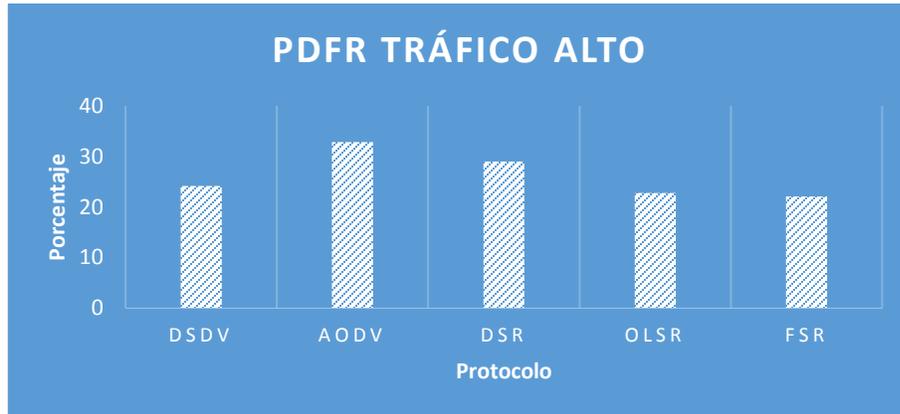


Gráfico 8.1: Porcentaje de Entrega UDP en tráfico Alto Autopista

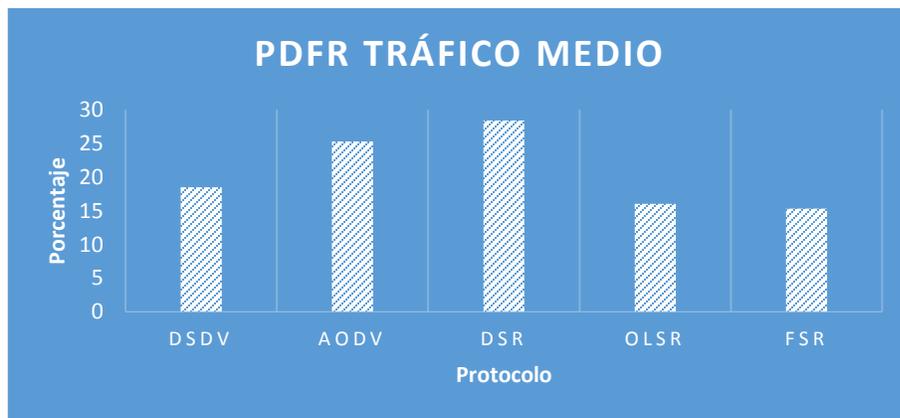


Gráfico 8.2: Porcentaje de Entrega UDP en tráfico Medio Autopista

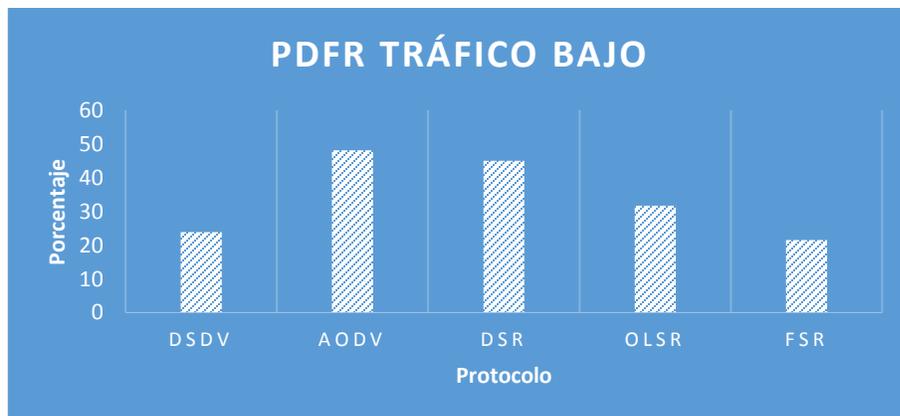


Gráfico 8.2: Porcentaje de Entrega UDP en tráfico Bajo Autopista



Gráfico 9.1: Retardo UDP en tráfico Alto Autopista

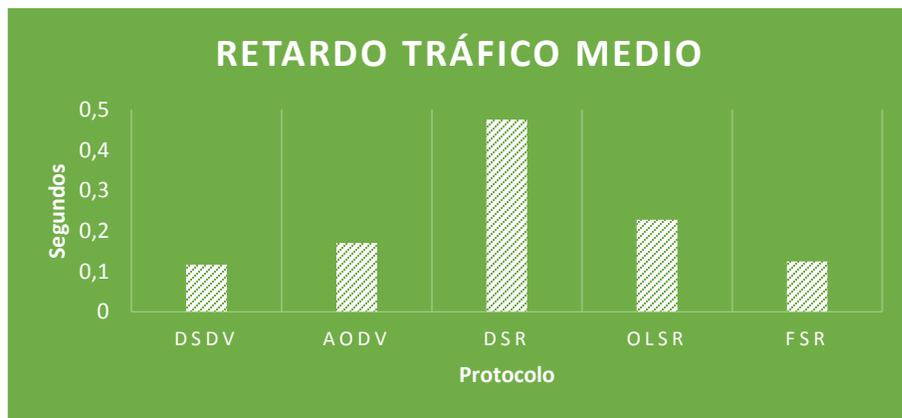


Gráfico 9.2: Retardo UDP en tráfico Medio Autopista



Gráfico 9.3: Retardo UDP en tráfico Bajo Autopista

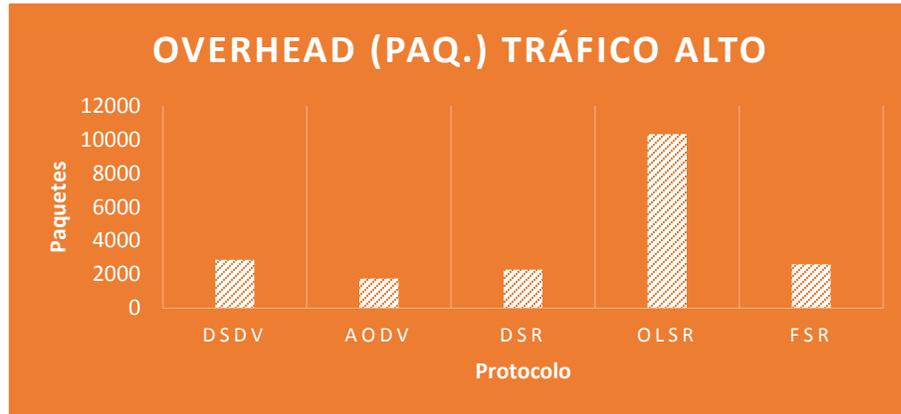


Gráfico 10.1: Sobrecarga UDP en tráfico Alto Autopista

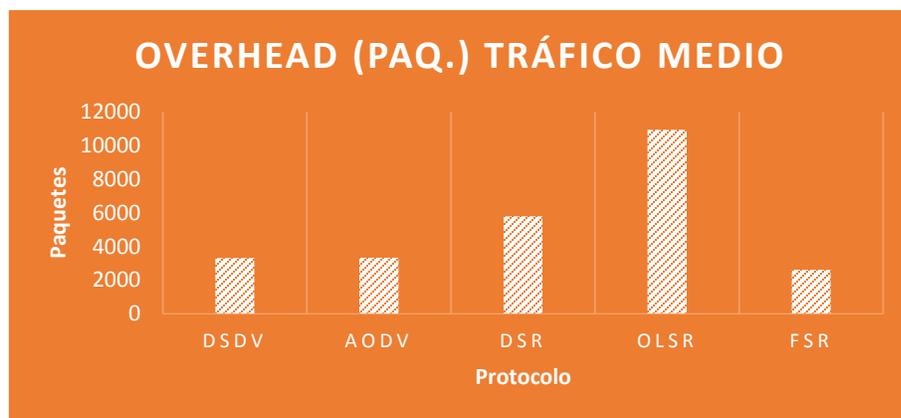


Gráfico 10.2: Sobrecarga UDP en tráfico Medio Autopista

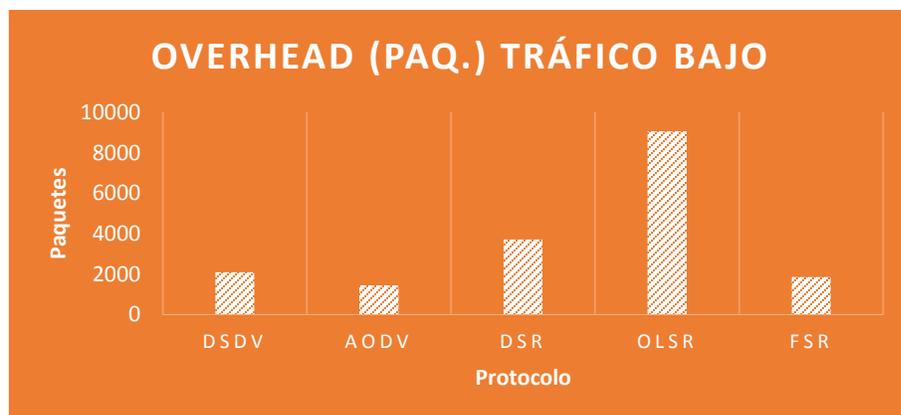


Gráfico 10.3: Sobrecarga UDP en tráfico Bajo Autopista



Observando las gráficas se puede concluir que para la autopista y densidades de tráfico UDP definidos, se obtienen buenos comportamientos con densidad de tráfico baja, cuando en circuitos urbanos este resultado sucedía en densidades medias. En autopistas y con densidades bajas es donde el porcentaje de éxito se acerca al 50% en el caso de los protocolos reactivos en particular AODV. Este protocolo, de forma clara, se puede presentar como claro vencedor en el estudio de UDP, seguido del otro protocolo reactivo DSR. Véase en detalle las conclusiones de cada conjunto de gráficas:

- **Porcentaje de Entregas (*pdfr*):** En comparación con los comportamientos en circuitos urbanos se observa que en densidades altas y medias no existen superioridades en cuanto a los protocolos y se puede decir que tienen comportamientos parecidos, exceptuando densidades bajas, que es donde en autopistas los protocolos poseen los números más altos de *pdfr*. Quizá debido a las altas velocidades de los nodos y que los paquetes viajan con números menores de saltos para llegar a su destino, teniendo así menos probabilidades de pérdidas de paquetes. El mejor comportamiento lo tiene AODV, seguido de DSR, DSDV, OLSR y FSR.
- **Retardos:** AODV presenta retardos aceptables entre 0,1 y 0,3 inferiores incluso a los de algún protocolo proactivo como OLSR con mucho menor porcentaje de éxito. También se ve a simple vista unos valores bastante elevados de retardo de DSR, sobre todo para densidades de tráfico medias. Esto puede ser debido a que este protocolo consiga enviar tráfico utilizando una ruta con elevado número de saltos, aumentando la media de este indicador considerablemente. En comparación con circuitos urbanos, los protocolos poseen menores tiempos de retardos en autopista con FSR como excepción, que posee retardos considerablemente menores en circuitos urbanos. En este caso en particular se tiene que DSDV posee los mejores tiempos de retardo en autopista.
- **Overhead:** El protocolo con mejor comportamiento es AODV, seguido de FSR, DSDV y DSR. Una vez más OLSR genera una sobrecarga en la red desproporcionada para los rendimientos que proporciona. Al igual que pasaba en el circuito urbano los protocolos proactivos mantienen un *overhead* casi constante al variar la densidad de nodos mientras que los reactivos son más sensibles a esta variación, pero en comparación con los resultados de circuitos urbano, la sobrecarga de paquetes introducidos por los protocolos en autopista es mucho mayor que la de los mismos protocolos en urbano.

5.4. Resultado y Análisis de Métricas Circuito Autopista UDP

Nivel de Tráfico Alto

Protocolo	Paq. Enviados	% Entrega	Retardo	Overhaed (Paq.)
DSDV	1562	99,36	0,1339	2879
AODV	1962	98,27	0,1184	1009
DSR	1805	98,67	0,1596	63
OLSR	1407	98,58	0,204	10293
FSR	1396	98,88	0,1238	2604

Cuadro 13: TCP Autopista tráfico Alto

Nivel de Tráfico Medio

Protocolo	Paq. Enviados	% Entrega	Retardo	Overhaed (Paq.)
DSDV	1131	98,59	0,1168	3039
AODV	1687	96,92	0,1802	2195
DSR	1338	99,78	0,1746	87
OLSR	971	99,18	0,1277	11301
FSR	993	97,28	0,1541	2604

Cuadro 14: TCP Autopista tráfico Medio

Nivel de Tráfico Bajo

Protocolo	Paq. Enviados	% Entrega	Retardo	Overhaed (Paq.)
DSDV	5274	99,77	0,14	2200
AODV	5742	99,96	0,1347	87
DSR	5399	99,02	0,1348	43
OLSR	4763	99,56	0,1532	8856
FSR	4819	99,46	0,1194	1860

Cuadro 15: TCP Autopista tráfico Bajo

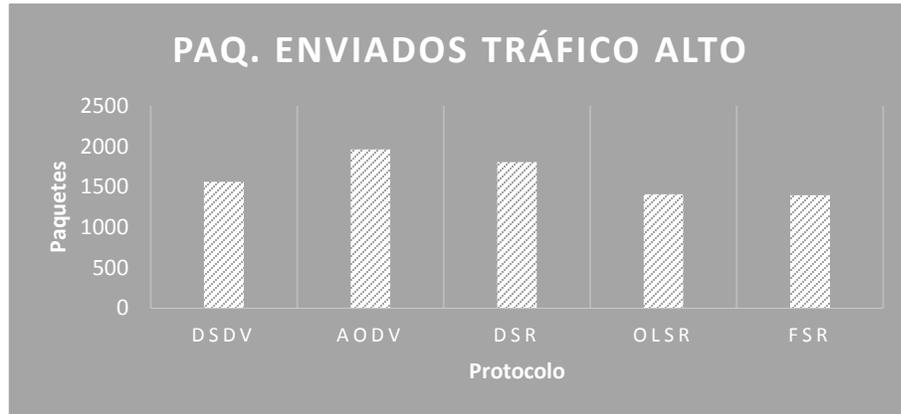


Gráfico 11.1: *Throughput* TCP en tráfico Alto Autopista

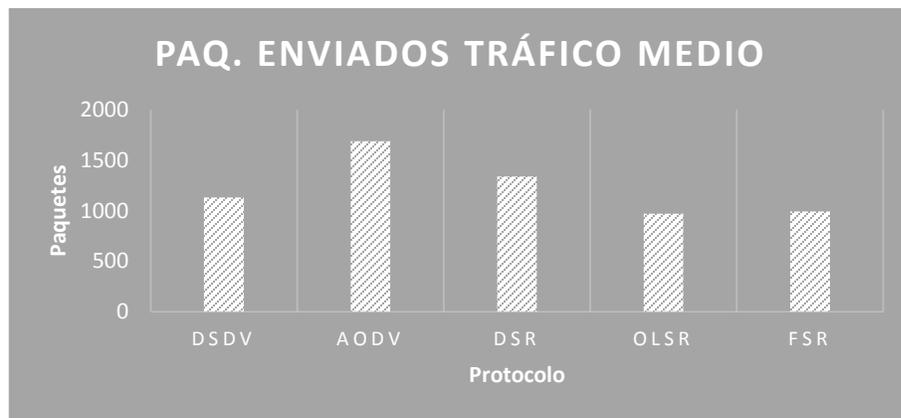


Gráfico 11.2: *Throughput* TCP en tráfico Medio Autopista

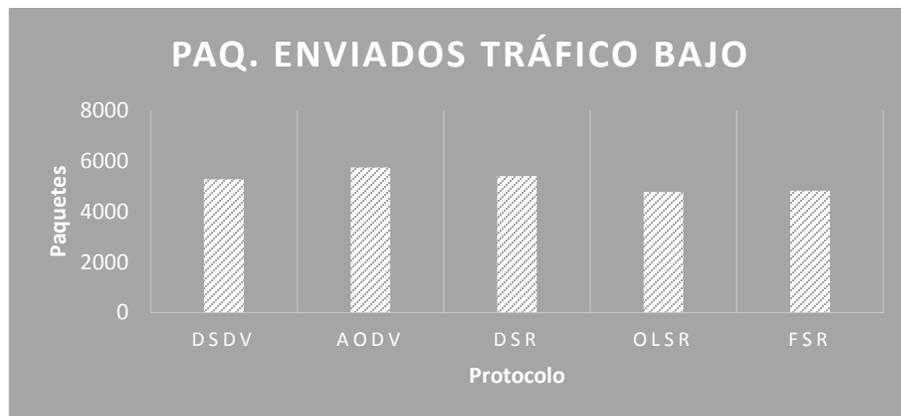


Gráfico 11.3: *Throughput* TCP en tráfico Bajo Autopista

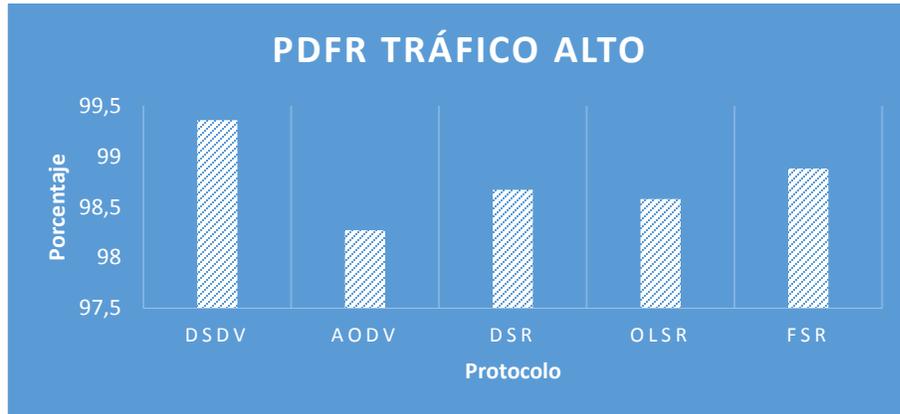


Gráfico 12.1: Porcentaje de Entregas TCP en tráfico Alto Autopista

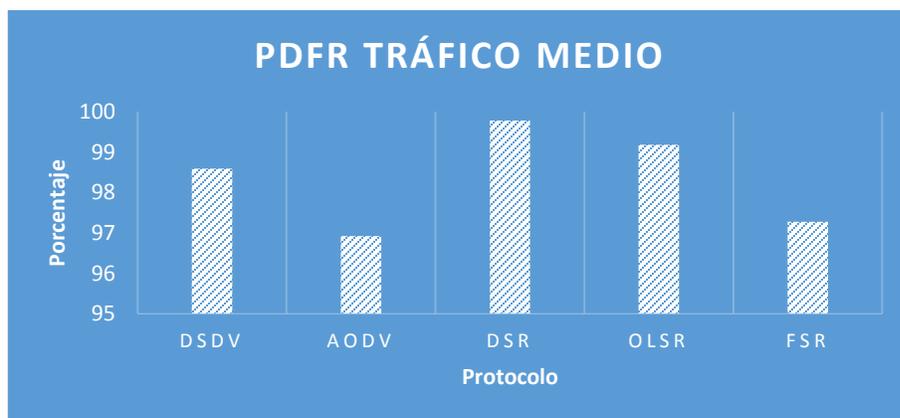


Gráfico 12.2: Porcentaje de Entregas TCP en tráfico Medio Autopista

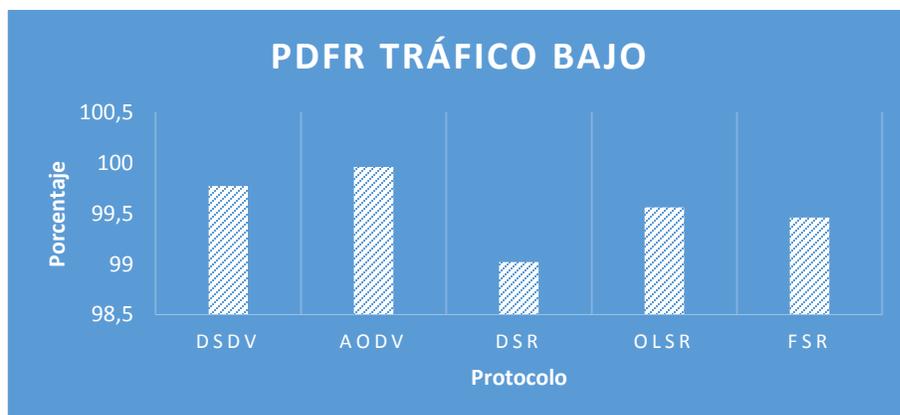


Gráfico 12.3: Porcentaje de Entregas TCP en tráfico Bajo Autopista



Gráfico 13.1: Retardo TCP en tráfico Alto Autopista

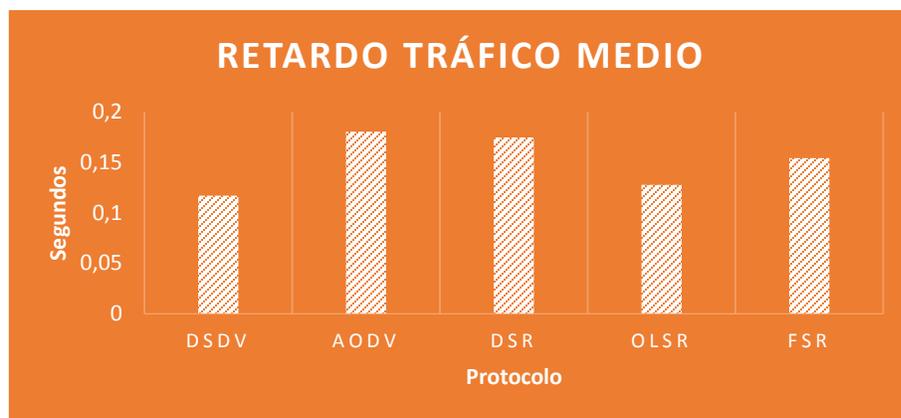


Gráfico 13.2: Retardo TCP en tráfico Medio Autopista



Gráfico 13.3: Retardo TCP en tráfico Bajo Autopista

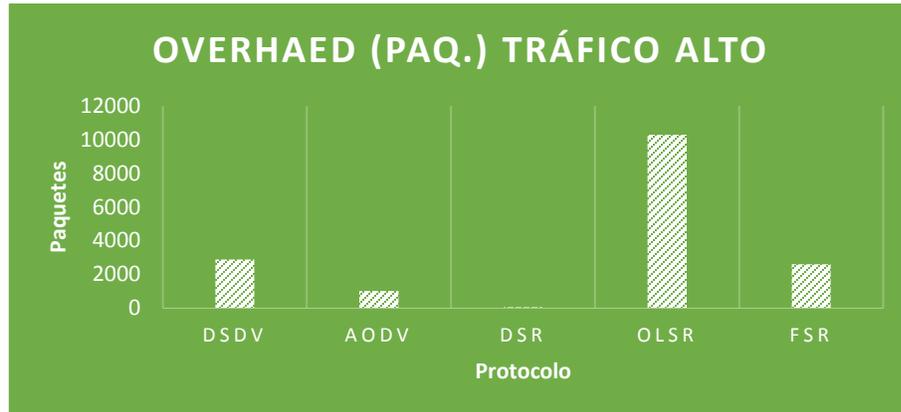


Gráfico 14.1: Sobrecarga TCP en tráfico Alto Autopista

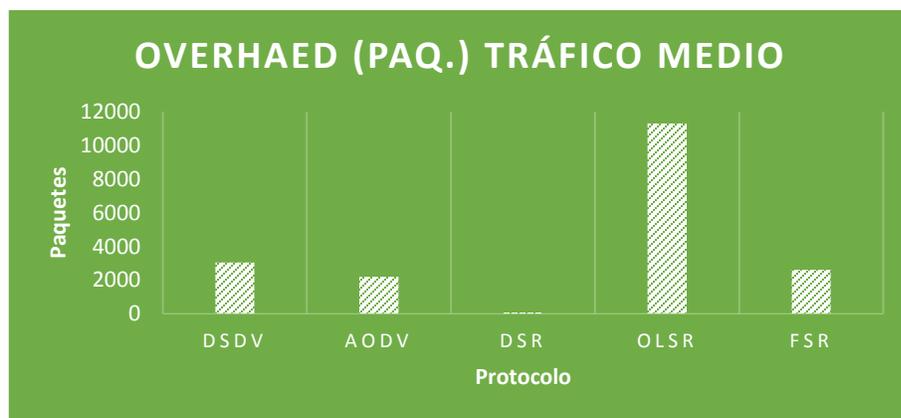


Gráfico 14.2: Sobrecarga TCP en tráfico Medio Autopista

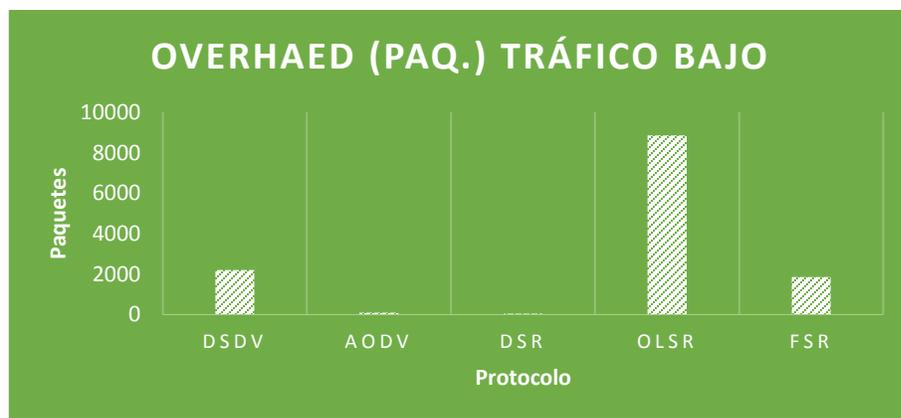


Gráfico 14.3: Sobrecarga TCP en tráfico Bajo Autopista



Se observa que los mejores valores de transferencia de datos TCP en autopista se dan con la densidad de tráfico bajo donde todos los protocolos aumentan considerablemente su rendimiento, esto ya deja claro que ambientes con bajas velocidades y pocos nodos los protocolos presentan sus mejores desempeños, inclusive frente a sus resultados en un ambiente urbano. Una vez más, AODV seguido de FSR dan los mejores resultados en este estudio. Véase más en detalle las conclusiones de cada conjunto de gráficas:

- **Throughput:** Todos los protocolos siguen la misma línea general con una diferencia no muy grande entre todos ellos, esto significa que la mayor parte de los datos transmitidos lo han hecho en circunstancias en las que todos los protocolos funcionan igual, probablemente con el mismo número de nodos intermedios.
- **Porcentaje de Entrega:** Se puede apreciar que AODV tiene un mal porcentaje en comparación con el resto para densidades altas y medias. En cambio DSR y OLSR presentan buenos porcentajes. El protocolo más estable en este caso es el proactivo DSDV.
- **Retardo:** Los protocolos reactivos tienen un máximo para densidades de tráfico medio, esto significa que consiguen enviar información por una ruta que tarda más tiempo en llegar al destino, pero lo alcanza, aumentando en promedio este indicador. En cambio protocolos proactivos como DSDV y FSR mantienen un tiempo de retardo constantes indiferentemente de la densidad de tráfico y aumento o disminución de las velocidades de los nodos.
- **Overhead:** Vuelve a aparecer la excesiva carga de la red por parte de OLSR, manteniéndose baja la carga de ambos protocolos reactivos. FSR presenta una sobrecarga realmente baja. Pero el caso del reactivo DSR es totalmente interesante con sobrecarga de paquetes realmente bajos lo que contrasta de manera curiosa con su desempeño con comunicaciones UDP.



5.5. Análisis Final

La principal conclusión que se puede desprender de este estudio es que en los entornos de topología muy variable, los protocolos reactivos AODV y DSR presentan prestaciones mayores que los protocolos proactivos. Resulta más eficiente llevar a cabo un mecanismo de descubrimiento de rutas sobre demanda que intentar tener tablas actualizadas de ruta en cada momento. En efecto, con un protocolo proactivo, si los cambios de topología son rápidos, se multiplican los mensajes de control introduciendo una sobrecarga excesiva, se pueden perder más paquetes y causar más retardo.

Dentro de los protocolos reactivos, AODV presenta, en la mayoría de las topologías, las mejores tasas de transferencia; bajos retardos y no genera grandes cantidades de paquetes de control para la creación y el mantenimiento de las rutas, se concluye que es el más eficiente para estas comunicaciones en general.

En todos los estudios se observa que el perfil de tráfico óptimo es la densidad media. Si aumenta la densidad de tráfico la VANET se satura y si disminuye el número de nodos se reducen las rutas para llegar al destino en ciertos instantes de tiempo lo que supone pérdidas de paquetes o aumento de los retardos de transmisión.

Tanto en UDP como en TCP, los protocolos reactivos logran mandar mayor cantidad de datos. AODV alcanza en UDP un porcentaje de éxito de 50 % en el mejor de los casos (densidad baja en autopistas) y un *throughput* de alrededor de 6000 paquetes en TCP. Estos valores se observan durante el tiempo que duran las simulaciones (100 s), por lo tanto equivalen a tasas medias de 30kbps para UDP y 70kbps para TCP aproximadamente.

Se comprobó que en muchas ocasiones, los retardos son inversamente proporcionales a la cantidad de información transmitida ya que para conseguir que el paquete llegue a su destino los protocolos se ven muchas veces obligados a mandar el tráfico por rutas más largas, aumentando así la media de los retardos de transmisiones. Aun así, los retardos que presenta AODV, excepto en el circuito urbano con conexión UDP y perfil de tráfico alto, se mantienen entre 100 y 200 ms para todas las topologías.

En cuanto a *overhead*, los protocolos proactivos presentan una mayor carga a la red siendo casi constante al modificar los perfiles de tráfico mientras que los reactivos introducen menos carga y son más sensibles a la densidad de nodos. Esta sensibilidad hace que el porcentaje de paquetes de control vaya del 2 % en el mejor de los casos hasta al 30 % en el peor de los casos. DSR destaca por su buen comportamiento en comunicaciones TCP.



6. CONCLUSIONES

En primer lugar ya introducidos en las redes ad-hoc y VANET para conocer sus características y posibles aplicaciones se creó una plataforma con herramientas para simular redes inalámbricas vehiculares Ad-hoc VANET con la que se pudo analizar y elegir los protocolos de enrutamiento con código disponible para simular, midiendo así todos los parámetros, escenarios, métricas y casos a evaluar en las simulaciones. Esta plataforma nos ha arrojado los resultados deseados y finalmente después de obtenerlos y analizarlos se pudo realizar las conclusiones pertinentes, determinando que protocolo de enrutamiento es más eficiente a la hora de la implementación práctica de una red inalámbrica vehicular Ad-hoc VANET.

6.1. Logros

La principal conclusión obtenida a lo largo de este proyecto es que el despliegue de una plataforma de comunicaciones en el entorno automóvil puede aportar muchas mejoras a las arquitecturas de comunicaciones inalámbricas tradicionales, y en esto se puede incluir a las telefónicas como a la sociedad en su conjunto.

La plataforma permitirá la implementación de servicios de seguridad vial ayudando al conductor y pudiendo contribuir a disminuir las cifras de accidentes en la carretera. Gracias a sistemas de conducción basados en la plataforma de comunicaciones VANET se puede esperar una gestión más eficiente del tráfico basados en servicios de valor añadido para los ocupantes de los vehículos. Tales avances son de indudable utilidad para la sociedad en general. Además de la mejoría y calidad de servicio que el transporte público podrá ofrecer a los ciudadanos.

La comparación de protocolos de encaminamiento llevada a cabo en este proyecto ha permitido valorar la eficiencia de los protocolos en varios escenarios de comunicación típicos. Se ha obtenido resultados que se podrán reutilizar en otros proyectos.

Uno de los objetivos que se habían propuesto era la definición de una plataforma de simulación para el entorno automóvil. Este objetivo se ha cumplido ya que ahora se dispone de una plataforma de simulación eficiente que ha permitido comparar distintos protocolos de comunicaciones. Esa plataforma se podrá reutilizar en otros proyectos que con pequeñas modificaciones podrán abordar otras cuestiones y problemáticas con otras condiciones.



Durante el proyecto se fue capaz de identificar los escenarios de comunicación típicos y útiles para el despliegue de servicios en redes VANET básica en la ciudad. Además para cada escenario se han definido indicadores de red para medir el rendimiento de los protocolos. Este logro no debe encasillarse en el marco de este proyecto. Es útil tener una serie de test y de medidas que se pueden reutilizar en otras simulaciones o en otras pruebas de terreno.

A nivel de simulaciones de protocolos se ha podido comprobar que las redes VANET funcionan con rendimientos aceptables, a pesar de los ambientes inestables de muy alta movilidad en los cuales operan. Se ha demostrado que en escenarios de comunicaciones unicast, AODV es claro opera con un gran rendimiento frente a otros protocolos. Esos resultados son fundamentales porque es imprescindible valorar la viabilidad de una solución mediante simulación antes de poder implementar una solución.

Para resumir, el presente estudio participa en la construcción de un entorno de trabajo riguroso para simular comunicaciones VANET. Los resultados obtenidos son confortables en la idea de que sería viable implementar un prototipo de comunicación VANET. Se han definidos herramientas de trabajo, tales como escenarios e indicadores de red, que definen un marco para futuras simulaciones.

6.2. Futuros Trabajos

Aunque se ha avanzado en la investigación acerca de las redes vehiculares, queda mucho camino por recorrer y es necesario seguir haciendo esfuerzos en la investigación de este campo.

Primero, es necesario seguir investigando para proponer una plataforma de simulación más realista y completa. Tomando en cuentas datos probabilísticos que permitan conseguir resultados más veraces y más aproximados a la realidad. Pasando también por la implementación de nuevos protocolos de encaminamiento dentro del simulador NS-2. En un futuro resultará interesante implementar un protocolo basado en AODV, quizá híbrido y hasta con respaldo UMTS, de forma rigurosa para poder afinar los resultados presentados en este documento.

Además, será necesario implementar otros protocolos de encaminamiento, ya que los protocolos basados en información geográfica prometen mejorar de forma significativa



los rendimientos en redes VANET. A partir de mapas, horarios de autobuses, sensores en la carretera, se recoge una información útil al protocolo de encaminamiento. Una vez implementados esos protocolos en NS-2, será necesario volver a realizar una comparativa con los protocolos existentes para comprobar si la información geográfica introduce mejoras en el encaminamiento vehicular.

Otra línea de trabajo interesante será desarrollar un demostrador para poder realizar pruebas de campo y validar los resultados obtenidos mediante simulación. Las pruebas de terreno son imprescindibles antes de empezar con el desarrollo de una solución ya que las simulaciones, aunque dan buenas indicaciones de rendimiento, no son más que una aproximación de la realidad. Por lo tanto, el siguiente paso en este trabajo será desarrollar un nodo de comunicación híbrido AODV/respaldo UMTS para poder comprobar la viabilidad de la solución en escenarios reales. El nodo híbrido se deberá empotrar en un sistema embarcado dentro del vehículo. El desarrollo de este, supone un trabajo adicional significativo ya que se tendría que implementar un protocolo de encaminamiento nuevo y desarrollar los servicios para probar la nueva plataforma de comunicación.



REFERENCIAS

- Draves, R., Padhye, J., & Zill, B. (2004). *Routing in multi-radio, multi-hop wireless mesh networks*. *ACM Annual International Conference on Mobile Computing and Networking (MOBICOM)*, (págs. 114-128).
- Perkins, C. (2001). *Ad-hoc networking: an introduction*. Boston, MA: Addison-Wesley Longman Publishing.
- Das S. Perkins, C. E. Belding-Royer (2003) E.M. *Ad-hoc On-demand Distance Vector (aodv) Routing. RFC 3561*, IETF Network Working Group.
- Bommer, J. (2007). *Joint Networking Command and Control (C2) Communications among Distributed Operations, JCAS and Joint Fires*. Tesis de Maestría, Naval Postgraduate School, Monterrey, California.
- Nitin H. Vaidya, (2006). *Tutorial on Mobile Ad Hoc Networks: Routing, MAC and Transport Issues*. Disponible en <http://www.crhc.uiuc.edu/wireless/tutorials.html>
- Akyildiz, I., Wang, X., & Wang, W. (2005). *Wireless mesh networks: a survey*. *Computer Networks*, 47 (4), 445-487.
- WWRF. (2008). *Technologies for the Wireless Future: Wireless World Research Forum*. San Francisco: Wiley.
- Srivastava, V., Hilal, A., Thompson, M., Chattha, J., MacKenzie, A., & DaSilva, L. (2008). *Characterizing mobile Ad-hoc networks: the maniac challenge experiment*. *Proceedings of the third ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, (págs. 65-72). San Francisco, California.
- Rohde & Schwarz. (2007). *WLAN 802.11p Measurements for Vehicle to Vehicle (V2V) DSRC*. Munich, Alemania.
- Daniel Lang. (2003). *A comprehensive overview about selected Ad-hoc Networking Routing Protocols*. Munich, Alemania. E-Mail: dl@leo.org.
- Eichler, S., Schroth, C., Eberspächer, J., *Car-to-Car Communication*, VDE Congress "Innovations for Europe", Poster session, Institute of Communication Networks, Technische Universität München, München, 2006.

