



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

**ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES
Y REDES**

**“EVALUACIÓN DE PROTOCOLOS DE ENCAMINAMIENTO EN
REDES VANET CONECTADOS MEDIANTE 802.11P
CONSIDERANDO MODELOS DE MOVILIDAD SINTÉTICOS Y
GENERADORES DE TRÁFICO VEHICULAR”**

TRABAJO DE TITULACIÓN: PROYECTO TÉCNICO

Para optar al Grado Académico de:

**INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y
REDES**

AUTOR: TIERRA QUISHPILLO DANNI OSWALDO

TUTOR: ING. RAÚL MARCELO LOZADA YÁNEZ

Riobamba – Ecuador

2018

@2018, Danni Oswaldo Tierra Quishpillo

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y

REDES

El tribunal del trabajo de titulación certifica que: El trabajo de titulación: EVALUACIÓN DE PROTOCOLOS DE ENCAMINAMIENTO EN REDES VANET CONECTADOS MEDIANTE 802.11P CONSIDERANDO MODELOS DE MOVILIDAD SINTÉTICOS Y GENERADORES DE TRÁFICO VEHICULAR”, de responsabilidad del señor TIERRA QUISHPILLO DANNI OSWALDO, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

NOMBRE

FIRMA

FECHA

Dr. Julio Santillán

VICEDECANO DE LA FACULTAD DE

INFORMÁTICA Y ELECTRÓNICA

Ing. Patricio Romero

DIRECTOR DE LA ESCUELA DE

INGENIERÍA ELECTRÓNICA EN

TELECOMUNICACIONES Y REDES

Ing. Raúl Marcelo Lozada Yáñez.

DIRECTOR DEL TRABAJO DE

TITULACIÓN

Ing. Alberto Arellano Aucancela.

MIEMBRO DEL TRIBUNAL

Yo, Tierra Quishpillo Danni Oswaldo, con cédula de identidad 230013611-2, declaro que el presente trabajo de titulación es de mi autoría y que los resultados del mismo son auténticos y originales. Los textos constantes en el documento que provienen de otra fuente están debidamente citados y referenciados.

Como autor, asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación.

Riobamba, 08 de agosto de 2018

Danni Oswaldo Tierra Quishpillo

ESTUDIANTE

Ing. Raúl Marcelo Lozada Yáñez

DIRECTOR TRABAJO DE TITULACIÓN

Ing. Alberto Arellano Aucancela

MIEMBRO TRIBUNAL

Yo, **TIERRA QUIHPILLO DANNI OSWALDO**, soy responsable de las ideas, doctrinas y resultados, expuestos en este trabajo, y el patrimonio intelectual del Trabajo de Titulación pertenece a la **ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**.

Tierra Quishpillo Danni Oswaldo

DEDICATORIA

El presente trabajo de titulación va dedicado con mucho amor, a mi madre Silvia Quishpillo, quien con sus palabras de aliento y motivación lograron que día a día me esfuerce para cumplir nuevas metas y cuya partida ha dejado un gran vacío en mi alma pero al mismo tiempo sus consejos y amor perduraran por siempre, a mi padre Ángel Tierra que con su esfuerzo y sacrificio me brindo la ayuda necesaria para continuar mis estudios y sigue siendo un pilar fundamental en mi formación personal, este logro no habría sido posible sin los valores inculcados por mis padres.

A mi hermana mayor Lourdes, quien se ha convertido en mi segunda madre y representa un gran apoyo en mi vida, a mi hermana menor Paola que con sus ocurrencias provoca sonrisas y alegría en nuestras vidas, para ellas quiero ser un ejemplo de superación y apoyarlas para que cumplan nuevas metas.

A mi familia y amigos que me han brindado su apoyo incondicional y han aportado en mi vida compartiendo momentos de felicidad y tristeza permitiendo superar obstáculos y cumplir metas, con la confianza de siempre poder contar con ellos no me queda más que decirles gracias y que Dios los bendiga.

Danni

AGRADECIMIENTO

Agradezco a Dios por darme la oportunidad de culminar esta etapa de mi vida y brindarme la fortaleza y perseverancia para seguir adelante ante los obstáculos que se han presentado a lo largo de mi vida. A mi familia, ya que gracias a ellos he logrado llegar a la meta y han sido el pilar fundamental. A mis amigos con los que compartimos buenos momentos, aprendimos unos de otros, me extendieron la mano y ofrecieron su ayuda incondicional.

A todos los docentes que formaron parte de este largo camino, quienes compartieron su experiencia y conocimiento sobre las materias de la carrera y con los cuales se ha establecido un lazo de amistad y respeto mutuo. Sobre todo, al Ing. Raúl Lozada, ya que gracias a su guía he logrado culminar con éxito el presente trabajo de titulación y al Ing. Alberto Arellano, quien aportó al planteamiento del tema del trabajo de titulación.

Danni

TABLA DE CONTENIDOS

RESUMEN	xx
SUMMARY	xxi
INTRODUCCIÓN	1

CAPÍTULO I

1.	MARCO TEÓRICO REFERENCIAL	5
1.1.	Protocolo IEEE 802.11	5
1.1.1.	Introducción.....	5
1.1.2.	Capa física (Layer 1).....	6
1.1.2.1.	Espectro esparcido por salto de frecuencia (FHSS)	6
1.1.2.2.	Espectro esparcido por secuencia directa (DSSS)	8
1.1.2.3.	Multiplexación por División de Frecuencia Ortogonal (OFDM)	9
1.1.3.	Capa de enlace de datos (Layer 2)	9
1.1.3.1.	Capa de control de acceso al medio (MAC).....	9
1.1.3.2.	Capa de control lógico de enlace (LLC)	11
1.1.4.	Protocolo CSMA/CA y MACA	11
1.1.4.1.	CSMA/CA	11
1.1.4.2.	MACA	12
1.2.	Enmiendas IEEE 802.11	13
1.2.1.	Enmienda 802.11b	14
1.2.2.	Enmienda 802.11a	14
1.2.3.	Enmienda 802.11g	15
1.2.4.	Enmienda 802.11n	15
1.3.	Redes VANET	16
1.3.1.	Introducción.....	16

1.3.2.	¿Qué es una red VANET?.....	17
1.3.3.	Características de una VANET	18
1.3.4.	Tipos de comunicación de una VANET.....	18
1.3.4.1.	Comunicación Vehículo – Vehículo (V2V).....	19
1.3.4.2.	Comunicación Vehículo – Infraestructura (V2I).....	20
1.3.5.	Estándar IEEE 802.11p.....	21
1.4.	Clasificación de protocolos de enrutamiento para VANET	23
1.4.1.	Protocolos de Enrutamiento	23
1.4.2.	Según la planificación.....	24
1.4.2.1.	Protocolos Reactivos.....	24
1.4.2.2.	Protocolos Proactivos.....	24
1.4.3.	Según el alcance	25
1.4.3.1.	Unicast.....	25
1.4.3.2.	Multicast	25
1.4.3.3.	Geocast	25
1.4.3.4.	Broadcast	26
1.4.4.	Según el algoritmo de ruteo	26
1.4.4.1.	Basado en el algoritmo del vector distancia	26
1.4.4.2.	Basado en el algoritmo de estado de enlace	27
1.4.4.3.	Basado en el encaminamiento desde el origen	27
1.4.5.	Según el cálculo de rutas.....	28
1.4.5.1.	Centralizado.....	28
1.4.5.2.	Distribuido	28
1.5.	Protocolos de enrutamiento para redes VANET	28
1.5.1.	Protocolos de enrutamiento proactivos para redes VANET	28
1.5.1.1.	DSDV (Destination Sequenced Distance Vector)	28
1.5.1.2.	CGSR (Cluster-Head Gateway Switch Routing Protocol)	29
1.5.1.3.	WRP (Wireless Routing Protocol)	29

1.5.1.4.	HSR (Hierarchi).....	30
1.5.1.5.	OLSR (Optimized Link State Routing).....	30
1.5.1.6.	FSR (Fisheye State Routing).....	30
1.5.2.	Protocolos de enrutamiento reactivos para redes VANET	31
1.5.2.1.	AODV (Ad-hoc On Demand Vector)	31
1.5.2.2.	AOMDV (Ad-hoc On-demand Multipath Distance Vector Routing).....	32
1.5.2.3.	DSR (Dynamic Source Routing).....	33
1.5.2.4.	ABR (Associativity Based Routing Protocol).....	33
1.5.2.5.	TORA (Temporally Ordered Routing Algorithm)	34
1.6.	Modelos de movilidad de redes VANET	35
1.6.1.	Modelos de movilidad sintéticos.....	36
1.6.1.1.	Modelo random waypoint (RWP).....	37
1.6.1.2.	Modelo random walk (RWM).....	38
1.6.1.3.	Modelo Manhattan	38
1.6.2.	Modelos de movilidad basados en simuladores de tráfico.....	39
1.6.2.1.	Modelo de movilidad Krauss	40
1.6.2.2.	Modelo de movilidad Kerner	40
1.6.2.3.	Modelo de conducción inteligente (IDM)	41
1.7.	Simuladores para el desarrollo de redes VANET.....	42
1.7.1.	Simuladores de tráfico vehicular.....	43
1.7.1.1.	CityMob for RoadMaps (C4R)	44
1.7.1.2.	Simulation of Urban Mobility (SUMO).....	44
1.7.1.3.	Street Random Waypoint (STRAW).....	45
1.7.1.4.	VanetMobiSim.....	45
1.7.2.	Simuladores de red.....	46
1.7.2.1.	OMNeT++	46
1.7.2.2.	Network Simulator 2 (NS-2).....	47
1.7.2.3.	Optimized Network Engineering Tool (OPNET Modeler)	47

1.7.3.	Simuladores híbridos.	48
1.7.3.1.	Traffic and Network Simulator environment (Trans).....	48
1.7.3.2.	Vehicules in Network Simulation (Veins)	49
1.7.3.3.	Mobility Model Generator for Vehicular Networks (MOVE).....	49

CAPÍTULO II

2.	MARCO METODOLÓGICO	50
2.1.	Introducción.....	51
2.2.	Determinación de la zona geográfica	51
2.3.	Flujo de tráfico vehicular	52
2.4.	Plataformas empleadas.....	55
2.4.1.	Simulador de tráfico vehicular	55
2.4.2.	Simulador de Red.....	57
2.4.3.	Simulador Híbrido	59
2.5.	Modelos de movilidad vehicular Empleados.....	60
2.5.1.	Modelos de movilidad vehicular sintéticos	60
2.5.2.	Modelos de movilidad vehicular basados en generadores de tráfico vehicular	62
2.6.	Modelo de propagación empleado	64
2.7.	Protocolos de enrutamiento empleados	64
2.8.	Parámetros de simulación	66
2.9.	Descripción de la interfaz gráfica de MOVE	67
2.10.	Simulación del entorno VANET basado en el modelo generador de tráfico IDM..	70
2.10.1.	Comunicación V2V	71
2.10.1.1.	Simulación de la sección de movilidad vehicular.....	71
2.10.1.2.	Simulación de la sección de red de datos	78
2.10.2.	Comunicación V2I.....	83
2.10.2.1.	Simulación de la sección de movilidad vehicular.....	83
2.10.2.2.	Simulación de la sección de red de datos	85

CAPÍTULO III

3.	MARCO DE RESULTADOS DE LAS SIMULACIONES	88
3.1.	Análisis de resultados de las simulaciones	88
3.1.1.	Análisis de Comunicación V2V	89
3.1.1.1.	Simulación del Escenario 1	89
3.1.1.2.	Simulación del Escenario 2.....	90
3.1.1.3.	Simulación del Escenario 3.....	90
3.1.2.	Análisis de Comunicación V2I	91
3.1.2.1.	Simulación del Escenario 4.....	91
3.1.2.2.	Simulación del Escenario 5.....	92
3.1.2.3.	Simulación del Escenario 6.....	93
3.2.	Comparación de los resultados.....	93
3.2.1.	Comparación de Comunicación V2V.....	94
3.2.1.1.	PDR o PDF (Relación de entrega de paquetes).....	94
3.2.1.2.	Throughput (Rendimiento)	94
3.2.1.3.	NRL (Carga de enrutamiento normalizada)	95
3.2.1.4.	Average End to End Delay (Retardo promedio).....	96
3.2.2.	Comparación de Comunicación V2I.....	96
3.2.2.1.	PDR o PDF (Relación de entrega de paquetes).....	96
3.2.2.2.	Throughput (Rendimiento)	97
3.2.2.3.	NRL (Carga de enrutamiento normalizada)	98
3.2.2.4.	Average End to End Delay (Retardo promedio).....	98
	CONCLUSIONES.....	100
	RECOMENDACIONES.....	102
	BIBLIOGRAFÍA	
	ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-1: Información del canal inalámbrico V2X.....	22
Tabla 2-1: Protocolos proactivos para VANET.....	31
Tabla 3-1: Protocolos reactivos para VANET.....	34
Tabla 4-1: Clases del modelo sintético.....	37
Tabla 5-1: Tipos de conductores en el modelo IDM.....	42
Tabla 1-2: Flujo de tráfico vehicular de las distintas rutas de la zona de simulación.....	54
Tabla 2-2: Características de algunos simuladores de tráfico vehicular.....	56
Tabla 3-2: Selección del simulador de tráfico vehicular.....	56
Tabla 4-2: Características de Simuladores de red de datos.....	57
Tabla 5-2: Selección del simulador de red.....	58
Tabla 6-2: Características de Simuladores híbridos.....	59
Tabla 7-2: Selección del simulador híbrido.....	60
Tabla 8-2: Características de algunos modelos de movilidad sintéticos.....	61
Tabla 9-2: Selección del modelo de movilidad sintético.....	61
Tabla 10-2: Modelos de movilidad basados en un generador de tráfico vehicular.....	62
Tabla 11-2: Selección del modelo de movilidad basado en un generador de tráfico vehicular.....	63
Tabla 12-2: Comparación de protocolos proactivos.....	64
Tabla 13-2: Comparación de protocolos reactivos.....	65
Tabla 14-2: Parámetros de simulación del entorno VANET.....	66
Tabla 15-2: Descripción de las subsecciones de Mobility Model en MOVE.....	68
Tabla 16-2: Descripción de las subsecciones de Traffic Model en MOVE.....	70

ÍNDICE DE FIGURAS

Figura 1-1: Técnica FHSS	7
Figura 2-1: Codificación de información con secuencia de Barker	8
Figura 3-1: Protocolo WLAN de 802.11	10
Figura 4-1: Evolución de IEEE 802.11	14
Figura 5-1: Arquitectura de una red VANET	17
Figura 6-1: Red VANET-Comunicación V2V	20
Figura 7-1: Red VANET-Comunicación V2I-I2V	21
Figura 8-1: Clasificación de protocolos de enrutamiento MANET	23
Figura 9-1: Clasificación de movilidad para VANET	36
Figura 1-2: Metodología aplicada en la simulación	50
Figura 2-2: Geográfica de la simulación	52
Figura 3-2: Identificadores para las secciones de la zona geográfica	52
Figura 4-2: Recomendaciones para IEEE 802.11p	67
Figura 5-2: GUI de herramienta MOVE	67
Figura 6-2: GUI de las subsecciones de Mobility Model	68
Figura 7-2: Subsecciones de Traffic Model	70
Figura 8-2: GUI de la herramienta MOVE	71
Figura 9-2: GUI Mobility Model para configurar los parámetros de red vehicular	72
Figura 10-2: Configuración de nodos (intersecciones)	72
Figura 11-2: Configuración del tipo de sección de calle	73
Figura 12-2: Configuración de las secciones que forman las calles	73
Figura 13-2: Rutas de los archivos para crear la topología de la zona de simulación	74
Figura 14-2: Subiendo archivo .netc.cfg	74
Figura 15-2: Generación exitosa del mapa	75
Figura 16-2: Parámetros para las rutas del modelo IDM	75

Figura 17-2: Parámetros para el tipo de conductor normal de IDM	76
Figura 18-2: Creación de los archivo .sumo.tr y .sumo.cfg.....	76
Figura 19-2: Cargando archivo .sumo.cfg en la herramienta SUMO	77
Figura 20-2: Simulación vehicular con 120 nodos.....	77
Figura 21-2: Cargando archivo .sumo.cfg para luego poder exportar los datos a NS-2	78
Figura 22-2: GUI de herramienta MOVE	78
Figura 23-2: GUI de la sección Traffic Model	79
Figura 24-2: Importación de datos desde SUMO a NS-2 (.sumo.tr)	79
Figura 25-2: Importación de datos desde SUMO a NS-2 (.net.xml)	79
Figura 26-2: Configuración de parámetros para la red de datos en NS-2.....	80
Figura 27-2: Las 60 Conexión TCP (archivo.tcl)	81
Figura 28-2: Ejecución del archivo .tcl con la herramienta NS-2.....	81
Figura 29-2: GUI de NAM (extensión de NS-2).....	82
Figura 30-2: Simulación de VANET completa (SUMO-NS2).....	82
Figura 31-2: Configuración de intersecciones (V2I)	83
Figura 32-2: Topología para comunicación V2I	84
Figura 33-2: Declaración de las 5 Infraestructuras adicionales a los 120 vehículos	85
Figura 34-2: Posicionamiento estratégico de las Infraestructuras en la topología	86
Figura 35-2: Ejecución del archivo tcl con 120 vehículos y 5 infraestructuras	86
Figura 36-2: Comunicación V2I con 120 vehículos y 5 infraestructuras (AOMDV)	87
Figura 37-2: Comunicación V2I con 120 vehículos y 5 infraestructuras (DSDV)	87

_Toc519970515

ÍNDICE DE GRÁFICOS

Gráfico 1-3: Parámetro PDR de 5, 10 y 30 conexiones V2V.....	94
Gráfico 2-3: Parámetro Throughput de 5, 10 y 30 conexiones V2V.	95
Gráfico 3-3: Parámetro NRL de 5, 10 y 30 conexiones V2V.	95
Gráfico 4-3: Parámetro Delay End to End de 5, 10 y 30 conexiones V2V.....	96
Gráfico 5-3: Parámetro PDR de 5, 10 y 30 conexiones V2I.....	97
Gráfico 6-3: Parámetro Throughput de 5, 10 y 30 conexiones V2I.....	97
Gráfico 7-3: Parámetro NRL de 5, 10 y 30 conexiones V2I.....	98
Gráfico 8-3: Parámetro Delay End to End de 5, 10 y 30 conexiones V2I.....	99

ÍNDICE DE ECUACIONES

Ecuación 1-1: Descripción de la velocidad de un vehículo según el modelo Manhattan.....	39
Ecuación 2-1: Restricción sobre los límites de velocidad en el modelo Manhattan.....	39
Ecuación 3-1: Velocidad según el modelo Krauss	40
Ecuación 4-1: Distancia entre vehículos según el modelo IDM	41
Ecuación 1-2: Fórmula de muestreo para poblaciones conocidas.....	53

ABREVIATURAS

AD-HOC Para esto

VANET Red Ad-Hoc Vehicular (Vehicular Ad-Hoc Network)

ITS Sistema Inteligente de Transporte Terrestre (Intelligent Transportation Systems)

RSU Unidad Situada en la Carretera (Road-Side Unit)

OBU Unida de Comunicación a bordo (On Board Unit)

MANET Red Ac-Hoc Móvil (Mobile Ad hoc Network)

IEEE Instituto de Ingeniería Eléctrica y Electrónica (Institute of Electrical and Electronics Engineers)

MAC Control de Acceso al Medio (Medium Access Control)

PHY Capa Física (Physical Layer)

WLAN Red de Área Local Inalámbrica (Wireless Local Area Network)

CSMA/CA Acceso Múltiple con Detección de Operador con Prevención de Colisiones (Carrier-sense multiple access with collision avoidance)

OSI Interconexión de Sistemas Abiertos (Open Systems Interconnection)

FHSS Espectro Esparcido por Salto de Frecuencia (Frequency Hopping Spread Spectrum)

DSSS Espectro Esparcido por Secuencia Directa (Direct Sequence Spread Spectrum)

OFDM Modulación por División de Frecuencias Ortogonales (Orthogonal Frequency-Division Multiplexing)

LLC Control Lógico de Enlace (Logical Link Control)

ACK Acuse de Recibido (acknowledgement)

PCF Función de coordinación puntual (Punctual Coordination Function)

DFC Función de coordinación distribuida (Distributed Coordination Function)

BSS Conjunto Básico de Servicios (Business Support System)

MACA Acceso Múltiple con Prevención de Colisiones (MultiAccess Collision Avoidance)

IFS Espacio Entre Tramas

CW Ventana de Contienda

SLOT TIME Ranuras Temporales

RTS Petición de envío

CTS Borrar para enviar

MIMO Múltiple entrada múltiple salida

V2V Comunicación VANET vehículo a vehículo

V2I Comunicación VANET vehículo a infraestructura

AU Unidad de aplicación

DSDV Protocolo de ruteo proactivo (Destination Sequenced Distance Vector)

CGSR Protocolo de ruteo proactivo (Cluster-Head Gateway Switch Routing Protocol)

WRP Protocolo de ruteo proactivo (Wireless Routing Protocol)

HSR Protocolo de ruteo proactivo (Hierarchi)

OLSR Protocolo de ruteo proactivo (Optimized Link State Routing)

FSR Protocolo de ruteo proactivo (Fisheye State Routing)

AODV Protocolo de ruteo reactivo (Ad-hoc On Demand Vector)

AOMDV Protocolo de ruteo reactivo (Ad-hoc On-demand Multipath Distance Vector Routing)

DSR Protocolo de ruteo reactivo (Dynamic Source Routing)

TORA Protocolo de ruteo reactivo (Temporally Ordered Routing Algorithm)

RWP Modelo de movilidad random waypoint

RWM Modelo de movilidad random walk

IDM Modelo de conducción inteligente

SUMO Simulador de red vehicular (Simulation of Urban Mobility)

STRAW Simulador de red vehicular (Street Random Waypoint)

NS-2 Simulador de red de datos (Network Simulator 2)

MOVE Simulador híbrido (Model Generator for Vehicular Networks)

VEINS Simulador híbrido (Vehicles in Network Simulation)

RESUMEN

El trabajo de titulación tuvo como objetivo evaluar los protocolos de encaminamiento en redes VANET conectados mediante 802.11p considerando los modelos de movilidad sintéticos y generadores de tráfico vehicular. En base a la recopilación de información de modelos de movilidad sintéticos y generadores de tráfico se seleccionan el modelo Manhattan e IDM respectivamente. Se instalaron las siguientes plataformas en el sistema operativo Ubuntu 12.04: SUMO para el tráfico vehicular, NS-2 para la red de datos y MOVE para exportar la información de SUMO a NS-2. Se hizo uso de la clasificación de los protocolos de enrutamiento proactivos y reactivos, para posteriormente aplicar los protocolos DSDV y AOMDV ya que al realizar la comparación estos fueron los que evidenciaron mejores características. En los escenarios de simulación se aplica un número bajo, medio y alto de conexiones con dos tipos de comunicación (V2V, V2I) para redes VANET. Los indicadores que se analizaron para determinar el rendimiento de los protocolos fueron Throughput, PDF, Average End to End Delay y NRL. Se determinó que para comunicaciones V2V, independientemente de las conexiones o modelo de movilidad se puede emplear AOMDV o DSDV ya que la diferencia es mínima siendo AOMDV superior y para comunicaciones V2I el mejor protocolo es AOMDV con una superioridad ante DSDV en la mayoría de casos. Se concluye que el modelo más cercano a la realidad es IDM ya que está basado en un generador de tráfico, el modelo Manhattan al ser sintético y estar basado en fórmulas matemáticas ofrece resultados ambiguos. Al usar MOVE se recomienda compilar el archivo .sumo.tr para exportar los datos a NS-2 sin inconvenientes y establecer las conexiones de los nodos por líneas de comando en el archivo .tcl para optimizar el tiempo de configuración.

PALABRAS CLAVE: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <REDES>, <VEHICULAR AD-HOC NETWORK (VANET)>, <PROTOCOLOS DE ENRUTAMIENTO>, <INTELLIGENT DRIVER MODEL (IDM)>, <SIMULATION OF URBAN MOBILITY (SUMO)>, <MODEL GENERATOR FOR VEHICULAR NETWORKS (MOVE)>, <DESTINATION SEQUENCED DISTANCE VECTOR (DSDV)>, <AD-HOC ON-DEMAND MULTIPATH DISTANCE VECTOR ROUTING (AOMDV)>

SUMMARY

This research work aimed to evaluate the routing protocols in VANET networks connected through 802.11 p, by considering the synthetic mobility models and vehicular traffic generators. Based on the data collection, the Manhattan and IDM models were selected, thus several platforms were installed as well at the OS Ubuntu 12.04: SUMO for the vehicular traffic, NS-2 for data network and MOVE for exporting the SUMO information to NS-2. Some routing agreements were used such as reactive and proactive in order to apply the DSDV and AOMDV protocols, hence by making a comparison these ones were which shown the best features. At the simulation sceneries, a low, medium and high connection numbers were applied, with two kinds of communication (V2V, V2I) for VANET networks. The indicators used for determining the protocols performance were Throughput, PDF, Average End to End Delay and NRL. For V2V communications regardless of connections or mobility models, AOMDV or DSDV protocols can be easily used because there is a minimal difference, being AOMDV a bit higher than DSDV and suitable for V2I communications. As conclusion, the most real model is IDM due to it is based on a traffic generator, the Manhattan model offers ambiguous results because it is synthetic and uses mathematical formulas. When using MOVE, it is recommendable to compile the file .sumo.tr in order to export data to NS-2 without drawbacks and establishing the nodes connections by means of command designs at .tcl file to optimize the setting timing.

KEYWORDS: <ENGINEERING SCIENCES AND TECHNOLOGY> <NETWORKS>
<VEHICULAR AD-HOC NETWORK (VANET)> <ROUTING PROTOCOLS>
<INTELLIGENT DRIVER MODEL (IDM)> <SIMULATION OF URBAN MOBILITY (SUMO)>
<MODEL GENERATOR FOR VEHICULAR NETWORKS (MOVE)>
<DESTINATION SEQUENCED DISTANCE VECTOR (DSDV)> <AD-HOC ON-DEMAND MULTIPATH DISTANCE VECTOR ROUTING (AOMDV)>

INTRODUCCIÓN

Las mejoras en las tecnologías existentes han proporcionado el funcionamiento de nuevos tipos de redes que trabajan en diferentes rangos de frecuencias que no se encuentran congestionadas, permitiendo un mejor rendimiento en el envío de datos, además que también se optimiza la velocidad de transmisión y reducción en los costos de implementación por la oferta que existe en el mercado.

Una de estas mejoras se está aplicando al entorno del transporte terrestre para salvar vidas humanas mediante la transmisión de datos entre vehículos, esto es posible gracias a los protocolos de enrutamiento, los mismos que deben tener una capacidad adaptativa rápida y eficiente frente a los cambios topológicos que se susciten dentro de la red, al mismo tiempo tienen que utilizar la mínima cantidad de memoria, ancho de banda y potencia de transmisión para el estudio de este anteproyecto se ha realizado la revisión de las investigaciones llevadas a cabo a lo largo de los últimos años.

En la ciudad de Valencia, España se realizó el trabajo investigativo titulado “Configuración Óptima del Protocolo de Encaminamiento OLSR para VANETs Mediante Evolución Diferencial” publicado por Jamal Toutouh, José M. García-Nieto y Enrique Alba en el que se plantea que el encaminamiento en una red VANET debe ser eficiente para garantizar la calidad de servicio y se busca una configuración eficiente del protocolo OLSR para VANET basado en la tasa de envío de paquetes, carga de gestión de encaminamiento y retardo medio punto a punto. (Toutouh, Garcia-Nieto y Alba 2010, p. 463-471)

Un trabajo similar, realizado en Sudamérica es el elaborado por Jiménez Pinto Gonzalo, Danilo López y Luis Pedraza en el 2016 en la ciudad de Bogotá, Colombia, mismo que se titula “Simulación y análisis de desempeño de protocolos unicast para Redes VANET”. En esta publicación, se hace énfasis en los desafíos tecnológicos que representa el enrutamiento en las redes VANET, esto se ocasiona al estar sus nodos en constante movimiento y lo cual conlleva que la topología de red cambie constantemente por lo que se deben implementar simulaciones en software de red para encontrar los protocolos de enrutamiento y los modelos de movilidad que ofrezcan soluciones a estos inconvenientes a la hora de intercambiar paquetes entre los nodos que conforman la red. (Jiménez Pinto, López y Pedraza 2012, p. 66-75)

En el Ecuador, una de las últimas tesis publicadas a cerca de redes VANET fue realizada por Estalin Fuentes y Andrés Valencia estudiantes de la Escuela Superior Politécnica de Chimborazo que lleva como tema “Estudio de las VANET para Control de los Vehículos en las Carreteras en el Ecuador”, este documento de titulación, da una descripción y comparación de algunos algoritmos de enrutamiento reactivo y proactivo utilizados en las redes VANET, además, hace uso del modelo de movilidad basado en simulador de tráfico. (Fuentes Salazar y Valencia Enríquez 2015, p. 30-52)

FORMULACIÓN DEL PROBLEMA

¿Qué protocolo de encaminamiento es más eficiente para una red VANET considerando el estándar IEEE 802.11p, así como los modelos de movilidad sintéticos y generadores de tráfico vehicular?

JUSTIFICACIÓN

JUSTIFICACIÓN TEÓRICA

El avance tecnológico en el ámbito de las redes inalámbricas en la última década lograron que se desarrollen nuevas tecnologías que se aplican a todos los ámbitos de la vida cotidiana, una de las zonas en la que más necesitamos ayuda de la tecnología es en el campo del transporte terrestre, mediante la simulación de una red VANET y varios protocolos de enrutamiento, podremos obtener datos sobre el rendimiento y eficiencia de dichos protocolos, hecho que permitirá emitir un criterio acerca de cuál es el protocolo más eficiente y que mejor se ajusta a las características especiales que presentan las redes VANET.

Siendo que el enrutamiento es una función importante en una red de comunicaciones ya que es la encargada de determinar la mejor ruta que seguirán los paquetes desde su origen hasta su destino, garantizando un efectivo envío de la información. En el contexto de esta realidad, cabe señalar que, para entornos móviles, esta importante función de red, se vuelve una tarea compleja debido al dinamismo que produce el cambio rápido de posición de los nodos de la red, así como la variabilidad del canal inalámbrico y la fluctuación de los enlaces (Guevara, Chavez y Calderon 2015, p. 23-30). Más aún, esta tarea se complica si consideramos las características del entorno

en el que se movilizan los vehículos, mismos que deben respetar un conjunto de reglas (velocidad, señalética, seguridad, etc.) que determinan el comportamiento de cada vehículo. En este sentido, los modelos de movilidad, se presentan como un complemento de las redes VANET que se encarga de describir cómo tiene que desplazarse un vehículo y qué decisiones tiene que tomar. (Calvo et al. 2011, p. 1-22)

La investigación propuesta, se realiza ya que existe la necesidad de determinar cuál de los protocolos de enrutamiento para redes VANET existentes, presenta una mejor capacidad adaptativa eficiente y rápida ante los cambios en la topología de la red, además, utilizando el mínimo ancho de banda, optimizando el uso de memoria, y potencia de transmisión, considerando además los modelos de movilidad sintéticos y basados en simuladores de tráfico, utilizando para ello simulaciones que permitan obtener métricas para comparar el rendimiento de los protocolos de enrutamiento para VANET más utilizados.

JUSTIFICACIÓN APLICATIVA

Las personas están propensas a cometer errores, además, según la (Dirección General de Tránsito de España 2009, p. 1-20), es fácil distraerse mientras se conduce un vehículo ya sea porque se escucha música a alto volumen, por usar el celular, llevar bebidas y comida dentro del auto, estar bajo estrés, y un sin número de motivos que hacen que sea fácil perder la concentración al estar conduciendo, este hecho ocasiona que se infrinjan las reglamentaciones de tránsito pudiendo llegar a provocar un accidente además, otro problema que se genera en las metrópolis son los embotellamientos en las principales vías de movilización pudiendo evitar más afluencia de vehículos si se comunica al resto de conductores que tomen vías alternativas.

El estudio presentado pretende analizar y comparar, mediante el uso de escenarios simulados, el rendimiento de varios protocolos de enrutamiento en redes VANET como los modelos de movilidad sintéticos y basados en simuladores de tráfico, con el objetivo de determinar cuál de los protocolos estudiados presenta el mejor rendimiento en base a métricas (throughput, tiempo de retardo, paquetes perdidos y otros parámetros), tomadas a partir de cada simulación, en las que se considerarán los modelos de movilidad sintéticos y basados en Simuladores de tráfico. Para las simulaciones se empleará una representación de un entorno urbano, específicamente para un área del sector céntrico de la ciudad de Riobamba, en Ecuador ya que cuenta con una afluencia de tráfico vehicular considerable.

El análisis y comparación de los valores obtenidos para cada protocolo y por cada modelo de movilidad considerado, permitirán determinar cuál de los protocolos analizados se presenta con el mejor rendimiento, los descubrimientos realizados, podrán ser empleados en otros trabajos de investigación y por otras Instituciones Educativas.

OBJETIVOS

OBJETIVOS GENERALES

Evaluar los protocolos de encaminamiento en redes VANET conectados mediante 802.11p considerando los modelos de movilidad sintéticos y generadores de tráfico vehicular.

OBJETIVOS ESPECÍFICOS

- Investigar sobre los diferentes protocolos de enrutamiento existentes para redes VANET y los modelos de movilidad tanto sintéticos como generadores de tráfico vehicular.
- Caracterizar cada uno de los protocolos de enrutamiento existentes para redes VANET y los modelos de movilidad seleccionados.
- Seleccionar la plataforma a utilizar en las simulaciones de la red VANET bajo el estándar IEEE 802.11p.
- Definir en base al Throughput (rendimiento), PDF (relación de entrega de paquetes), Average End to End Delay (retardo promedio) y NRL (carga de enrutamiento normalizada), cual es el protocolo de enrutamiento más eficiente para un entorno de simulación de red VANET.

CAPÍTULO I

1. MARCO TEÓRICO REFERENCIAL

1.1. Protocolo IEEE 802.11

1.1.1. *Introducción*

La base para que surja el estándar IEEE 802.11 se debe a las necesidades de dar solución a los problemas generados en las redes sin que dependan de soluciones propietarias y permitan la interoperabilidad de la red entre componentes de diferentes fabricantes, lo cual, permitió obtener mejores precios y equipos eficientes que se pueden acoplar unos con otros sin depender de la marca de los equipos. (Yunqueira 2005, p. 36-37)

El instituto de ingenieros electrónicos es una organización sin fines de lucro, líder en la promoción de estándares internacionales, el protocolo IEEE 802.11 se considera para ethernet inalámbricas, fue lanzado en 1997 y basa su acceso al medio en el método CSMA/CA, este método de acceso se caracteriza por sacrificar el ancho de banda para lograr conseguir una transmisión de datos más confiable, además, trabaja en la frecuencia libre 2.4Ghz. (Werchau y Nazar 2013, p. 1-12)

El estándar IEEE 802.11 ha recibido diferentes aportes con algunas modificaciones particulares permitiendo aumentar la velocidad de transmisión/recepción, usar diferentes tecnologías de modulación, ancho de banda y número de canales disponibles, según (Tanenbaum y Wetherall 2012, p. 67-69) hoy en día existen varias extensiones del protocolo IEEE 802.11 entre las que se tienen:

- IEEE 802.11 b
- IEEE 802.11 a
- IEEE 802.11 g

- IEEE 802.11 e
- IEEE 802.11 h
- Entre otros

1.1.2. *Capa física (Layer 1)*

Esta capa del modelo OSI conocida típicamente como capa física se encarga de codificar y modular los datos que el emisor va a enviar al receptor. La modulación de los datos juega un papel importante en la transmisión pues de esta depende la posibilidad de enviar una mayor tasa de bits en un mismo ancho de banda. (Werchau y Nazar 2013, p. 6)

Las técnicas más utilizadas son Frequency Hopping Spread Spectrum espectro esparcido por salto de frecuencia (FHSS), Direct Sequence Spread Spectrum espectro esparcido por secuencia directa (DSSS) y Orthogonal Frequency-Division Multiplexing modulación por división de frecuencias ortogonales (OFDM). (Werchau y Nazar 2013, p. 6-7)

Las técnicas basadas en el ensanchamiento del espectro transmiten la señal de información en todo el ancho de banda del cual se dispone y no solo enfoca la energía en una portadora en particular, lo que hace posible distribuir la energía de las señales por todo el ancho de banda del cual se dispone. (Stallings 2004, p. 286-287)

1.1.2.1. *Espectro esparcido por salto de frecuencia (FHSS)*

Si se aplica la técnica del espectro esparcido por salto de frecuencia, lo que ocurre es que en un intervalo de tiempo que debe ser menor a los 400ms, se envía un fragmento de la señal de información a través de una determinada frecuencia. Una vez que la transmisión llega al límite de tiempo la frecuencia se cambia y continúa el proceso de transmisión del resto de la información hasta conseguir transmitir toda la señal. (Stallings 2004, p. 287-292)

Al transmitir mediante FHSS es como que se realizara una fragmentación de la información en el dominio del tiempo (Fig. 1-1) y para poder realizar este proceso se utiliza una portadora de banda

estrecha que va cambiando a medida que se cambia de frecuencia, aun cuando pareciera que las frecuencias son aleatorias no es del todo cierto, pues están basadas en una secuencia pseudoaleatoria que comparte el transmisor con el receptor para sincronizar los saltos de frecuencia. (Stallings 2004, p. 287-292)

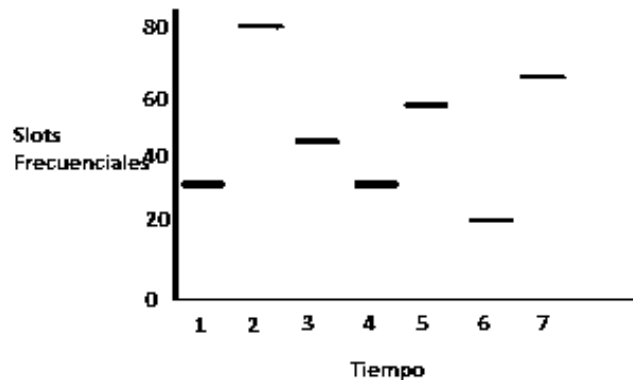


Figura 1-1: Técnica FHSS

Fuente: <http://www.dit.upm.es/~david/tar/trabajos2002/08-802.11-Francisco-Lopez-Ortiz-res.pdf>

Según, (Stallings 2004, p. 287-292) las ventajas que presenta una transmisión mediante la aplicación de la técnica FHSS son:

- Las señales tienen un alto grado de inmunidad frente a interferencias y ruido.
- Presentan gran complejidad de interceptación para los usuarios que no tengan la secuencia que comparte el transmisor al enviar la señal al receptor, con lo que un intruso solo recibiría una parte de la señal la que sería interpretada como un ruido de corta duración.
- Permite que en una misma zona geográfica se pueda tener más de un punto de acceso, solo si las comunicaciones no usan la misma frecuencia portadora al mismo tiempo.

La desventaja que principalmente afecta a una transmisión por FHSS es el alto consumo de ancho de banda que se necesita para enviar las señales de información. (Stallings 2004, p. 287-292)

1.1.2.2. Espectro esparcido por secuencia directa (DSSS)

Este tipo de tecnología opera de forma que genera un patrón redundante de bits para cada bit de información y cuan mayor sea el patrón mayor es la resistencia a interferencias, dicho patrón recibe el nombre de *señal de chip*, mientras que para la modulación de la señal de salida se emplea una portadora de radio frecuencia(RF). (Stallings 2004, p. 292-296)

En la modulación de los bits de información se emplea la secuencia de Barker o también conocida como código de pseudoruido, DSSS permite que el receptor reconstruya la señal de información aun cuando parte de la señal que recepto este afectada por interferencia, la velocidad de transferencia hasta la que puede llegar es de 2Mbps dependiendo de la modulación que se emplee. (Stallings 2004, p. 292-296)

Se puede apreciar en la Fig. 2-1 como es el proceso de codificación de los bits de información mediante la secuencia de Barker +1, -1, +1, +1, -1, +1, +1, -1, -1, -1.

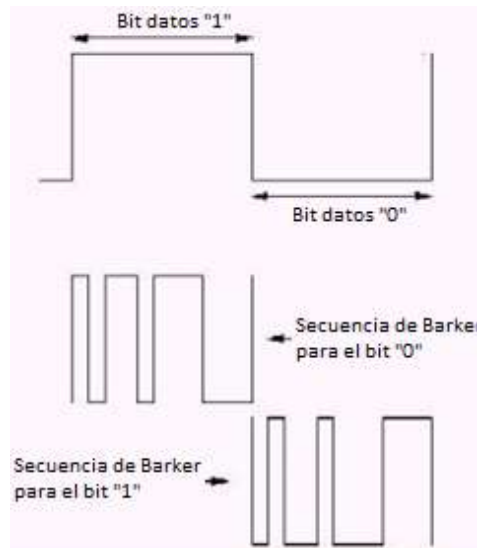


Figura 2-1: Codificación de información con secuencia de Barker

Fuente: <http://www.dit.upm.es/~david/tar/trabajos2002/08-802.11-Francisco-Lopez-Ortiz-res.pdf>

En Europa y Estados Unidos la frecuencia de operación de la tecnología DSSS es desde 2.4Ghz a 2.4835Ghz, posee un ancho de banda de 83.5Mhz que se encuentran divididos en 14 canales,

cada canal posee un ancho de banda de 5Mhz. El uso del subconjunto de canales depende de las respectivas regulaciones de los países en los que se van a utilizar. (Stallings 2004, p. 292-296)

1.1.2.3. *Multiplexación por División de Frecuencia Ortogonal (OFDM)*

La técnica *OFDM* (multiplexación por división de frecuencia ortogonal) es un caso particular de transmisión multicarrier, misma que consiste en enviar los datos a través de varias sub portadoras con menor tasa de tal manera que si existe interferencia no se perderá todo en enlace. Principalmente se utiliza *OFDM* ya que soluciona el problema de desvanecimiento por interferencia de banda estrecha o frecuencia selectiva. (Prasad 2004, p. 11-13)

En la técnica *OFDM* la palabra ortogonal nos indica que hay una relación matemática precisa entre las frecuencias portadoras (52 en total, 48 para datos y 4 de sincronización), además, es posible organizar las portadoras provocando que las bandas laterales se superpongan y así seguir recibiendo los datos sin la interferencia de la portadora adyacente y el receptor ejerce como un banco de demoduladores. (Prasad 2004, p. 11-13)

1.1.3. *Capa de enlace de datos (Layer 2)*

La capa enlace de datos está compuesta por dos subcapas:

- Control de acceso al medio (MAC)
- Control lógico del enlace (LLC)

La subcapa MAC presenta un pequeño cambio para adecuarla a los medios inalámbricos ya que emplea un acuse de recibo de tramas llamado ACK para acceder al medio y reducir las colisiones. (Werchau y Nazar 2013, p. 8)

1.1.3.1. *Capa de control de acceso al medio (MAC)*

Para el estándar IEEE 802 los métodos de acceso fueron diseñados según el modelo OSI y se encuentra ubicados en la capa física y en la parte inferior de la capa de enlace. La MAC es la

encargada de definir el acceso de las tarjetas de red con el medio físico para intercambiar datos. (Tanenbaum y Wetherall 2012, p. 257-267)

Según (Tanenbaum y Wetherall 2012, p. 257-267), la arquitectura MAC del estándar IEEE 802.11 está formada de dos funciones básicas:

- Función de coordinación puntual (PCF): Enfocado para servicios libres de contienda es decir accesos deterministas. Es 100% compatible con DFC y está enfocado para servicios síncronos que no toleran retardos aleatorios al acceder al medio.
- Función de coordinación distribuida (DCF): En un conjunto básico de servicios (BSS) determina cuando una estación recibe o transmite datos de protocolo a nivel MAC por el medio inalámbrico para servicios de acceso aleatorio o con contienda. Las características que más resaltan del DFC es que utiliza MACA para acceder al medio, además de ACKs para verificar que se entregó la información correctamente y soporta broadcast sin ACKs.

En la Fig. 3-1 se muestra el formato de trama del estándar IEEE 802.11.

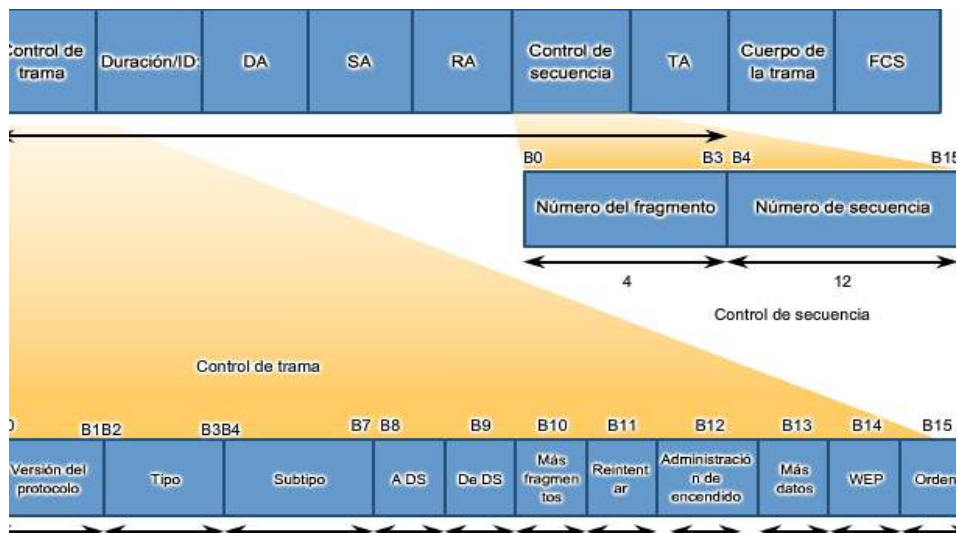


Figura 3-1: Protocolo WLAN de 802.11

Fuente: <http://pantheanet.blogspot.com/2012/01/tema-7-capa-enlace-de-datos.htm>

1.1.3.2. *Capa de control lógico de enlace (LLC)*

La capa LLC es la encargada de controlar el flujo de las tramas e identificar que protocolo de capa de red se está utilizando en la comunicación, de tal modo que proporciona un formato único y una interfaz con la capa de red, con lo que permite la compatibilidad entre todo tipo de red 802. Está compuesta por la mitad superior de la capa de enlace y la subcapa de MAC que se encuentra debajo del LLC. (Tanenbaum y Wetherall 2012, p. 291)

LLC se desempeña de la siguiente manera. El emisor en su capa de red envía un paquete al LLC, luego la subcapa LLC adiciona un header LLC que tiene la información de los números de secuencia y confirmación de recepción, esta estructura es introducida en el campo de carga útil de la trama 802 y es transmitida, mientras que el receptor realiza el proceso inverso. (Tanenbaum y Wetherall 2012, p. 291)

1.1.4. *Protocolo CSMA/CA y MACA*

CSMA/CA y MACA son protocolos de acceso al medio que se basan en un algoritmo de características similares a las implementadas en el estándar IEEE 802.3, además que se encargan de las retransmisiones de tramas. (Tanenbaum y Wetherall 2012, p. 257-267)

1.1.4.1. *CSMA/CA*

Según (Tanenbaum y Wetherall 2012, p. 257-267), el algoritmo CSMA/CA desempeña sus funciones de acuerdo a la siguiente descripción:

- 1) Como primer paso debe determinar el estado del medio, es decir, si se encuentra ocupado o libre para de esta manera proceder a la transmisión de la información.
- 2) En el caso de que el medio este libre y no lo tenga en uso ninguna trama la estación procede a ejecutar una espera adicional que se conoce como *IFS (espaciado entre tramas)*.

- 3) Si desde el inicio o durante el intervalo de espera adicional, se determina que el medio se encuentra en uso, la estación tiene que esperar hasta que se termine dicha transacción y luego puede proceder a ejecutar cualquier otra acción.

Al terminar la espera del medio que se encontraba ocupado, la estación se encarga de ejecutar el algoritmo de *Backoff*, este algoritmo ingresa una espera aleatoria que es escogida uniformemente dentro del intervalo *CW* (ventana de contienda). El conjunto de slot time (ranuras temporales) son los encargados de disminuir la probabilidad de colisión, misma que es máxima cuando están varias estaciones esperando a que se libere el medio, el número de slots son enteros aleatorios y son generados por el algoritmo de Backoff. (Rodríguez Avalos y Tucto Succhil 2014, p. 24-25)

Mientras está en ejecución la espera ingresada por el algoritmo de Backoff se sigue testeando el medio y si se determina que se encuentra libre por al menos un espacio entre tramas, la espera sigue avanzando temporalmente hasta que son consumidos por parte de la estación todos los slots que se asignaron. Por lo contrario, si el medio se determina en uso y no está libre por al menos un tiempo igual al IFS se suspende el algoritmo de Backoff hasta cumplir con dicha condición. (Rodríguez Avalos y Tucto Succhil 2014, p. 24-25)

1.1.4.2. *MACA*

Según (Rodríguez Avalos y Tucto Succhil 2014, p. 24-25), al aplicar el protocolo CSMA/CA en un ambiente celular e inalámbrico se presentan dos principales problemas que se describirán a continuación:

- **Nodo oculto:** Se presenta cuando una estación determina que se encuentra libre el canal, pero es el caso contrario, lo que sucede es que no logra oír al nodo que está haciendo uso del canal.
- **Nodo expuesto:** Sucede cuando la estación determina que el canal está en uso, pero se encuentra libre porque el nodo que oye no le interferiría para transmitir a otro destino.

El estándar IEEE 802.11 propone como solución, el uso del protocolo MACA, el mismo consiste en que antes de comenzar la transmisión se envía una trama *RTS* (Request to Send) desde el transmisor al receptor, este último responde con una trama *CTS* (Clear to Send). Tanto la trama

RTS como la CTS indican la longitud de los datos. El emisor al recibir el CTS empieza el envío de datos. (Rodríguez Avalos y Tucto Succhil 2014, p. 24-25)

Los nodos siguen las siguientes normas para evitar nodos ocultos y expuestos:

- Después de escuchar un RTS, tiene que esperar un tiempo por la trama CTS.
- Después de escuchar un CTS, tiene que esperar según la longitud de los datos que se envían.

La solución aplicada en IEEE 802.11 es el uso de MACA y CSMA/CA para el envío de las tramas RTS y CTS. (Rodríguez Avalos y Tucto Succhil 2014, p. 24-25)

1.2. Enmiendas IEEE 802.11

El estándar 802.11 para redes LAN inalámbricas tiene varias enmiendas que se diferencian por las técnicas de modulación, la frecuencia a la que operan, velocidad y rango de cobertura que alcanzan, este protocolo cubre tanto la capa física (L1) como la capa de enlace de datos (L2) del modelo OSI (Open Systems Interconnection). (Tanenbaum y Wetherall 2012, p. 257-267)

A continuación, se describen algunas evoluciones de IEEE 802.11 que se observan en la Fig. 4-1 para apreciar las mejoras que se dieron en las diferentes enmiendas del protocolo con el paso del tiempo.

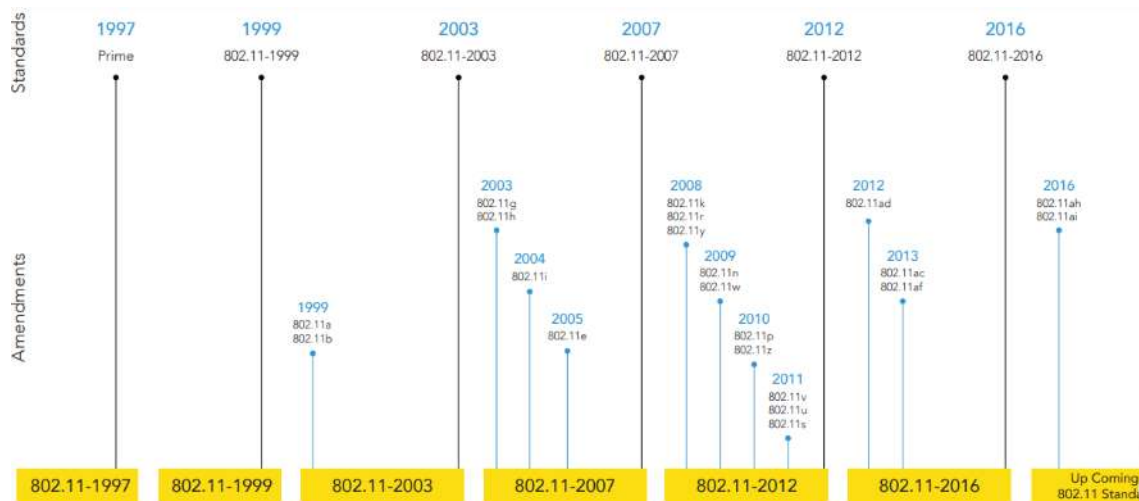


Figura 4-1: Evolución de IEEE 802.11

Fuente: http://grouper.ieee.org/groups/802/11/Reports/802.11_Timelines.htm

1.2.1. *Enmienda 802.11b*

El estándar IEEE 802.11b se aprobó en 1999 y también se lo conoce como Wi-Fi, el mismo consta de un ancho de banda de canal de 20Mhz, tasa de transferencia de 11Mbps que se ve reducida al momento de presentarse atenuaciones en el canal o interferencias provocando una caída a 5.5Mbps, 2Mbps y llega hasta 1Mbps. Trabaja en la frecuencia de 2.4 Ghz. Se considera como uno de los mejores estándares para la aplicación en interiores. (Tanenbaum y Wetherall 2012, p. 257-267)

Usa de manera exclusiva la técnica DSSS con codificación CCK y el principal problema que presenta es la interferencia que se ocasionan debido a que varios estándares operan bajo la frecuencia de 2.4Ghz. (Tanenbaum y Wetherall 2012, p. 257-267)

1.2.2. *Enmienda 802.11a*

IEEE 802.11a también llamado Wi-Fi5 fue publicado en 1999 y tiene un costo relativamente alto, el mismo hace uso de las 52 portadoras con las que cuenta la técnica de acceso al medio OFDM, tasa de transferencia máxima de 54Mbps, pero también funcionan a 48, 36, 24, 18,12 y 6Mbps. (Tanenbaum y Wetherall 2012, p. 257-267)

Trabaja en la frecuencia de 5Ghz lo que provoca que su capacidad de cobertura sea la mitad del IEEE 802.11b ya que su banda de frecuencia es un poco más del doble, además que al usar una frecuencia más alta necesita mayor capacidad de energía para la transmisión. El estándar presenta problemas al tratar de implementarse tanto en Japón y algunos países europeos debido al tema de licencias. Cabe recalcar que el estándar IEEE 802.11a no es compatible con 802.11b. (Tanenbaum y Wetherall 2012, p. 257-267)

1.2.3. *Enmienda 802.11g*

El estándar IEEE 802.11g se considera como la evolución del 802.11b y se ratificó en junio del 2003, opera en la banda de 2.4Ghz y puede alcanzar tasas de transferencia de 54, 48, 36, 24, 18, 11, 5.5, 2 o 1Mbps dependiendo de la técnica de acceso al medio que emplee (DSSS para llegar hasta los 11Mbps y OFDM para velocidades superiores a 11Mbps). (Yunqueira 2005, p. 40-43)

Es compatible con 802.11 y 802.11b permitiendo usar la misma infraestructura de esta última, se considera que llegará a ser la competencia directa del estándar 802.11a pero que al igual que 802.11b sufrirá interferencias a operar en la banda de 2,4Ghz. (Tanenbaum y Wetherall 2012, p. 257-267)

1.2.4. *Enmienda 802.11n*

El estándar IEEE 802.11n fue estructurado en base a normas anteriores de la familia 802.11 mejorando la confiabilidad, velocidad y rendimiento en comparación a 802.11 a/b/g logrando llegar aproximadamente hasta los 600Mbps, se aprobó en septiembre del 2009. (Lara Cueva, Fernández Jiménez y Morales Maldonado 2016, p. 3-5)

IEEE 802.11n es compatible con equipamiento 802.11 a/b/g, ofrece poco consumo de energía y trabaja con modulación OFDM contando con 127 tasas de transferencia diferentes. La alta tasa de transferencia se logró añadiendo características de múltiple entrada múltiple salida (MIMO), el conjunto de antenas MIMO provoca una transmisión multi trayectoria alcanzando los 600Mbps. (Lara Cueva, Fernández Jiménez y Morales Maldonado 2016, p. 3-5)

1.3. Redes VANET

1.3.1. Introducción

Desde hace varios años, las redes inalámbricas han evolucionado para ofrecer varios servicios a los usuarios, satisfaciendo necesidades en distintos campos de aplicación, ampliando al mismo tiempo las opciones de comunicación con las que este tipo de redes pueden funcionar. De entre estas aplicaciones, la comunicación entre vehículos ha despertado el interés entre investigadores y desarrolladores de todo el mundo. (Guevara, Chavez y Calderon 2015, p. 23-30)

El interés se debe a que las denominadas Redes Ad hoc Vehiculares (VANET, de su acrónimo en inglés Vehicular Ad hoc Network), al permitir la comunicación entre vehículos en movimiento, proporcionan acceso a información referente a los vehículos cercanos (nodos) e intercambian información con las infraestructuras que se encuentran junto a las vías. (Guevara, Chavez y Calderon 2015, p. 23-30)

Esta tecnología ha logrado un importante reconocimiento debido en gran medida a que ofrecen una solución a los problemas de seguridad y tránsito vehicular, sin representar un incremento considerable en el precio final de los vehículos (nodos). Además, al no necesitar un enrutador centralizado, la flexibilidad de la red se adapta a las necesidades de los nodos que la conforman. (Hartenstein y Laberteaux 2010, p. 1-5)

Lo que permite que los sensores incorporados en el vehículo puedan intercambiar la información directamente de nodo a nodo o con la infraestructura Road-Side Unit (RSU) que se encuentra a los lados de las vías. Investigaciones revelan que estas características de las redes VANET han permitido reducir los índices de accidentes de tráfico y reducir el congestionamiento vehicular en las ciudades y metrópolis donde se implementaron. (Hartenstein y Laberteaux 2010, p. 1-5)

1.3.2. ¿Qué es una red VANET?

Una red VANET es un tipo de red móvil ad hoc que se considera como un caso especial de red ad hoc móvil (MANET). En la Fig. 5-1 se aprecia su arquitectura, misma que se enfocan en la comunicación de redes conformadas por vehículos. (Hartenstein y Laberteaux 2010, p. 1-5)

La red VANET permite la comunicación entre los nodos que conforman la red, permitiendo alertar a los conductores de cualquier incidente suscitado en las carreteras o advirtiéndoles una posible colisión del automóvil por imprudencia del conductor, así como diferentes aplicaciones que mejoran el transporte terrestre y ayudara salvar vidas humanas. (Hartenstein y Laberteaux 2010, p. 1-5)

Los tipos de comunicación que nos ofrece la tecnología VANET son vehículo-vehículo (V2V) y vehículo-infraestructura (V2I) cada uno con sus características. La tecnología VANET trabaja bajo el estándar IEEE 802.11p el cual es una mejora del IEEE 802.11a; se diferencia del tradicional IEEE 802.11 ya que se adapta rápidamente a los cambios de la topología de la red ocasionados por la movilidad que poseen sus nodos. (Fuentes Salazar y Valencia Enríquez 2015, p. 40)

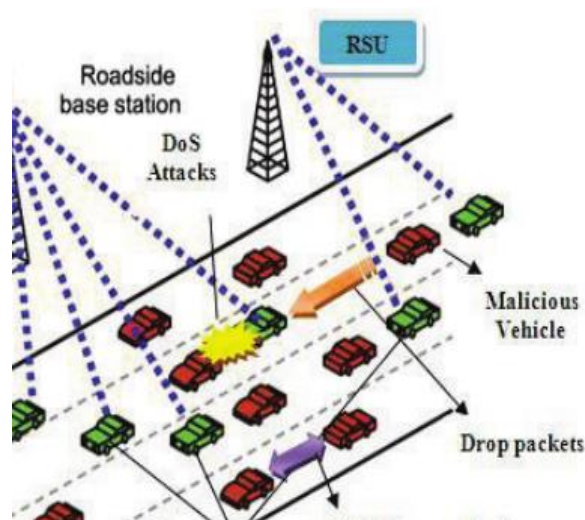


Figura 5-1: Arquitectura de una red VANET

Fuente: www.sciencedirect.com/science/article/pii/S0140366414000863

IEEE 802.11p trabaja a una frecuencia de 5.9 GHz la cual no tiene un gran número de equipos trabajando en esa frecuencia y así evita interferencia lo cual sería un problema considerable si

funcionara en el espectro de 2.4 GHz, las redes VANET al trabajar en una frecuencia tan alta se ve afectada por el alcance de su comunicación, debiendo recordar que a mayor frecuencia menor es el alcance de la comunicación de los equipos, es decir, son inversamente proporcionales. (Ruiz et al. 2013, p. 77)

1.3.3. *Características de una VANET*

Según (Mahmoud Al-Qutayri, Chan Yeun 2010, p. 251-264), las principales características con las que cuenta una red VANET son:

- La naturaleza de la comunicación: La comunicación en una VANET puede ser distribuida al pasarse información entre nodos o centralizada en donde un nodo posee mayor nivel de control.
- Autonomía: Cada nodo es considerado independiente ya que cuenta con la capacidad de procesar la información que recibe.
- Movilidad y naturaleza dinámica: Al ser un caso especial de MANET heredan la característica de movilidad. Los nodos en una VANET se desplazan (excepto las RSU) en diferentes direcciones y con velocidades diferentes pero estos parámetros se encuentran ligados a la carretera en la que se desplazan.
- Intercambio frecuente de información: Al estar conformada por nodos móviles, un gran número de nodos intercambian información con otros.
- Auto organización y procesamiento en tiempo real: Debe adaptarse a los cambios topológicos que presente la red y reorganizarse para establecer las comunicaciones.
- Sin infraestructura: Entorno totalmente inalámbrico.

1.3.4. *Tipos de comunicación de una VANET*

Según el artículo (Orozco, Chavarro y Calderón 2014, p. 62-70), una red VANET básicamente tiene dos tipos de comunicación que son V2V y V2I, pero para llevarlos a cabo es necesario que existan ciertos componentes, mismos que se detallan a continuación:

- On Board Unit (OBU): Se ubican en el interior del vehículo y es el dispositivo que permiten la comunicación con los demás nodos y unidades al borde de las carreteras.
- Application Unit (AU): La unidad de aplicación permite al usuario la visualización de la información.
- Road-Side Unit (RSU): Se ubican en los bordes de las carreteras y permiten alcanzar nodos que no se encuentren directamente en el rango de cobertura.

1.3.4.1. *Comunicación Vehículo – Vehículo (V2V)*

En este tipo de comunicación se encuentra integrado un módulo de comunicación de radio de corto alcance en el automóvil junto con la unidad de aplicación (AU) permitiendo el intercambio de información entre OBUs de diferentes vehículos, e intercambian información de la ubicación y posición de los mismos. (Caiza Changoluisa 2017, p. 11-13)

En la comunicación V2V se reciben alertas gracias a las señales de los sensores con los que están equipados los vehículos permitiendo evitar choques ocasionados por los puntos ciegos del conductor, o por un frenado brusco por parte del vehículo que se encuentra delante. (Caiza Changoluisa 2017, p. 11-13)

Además, se puede intercambiar información sobre el estado de las carreteras por si sucede algún desastre natural o en zonas urbanas nos ayuda a disminuir la congestión de tráfico en las carreteras con mayor afluencia ya que los datos que intercambian los vehículos son en tiempo real. (Caiza Changoluisa 2017, p. 11-13)

La comunicación de vehículo a vehículo se logra gracias a la antena omnidireccional que posee la OBU, la conexión es directa si estos nodos se encuentran en su rango de cobertura, pero de no ser el caso simplemente se conecta al nodo más cercano para realizar el envío de la información a su destino por medio de múltiples saltos de información hasta llegar al nodo destino. (Bitam y Mellouk 2014, p. 2)

Cabe recalcar que tanto en una comunicación V2V (Fig. 6-1) como en la V2I la red VANET utiliza el mismo protocolo de enrutamiento para establecer la comunicación entre el emisor y el destino al cual se desea enviar los datos. Algunos de estos protocolos de enrutamiento más

conocidos son: DSDV, DSR, AODV, AOMDV, FSR, TORA, entre otros. (García Campos 2014, p. 31-78)

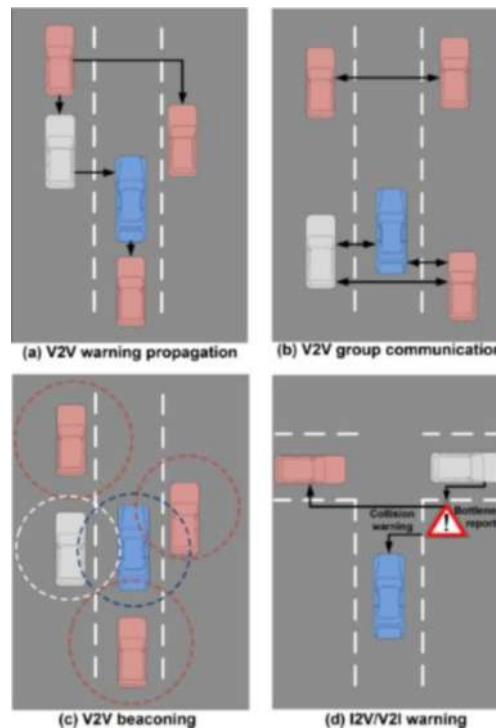


Figura 6-1: Red VANET-Comunicación V2V

Fuente: www.researchgate.net/publication/47410831_Overview_of_Security_Issues_in_Vehicular_Ad-hoc_Networks

1.3.4.2. Comunicación Vehículo – Infraestructura (V2I)

La comunicación se establece entre los vehículos y la infraestructura al borde de las carreteras es decir entre la OBU y la RSU, se aplica esencialmente en casos en los que algunos nodos se encuentren fuera del área de cobertura y no sea posible aplicar saltos entre vehículos para llegar al nodo destino, por lo que se usan las RSUs y se concentra en las aplicaciones de seguridad vial que la comunicación V2V no logra cubrir eficientemente. (Caiza Changoluisa 2017, p. 11-13)

En el esquema de comunicación V2I (Fig. 7-1), resulta más costosa su implementación ya que es necesaria la ubicación de muchos terminales RSU en las carreteras para permitir que los nodos de la red VANET se conecten con la infraestructura, la cual se encarga de negociar las conexiones de los nodos y puede permitir el acceso a internet y más servicios adicionales con redes externas a la VANET. (Al-Doori 2011, p. 87)

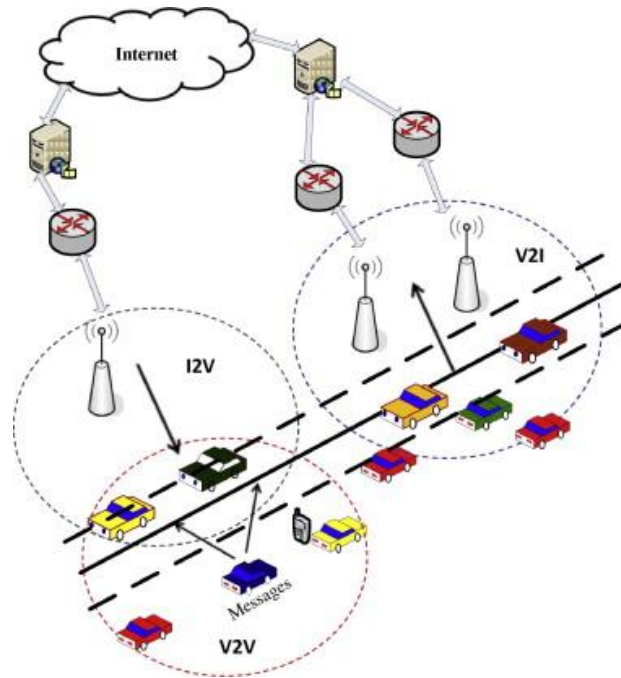


Figura 7-1: Red VANET-Comunicación V2I-I2V

Fuente: www.researchgate.net/figure/261156564_fig1_Figure-1-DoS-attacks-on-a-VANET-infrastructure-Most-existing-defense-schemes-against-IP

1.3.5. *Estándar IEEE 802.11p*

En el año de 2002 la Sociedad Americana de Transporte Terrestre (*ITSA*, Intelligent Transportation Society of America) realizó una recomendación para establecer un estándar para el entorno de las redes VANET, dicho estándar es la modificación del IEEE 802.11a y cuenta con una ligera modificación en la capa MAC, el resultado de la modificación dio como resultado el estándar conocido como IEEE 802.11p. (Ruiz et al. 2013, p. 71-93)

IEEE 802.11p posee características como un bajo overhead y un uso más eficiente del espectro, además se eliminó el intercambio de tramas para poder establecer el canal por lo que la comunicación se puede realizar en el instante en el que transmisor y el receptor están en el mismo rango de cobertura. (Ruiz et al. 2013, p. 71-93)

Además, cuenta con 7 canales no solapados, 6 destinados para el servicio (SCH) y uno para control (CCH), mismo que hace uso del protocolo de mensajes cortos WAVE (*WSMP*, WAVE Short Message Protocol) para disminuir el tamaño de cabecera en los paquetes y ofrecer eficiencia en las transmisiones, las comunicaciones de PHY se basan en OFDM con tasas de transmisión de

entre 3-27 Mbps, se desempeña en la banda de 5.9Ghz con canales de 10Mhz. (Caiza Changoluisa 2017, p. 15-17)

También cuenta con 64 subportadoras y puede utilizar modulaciones BPSK, QPSK, 16-QAM, o 64-QAM. Emplea 52 sub portadoras para transmisión de las que 48 se destinan para datos y 4 se emplean para el rastreo los ruidos de fase y desplazamiento de frecuencia. (Caiza Changoluisa 2017, p. 15-17)

En la Tabla 1-1 se muestra la información sobre el canal inalámbrico V2X del protocolo IEEE 802.11p.

Tabla 1-1: Información del canal inalámbrico V2X

Indicative wireless data link characteristic	802.11p WAVE
Bit rate	3 - 27 Mbps
Communication range	< 1000 m
Transmission power for mobile (maximum)	760 mW (US) 2 W (EU)
Channel bandwidth	10 Mhz
Allocated spectrum	75 Mhz (US) 30 Mhz (EU)
Frecuency bands	5.86 -5.93 Ghz

Fuente: (Emmelmann, y otros, 2010)

1.4. Clasificación de protocolos de enrutamiento para VANET

1.4.1. *Protocolos de Enrutamiento*

Al tener la necesidad de ser adaptables a los continuos y repentinos cambios en la topología de red se hace necesaria la utilización de protocolos de enrutamiento eficientes que puedan adaptarse a diferentes ambientes y que, además, consuman la menor cantidad de recursos de la red MANET. (Tanenbaum y Wetherall 2012, p. 375-376)

También llamados protocolos de ruteo, como su nombre lo indica son los que tienen que cumplir con la tarea de organizar la ruta más confiable y eficiente por la cual se van a transmitir los datos desde un nodo origen a un nodo destino. La selección de estos caminos o rutas tienen que garantizar un consumo mínimo del ancho de banda y una sobrecarga mínima en el header de los paquetes de control. (García Campos 2014, p. 17)

En la Fig. 8-1 se representan algunas de las clasificaciones más empleadas para los protocolos de ruteo en redes MANET.

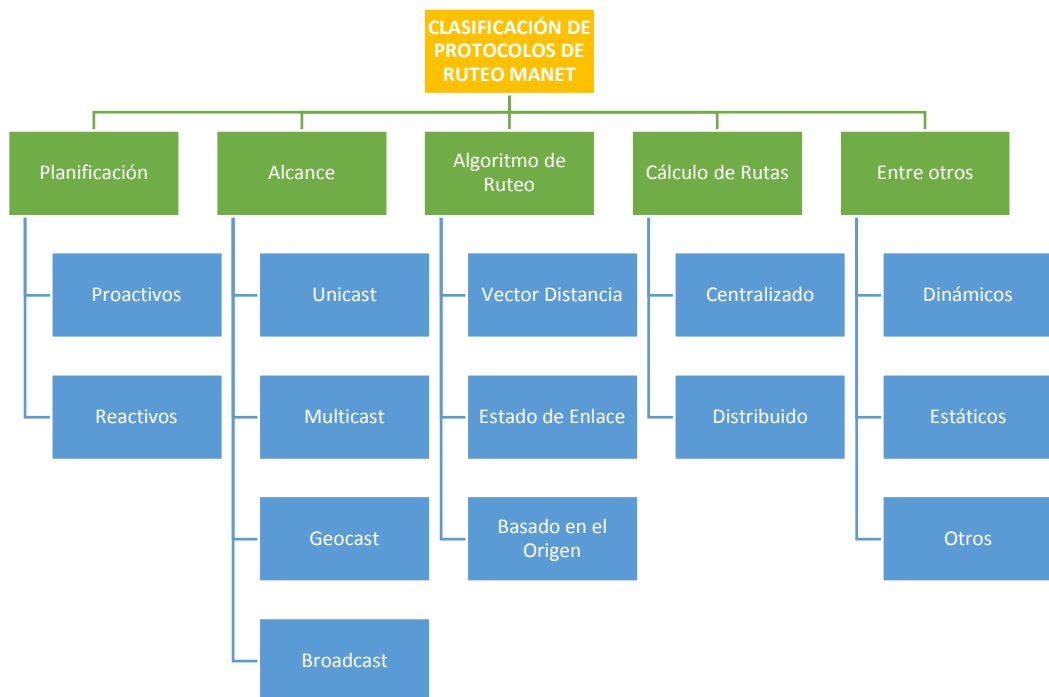


Figura 8-1: Clasificación de protocolos de enrutamiento MANET

Realizado por: Tierra Danni, 2018.

1.4.2. *Según la planificación*

Este tipo de clasificación es el más empleado en los documentos que tratan temas de red MANET y toma como criterio la manera en la que los protocolos realizan el cálculo de la ruta origen-destino. (Albero 2012, p. 24-27)

1.4.2.1. *Protocolos Reactivos*

Realizan el proceso de descubrimiento de ruta solo cuando el nodo origen realiza la petición, por tal motivo también se conocen como protocolos iniciados en el origen bajo demanda y esto permite que el tiempo de entrega de paquetes sea menor ya que el descubrimiento de la ruta se lo realiza solo para determinado nodo. Establecen una ruta y la mantienen activa durante el tiempo que la mantienen en uso o hasta que sea inalcanzable el destino para luego liberar dicha ruta. (Albero 2012, p. 24-27)

En los protocolos reactivos el primer paquete tiene una latencia elevada y sus rutas tienen cierta independencia es decir que, si la topología de la red cambia, una ruta individual se mantiene por cierto tiempo, además utiliza los algoritmos de enrutamiento vector distancia y encaminamiento en origen. (Albero 2012, p. 24-27)

1.4.2.2. *Protocolos Proactivos*

Estos protocolos siempre mantienen actualizadas sus tablas de rutas por lo que provocan un mayor consumo de ancho de banda en la red al intercambiar mensajes de actualización de tablas entre nodos. Las propagaciones de rutas se realizan de manera proactiva desde cada nodo que conforma la red hacia el resto de nodos. (Albero 2012, p. 24)

Mediante el uso de los protocolos proactivos cada uno de los nodos puede tener una o más tablas con la información que les permite enrutar los datos a cualquier nodo que forme parte de la red. Su aplicación es recomendada en redes en las cuales el proceso de descubrimiento de ruta no presente una latencia excesiva y al mismo tiempo disponga de un ancho de banda elevado y gran capacidad de energía. (Albero 2012, p. 26)

1.4.3. *Según el alcance*

Toma en consideración los parámetros que indican a que nodos están dirigidos los paquetes, no siempre se enviara la solicitud para establecer una ruta a todos los nodos, por lo que se pueden aplicar mecanismos que permiten restringir el grupo de nodos destino que recibirán dichas solicitudes. (Tanenbaum y Wetherall 2012, p. 369)

1.4.3.1. *Unicast*

Este alcance hace referencia a una comunicación en la que simplemente intervienen un nodo origen y un nodo destino. Es el modo de alcance más común que se establecen en una red Ad-Hoc. (Tanenbaum y Wetherall 2012, p. 369)

1.4.3.2. *Multicast*

En el caso del método multicast se realiza el envío de datos desde un nodo hacia varios nodos que necesitan la misma información, es decir que se envía la información a un grupo determinado de nodos, al momento de enviar gran cantidad de información nos ofrece optimizar el uso del ancho de banda de la red ya que los datos son enviados una sola vez en la mayor parte de los casos. Al llegar la información por una interfaz de un nodo este se encarga de reenviar la información por el resto de interfaces, exceptuando la interfaz por la que recibió la información. (Tanenbaum y Wetherall 2012, p. 369)

1.4.3.3. *Geocast*

Se considera como un caso especial de multicast porque la información se la envía a todos los nodos que se encuentran organizados de acuerdo a una zona geográfica. Este protocolo permite descubrir las rutas con mayor facilidad porque nos brinda información de la ubicación física de los nodos y con esto se puede saber si algún nodo abandono el grupo o caso contrario se adiciono un nodo. (Tanenbaum y Wetherall 2012, p. 370)

1.4.3.4. *Broadcast*

Provoca un alto consumo de ancho de banda de manera innecesaria si se lo aplica exclusivamente, esto se debe principalmente a que la información es enviada desde un nodo hacia todos los demás. Además, para proceder al envío de los paquetes es necesario que el emisor tenga una lista completa de todos los nodos destino. (Tanenbaum y Wetherall 2012, p. 370)

1.4.4. *Según el algoritmo de ruteo*

1.4.4.1. *Basado en el algoritmo del vector distancia*

Poseen una tabla de información que comparte periódicamente con cada nodo vecino, dicho intercambio de información se lo realiza por difusión, luego de haber recibido la información vuelve a calcular las rutas y actualiza su vector distancia. Este protocolo presenta una convergencia lenta y tiende a generar bucles. (Tanenbaum y Wetherall 2012, p. 357)

Según (Tanenbaum y Wetherall 2012, p. 357), la tabla de información está compuesta por:

- Dirección ip del destino
- Dirección ip del siguiente salto
- Distancia más corta para llegar al siguiente salto

Estos protocolos pueden utilizar como métrica para la distancia: el número de saltos, tiempo de retardo en milisegundos, total de paquetes encolados en la ruta o algo parecido. Cuando la métrica son los saltos, la distancia es un salto; cuando es longitud de cola de paquetes se examinan las colas y si la métrica es el retardo se hace uso de paquetes especiales de eco que el receptor marca con la hora que lo recibe y reenvía tan rápido como le es posible. (Tanenbaum y Wetherall 2012, p. 357)

1.4.4.2. *Basado en el algoritmo de estado de enlace*

Trasmiten paquetes LSP que se encargan de llevar la información de los nodos vecinos de cada nodo que conforma la red para que cada nodo tenga la capacidad de tener la topología global de la red y establece una ruta basado en el costo del enlace. La red es inundada de paquetes LSP con cualquier modificación de enlace que sufra la red. Pueden surgir bucles si la red sufre cambios de topología rápidamente. (Tanenbaum y Wetherall 2012, p. 360-363)

Según (Tanenbaum y Wetherall 2012, p. 363-365), este algoritmo básicamente cumple el siguiente proceso:

- 1) Descubre a sus vecinos y sus direcciones de red mediante el empleo de paquetes HELLO enviados a cada línea punto a punto y recibe una respuesta con la información del nodo del otro extremo.
- 2) Mide el retardo de cada uno de sus vecinos empleando los paquetes ECHO que son enviados por una línea y son devueltos inmediatamente por lo que se mide el tiempo de ida y regreso y se divide para dos, asumiendo que los retardos son simétricos, lo cual no se siempre es cierto.
- 3) Construye un paquete con toda la información recogida en los pasos 1 y 2.
- 4) Envía el paquete a todos los demás nodos.
- 5) Calcula la ruta más corta a todos los demás nodos empleando localmente el algoritmo de Dijkstra.

1.4.4.3. *Basado en el encaminamiento desde el origen*

En la cabecera de todos los paquetes de datos contienen la información de la ruta a seguir para llegar a su destino, esta información es administrada directamente por el nodo origen. No es empleado este protocolo porque genera demasiada sobrecarga en la red. (Albero 2012, p. 25)

1.4.5. *Según el cálculo de rutas*

Según (Albero 2012, p. 26), los protocolos de ruteo dinámico se pueden clasificar en centralizados o distribuidos dependiendo de cuantos nodos toman la decisión de ruteo de la información.

1.4.5.1. *Centralizado*

El nodo central se encarga de tomar las decisiones para el enrutamiento de los paquetes desde su origen hasta su destino, solo en nodo central designa la ruta a utilizar. (Albero 2012, p. 26)

1.4.5.2. *Distribuido*

Las decisiones del enrutamiento se llevan a cabo por todos sus nodos, pero de manera independiente, es decir cada nodo toma su decisión y contribuye a la designación de la ruta más óptima para utilizar. (Albero 2012, p. 26)

1.5. **Protocolos de enrutamiento para redes VANET**

Uno de los desafíos en el estudio de los protocolos de enrutamiento en redes VANET es la adaptabilidad al constante cambio topológico debido al movimiento, puesto que la mayoría de protocolos de enrutamiento no han sido desarrollados para topologías que cambien, en las que los nodos son los responsables del mantenimiento de la red y de su propio enrutamiento. (Jiménez, Santos y Navas 2013, p. 109-116)

1.5.1. *Protocolos de enrutamiento proactivos para redes VANET*

1.5.1.1. *DSDV (Destination Sequenced Distance Vector)*

Es un protocolo de tipo proactivo basado en la familia de algoritmos de vector distancia y surge como una modificación del algoritmo Bellman-Ford. Cada nodo que es parte de la red contiene

una tabla de rutas con la información de cuantos saltos debe atravesar y la sucesión de los mismos hasta llegar a cada destino deseado. (Albero 2012, p. 29)

La actualización de las tablas se realiza con el intercambio de información entre nodos cercanos y al replicarse el algoritmo de menor costo. Los caminos tienen una secuencia de números como etiqueta para que no se generen bucles e indican el tiempo válido del camino, si la secuencia es más alta esto nos indica que el camino es más fiable. Si se da el caso de que dos caminos posean el mismo número de secuencia viene elegido en la tabla de rutas el de menor costo. (Albero 2012, p. 29-30)

Si el trayecto a un nodo destino se cae, se le asigna un número de secuencia impar que significa que el camino es inalcanzable (par indica que el camino es alcanzable) y un número de salto alto que significa infinito. (Albero 2012, p. 29)

1.5.1.2. *CGSR (Cluster-Head Gateway Switch Routing Protocol)*

Protocolo de enrutamiento proactivo basado en la métrica vector distancia que realiza la difusión de información a través de broadcast a todos los nodos que conforman parte de la red y los clasifica en niveles es decir de manera jerárquica, lo que ocasiona que los nodos de nivel inferior dependan directamente del desempeño de su respectivo nodo superior. Su escalabilidad es buena pero no cuenta con las características necesarias para adaptarse a redes que sean altamente dinámicas. (García Campos 2014, p. 25-27)

1.5.1.3. *WRP (Wireless Routing Protocol)*

WRP tiene como métrica para calcular el siguiente salto de su enrutamiento el vector distancia, es proactivo y todos los nodos que están bajo este protocolo están al mismo nivel lo que conlleva que al existir un fallo en algún nodo este no provoca un fallo en cascada para todos los nodos que conforman la red. Su escalabilidad es muy limitada, solo realiza el cálculo de rutas únicas y la difusión que aplica es a manera de broadcast y no se adapta eficientemente a redes con una alta dinámica en su topología. (García Campos 2014, p. 25-27)

1.5.1.4. *HSR (Hierarchi)*

Protocolo basado en la métrica estado de enlace que realiza la difusión de información a través de broadcast a los nodos que conforman parte de la red, es proactivo y realiza una clasificación jerárquica de los nodos. Su escalabilidad es buena y cuenta con las características necesarias para adaptarse a redes que sean altamente dinámicas, pero una de las deficiencias que presenta es que no realiza el descubrimiento de rutas de respaldo. (García Campos 2014, p. 25-27)

1.5.1.5. *OLSR (Optimized Link State Routing)*

Este protocolo de enrutamiento proactivo presenta una escalabilidad buena pero su capacidad de adaptarse a topologías dinámicas puede verse afectadas ya que para el cálculo de las rutas se basa en el estado de enlace y una red VANET al ser altamente dinámica provocaría que la red se inunde de paquetes de control propios del protocolo, mermando los paquetes de información y que no se haga un uso eficiente del ancho de banda. Su modelo de transmisión es el broadcast y puede realizar el cálculo de rutas múltiples. (García Campos 2014, p. 25-27)

1.5.1.6. *FSR (Fisheye State Routing)*

Protocolo de encaminamiento basado en la “técnica de ojo de pez”, es proactivo y mantiene una tabla de rutas y una lista de vecinos. Se diferencia de los protocolos de estado de enlace ya que para actualizar las rutas no realiza broadcast y solo envía los paquetes a los vecinos de su lista, utiliza los números de secuencia para identificar la información más reciente y sus mensajes de actualización contienen la dirección del nodo destino y la lista de vecinos. (Valencia Ayala y Risueño Benítez 2017, p. 34-36)

Las actualizaciones tienen periodos diferentes, mientras más cerca este el nodo vecino la información tiene periodicidad más alta y los nodos lejanos tienen periodicidad baja por lo que la información no es la más reciente y es imprecisa. FSR tiene potencial para su uso en redes grandes ya que al ocurrir cambios en los nodos lejanos no se produce el mismo tráfico que con un protocolo proactivo. (Valencia Ayala y Risueño Benítez 2017, p. 34-36)

A continuación, se presenta en la tabla 2-1 un resumen de las características de los protocolos proactivos mencionados.

Tabla 2-1: Protocolos proactivos para VANET

Protocolo	DSDV	CGSR	WRP	HSR	OLSR	FSR
Métrica para su ruta	Vector distancia	Vector distancia	Vector distancia	Estado de enlace	Estado de enlace	Estado de enlace
Transmisión	Broadcast	Broadcast	Broadcast	Broadcast	Broadcast	Broadcast
Clasificación de nodos	Plano	Jerárquico	Plano	Jerárquico	Plano	Plano
Rutas	Única	Única	Única	Única	Multiple	Multiple
Adaptabilidad a red dinámica	No	No	No	Si	Si	Si
Escalabilidad	Media	Buena	Limitada	Buena	Buena	Limitada
Documentación	Buena	Media	Limitada	Media	Media	Limitada

Fuente: (García Campos 2014)

1.5.2. *Protocolos de enrutamiento reactivos para redes VANET*

1.5.2.1. *AODV (Ad-hoc On Demand Vector)*

Protocolo reactivo diseñado por Johnson y Maltz a partir del protocolo proactivo DSDV, en el cual se realizaron mejoras para minimizar el broadcast que se generaba al realizar la búsqueda de las rutas, además que hace un mejor uso del ancho de banda y se adapta con rapidez a los cambios topológicos de red y permite transmitir en modo unicast o multicast. (Albero 2012, p. 31)

Al ser reactivo mantiene la información de ruteo más reciente en una tabla de rutas y como está basado en DSDV también cuenta con un numero de secuencia y un identificador único dentro de

la red que permiten identificar la información nueva de la antigua, pero al no utilizar la memoria de rutas el tamaño de los datos es pequeño. En redes de movilidad elevada tiene problemas de escalabilidad. (Albero 2012, p. 32-33)

Según (Albero 2012, p. 34-36), AODV hace uso de tres tipos de mensajes que son enviados mediante UDP:

- Solicitud de ruta (RREQ – Route Request): Los paquetes RREQ se envían a manera de broadcast al momento que un nodo requiera la ruta para un determinado destino. La ruta se determina al momento que el RREQ llega al destino.
- Contestación de ruta (RREP – Route Reply): Para establecer la ruta los RREP se envían de manera unicast hasta el nodo que realizó la petición.
- Error de ruta (RERR – Route Error): Son enviados cuando un enlace es inalcanzable es decir que un vecino que forma parte de la ruta no responde, estos mensajes no son enviados por broadcast, sino que cada nodo los envía a sus precursores hasta que lleguen a todos los nodos que conforman la ruta.

1.5.2.2. AOMDV (*Ad-hoc On-demand Multipath Distance Vector Routing*)

AOMDV es considerado como una extensión de AODV que permiten encontrar múltiples rutas sin bucles y enlaces disjuntos. Con la particularidad de que la entrada de enrutamiento de cada destino posee una lista de secuencia de saltos junto con la correspondencia del número de saltos y todos los siguientes saltos tienen el mismo número de secuencia para establecer el seguimiento de la ruta. (Marina y Das 2006, p. 969-988)

El número máximo de saltos para todas las rutas se establece como el número de saltos por cada destino que mantiene un nodo origen. Al realizar el envío de información mediante flooding existen mensajes duplicados y anuncios de una ruta recibida que se definen como rutas alternativas al nodo destino, al recibir el anuncio de ruta en un nodo destino con número de secuencia mayor la información del siguiente salto y el número de saltos anunciado se reinician. (Marina y Das 2006, p. 969-988)

AOMDV realiza el proceso de descubrimiento de enlaces, rutas disjuntas y nodos. Para las rutas de nodos disjuntos el nodo no procede a rechazar de inmediato el mensaje RREQ duplicados. Los nodos vecinos (menos el nodo origen) que envían paquetes RREQ definen una trayectoria de nodo disjunto, esto se da ya que no se pueden enviar paquetes RREQ duplicados entre nodos esto implica que los RREQ duplicados no pueden pasar un mismo nodo intermedio, los paquetes RREP solo se envían a los nodos vecinos de los que se recibieron los RREQ duplicados para establecer múltiples rutas. (Marina y Das 2006, p. 969-988)

1.5.2.3. *DSR (Dynamic Source Routing)*

DSR es un protocolo reactivo que se basa en el encaminamiento desde el origen por lo que los nodos fuente tienen que conocer la secuencia de nodo a seguir para llegar al destino. La información de los caminos que se deben tomar se almacenan en una memoria de enrutamiento por lo que está establecido para topologías de hasta 200 nodos ya que si se supera este límite la memoria de enrutamiento se sobrecarga y el funcionamiento no es correcto. (Chalmeta Ugas 2009, p. 31-34)

El proceso de descubrimiento de las rutas empieza con el envío de mensajes RREQ por parte del nodo origen y si el nodo destino tiene trayectos válidos responden con un mensaje RREP al nodo solicitante. Si la conexión se pierde se envían paquetes RERR para que todos los nodos actualicen la memoria de enrutamiento. (Chalmeta Ugas 2009, p. 31-34)

1.5.2.4. *ABR (Associativity Based Routing Protocol)*

Protocolo de enrutamiento reactivo, basado en el estado de enlace que presenta una buena escalabilidad y adaptación a redes dinámicas, además cuenta con un modelo de transmisión tanto unicast como multicast para establecer rutas únicas en la comunicación de sus nodos, mismos que se encuentran al mismo nivel es decir que carecen de jerarquía. (García Campos 2014, p. 28-29)

1.5.2.5. TORA (*Temporally Ordered Routing Algorithm*)

TORA es un protocolo de tipo reactivo que crea, mantiene y elimina las rutas, su principal característica es que le permite al nodo origen conocer múltiples caminos para que pueda llegar al destino, asocia pesos respecto al destino para cada nodo de la red y se envían los mensajes desde un nodo de mayor peso a uno de menor peso, descubre las rutas mediante mensajes QRY (Query) que se mantienen actualizadas con mensajes UPD (Update). (Valencia Ayala y Risueño Benítez 2017, p. 40)

Para conocer una ruta se envía mensajes QRY mediante broadcast hasta alcanzar el destino y todos los nodos que forman parte de la ruta responden con un mensaje UDP que se agregan a su peso. Si el destino es inalcanzable su peso aumenta al valor máximo consentido y si un nodo cualquiera tiene un nodo cercano con un peso máximo debe cambiar de nodo para poder establecer la ruta. (Valencia Ayala y Risueño Benítez 2017, p. 40-41)

Los paquetes CLR (Clear) solo se envían para los casos en los que se tienen que resetear la información del estado de enlaces de una porción de la red que no es alcanzable provocando la eliminación de la ruta y se procede a encontrar una ruta alternativa para llegar al nodo destino, dicha ruta alternativa es construida en un solo paso del algoritmo TORA. (Valencia Ayala y Risueño Benítez 2017, p. 40-41)

En la tabla 3-1 se presenta un resumen de las características de los protocolos reactivos mencionados en este documento de titulación.

Tabla 3-1: Protocolos reactivos para VANET

Protocolo	AODV	AOMDV	DSR	ABR	TORA
Métrica para su ruta	Vector distancia	Vector distancia	Estado de enlace	Estado de enlace	Inversión de enlace
Transmisión	Unicast Broadcast	Unicast Broadcast	Unicast	Unicast Broadcast	Broadcast

Clasificación (nodos)	Plano	Plano	Plano	Plano	Plano
Rutas	Única	Múltiple	Múltiple	Única	Múltiple
Adaptabilidad a red dinámica	Si	Si	No	Si	Media
Escalabilidad	Buena	Buena	Media	Buena	Buena
Documentación	Buena	Media	Buena	Limitada	Media

Fuente: (García Campos 2014)

1.6. Modelos de movilidad de redes VANET

En el entorno de las redes VANET los modelos de movilidad son los encargados de referenciar los patrones de movimiento que ocurren en los nodos que conforman la red Ad-Hoc vehicular y determinar su localización en la topología de la red en un tiempo determinado, además, describen el cambio de posición, velocidad y aceleración de los nodos en un instante de tiempo. (Orozco, Chavarro y Calderón 2014, p. 62-70)

Según (Orozco, Chavarro y Calderón 2014), existen varios tipos de clasificación para los modelos de movilidad entre los que destacan:

- Macroscópicos, microscópicos y mesoscópicos.
- Aleatorios, de tráfico, comportamiento y de flujo.
- Sintéticos, basados en datos, basados en trazas y basados en simuladores de tráfico.
- Entre otras clasificaciones más.

En la Fig. 9-1 se observa una de las clasificaciones que se emplean para los modelos de movilidad VANET y a continuación, se realiza una descripción de los distintos modelos que forman parte de esta clasificación.

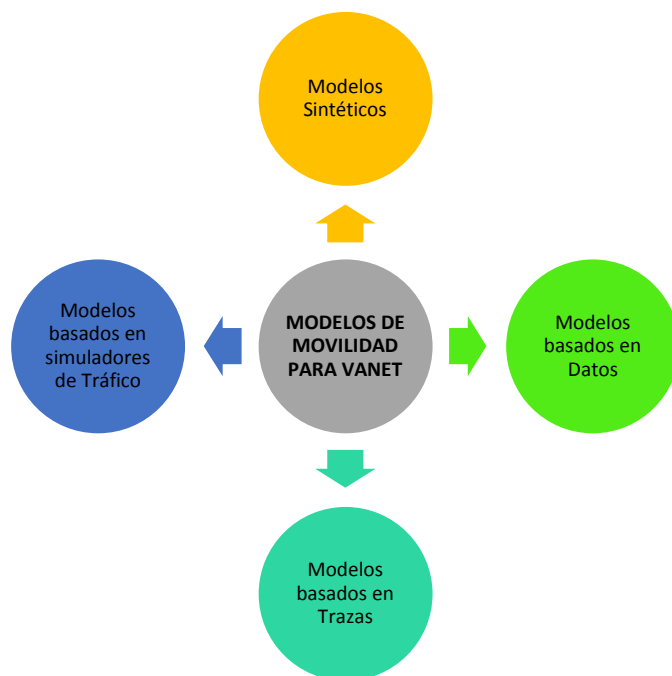


Figura 9-1: Clasificación de movilidad para VANET

Realizado por: Danni Tierra, 2018

1.6.1. *Modelos de movilidad sintéticos*

El modelo sintético es una de las clasificaciones más conocidas la cual separa a los modelos en varias clases, pero tiene como limitante que en la mayoría de estos modelos es complejo modelar la reacción del ser humano ante un imprevisto en las carreteras debido a que el comportamiento no es el mismo en todos los casos. (Roy 2011, p. 4,24)

Están basados en modelos matemáticos que tratan de recrear los efectos físicos reales de la movilidad de los nodos (vehículos), los modelos matemáticos fueron expuestos a pruebas para comprobar y garantizar que se acercan a la realidad por lo que hacen usos de trazas obtenidas durante mediciones de movimientos de entornos reales. (Roy 2011, p. 24)

En la Tabla 4-1, se pueden apreciar las clases de modelos sintéticos que son más conocidos en el ámbito de redes VANET.

Tabla 4-1: Clases del modelo sintético

CLASES DEL MODELO DE MOVILIDAD SINTÉTICO	Modelos estocásticos: Movimientos aleatorios.
	Modelos de flujo de tráfico: Basados en fenómenos de hidrodinámica.
	Modelos de coche seguidor: Se modela de acuerdo al movimiento del vehículo precedente.
	Modelos de colas: Modela carreteras como colas FIFO.
	Modelos de comportamiento: Normas de comportamiento.

Fuente: (Roy 2011)

Realizado por: Danni Tierra, 2018

Entre los modelos sintéticos más conocidos se tienen: random waypoint, random walk, random, random direction, Freeway y Manhattan.

1.6.1.1. *Modelo de movilidad random waypoint (RWP)*

El modelo de movilidad Random waypoint o punto de referencia aleatorio incluye pausas aleatorias uniformes al momento de cambiar de velocidad o dirección, es considerado como simple y conveniente para la evaluación de los protocolos de encaminamientos en redes VANET, pero es poco realista para implementar en casos reales. (Roy 2011, p. 65-66)

Al iniciar la simulación con este modelo cada nodo toma una ubicación de destino de manera aleatoria y que se encuentra dentro del campo de simulación, luego se desplazan hacia el destino con una velocidad constante aleatoria y de distribución uniforme que se encentra dentro del rango de $[v_{min}, v_{max}]$, al llegar al destino esperan un tiempo distribuido uniformemente en el intervalo de $[P_{min}, P_{max}]$ y procede a elegir un nuevo destino y velocidad para repetir el proceso hasta que se complete el tiempo de simulación asignado. La densidad de los nodos no es uniforme y los

nodos se encuentran más cerca del medio que del borde del campo de simulación. (Roy 2011, p. 66-68)

1.6.1.2. *Modelo de movilidad random walk (RWM)*

Este modelo de movilidad fue propuesto en un principio para emular las direcciones impredecibles que pueden tomar las partículas en física, este modelamiento considera que los campos de simulación están formados por celdas y un nodo puede trasladarse a cualquier celda vecina. Tanto RWP como RWM son similares ya que los nodos tienen movimientos aleatorios distribuidos aleatoriamente. (Roy 2011, p. 35-36)

RWM se considera un caso particular de RWP en donde el tiempo de espera es nulo, al iniciar la simulación cada nodo elige una dirección aleatoria con una distribución uniforme en un intervalo de $[0, 2\pi]$ y una velocidad aleatoria uniforme de entre $[v_{min}, v_{max}]$, luego se desplaza durante un tiempo t con dicha velocidad. Por cada nuevo intervalo de tiempo t los nodos vuelven a elegir su velocidad y dirección. (Roy 2011, p. 37-40)

1.6.1.3. *Modelo Manhattan*

Generalmente se usa para emular el movimiento de los nodos móviles en las calles con la forma de una sección de la ciudad de Manhattan y por ello el nombre del modelo. Este modelo se caracteriza por su topología de camino en cuadrícula. Formado por calles horizontales y verticales. (Roy 2011, p. 173-174)

Además, para el desplazamiento de sus nodos emplea un modelo probabilístico que asigna una probabilidad del 50% de que el nodo continúe una trayectoria recta, 25% a que gire en una intersección a la derecha y el otro 25% es la probabilidad de que gire a la izquierda de la intersección. En los intervalos de tiempo la velocidad de un nodo depende del nodo que lo precede en el mismo carril. (Roy 2011, p. 173-174)

Las ecuaciones que describen la velocidad en el modelo Manhattan son las siguientes:

Ecuación 1-1: Descripción de la velocidad de un vehículo según el modelo Manhattan.

$$v_i(t + \Delta t) = \begin{cases} v_{i+1}(t) - a / 2, & \text{Si } \Delta x_i(t) \leq \Delta x_{\min} \\ \tilde{v}_i(t + \Delta t), & \text{En otro caso} \end{cases}$$

La *Ecuación 1-1* añade la restricción para que los vehículos no se solapen y evitar colisiones entre ellos. Esta ecuación implica que, si el nodo alcanza la distancia de seguridad mínima, este debe frenar y de no hacerlo se aplica la forma estándar para calcular la velocidad. (García Campos 2014, p. 83-84)

Ecuación 2-1: Restricción sobre los límites de velocidad en el modelo Manhattan.

$$\tilde{v}_i(t + \Delta t) = \min \left\{ \max \left\{ v_i(t) + \eta a \Delta t, v_{\min} \right\}, v_{\max} \right\}$$

La *Ecuación 2-1* supone que la velocidad de cualquier nodo no puede ser menor que el valor mínimo de v_{\min} ni puede ser mayor que el valor máximo de v_{\max} durante cualquier ubicación del trayecto que recorre. En la *Ecuación 1-1* está incluida la *Ecuación 2-1*, lo que provoca que cuando un nodo no ha alcanzado la distancia mínima de seguridad el parámetro η es la misma variable aleatoria uniforme introducida en la *Ecuación 1-1*. Por lo que el modelo Manhattan añade la aceleración de manera aleatoria a la descripción de la velocidad del nodo. (García Campos 2014, p. 83-84)

1.6.2. *Modelos de movilidad basados en simuladores de tráfico*

Estos modelos tienen una validación basada en trazas reales y datos de comportamiento basados en la trayectoria y la interacción con los semáforos, cruces, intersecciones, entre otros. Este tipo de modelo tiene simuladores de tráfico microscópicos que permiten la configuración a detalle de las características de los nodos y macroscópicos que permiten la configuración de los nodos por grupo y no de manera individual como lo permiten los simuladores microscópicos. (Ripoll Cerezo 2012, p. 25-26)

Se consideran como la refinación de los modelos sintéticos por lo que son los más empleados en la comunidad dedicada al estudio de redes VANET ya que cuentan con distintas herramientas que permiten modelar de mejor manera el comportamiento vehicular, pero tiene como limitación que la mayoría de estos modelos hacen uso de software propietario y se vuelve necesaria la adquisición de licencias para poder utilizar todas las herramientas. (Ripoll Cerezo 2012, p. 25-26)

Entre los modelos de movilidad que se pueden encontrar en un simulador de tráfico vehicular se tienen: modelo de Krauss, Wagner, Kerner, IDM (*Intelligent Driver Model*) y otros más.

1.6.2.1. *Modelo de movilidad Krauss*

Fue propuesto por Stefan Krauss, es un modelo de movilidad microscópico que está basado en el ajuste de la velocidad de los nodos dependiendo de la velocidad de su predecesor para prevenir solapamientos (colisiones) y mantener la distancia de seguridad. (García Campos 2014, p. 84)

A continuación, se presenta la *Ecuación 3-1* que define la velocidad de este modelo.

Ecuación 3-1: Velocidad según el modelo Krauss

$$v(t+1) = v_1(t) + \frac{g(t) - v_1(t)}{\tau_b(t) + 1} - \eta(t)$$

Donde v es la velocidad, t el tiempo, τ el tiempo de reacción y η una variable aleatoria de ruido entre 0 y 1 que se introduce estocásticamente. En la ecuación se puede apreciar que la velocidad se rige continuamente por los vehículos de alrededor (apreciable en los parámetros v_1 y $g(t)$). (García Campos 2014, p. 84-85)

1.6.2.2. *Modelo de movilidad Kerner*

Según (García Campos 2014, p. 84), el modelo de Kerner también se lo conoce como teoría del tráfico trifásico, porque para modelar el tráfico vehicular lo separa en tres fases: flujo libre, flujo sincronizado y flujo de amplio congestionamiento.

- Flujo libre: Como su nombre lo indica el flujo está libre y los vehículos se puede circular sin ninguna congestión,
- Flujo sincronizado: Su característica es que la densidad de tráfico permite que los vehículos se desplacen con fluidez hasta determinada distancia actuando como si se tratara de flujo libre, superada esta distancia se empieza a bajar la marcha y los vehículos comienzan a formar hileras que crean cuellos de botella.
- Amplio congestionamiento: El tráfico es congestionado y es provocado porque la densidad de tráfico es extremadamente alta y la velocidad de desplazamiento vehicular es baja.

1.6.2.3. Modelo de conducción inteligente (IDM)

Es considerado como un modelo determinista, en el que la aceleración no solo depende de la distancia hasta el vehículo predecesor, sino que también depende del propio vehículo y de los que le rodean. IDM permite simular aspectos relacionados con las características del conductor, del vehículo y su entorno. La *Ecuación 4-1* describe la distancia que debe existir entre vehículos basados en el modelo IDM. (García Campos 2014, p. 85)

Ecuación 4-1: Distancia entre vehículos según el modelo IDM

$$\delta = x(v, \Delta v) = \Delta x_{\min} + vT + \frac{v\Delta v}{2\sqrt{ab}}$$

Cada parámetro de la *Ecuación 4-1* se describe a continuación según (García Campos 2014, p. 85).

- **Velocidad deseada v :** Velocidad máxima que el conductor desea alcanzar en cuando la densidad de tráfico permite circular con rapidez. El conductor tímido circula con valores de velocidad por debajo del límite legal, el normal circularía a la máxima velocidad que es permitida y el conductor agresivo supera el límite de velocidad legal.
- **Tiempo espacial T deseado:** Hace referencia a la distancia preferida por un conductor a una velocidad v , es considerada como la distancia que hay entre un vehículo y otro. Este factor tiene un valor típico de 1.4 s.

- **Distancia mínima entre parachoques Δx_{min} :** Esta distancia es inversamente proporcional a la agresividad del conductor, mientras más agresivo es el conductor la distancia entre parachoques es menor.
- **Aceleración deseada a :** El valor de este parámetro es directamente proporcional a la agresividad del conductor puesto que mientras más agresivo es la aceleración deseada será mayor, se tiene que considerar que esta aceleración está ligada a restricciones físicas, por lo que la aceleración en un camión necesariamente será, en general, menor que la de un coche.
- **Frenado confortable b :** El valor de b describe la deceleración de un vehículo al acercarse a un obstáculo en la carretera. Los conductores agresivos necesitan aplicar una desaceleración mayor a la de un conductor tímido puesto que se aproximan a grandes velocidades. Al ser un conductor más agresivo, el valor de desaceleración es mayor.

En la Tabla 5-1 se observa los tipos de conductores que existen en el modelo de movilidad IDM.

Tabla 5-1: Tipos de conductores en el modelo IDM

Parámetros IDM	Normal	Tímido	Agresivo	Camión
Velocidad deseada v_0 (en Km/h)	120	100	140	85
Tiempo espacial T deseado (en s.)	1.5	1.8	1.0	2.0
Distancia de seguridad s_0 (en m.)	2.0	4.0	1.0	4.0
Aceleración a máxima (en m/s²)	1.4	1.0	2.0	0.7
Desaceleración b (en m/s²)	2.0	1.0	3.0	2.0

Fuente: (García Campos 2014)

1.7. Simuladores para el desarrollo de redes VANET

El uso de los simuladores es indispensable para caracterizar un escenario próximo a la realidad sin el inconveniente que representar la inversión económica en infraestructura. Además, que Se vuelve necesaria la aplicación de software libre para no tener la necesidad de adquirir licencia de los programas que son propietarios. (Caiza Changoluisa 2017, p. 1)

Existen diferentes tipos de softwares de simulación libre y de pago que se pueden utilizar para el desarrollo de una red VANET, pero cabe recalcar que también es necesario utilizar software de movilidad para recrear las condiciones de congestión vehicular, software de red para la comunicación de los nodos que conforman la VANET y adicionalmente se utilizará un software híbrido que ayudará a la interoperabilidad entre el simulador de tráfico vehicular y de datos. (Caiza Changoluisa 2017, p. 1-3)

Existen varios simuladores de congestión vehicular entre los que se tiene: CityMob for RoadMaps (C4R), Simulation of Urban Mobility (SUMO), Street Random Waypoint (STRAW), VanetMobiSim, Bonnmotion, entre otros. (Boukenadil 2014, p. 175-182)

Del mismo modo se existen varios simuladores de red aceptados por la comunidad orientada al estudio de las VANET entre los que destacan: Qualnet, OMNeT++, NCTUns, OPNET y NS-2. Los programas más utilizados para facilitar la exportación de los datos de movilidad hacia el simulador de redes son: Traffic and Network Simulator Environment (Trans), Veins y Mobility Model Generator for Vehicular Networks (MOVE). (Boukenadil 2014, p. 175-182)

1.7.1. *Simuladores de tráfico vehicular.*

Es un programa que permite la simulación de entornos de tráfico vehicular considerando varios parámetros que pueden ocasionar un mayor índice de congestión de tránsito; entre las variables que influyen en la simulación se tiene el número de vehículos, la velocidad de movilización, el número de intersecciones la cantidad de calles que intervienen en el escenario, entre otros. (Boukenadil 2014, p. 175-182)

Las herramientas de simulación de tráfico más reconocidas son: SUMO, SmartPath, FreeSim, MobilTools, Bonnmotion, STRAW (*Street Random Waypoint*) y entre otros; los programas que son free muchas veces no cuentan con la misma funcionalidad que los pagados y esto resulta ser un inconveniente a la hora de obtener datos cercanos al entorno real que se pretende simular. (Boukenadil 2014, p. 175-182)

Debido a los progresivos trabajos de investigación y desarrollo los softwares free ahora tienen un mejor rendimiento y más herramientas para obtener datos más eficientes al realizar el proceso de simulación de tráfico. (Roy 2011)

1.7.1.1. *CityMob for RoadMaps (CAR)*

Cuenta con diferentes modelos de movilidad para redes VANET que permiten analizar el impacto en la comunicación entre los vehículos, es open source, permite la exportación de mapas reales, generar rutas y los resultados de su comunicación son compatibles con el simulador de red NS-2. Una de las principales fortalezas es que cuenta con una interfaz gráfica intuitiva que facilita el modelamiento de las redes VANET. (Anggoro et al. 2016, p. 25-29)

Mantiene una relación directa con el simulador SUMO (Simulation of Urban MOBility), ya que lo emplea para permitir al usuario la visualización del flujo creado en base a los modelos de movilidad vehicular. Está basado en lenguaje de programación Java y se distribuye bajo licencia GNU/GPL permitiendo modificar sus archivos de simulación, además, cuenta con un manual de usuario que ayuda a la interpretación más eficiente de su funcionamiento. (Anggoro et al. 2016, p. 25-29)

1.7.1.2. *Simulation of Urban Mobility (SUMO)*

SUMO fue desarrollado por la DRL que es el centro aeroespacial nacional de Alemania, es considerado como un software de movilidad microscópico que trabaja en tiempo discreto y espacio continuo, es de código abierto y entre las prestaciones que más resaltan son el uso de diferentes formatos de mapas reales, cuenta con un gran número de modelos de movilidad, permite la configuración de varios parámetros para tener una simulación más realista (semáforos, intersecciones, tipos de vehículos, número de carriles, etc.). (Anggoro et al. 2016, p. 25-29)

SUMO antes de comenzar la simulación ya tiene computarizadas las rutas por lo que consumen un alto grado de la memoria del sistema y está conformado por un paquete de subprogramas que desempeñan diferentes funciones que se complementan entre sí y permiten la creación, gestión y modelamiento de redes VANET con gran número de nodos (pocos miles) que podrían provocar

una congestión de tráfico en las calles del mapa digital por lo que se ayuda del algoritmo “dynamic user assignment” que fue desarrollado por Christian Gawron. (Chirinos Cadavid 2013, p. 4-30)

SUMO goza de gran popularidad por parte de académicos e investigadores debido a las herramientas que ofrece a la hora de simular redes VANET y esa popularidad aumenta aún más al ser compatible con el simulador de red NS-2 que también es apreciado como simulador de red para aplicaciones del mismo entorno que se desempeña fundamentalmente SUMO. (Chirinos Cadavid 2013, p. 4-30)

1.7.1.3. *Street Random Waypoint (STRAW)*

Su desarrollo se fundamentó en el entorno de tráfico de algunas ciudades de Estados Unidos de America, por lo que sus datos de salida se acercan mucho a un entorno real, se desarrolló para que esté integrado al simulador de redes JiST/SWANS por lo que se vuelve necesario tener instalado previamente el simulador JiST/SWANS antes de empezar a simular un escenario VANET. La movilidad de sus nodos está en función de la congestión vehicular. (Boukenadil y Feham 2015)

Uno de sus limitantes es que se necesita convertir los mapas digitales a un formato que sea compatible con STRAW, además es dependiente de SWANS y no cuenta con una interfaz para la correcta visualización de la simulación y al realizar las simulaciones con base en las ciudades de Estados Unidos y no permitir crear nuevos mapas no es recomendable su utilización si no es una VANET enfocada a la simulación de una ciudad de Estados Unidos. (Boukenadil y Feham 2015)

1.7.1.4. *VanetMobiSim*

VanetMobiSim es una plataforma de distribución abierta destinada a la simulación de movilidad vehicular y se puede emplear a nivel macroscópico o microscópico, lo que permite controlar parámetros de manera específica para recrear el comportamiento de los conductores (semáforos, velocidad carriles, flujo de tráfico, entre otros) o si se emplea a nivel macroscópico solo se controlan ciertos parámetros de los conductores pero solo de manera general y es compatible con varios simuladores de red de datos (NS-2, GloMoSim, QualNet, NET). (Ripoll Cerezo 2012, p. 36)

Este simulador es considerado como la extensión del software CanuMobiSim (*CANU Mobility Simulation Environment*) por lo que es indispensable que CANU esté instalado previo a añadir la extensión VanetMobiSim, permite la importación de mapas bajo el estándar europeo de carreteras GDF (*Geographical Data Files*) o también puede crear mapas aleatoriamente, está basado en Java para el manejo de archivos y librerías, además es necesario contar con las herramientas Java Run Time Environment y Apache Ant para la instalación. (Ripoll Cerezo 2012, p. 36)

Presenta como desventaja la necesidad de editar la configuración de los parámetros nodos, calles y rutas en archivos XML por lo que el nivel de lenguaje de programación resulta un poco complicado de emplear al iniciar el uso de este lenguaje. (Ripoll Cerezo 2012, p. 36-37)

1.7.2. *Simuladores de red*

Este tipo de simulador permite diseñar una red conformada por switches, routers y equipos terminales que interactúan entre sí para simular el funcionamiento de una red de datos real pudiendo obtener datos sobre el rendimiento y modificar los diferentes parámetros que forman parte de la red como son los equipos físicos, los protocolos de enrutamiento, la cantidad de elementos que conforman la red, el direccionamiento ip asignado a los equipos y varios parámetros adicionales. (Ripoll Cerezo 2012)

1.7.2.1. *OMNeT++*

OMNeT++ es un software modular y extensible que surgió como un proyecto del Departamento de Telecomunicaciones de la Universidad de Tecnología y Economía de Budapest en Hungría y fue publicado en 1997. Y con los años ha ganado gran popularidad gracias a sus prestaciones y herramientas es aplicado en el ámbito científico y académico. (Ripoll Cerezo 2012, p. 47)

Simula eventos discretos y se basa en componentes de bibliotecas C++ para modelar distintos tipos de redes de comunicación, posee un entorno de ejecución gráfico y también cuenta con un entorno de desarrollo basado en Eclipse. (Caiza Changoluisa 2017, p. 24)

Además, cuenta con módulos jerárquicos simples que se comunican y se agrupan con otros conformar módulos compuestos y el nivel superior que los contiene a todos es conocido como módulo de sistema, todos los módulos se comunican entre ellos a través de interfaces denominadas puertas, mismas que tienen conexiones entre sí. (Caiza Changoluisa 2017, p. 24)

1.7.2.2. *Network Simulator 2 (NS-2)*

Los inicios de su desarrollo datan de 1989, se considera como la variante del simulador REAL Network Simulator. Durante el año de 1995 llegó a manos de unos investigadores y desarrolladores de la universidad de California (Estados Unidos) en Berkeley. NS-2 se diseñó para la ayuda en la simulación de redes telemáticas y es empleado tanto para redes cableadas, inalámbricas, Ad-hoc y entre otras. (Chirinos Cadavid 2013, p. 18)

NS-2 es el simulador de redes más extendido hoy en día ya que su uso se expande desde proyectos de investigación hasta su aplicación como herramienta educativa y esto es consecuencia del desarrollo de los nuevos protocolos a distintos niveles para esta plataforma además de su condición de software libre. Además, permite el uso de distintos algoritmos para la planificación de colas para prevenir cuellos de botella y se puede transmitir en modos unicast o multicast. (Chirinos Cadavid 2013, p. 19)

Para desarrollar las simulaciones se utilizará lenguaje C++ para la configuración de los protocolos que se van a emplear en la red y se combina con lenguaje Otcl que se emplea en la programación de las variables que intervienen en la simulación. (Chirinos Cadavid 2013, p. 19)

1.7.2.3. *Optimized Network Engineering Tool (OPNET Modeler)*

Es una plataforma que se desarrolló en 1984 por investigadores y científicos del Instituto de Tecnología de Massachusetts, está orientada a la simulación de redes de telecomunicaciones y pone a disposición del programador varias librerías que permiten modelar distintos entornos de red, es un software propietario que ayuda a evaluar el desempeño de las redes de datos en el dominio del tiempo. (García Campos 2014, p. 94)

Posee una interfaz de usuario amigable y robusta que permite configurar distintos parámetros como pérdidas de conexión, fallos en los enlaces, definir el tipo de tráfico de la red, administración de paquetes entre otras opciones más ya que es posible realizar configuraciones desde la capa física hasta la capa de aplicación. (García Campos 2014, p. 94-95)

1.7.3. *Simuladores híbridos.*

Básicamente este tipo de programas son híbridos que permiten el acoplamiento tanto de un simulador de red de datos como el de un simulador de tráfico permitiendo que estos intercambien información, obteniéndose datos más precisos sobre el funcionamiento de la red de datos al integrar congestión de tráfico generado por los diferentes nodos que conforman la red VANET. (Orozco, Llano y Michoud 2012, p. 11-22)

Entre los simuladores VANET mas aceptados por la comunidad científica se tienen: Veins que ofrece una interfaz gráfica común permitiendo integrar el simulador de red OMNET++ y el de tráfico SUMO, TraNS al ser una opción Open Source que integra a NS-2 con SUMO también es una de las más utilizadas adicional que permite integrar mapas geográficos de Google Earth, NCTUNs entre otros. (Orozco, Llano y Michoud 2012, p. 11-22)

1.7.3.1. *Traffic and Network Simulator environment (Trans)*

Una de las principales fortalezas de este software es que cuenta con una interfaz gráfica que facilita la implementación de una red VANET, es capaz de servir como interfaz para el traslado de los datos que resultan de la herramienta de simulación de tráfico vehicular SUMO hacia el simulador de red de datos NS-2, tanto NS-2 como SUMO son herramientas empleadas por parte de la comunidad dedicada al estudio de VANETs lo que desencadena en que TraNS también sea popular ya que los vincula. (Ripoll Cerezo 2012, p. 39-40)

TraNS puede hacer uso de mapas TIGER importados o se pueden hacer modificaciones a estos mapas, por lo que los nodos tienen su ubicación en base al desplazamiento en las carreteras de estos mapas y son compatibles para ser exportados a NS-2, además cuenta con un plus de realismo ya que permite que durante la simulación los datos que se envían puedan modificar el desplazamiento de los vehículos. (Ripoll Cerezo 2012, p. 39-40)

1.7.3.2. *Vehicules in Network Simulation (Veins)*

Veins surge de un proyecto desarrollado en común entre el Departamento de Redes de Ordenadores y Sistemas de Comunicación, la Universidad de Erlangen-Nuremberg (Alemania), la Universidad de Innsbruck (Austria), el Departamento de Ordenadores y Sistemas de Comunicación y con la colaboración del Centro Aeroespacial Alemán. con uno de tráfico y que además no sea de pago por lo que se considera como un software híbrido que ayuda a la integración del simulador de datos OMNeT++ y el simulador de movimiento vehicular SUMO, estableciéndose como una interfaz de comunicación entre ambos. (Caiza Changoluisa 2017, p. 26-27)

En el año de 2008 se dio la publicación de su primera versión y desde entonces varios investigadores y desarrolladores han colaborado para la mejora de dicha plataforma.

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

MOVE al igual que el resto de simuladores híbridos surgió por la necesidad de brindar mayor realismo a los entornos de simulación VANET. Cuenta con una interfaz gráfica que ayuda a generar los scripts de manera rápida y así generar las intersecciones, carreteras, tipo de vehículo, velocidad de los vehículos sus rutas y demás parámetros que influyen en una red VANET. Cabe destacar que también permite realizar varios tipos de mapas (rejilla, tela de araña, completamente aleatorio) o incluso exportar mapas tipo TIGER. (Anggoro et al. 2016)

Esta plataforma integra el simulador SUMO, obteniendo las trazas del desplazamiento de los vehículos en el entorno del mapa y permitiendo su uso por los simuladores de redes NS-2 o QualNet. MOVE está conformada por dos secciones. (Anggoro et al. 2016, p. 25-29)

La primera sección se encarga de la configuración de los parámetros del simulador vehicular (mapa, vehículos, rutas, etc.) y la otra sección está destinada a la configuración de los parámetros relacionados al simulador de red (protocolos de transporte, aplicación, enlace, modelo de propagación, etc.). Además, permite la generación de movimiento de los vehículos de manera manual o automáticamente, definir la trayectoria, la posición de los semáforos o establecer el tipo de vehículo además de la frecuencia con la que los vehículos arribaran a su destino. (Boukenadil 2014, p. 175-182)

CAPÍTULO II

2. MARCO METODOLÓGICO

En la Fig. 1-2 se presenta un diagrama con la metodología que se aplicó al realizar la simulación de los distintos escenarios de este proyecto de titulación.

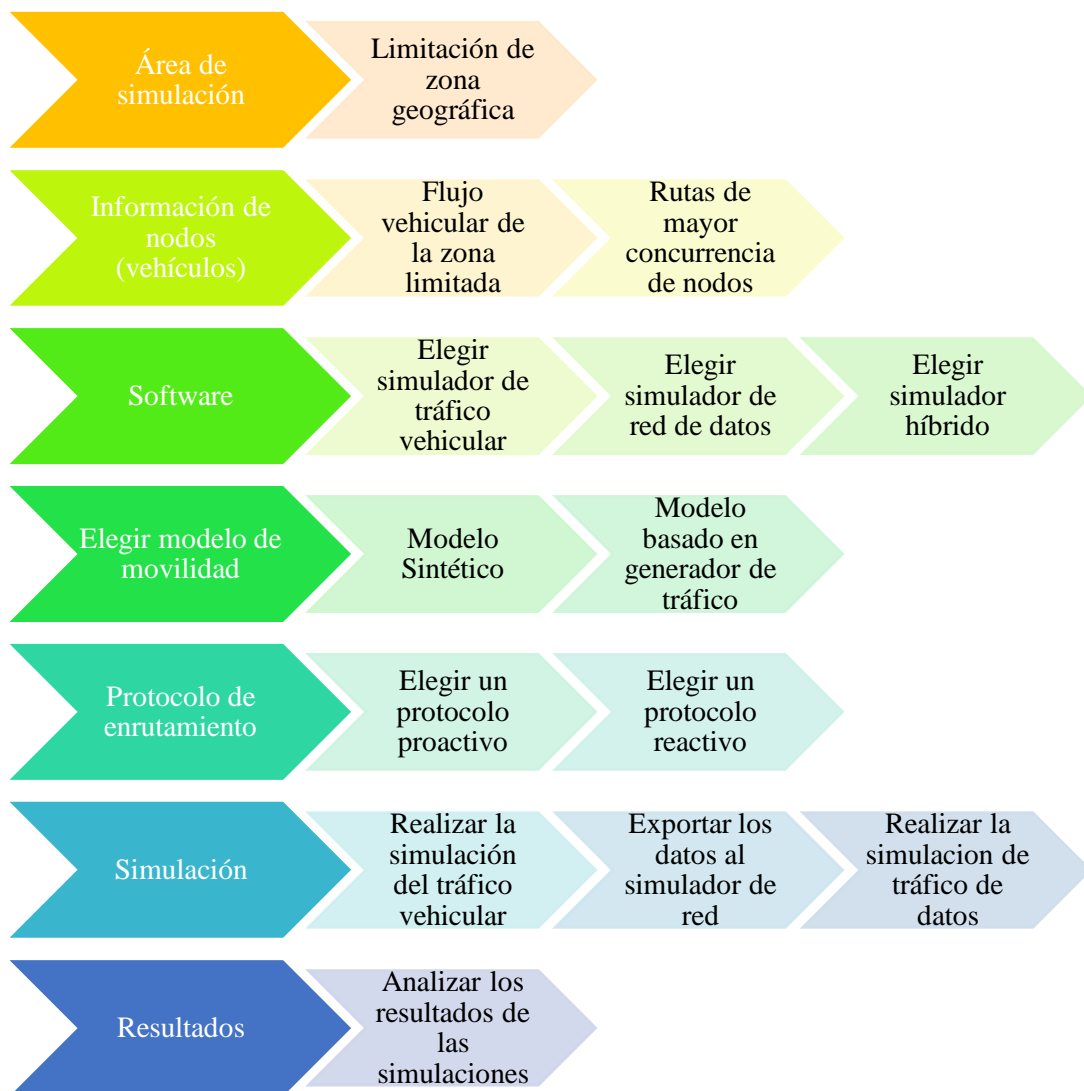


Figura 1-2: Metodología aplicada en la simulación

Realizado por: Danni Tierra, 2018

2.1. Introducción

En el capítulo 2 se describirá la justificación de la elección de la zona geográfica, del número de nodos para la simulación, las plataformas, los modelos de movilidad y protocolos de encaminamiento empleados para la realización de la simulación de un escenario VANET con sus dos tipos de comunicación: V2V y V2I.

Al terminar la simulación se obtendrán datos que permitan realizar una comparativa del rendimiento de los protocolos de enrutamiento basándonos en parámetros como throughput (rendimiento), PDF (relación de entrega de paquetes), Average End to End Delay (retardo promedio) y NRL (Carga de enrutamiento normalizada).

2.2. Determinación de la zona geográfica

La ciudad de Riobamba, es la capital de la provincia de Chimborazo, misma que desde hace algunos años cuenta con un gran número de parque automotor, cuya circulación provoca una alta densidad de tráfico vehicular especialmente en los lugares céntricos de la ciudad.

Se determinó que la zona delimitada por las calles Pichincha, 10 de agosto, España y José Joaquín de Olmedo poseen características adecuadas para realizar la simulación, ya que cuenta con semáforos en varias de sus intersecciones, gran afluencia de vehículos y se encuentra en la parte céntrica de la ciudad. Por lo tanto, se procederá a establecer una zona de 4 manzanas delimitadas por las calles ya mencionadas que cuentan con 9 intersecciones, 12 secciones de calles con 2 carriles cada sección y 3 semáforos.

Las secciones de la calle tienen una longitud aproximada de entre 75 a 90 metros dependiendo de cada sección, esta longitud se estableció basado en la medición realizada en Google Maps (Fig. 2-2) para tener resultados aproximados a la realidad durante la simulación.

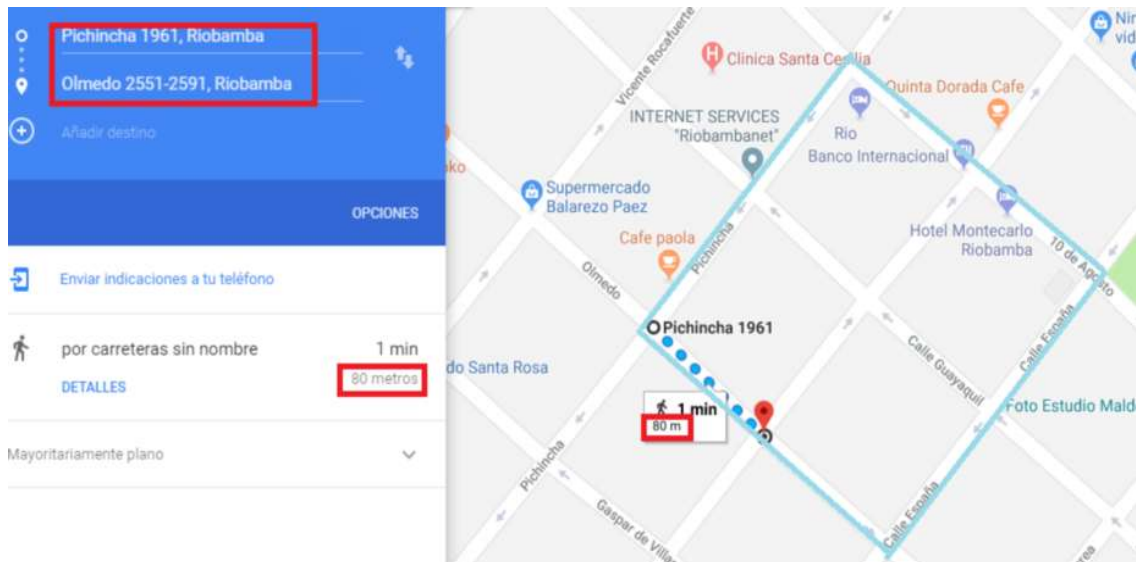


Figura 2-2: Geográfica de la simulación

Fuente: Google Maps

2.3. Flujo de tráfico vehicular

En la Fig. 3-2 se observa la zona que se escogió para la simulación de este proyecto, misma que se procedió a colocar identificadores para cada sección de la carretera para describir las rutas por las que el flujo de tráfico vehicular presenta una mayor densidad.



Figura 3-2: Identificadores para las secciones de la zona geográfica

Fuente: Google Maps

Para determinar el número de nodos a emplear en las simulaciones se procedió a la recopilación de datos de campo por 1 hora (12h00-13h00) durante 5 días, con lo que se obtuvo que el número total de vehículos que circulan por toda el área que se delimitó previamente (sección 2.2 Determinación de la zona geográfica) es de 600 vehículos.

En base a la fórmula estadística de muestreo para una población conocida (*Ecuación 1-2*) que nos genera un valor de 112 nodos (número mínimo de objetos de la muestra), que representan un 18.67%, se decide realizar la simulación con 120 nodos, mismos que representan el 20% de la población.

Ecuación 1-2: Fórmula de muestreo para poblaciones conocidas.

$$n = Z_{\alpha}^2 \frac{N \cdot p \cdot q}{i^2(N - 1) + Z_{\alpha}^2 \cdot p \cdot q}$$

Variables:

- **n:** Tamaño muestral
- **N:** Tamaño de la población, número total de historias
- **Z:** Valor correspondiente a la distribución de Gauss (1,96 para $\alpha= 0,05$)
- **p:** Prevalencia del fenómeno en estudio
- **q:** 1-p
- **“i” o “d”:** Tolerancia de error

En la tabla 1-2 se obtiene el 20% del universo (600) que genera un valor de 120 vehículos que representa el valor máximo a simular en este proyecto de titulación, al ser necesario realizar la comparación de los protocolos de enrutamiento se va a realizar la comparación con un valor intermedio (60) y un valor mínimo (10) de nodos. Los valores se redondearán a su inmediato superior ya que es evidente que para representar los vehículos los números deben ser enteros.

Tabla 1-2: Flujo de tráfico vehicular de las distintas rutas de la zona de simulación

	Nombre	Secciones	N° Vehículos	20%	Redondeo
1	ruta Pichincha	p1 p2	34	6,8	7
2	ruta García	g1 g2	76	15,2	15
3	ruta España	e1 e2	63	12,6	13
4	ruta Olmedo	j1 j2	95	19	19
5	ruta Guayaquil	gu1 gu2	84	16,8	17
6	ruta Diez de agosto	d1 d2	29	5,8	6
7	ruta Pichincha-olmedo	p1 p2 j1 j2	67	13,4	13
8	ruta García-diezagosto	g1 g2 d2	50	10	10
9	ruta García-diezagostobus	g1 g2	4	0,8	1
10	ruta Olmedo-garcía	j1 g1 g2	25	5	5
11	ruta Guayaquil-garcía	gu1 g2	30	6	6
12	ruta Guayaquil-pichincha	gu1 gu2 p2	20	4	4
13	ruta Diezagosto-españa	d1 d2 e1 e2	23	4,6	4
TOTAL			600	120	120

Realizado por: Danni Tierra,2018

Fuente: Datos de campo

Resulta prácticamente imposible que suceda un caso en donde todos los nodos se comuniquen al mismo tiempo para que generen un porcentaje de conexión del 100% por lo que el número de conexiones se representará con el 50% del número de nodos, es decir que para 120 nodos se realizaran 60 conexiones, para 60 nodos 30 conexiones y para 10 nodos se establecen 5 conexiones, dichas conexiones se establecen de manera aleatoria ya que trata de modelar un entorno real y es imposible predecir que nodo va a establecer comunicación con otro nodo.

El tiempo de simulación establecido en todas las simulaciones es de 500 segundos, esto se basa en que al sobrepasar este valor el simulador NS-2 presenta problemas en la ejecución de las simulaciones provocados por la necesidad de mayor capacidad de hardware generando un volcado de memoria en el equipo.

Se recalca que las rutas que se emplean en el modelo de movilidad Manhattan difieren de las utilizadas en el modelo de movilidad IDM, esto se da porque en el modelo Manhattan se basa en porcentajes de probabilidad de giro en las intersecciones por lo que para configurar sus rutas solo permite indicar la sección de salida y llegada del vehículo, es decir que las secciones de carretera intermedias son designadas por cálculos propios del modelo Manhattan.

2.4. Plataformas empleadas

Existen muchas plataformas que se desarrollaron para aplicarse en el estudio y simulación de las redes VANET por lo que se realizara una comparación de las principales características para determinar cuál es la que mejores recursos ofrece al momento de realizar nuestro proyecto.

Como ya se detalló en el capítulo 1, para la simulación se hace uso de tres tipos de simulador: simulador vehicular, simulador de red y simulador híbrido.

Para determinar las distintas plataformas de simulación, modelos de movilidad y los protocolos de enrutamiento se aplica ciertos conocimientos básicos de estadística, ya que en esta asignatura se estudian datos, mismos que representan los atributos o variables que describen hechos, y al analizarlos y procesarlos, estos se transforman en Información. Para poder hacer esto, es necesario comparar los datos entre sí y respecto de referencias. (Alaminos Chica y Castejón 2006, p. 98-100)

Este proceso de comparación requiere de escalas de medición donde situar los posibles valores que tomen los datos, y por las diferentes características de estos, existen diferentes tipos de escalas. Para el caso del presente estudio y por el tipo de dato obtenido en las simulaciones realizadas, se ha empleado la escala de Likert para el caso de variables cualitativas con una escala de medición de entre 3 a 1 en donde 1 representa un nivel mínimo, 2 un nivel medio y 3 el máximo nivel que puede asumir una variable cualitativa. (Alaminos Chica y Castejón 2006, p. 98-100).

2.4.1. *Simulador de tráfico vehicular*

Cada simulador de tráfico tiene sus características como se muestra en la tabla 2-2 y se procede a realizar una comparación para determinar la herramienta a emplear.

Tabla 2-2: Características de algunos simuladores de tráfico vehicular

Herramienta de Simulación	STRAW	VanetMobiSim	SUMO
Distribución	Libre	Libre	Libre
Compatibilidad con simuladores de red	Solo funciona con JiST/SWANS.	NS-2, GloMoSim, QualNet, NET	NS-2, QualNet, OMNET++
Mapas	Solo de Estados Unidos y no puede crear nuevos mapas	Aleatorios y mapas Europeos solo de formato GDP y TIGER.	Puede importar mapas de OpenStreetMaps o de formato TIGER y crear nuevos mapas.
Interfaz Grafica	Si	Si	Si
Documentación	Poco	Normal	Mucho
Desarrollo continuo	No	No	Si
Tipos de vehículos	No definidos	No posee tipos de vehículos	Posee varios tipos (bus, passenger, taxi, etc.)

Fuente: (Boukenadil y Feham 2015), (Ripoll Cerezo 2012) y (Chirinos Cadavid 2013)

Realizado por: Danni Tierra, 2018

Basados en la información expresada en la tabla 2-2 se procede a determinar el software que se utilizara en la simulación del tráfico vehicular. Se asignan valores de entre 1 a 3, en donde 3 representa que satisface totalmente cada parámetro, 2 parcialmente y 1 si es deficiente.

Tabla 3-2: Selección del simulador de tráfico vehicular

Herramientas de Simulación	STRAW	VanetMobiSim	SUMO
Distribución	3	3	3
Compatibilidad con simuladores de red	1	3	2

Mapas	1	2	3
Interfaz Grafica	3	3	3
Documentación	1	2	3
Desarrollo continuo	1	1	3
Tipos de vehículos	2	1	3
TOTAL	12	15	20

Realizado por: Danni Tierra,2018

Al analizar las características propias de cada herramienta de tráfico vehicular en la tabla 3-2 se concluye que SUMO ofrece una gama más variada de parámetros a configurar en el control de los vehículos que van a intervenir en la simulación por lo que se empleara SUMO para realizar este proyecto de titulación.

2.4.2. *Simulador de Red*

En la tabla 4-2 se representa ciertas características de algunos simuladores de red que se emplean en el estudio de redes VANET.

Tabla 4-2: Características de Simuladores de red de datos

Simulador de red	OMNeT++	NS-2	OPNET
Distribución	Libre	Libre	Pago
Compatibilidad con simuladores de tráfico vehicular	SUMO	SUMO, VanetMobiSim	SUMO
Consumo recursos de computador	Alto	Medio	Alto

Pila IEEE 802.11p	No soporta 802.11p pero tiene un apartado (Veins) para redes VANET.	Si	No
Documentación	Medio	Mucho	Poco
Escalabilidad	Alta	Media	Alta
Interfaz gráfica	Si	Si	Si

Fuente: (Caiza Changoluisa 2017), (Chirinos Cadavid 2013) y (García Campos 2014)

Realizado por: Danni Tierra, 2018

Basados en la información expresada en la tabla 4-2 se procede a determinar el software que se utilizara en la simulación del tráfico vehicular. Se asignan valores de entre 1 a 3, en donde 3 representa que satisface de manera excelente cada parámetro, 2 parcialmente y 1 si es deficiente.

Tabla 5-2: Selección del simulador de red

Simulador de red	OMNeT++	NS-2	OPNET
Distribución	3	3	1
Compatibilidad con SUMO	3	3	3
Ahorro de recursos de computador	1	2	1
IEEE 802.11p	1	3	1
Documentación	2	3	1
Escalabilidad	3	2	1
Interfaz gráfica	3	2	3
TOTAL	16	18	11

Realizado por: Danni Tierra, 2018

Al analizar las características propias de cada herramienta de simulación de redes en la tabla 5-2 se concluye que NS-2 ofrece mejores prestaciones que van a intervenir en la simulación por lo que se empleara este simulador para realizar este proyecto de titulación recalcando que uno de los parámetros fundamentales para optar por NS-2 es que cuenta con la pila de protocolos 802.11p.

2.4.3. *Simulador Híbrido*

Cada simulador híbrido es compatible con ciertas plataformas como se muestra en la tabla 6-2, al ya conocer las herramientas que se emplearan para el flujo vehicular y red de datos es necesario aplicar un simulador híbrido que sea compatible con SUMO y NS-2.

Tabla 6-2: Características de Simuladores híbridos

Simulador Híbrido	TRANS	VEINS	MOVE
Distribución	Libre	Libre	Libre
Compatibilidad con simuladores de tráfico vehicular y redes	SUMO y NS-2	SUMO, OMNET++	SUMO y NS-2
Consumo recursos de computador	Alto	Alto	Medio
Documentación	Medio	Medio	Alto
Soporta mapas	TIGER, Openstreetmap y aleatorios	TIGER y Openstreetmap	TIGER, Openstreetmap, y aleatorios

Fuente: (Ripoll Cerezo 2012), (Caiza Changoluisa 2017) y (Auparay, Anggoro y Ijtihadie 2018)

Realizado por: Danni Tierra,2018

Basados en la información expresada en la tabla 6-2 se procede a determinar el software que se utilizara en la exportación de datos desde SUMO a NS-2. Se asignan valores de entre 1 a 3, en donde 3 representa que satisface de manera excelente cada parámetro, 2 parcialmente y 1 si es deficiente.

Tabla 7-2: Selección del simulador híbrido

Simulador Híbrido	TRANS	VEINS	MOVE
Distribución	3	3	3
Compatibilidad con simuladores de tráfico vehicular y redes	3	1	3
Ahorro recursos de computador	1	1	2
Documentación	2	2	3
Soporta mapas	3	3	3
TOTAL	12	10	14

Realizado por: Danni Tierra,2018

Al analizar las características propias de cada herramienta híbrida en la tabla 7-2 se concluye que MOVE ofrece mejores prestaciones que van a intervenir en la simulación por lo que se empleara este simulador para realizar este proyecto de titulación recalcando que uno de los parámetros fundamentales para optar por MOVE es que tiene un menor consumo de recursos del computador.

2.5. Modelos de movilidad vehicular Empleados

2.5.1. Modelos de movilidad vehicular sintéticos

La simulación de la red VANET se llevará a cabo con dos modelos de movilidad, un sintético y otro basado en un simulador de tráfico vehicular para proceder a realizar la comparación de los modelos de movilidad en la tabla 8-2 se caracterizan los modelos sintéticos más conocidos.

Tabla 8-2: Características de algunos modelos de movilidad sintéticos

Modelo de movilidad sintético	RWP	RWM	Manhattan
Rutas	Aleatorias y tienden a tomar las rutas del centro del área de simulación	Aleatorias y tienden a tomar las rutas de los bordes del área de simulación	Semialeatorias, permiten establecer el inicio y fin de la ruta pero no la sección intermedia
Confiabilidad de resultados	Bajo y poco realista	Bajo y poco realista	Medio, se acerca un poco a la realidad
Topología del entorno de simulación	Aleatorio	Aleatorio	Forma de malla, coincide perfectamente con la topología de la ciudad de Riobamba
Velocidad de los nodos	Aleatoria uniforme	Aleatoria uniforme	Depende del nodo que lo precede en el mismo carril.

Realizado por: Danni Tierra, 2018

Fuente: (Roy 2011), (Roy 2011) y (García Campos 2014)

Considerando la información de la tabla 8-2 se procede a determinar el modelo de movilidad que se empleara en la simulación del entorno VANET. Se asignan valores de entre 1 a 3, en donde 3 representa que satisface de manera excelente cada parámetro, 2 parcialmente y 1 si es deficiente.

Tabla 9-2: Selección del modelo de movilidad sintético

Modelo de movilidad sintético	RWP	RWM	Manhattan
Tipo de rutas	1	1	2
Confiabilidad de resultados	1	1	2

Topología del entorno de simulación	1	1	3
Velocidad de los nodos	1	1	2
TOTAL	4	4	9

Realizado por: Danni Tierra, 2018

Para la red VANET basada en un modelo de movilidad sintético se determinó mediante la tabla 9-2 que el modelo Manhattan se adapta mejor al entorno del centro de la ciudad de Riobamba y se procederá a asignar los valores por defecto que se establecen para este modelo que son: 50% de probabilidad de seguir una ruta recta, 25% que el vehículo gire en la intersección a la derecha y 25% de probabilidad para que realice un giro a la izquierda. (Chirinos Cadavid 2013, p. 13)

2.5.2. Modelos de movilidad vehicular basados en generadores de tráfico vehicular

En la tabla 10-2 se caracterizan algunos de los modelos de movilidad basados en un simulador de tráfico vehicular.

Tabla 10-2: Modelos de movilidad basados en un generador de tráfico vehicular

Modelo de movilidad	Krauss	Kerner	IDM
Tipo	Microscópico	Macroscópicos	Microscópico
Confiabledad de resultados	Medio, se acerca un poco a la realidad	Bajo y poco realista	Medio, se acerca un poco a la realidad
Topología del entorno de simulación		Aleatorio	Forma de malla, coincide perfectamente con la topología de la ciudad de Riobamba

Velocidad de los nodos	Depende de su predecesor	Aleatoria uniforme	Depende del nodo que lo precede en el mismo carril.
-------------------------------	--------------------------	--------------------	---

Fuente: (García Campos 2014)

Realizado por: Danni Tierra,2018

Considerando la información de la tabla 10-2 se procede a determinar el modelo de movilidad que se utilizara en la simulación del entorno VANET. Se asignan valores de entre 1 a 3, en donde 3 representa que satisface de manera excelente cada parámetro, 2 parcialmente y 1 si es deficiente.

Tabla 11-2: Selección del modelo de movilidad basado en un generador de tráfico vehicular

	Krauss	Kerner	IDM
Modelo de movilidad	1	3	3
Confiabilidad de resultados	2	1	2
Topología del entorno de simulación	2	2	3
Velocidad de los nodos	2	1	3
TOTAL	7	7	11

Realizado por: Danni Tierra,2018

SUMO es un generador de tráfico vehicular que nos ofrece varias opciones de modelos de movilidad entre los que destacan el modelo de Krauss original, Krauss modificado, Kerner, Wagner, IDM entre otros. (Boukenadil 2014)

En base a los resultados de la tabla 11-2 se determina que el modelo de movilidad IDM es el óptimo para emplearse en nuestros distintos escenarios de simulación.

2.6. Modelo de propagación empleado

Existen varios tipos de propagación que se emplean de acuerdo al entorno en que se realizará la transmisión. El presente proyecto está encaminado al análisis de los protocolos de enrutamiento y no a evaluar y analizar los factores que influyen en la reflexión y difracción de las señales por lo que se empleara el modelo de propagación TwoRayGround Model.

Entre las principales necesidades de una VANET está el establecer comunicaciones eficientes a largas distancias y el modelo de propagación de dos rayos o conocido como TwoRayGround Model es el que mejor se adapta a esta necesidad. (Auparay, Anggoro y Ijtihadie 2018, p. 99-102)

2.7. Protocolos de enrutamiento empleados

Como se describió con anterioridad en el capítulo 1 en la sección 1.5 protocolos de enrutamiento para redes VANET, la clasificación que es más utilizada es en base a la planificación de la ruta por lo cual se procede a utilizar un protocolo de ruteo proactivo y un reactivo en base a las características que se describen en la tabla 2-1 y en la tabla 3-1, se asignan valores de entre 3 a 1 para cada parámetro, 3 si satisface eficientemente, 2 parcialmente y 1 si es deficiente.

Tabla 12-2: Comparación de protocolos proactivos

Protocolo	DSDV	CGSR	WRP	HSR	OLSR	FSR
Métrica para su ruta	3	3	3	1	1	1
Transmisión	2	2	2	2	2	2
Clasificación de nodos	3	1	3	1	3	3
Rutas	1	1	1	1	3	3

Menos paquetes de control (redes dinámicas)	3	3	3	2	2	2
Escalabilidad	2	3	1	3	3	1
Documentación	3	2	1	2	2	1
TOTAL	17	15	14	12	16	13

Realizado por: Danni Tierra,2018

Con los resultados de la tabla 12-2 se determina que el protocolo proactivo que se empleará en nuestros escenarios de simulación será el DSDV. En la tabla 13-2 se presenta la asignación de pesos para cada una de las características de los protocolos reactivos de la tabla 3-1.

Tabla 13-2: Comparación de protocolos reactivos

Protocolo	AODV	AOMDV	DSR	ABR	TORA
Métrica para su ruta	3	3	1	1	1
Transmisión	3	3	1	3	2
Clasificación de nodos	3	3	3	3	3
Rutas	1	3	3	1	3
Adaptabilidad a red dinámica	3	3	1	3	2
Escalabilidad	3	3	2	3	3
Documentación	3	2	3	1	2
TOTAL	19	21	14	15	16

Realizado por: Danni Tierra,2018

Con los resultados de la tabla 13-2 se determina que el protocolo de enrutamiento reactivo que mejores características presenta para un entorno VANET es el AOMDV, por lo que será utilizado para nuestros distintos escenarios de simulación.

Por otra parte, el protocolo de transporte que se empleará en los distintos escenarios de simulación será el TCP, ya que es un protocolo orientado a la comunicación lo que indica que los nodos reenviarán acuses de recibido cuando les llegue el paquete, de este modo se garantiza la entrega de la información. (Stallings 2004)

2.8. Parámetros de simulación

En la Tabla 14-2 se presentan los parámetros para la simulación del entorno VANET, tomando en consideración la información recopilada en el Capítulo 2 de este documento de titulación.

Tabla 14-2: Parámetros de simulación del entorno VANET.

Parámetros	Valor
Número de nodos (vehículos)	10, 60 y 120
Número de conexiones	5, 30 y 60
Modelos de movilidad	IDM, Manhattan
Tipos de comunicación	V2V, V2I
Tiempo de simulación	500 segundos (comunicación V2V y V2I)
Modelo de propagación	TwoRayGround
Protocolos de enrutamiento	AOMDV, DSDV
Protocolo de transporte	TCP
Frecuencia	5.9 Ghz (IEEE 802.11p)
Capa PHY/MAC	IEEE 802.11p

Realizado por: Danni Tierra, 2018

En la Fig. 4-2 se observan los valores recomendados por la IEEE para las variables de la capa MAC y PHY del estándar IEEE 802.11p, mismo que será aplicado en las simulaciones de este proyecto de titulación.

```

IEEE802-11p.tcl (-/ns-allinone-2.35/ns-2.35/tcl/ex/802.11) - gedit
IEEE802-11p.tcl (-/ns-allinone-2.35/ns-2.35/tcl/ex/802.11) - gedit
propagation.h ns-pgm.tcl plm-topo.tcl rmcc.tcl CtrMcast.tcl IEEE802-11p.tcl
#802.11p default parameters
Phy/WirelessPhyExt set C5Thresh_ 3.162e-12 ;# -85 dBm Wireless interface sensitivity (sensitivity defined in the standard)
Phy/WirelessPhyExt set Pt_ 0.001
Phy/WirelessPhyExt set Freq_ 5.9e+9
Phy/WirelessPhyExt set noise_floor_ 1.26e-13 ;# -99 dBm for 10MHz bandwidth
Phy/WirelessPhyExt set L_ 1.0 ;#default radio circuit gain/loss
Phy/WirelessPhyExt set PowerMonitorThresh_ 0.310e-14 ;# -102dBm power monitor sensitivity
Phy/WirelessPhyExt set HeaderDuration_ 0.000040 ;#40 us
Phy/WirelessPhyExt set BasicModulationScheme_ 0
Phy/WirelessPhyExt set PreambleCaptureSwitch_ 1
Phy/WirelessPhyExt set DataCaptureSwitch_ 0
Phy/WirelessPhyExt set SINR_PreambleCapture_ 2.5110; ;# 4 dB
Phy/WirelessPhyExt set SINR_DataCapture_ 100.0; ;# 10 dB
Phy/WirelessPhyExt set trace_dist_ 1e0 ;# PHY trace until distance of 1 Mio. km ("infty")
Phy/WirelessPhyExt set PHY_DBG_ 0

Mac/802_11Ext set CWMin_ 15
Mac/802_11Ext set CWMax_ 1023
Mac/802_11Ext set SlotTime_ 0.000013
Mac/802_11Ext set SIFS_ 0.000032
Mac/802_11Ext set ShortRetryLimit_ 7
Mac/802_11Ext set LongRetryLimit_ 4
Mac/802_11Ext set HeaderDuration_ 0.000040
Mac/802_11Ext set SymbolDuration_ 0.000008
Mac/802_11Ext set BasicModulationScheme_ 0
Mac/802_11Ext set use_802_11a_flag_ true
Mac/802_11Ext set RTSThreshold_ 2346
Mac/802_11Ext set MAC_DBG_ 0
  
```

Figura 4-2: Recomendaciones para IEEE 802.11p

Fuente: Archivo de la herramienta NS-2

2.9. Descripción de la interfaz gráfica de MOVE

En el Anexo A se adjuntan los comandos y el proceso que se debe seguir para la correcta instalación de las herramientas que se utilizaran para la simulación del entorno VANET.

Luego de instalar correctamente cada una de las herramientas se debe ejecutar el comando `java -jar MOVE.jar` en el terminal de Ubuntu y se observa la siguiente interfaz gráfica (GUI).

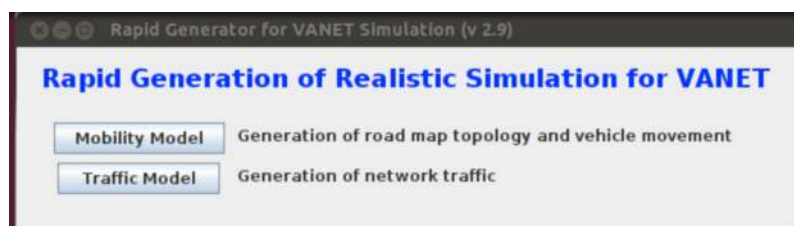


Figura 5-2: GUI de herramienta MOVE

Realizado por: Danni Tierra, 2018

La Fig. 5-2 se aprecian dos secciones: *Mobility Model* que nos va a permitir modelar los parámetros relacionados a la movilidad y características de los vehículos y la sección *Traffic Model* que se encargara de los parámetros que se establezcan en la red de datos luego de la importación de información de la primera sección mencionada.

Mobility Model a su vez cuenta con varias subsecciones (Fig. 6-2) que a continuación se detallan en la tabla 15-2 para facilitar el entendimiento de lo que se configura en cada subsección.

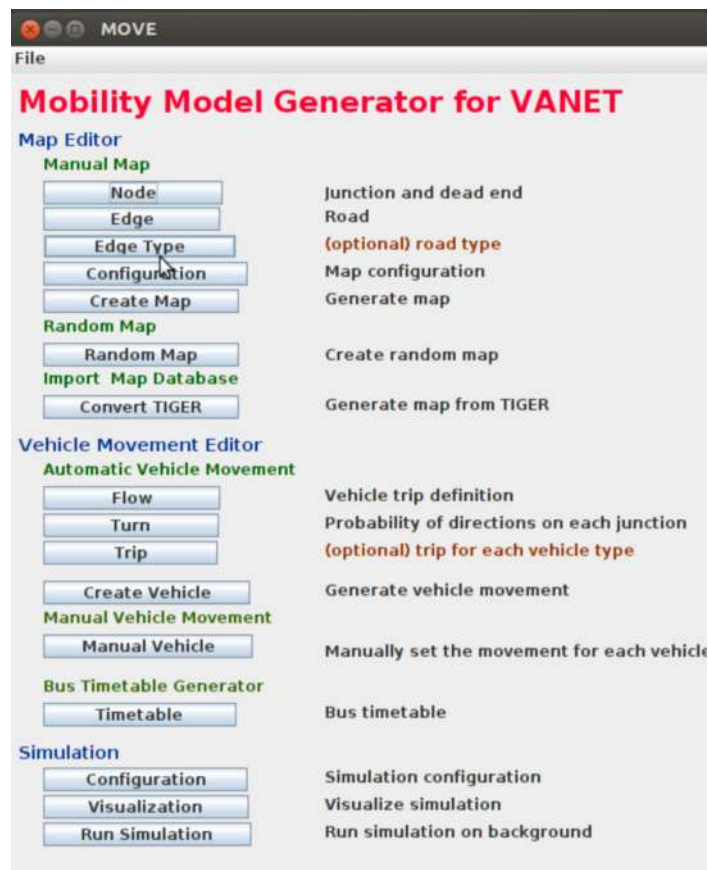


Figura 6-2: GUI de las subsecciones de Mobility Model

Realizado por: Danni Tierra, 2018.

Tabla 15-2: Descripción de las subsecciones de Mobility Model en MOVE

Node	Un nodo representan la esquina donde intersecan dos secciones de calle
Edge	Secciones de las calles que se unen entre sí en un nodo para formar las calles

Edge Type	Características de la sección de calle (primaria, secundaria, velocidad permitida)
Configuration	Carga los archivos necesarios para crear el mapa (node, edge y type edge que es opcional)
Create Map	Crea el mapa basado en la información de configuración
Random Map	Crea un mapa de manera aleatoria
Convert TIGER	Permite crear un mapa en base a datos TIGER
Flow	Características del flujo de tráfico vehicular (nombre, tipo, etc.)
Turn	Probabilidad de giro en las intersecciones
Trip	Establece las secciones de calle que conformaran una ruta
Create Vehicle	Crea las rutas basándose en la información del flow y turn o del flow y el trip que es opcional.
Manual Vehicle	Crea las rutas bajo configuraciones que establece el usuario
Timetable	Tabla de salida de buses
Configuration	Genera los archivos necesarios para la visualización en SUMO
Visualization	Visualiza la simulación en la GUI de SUMO
Run Simulation	Ejecuta el archivo de trazas de SUMO para poder exportar a NS-2

Realizado por: Danni Tierra, 2018

Fuente: MOVE

La sección *Traffic Model* también está compuesta por varias subsecciones (Fig. 7-2) que a continuación se detallan en la tabla 16-2 para facilitar el entendimiento de lo que se configura en cada subsección.

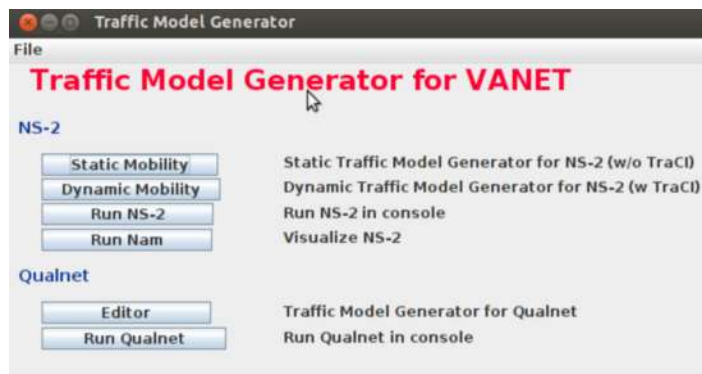


Figura 7-2: Subsecciones de Traffic Model

Realizado por: Danni Tierra, 2018

Tabla 16-2: Descripción de las subsecciones de Traffic Model en MOVE

Static Mobility Configurar los parámetros de red de datos cuando se simula sin la herramienta TraCI

Dynamic Mobility	Configurar los parámetros de red de datos cuando se simula con la herramienta TraCI
Run NS-2	Ejecuta los archivos tcl generados en Static mobility o Dynamic mobility
Run Nam	Ejecuta los archivos nam para la visualización de las simulación configurada en los archivos tcl
Editor	Configura los parámetros de red de datos si se utiliza Qualnet
Run Qualnet	Ejecuta Qualnet

Realizado por: Danni Tierra, 2018

Fuente: MOVE

2.10. Simulación del entorno VANET basado en el modelo generador de tráfico IDM

El proceso de simulación se realizó con 5 conexiones para 10 vehículos, 30 conexiones para 60 vehículos y 60 conexiones para 120 vehículos, recalando que el proceso es el mismo y lo que varía es el número de conexiones y nodos de acuerdo al escenario que se simule, a continuación, se detalla la simulación de 60 conexiones con 120 vehículos, en el Anexo B se adjuntarán las configuraciones para la simulación con el modelo Manhattan.

Para realizar la simulación para 60 vehículos con 30 conexiones o para 10 vehículos con 5 conexiones se siguen los mismos pasos que se describen en esta sección, solo se tiene que editar el número de vehículos y conexiones que se van a emplear.

2.10.1. *Comunicación V2V*

Paso 1 V2V. Para comenzar con la simulación se ejecuta el comando `java -jar MOVE.jar` desde el terminal de Ubuntu y nos aparecerá la interfaz de la Fig.8-2.

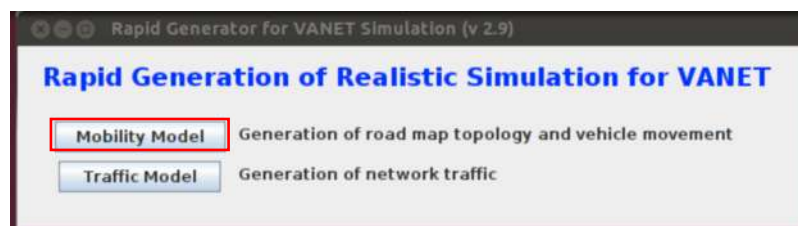


Figura 8-2: GUI de la herramienta MOVE

Realizado por: Danni Tierra, 2018.

2.10.1.1. *Simulación de la sección de movilidad vehicular*

Paso 2 V2V. El siguiente paso es seleccionar la opción *Mobility Model* (Fig. 8-2) que es la sección para modelar el tráfico vehicular, en la Fig. 9-2 se observa su interfaz con las subsecciones que la conforman.

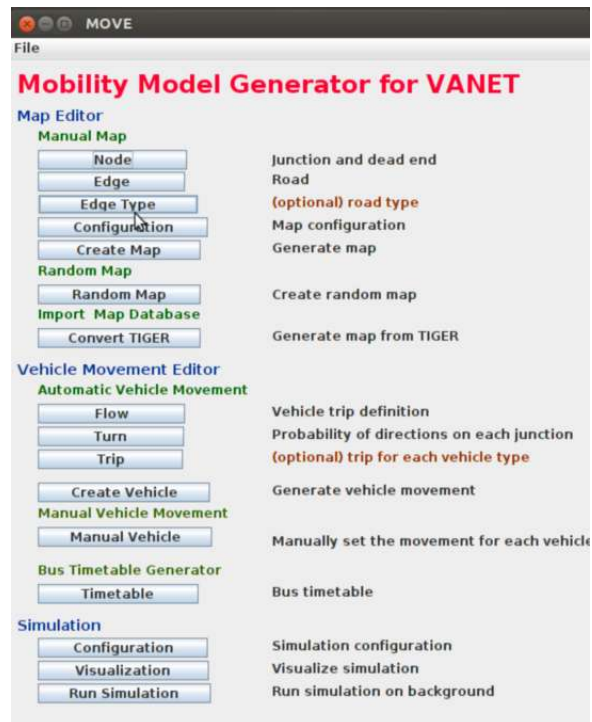


Figura 9-2: GUI Mobility Model

Realizado por: Danni Tierra,2018.

Paso 3 V2V. Seleccionar la opción *Node* y configurar los nodos con coordenadas (x, y) y distancias que se pueden obtener desde Google Maps, estos nodos serán los puntos de intersección de las secciones de calle que conformarán el mapa de la zona de simulación. En la Fig. 10-2 se observan los parámetros de configuración para las intersecciones.

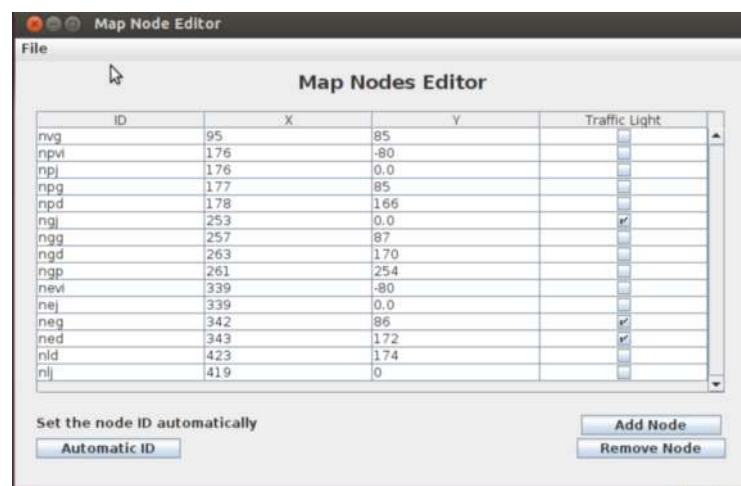


Figura 10-2: Configuración de nodos (intersecciones)

Realizado por: Danni Tierra,2018.

Paso 4 V2V. En la Fig. 11-2 se ve como configurar la opción *Edge Type* para definir los parámetros sobre el tipo de sección de carretera que se va a utilizar en la simulación. Para lo que se establecieron dos tipos de sección de calle que son primaria y secundaria con su respectivo id, nombre, número de carriles, prioridad, velocidad y capacidad.

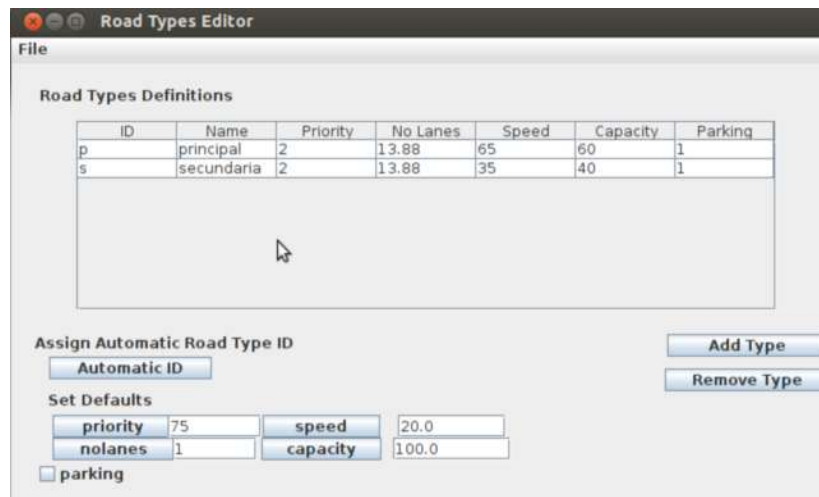


Figura 11-2: Configuración del tipo de sección de calle

Realizado por: Danni Tierra,2018.

Paso 5 V2V. En la Fig. 12-2 se observa la configuración de la opción *Edge*, colocando un id para cada sección de carretera, el id del tipo de sección (paso 4 V2V), el nodo inicio y fin que va a conectar la sección de carretera teniendo en cuenta que el orden en que se coloque el nodo para una sección define la dirección que pueden tomar los vehículos, es decir el sentido de la vía.

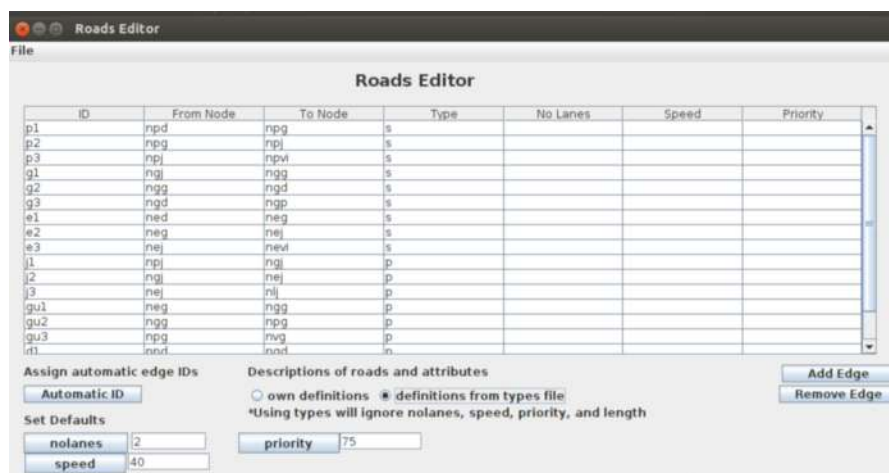


Figura 12-2: Configuración de las secciones que forman las calles

Realizado por: Danni Tierra,2018.

Paso 6 V2V. En la Fig. 13-2 se ven los parámetros de la subsección **Configuration** que permiten configurar el mapa que está basado en los archivos *.nod.xml*, *.edg.xml* y *.type.xml*. Se establecen las rutas a los archivos ya mencionados para crear un archivo *.net.xml* que será el que contiene la información de las vías que forman el mapa. Al guardar estas configuraciones se crea un nuevo archivo *.netc.cfg* que nos será útil para el Paso 7.

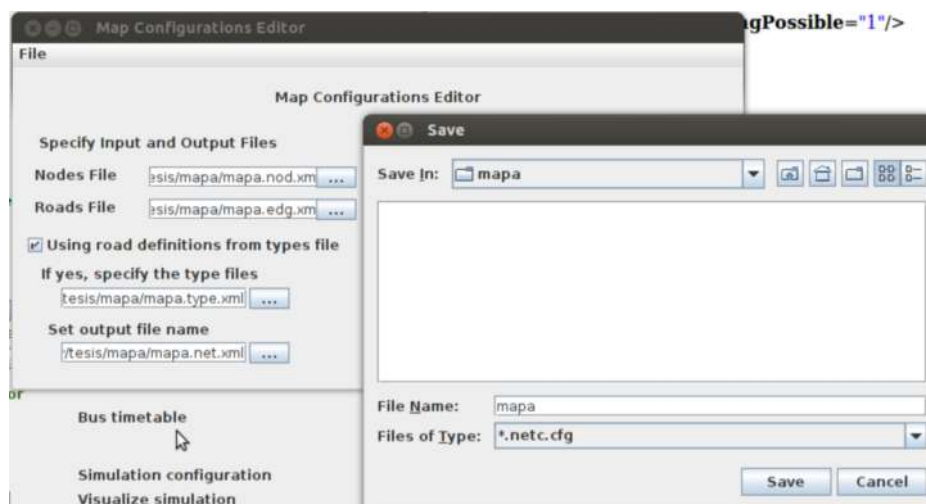


Figura 13-2: Rutas de los archivos para crear la topología de la zona de simulación

Realizado por: Danni Tierra,2018.

Paso 7 V2V. En la subsección **Create Map** se procede a cargar el archivo de configuración *.netc.cfg* (Fig. 14-2) que se creó en el Paso 6 V2V, mismo que contiene la ruta de los archivos necesarios para crear la topología del mapa. Luego click en *ok* para que comience el proceso de creación del mapa (Fig. 15-2).

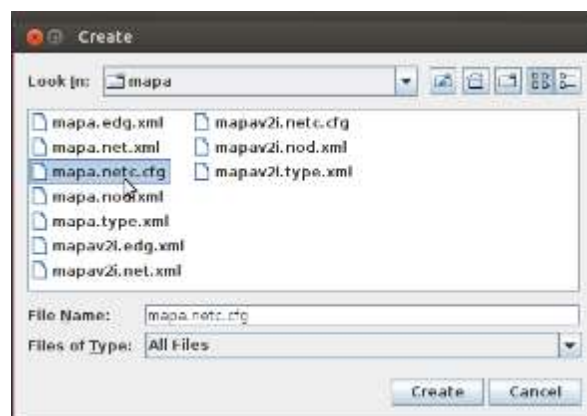


Figura 14-2: Subiendo archivo *.netc.cfg*

Realizado por: Danni Tierra,2018.



Figura 15-2: Generación exitosa del mapa

Realizado por: Danni Tierra,2018.

Al utilizar el modelo de movilidad IDM del simulador de tráfico SUMO se pueden controlar mas parametros que influyen en el comportamiento de los vehículos por lo que las rutas se proceden a crear siguiendo los pasos que se describen a continuacion.

Paso 8 V2V. En la Fig. 16-2 se observa la subsección *Manual Vehicle* que permite controlar el tipo de vehículos que van a circular, las secciones que van a conformar cada ruta y la asignación de las rutas para los vehículos. Estos parámetros dan como resultado un archivo *.rou.xml* que contiene la información de las rutas y vehículos que van a circular en el mapa creado anteriormente.

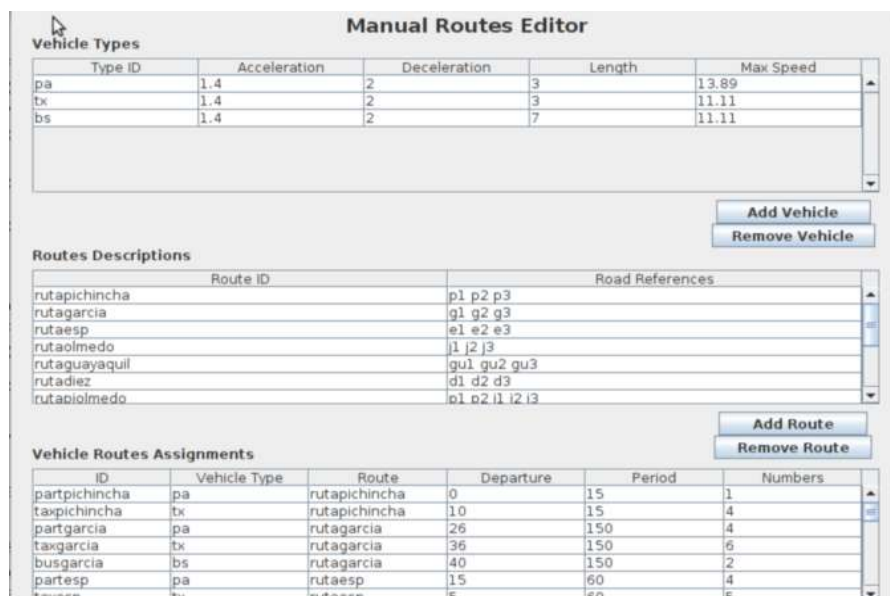
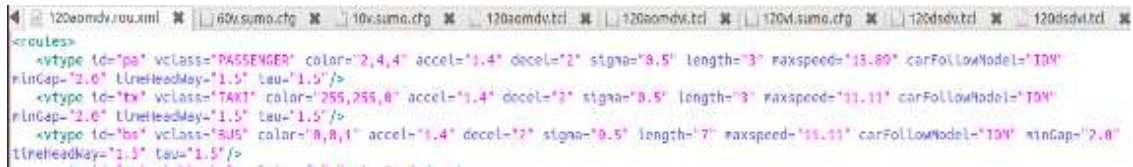


Figura 16-2: Parámetros para las rutas del modelo IDM

Realizado por: Danni Tierra,2018

Para realizar la simulación bajo los parámetros del modelo de movilidad IDM, en la simulación se debe editar el archivo `.rou.xml` como se aprecia en la Fig. 17-2.



```
<routes>
<route>
<vtype id="pa" vclass="PASSENGER" color="2,4,4" accel="1.4" decel="2" sigma="0.5" length="3" maxspeed="13.89" carFollowModel="IDM"
minGap="2.0" lTimeHeadway="1.5" tau="1.5"/>
<vtype id="tx" vclass="TAXI" color="255,255,0" accel="1.4" decel="2" sigma="0.5" length="3" maxspeed="11.11" carFollowModel="IDM"
minGap="2.0" lTimeHeadway="1.5" tau="1.5"/>
<vtype id="bs" vclass="BUS" color="0,0,1" accel="1.4" decel="2" sigma="0.5" length="7" maxspeed="11.11" carFollowModel="IDM" minGap="2.0"
lTimeHeadway="1.5" tau="1.5"/>
</route>
</routes>
```

Figura 17-2: Parámetros para el tipo de conductor normal de IDM

Realizado por: Danni Tierra,2018.

Paso 9 V2V. En la subsección *Configuration*, se carga el archivo `.net.xml` del Paso 6 V2V que contiene la información de la topología del mapa, el archivo `.rou.xml` del Paso 8 V2V que contiene toda la información de los vehículos, se indica el tiempo de simulación para que a su salida genere un archivo `.sumo.tr` (Fig. 18-2) y al guardar esta configuración se crea un archivo `.sumo.cfg`.

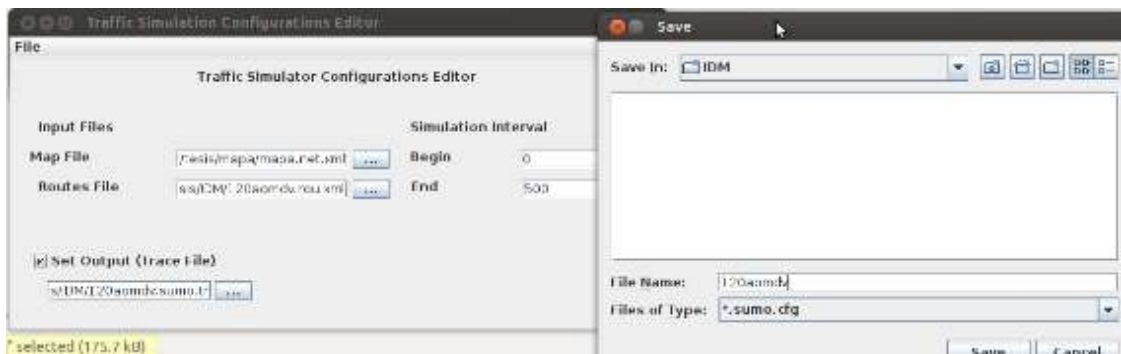


Figura 18-2: Creación de los archivo `.sumo.tr` y `.sumo.cfg`

Realizado por: Danni Tierra,2018.

Paso 10 V2V. La subsección *Visualization*, direcciona a la GUI de SUMO en donde se debe cargar el archivo `.sumo.cfg` del Paso 9 V2V (Fig. 19-2), mismo que contiene las rutas a los archivos necesarios para que SUMO permita la visualización de la simulación (Fig. 20-2)

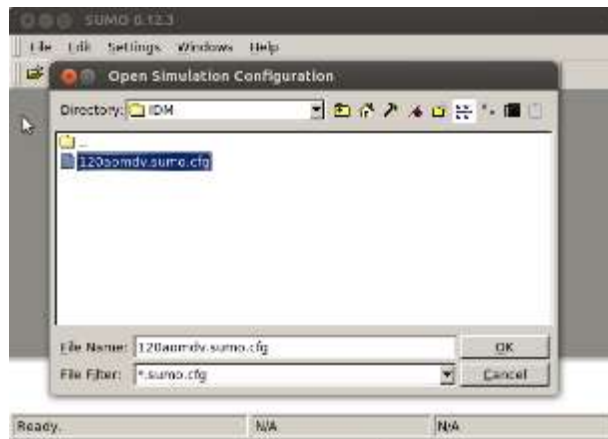


Figura 19-2: Cargando archivo .sumo.cfg en la herramienta SUMO

Realizado por: Danni Tierra,2018

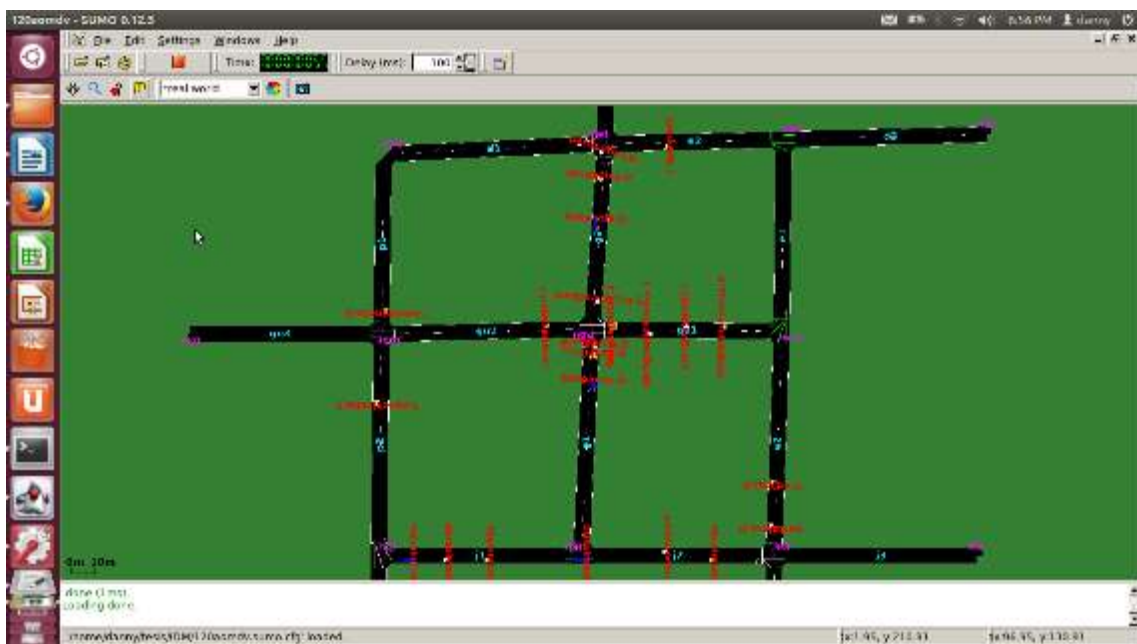


Figura 20-2: Simulación vehicular con 120 nodos

Realizado por: Danni Tierra,2018

Paso 11 V2V. Seleccionar la opción **Run Simulation** y cargar el archivo **.sumo.cfg** del Paso 9 V2V, esto se realiza para no tener inconvenientes al momento de exportar los datos hacia NS-2 y click en *ok* hasta que se visualice que el tiempo de simulación termine (Fig. 21-2).

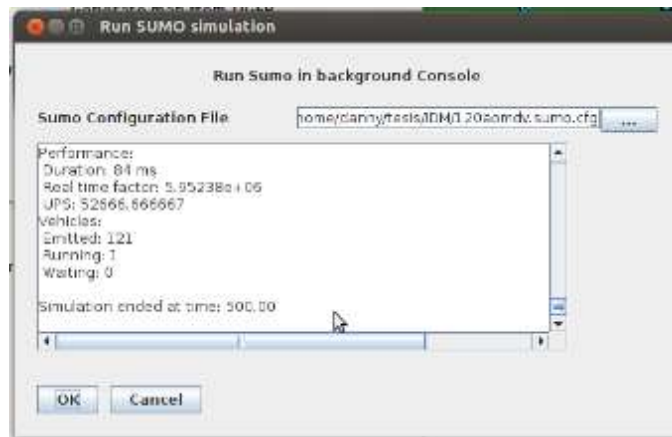


Figura 21-2: Cargando archivo .sumo.cfg para luego poder exportar los datos a NS-2

Realizado por: Danni Tierra,2018

2.10.1.2. Simulación de la sección de red de datos

Para la configuración de la sección de red de datos se debe colocar en la GUI de MOVE del paso 1 V2V y se sigue la intruccion del Paso 12 V2V.

Paso 12 V2V. Seleccionar la opción **Traffic Model** (Fig. 22-2) que es la sección para modelar la parte de red de datos. En la Fig. 23-2 se observa la interfaz que ayudará a importar los datos de sumo hasta NS-2.



Figura 22-2: GUI de herramienta MOVE

Realizado por: Danni Tierra,2018

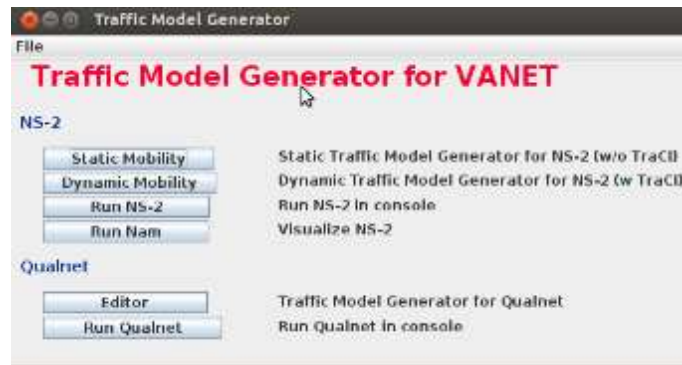


Figura 23-2: GUI de la sección Traffic Model

Realizado por: Danni Tierra,2018

Paso 13 V2V. Elegir la subsección *Static Mobility* y dar *click* en la opción importar datos desde SUMO, se sube el archivo *sumo.tr* del Paso 9 V2V como en la Fig. 24-2 y el archivo *.net.xml* del Paso 6 V2V como se indica en la Fig. 25-2.

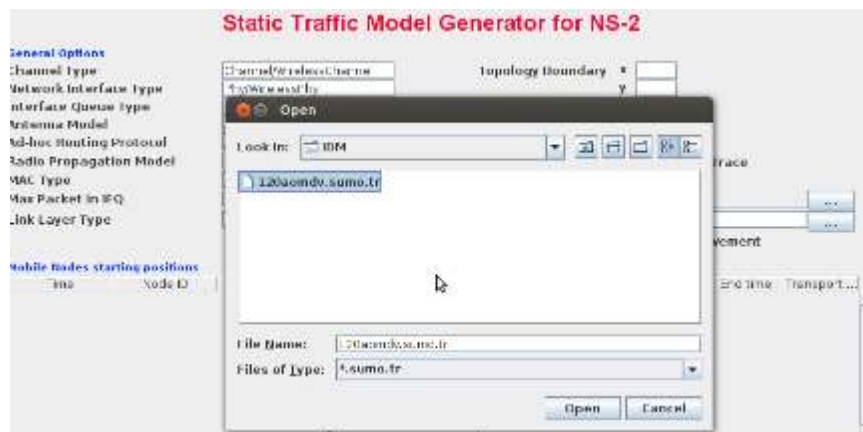


Figura 24-2: Importación de datos desde SUMO a NS-2 (.sumo.tr)

Realizado por: Danni Tierra,2018

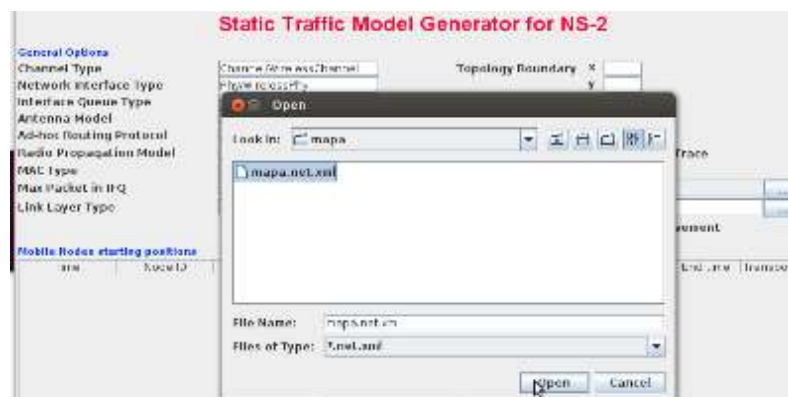


Figura 25-2: Importación de datos desde SUMO a NS-2 (.net.xml)

Realizado por: Danni Tierra,2018

Paso 14 V2V. Proceder a establecer los parámetros para la conexión de la red de datos. Establecer las capas PHY y MAC designadas para el uso del estándar IEEE 802.11p, configurar el protocolo de enrutamiento, que para este proyecto de titulación puede ser AOMDV o DSDV. Designar el modelo de propagación que se estableció previamente y seleccionar los ítems que nos permiten generar los archivos *.nam* y *.tr* con sus respectivas rutas. Además, configurar las características de los paquetes que se van a transmitir y las conexiones de los nodos.

En la Fig. 26-2 se observa toda la configuración de los parámetros descritos y solo una conexión del modo grafico ya que establecer las conexiones por lenguaje de programación resulta más rápido.

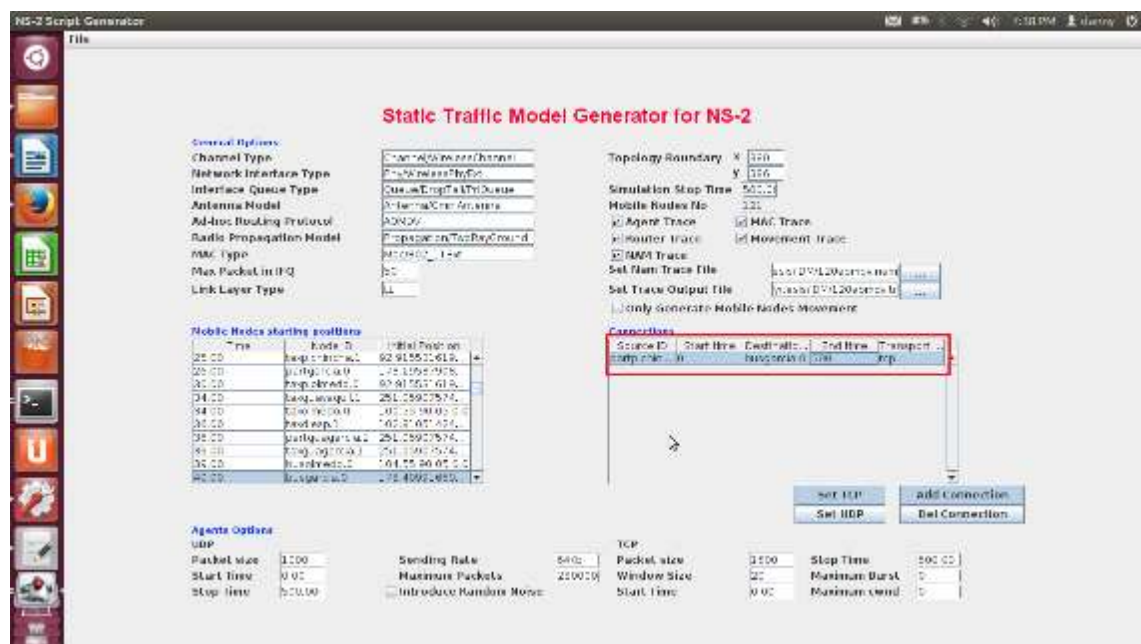


Figura 26-2: Configuración de parámetros para la red de datos en NS-2

Realizado por: Danni Tierra, 2018

Paso 15 V2V. Se establecen las 60 conexiones en el escenario con 120 nodos mediante el uso de lenguaje de programación (Fig. 27-2). Se procede a guardar el archivo *.tcl*.

```

120aomdv.tcl
$tcp57 set class_2
set sink57 [new Agent/TCPsink]
$ns_ attach-agent $node_(110) $tcp57
$ns_ attach-agent $node_(96) $sink57
$ns_ connect $tcp57 $sink57
set ftp57 [new Application/FTP]
$ftp57 attach-agent $tcp57
$ns_ at 300.0 "$ftp57 start"

$ns_ at 500.0 "$ftp57 stop"

set tcp58 [new Agent/TCP]
$tcp58 set class_2
set sink58 [new Agent/TCPsink]
$ns_ attach-agent $node_(108) $tcp58
$ns_ attach-agent $node_(116) $sink58
$ns_ connect $tcp58 $sink58
set ftp58 [new Application/FTP]
$ftp58 attach-agent $tcp58
$ns_ at 404.0 "$ftp58 start"

$ns_ at 472.0 "$ftp58 stop"

set tcp59 [new Agent/TCP]
$tcp59 set class_2
set sink59 [new Agent/TCPsink]
$ns_ attach-agent $node_(115) $tcp59
$ns_ attach-agent $node_(98) $sink59
$ns_ connect $tcp59 $sink59
set ftp59 [new Application/FTP]
$ftp59 attach-agent $tcp59
$ns_ at 472.0 "$ftp59 start"

$ns_ at 500.0 "$ftp59 stop"

```

Figura 27-2: Las 60 Conexión TCP (archivo.tcl)

Realizado por: Danni Tierra,2018

Paso 16 V2V. Ejecutar desde el terminal de Ubuntu el archivo *.tcl* mediante la herramienta NS-2 (Fig. 28-2) con el comando *ns nombre.tcl* para generar los archivos *.nam* y *.tr*. El archivo *.nam* nos servirá para la visualización de la simulación de red mientras que el archivo *.tr* es el que contiene toda la información del desempeño de la red, mismo que se utilizara luego para representar gráficamente en Excel la comparación de los protocolos de enrutamiento. En el anexo C se presenta uno de los archivos tcl generados en este proyecto de titulación.

```

danny-Dell-System-Inspiron-N4110: ~/te
danny@danny-Dell-System-Inspiron-N4110:~$ cd te
danny@danny-Dell-System-Inspiron-N4110:~/te$ ns 120aomdv.tcl
num_nodes is set 120
INITIALIZE THE LIST xListHead
Starting Simulation...
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 532.2
SORTING LISTS ...DONE!
NS EXITING...
danny@danny-Dell-System-Inspiron-N4110:~/te$ ns 120aomdv.tcl

```

Figura 28-2: Ejecución del archivo *.tcl* con la herramienta NS-2

Realizado por: Danni Tierra,2018

Paso 17 V2V. Ejecutar desde la consola el archivo *.nam* mediante la extensión NAM de la herramienta NS-2 con el comando *nam nombre.nam* para visualizar la simulación de red. En la

Fig. 29-2 se observa la GUI de nam al ejecutar el comando. Mientras que en la Fig. 30-2 se visualiza la simulación de la red VANET en la que se encuentran integrados la movilidad vehicular de SUMO y el tráfico de datos generado en NS-2.

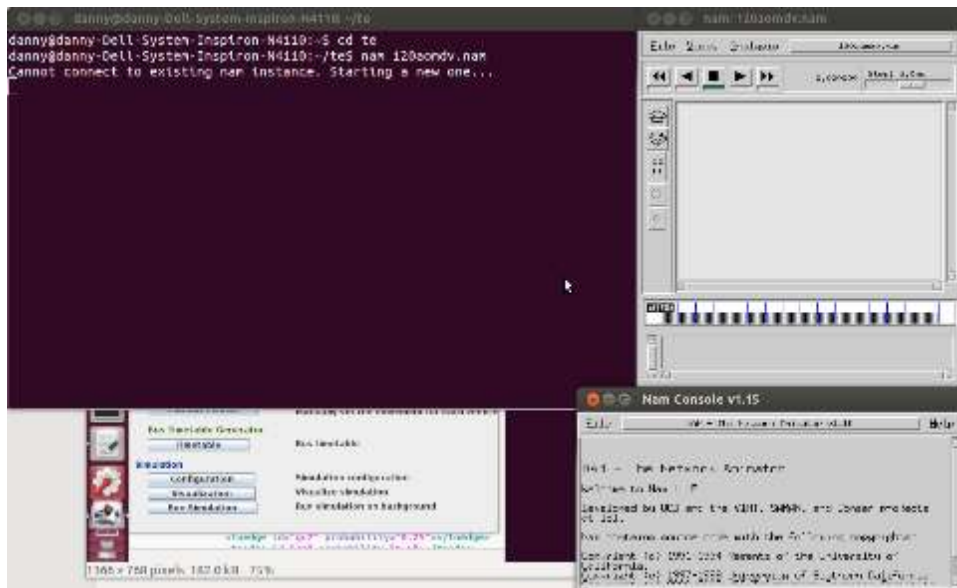


Figura 29-2: GUI de NAM (extensión de NS-2)

Realizado por: Danni Tierra, 2018

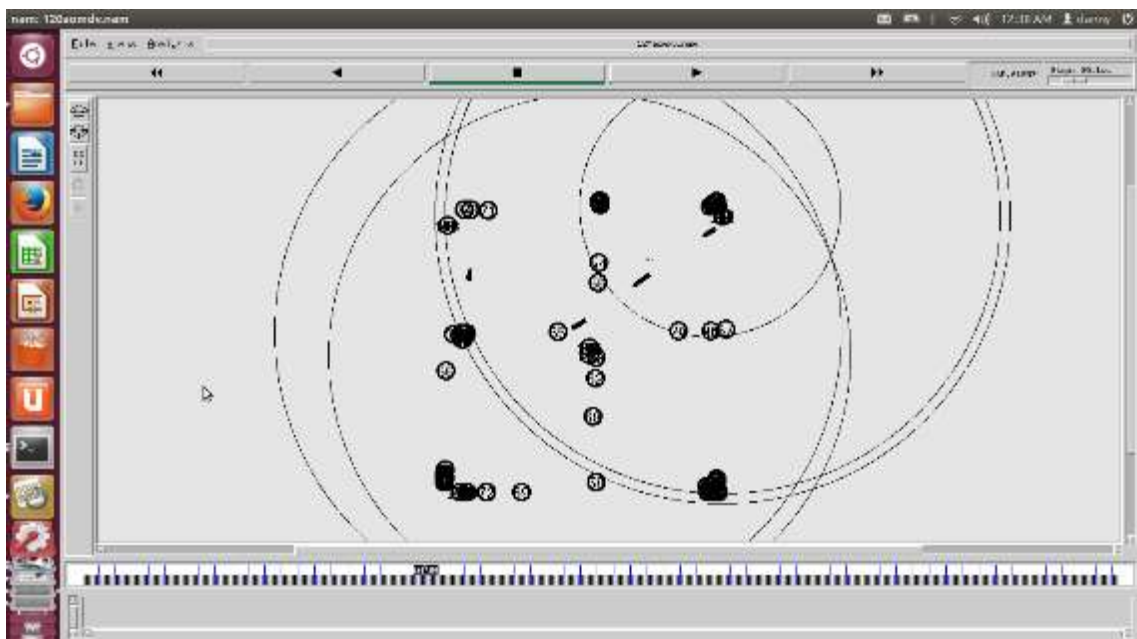


Figura 30-2: Simulación de VANET completa (SUMO-NS2)

Realizado por: Danni Tierra, 2018

2.10.2. Comunicación V2I

Los pasos a seguir son idénticos, con cierta variación en la creación del tamaño del mapa de la zona de simulación para que exista la necesidad del uso de las infraestructuras.

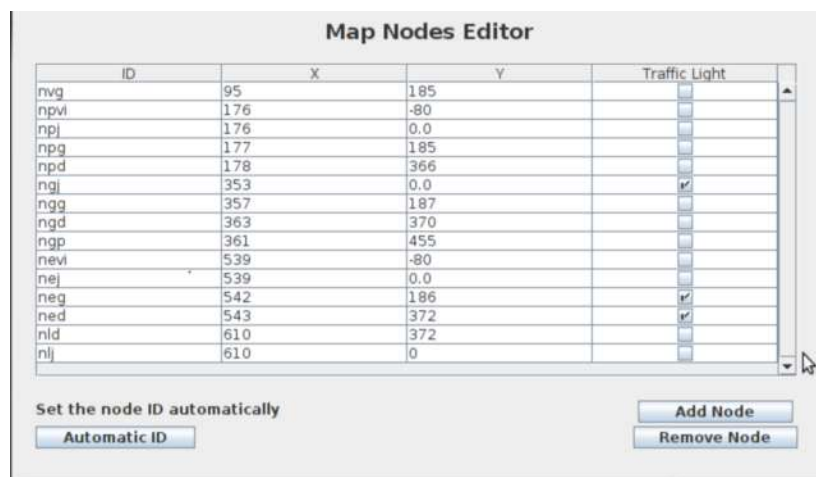
2.10.2.1. Simulación de la sección de movilidad vehicular

Paso 1 V2I. Se sigue el mismo proceso que se detalla en el Paso 1 V2V.

Paso 2 V2I. Se sigue el mismo proceso que se detalla en el Paso 2 V2V.

Paso 3 V2I. Se sigue el mismo proceso que se detalla en el Paso 3 V2V con la modificación en las coordenadas de los nodos, se aumentó 100 metros a cada sección de carretera (Fig. 31-2) para que se vuelva necesario el uso de las infraestructuras al no tener dentro del rango de cobertura otro nodo que permita la comunicación.

Al utilizar el estándar IEEE 802.11p y no considerar las pérdidas por las edificaciones, la cobertura alcanza aproximadamente los 200 metros y las secciones de calle llegan a tener una longitud de entre los 75 a 90 metros por lo que se optó por aumentar la distancia entre los nodos.



The screenshot shows a software window titled "Map Nodes Editor". It contains a table with the following data:

ID	X	Y	Traffic Light
nvq	95	185	<input type="checkbox"/>
npvi	176	-80	<input type="checkbox"/>
npj	176	0.0	<input type="checkbox"/>
npq	177	185	<input type="checkbox"/>
npd	178	366	<input type="checkbox"/>
ngj	353	0.0	<input checked="" type="checkbox"/>
ngq	357	187	<input type="checkbox"/>
ngd	363	370	<input type="checkbox"/>
ngp	361	455	<input type="checkbox"/>
nevi	539	-80	<input type="checkbox"/>
nej	539	0.0	<input type="checkbox"/>
neg	542	186	<input checked="" type="checkbox"/>
ned	543	372	<input checked="" type="checkbox"/>
nld	610	372	<input type="checkbox"/>
nj	610	0	<input type="checkbox"/>

Below the table, there are two buttons: "Automatic ID" and "Set the node ID automatically". At the bottom right, there are two buttons: "Add Node" and "Remove Node".

Figura 31-2: Configuración de intersecciones (V2I)

Realizado por: Danni Tierra, 2018

Paso 4 V2I. Se sigue el mismo proceso que se detalla en el Paso 4 V2V.

Paso 5 V2I. Se sigue el mismo proceso que se detalla en el Paso 5 V2V.

Paso 6 V2I. Se sigue el mismo proceso que se detalla en el Paso 6 V2V teniendo presente que se debe cargar el nuevo archivo de nodos en el que se modificaron las coordenadas.

Paso 7 V2I. Se sigue el mismo proceso que se detalla en el Paso 7 V2V y se guardan los nuevos archivos de la topología para un escenario de comunicación V2I.

Paso 8 V2I. Se sigue el mismo proceso que se detalla en el Paso 8 V2V.

Paso 9 V2I. Se sigue el mismo proceso que se detalla en el Paso 9 V2V, pero cargando los correspondientes archivos de la comunicación V2I.

Paso 10 V2I. Se sigue el mismo proceso que se detalla en el Paso 10 V2V, pero cargando los correspondientes archivos de la comunicación V2I. En la Fig. 32-2 se observa la simulación de 120 vehículos en la nueva topología para comunicación V2I.

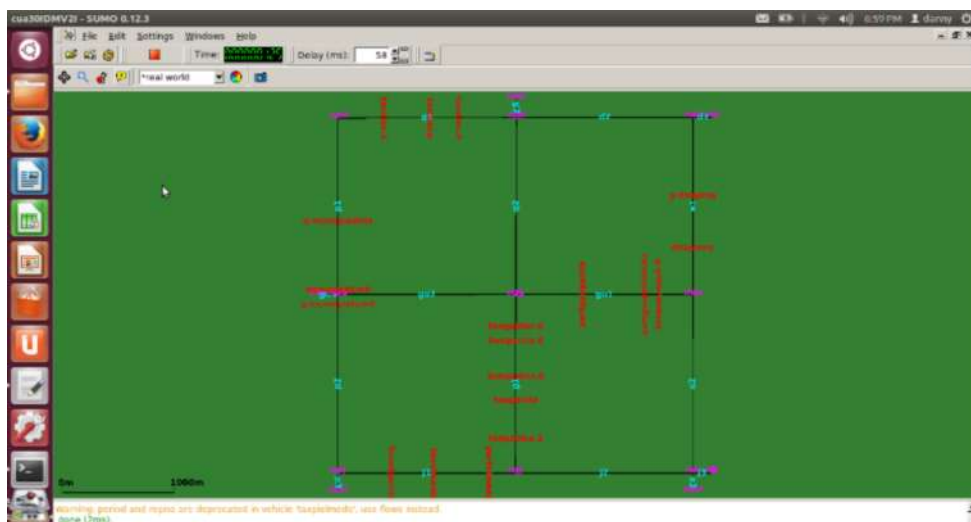


Figura 32-2: Topología para comunicación V2I

Realizado por: Danni Tierra, 2018

Paso 11 V2I. Se sigue el mismo proceso que se detalla en el Paso 11 V2V, pero ejecutando el correspondiente archivo de la comunicación V2I.

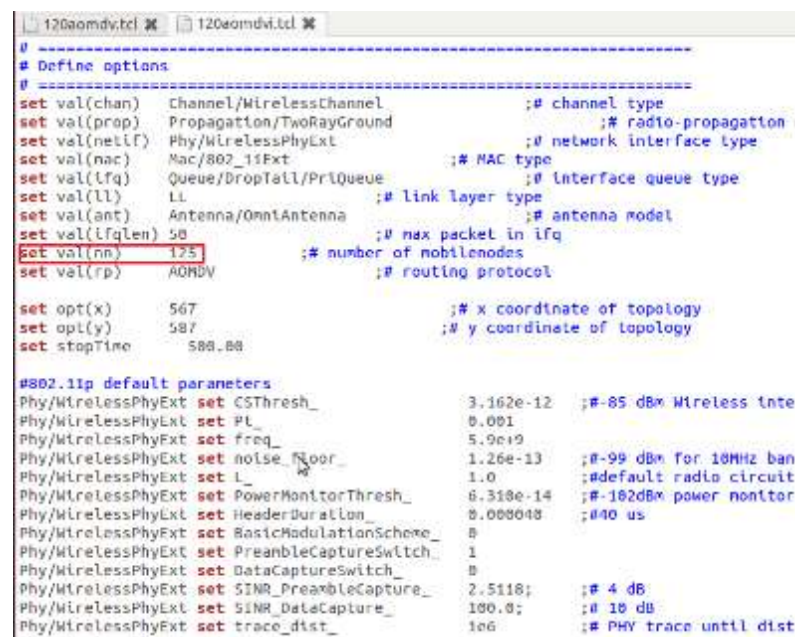
2.10.2.2. Simulación de la sección de red de datos

Paso 12 V2I. Se sigue el mismo proceso que se detalla en el Paso 12 V2V.

Paso 13 V2I. Se sigue el mismo proceso que se detalla en el Paso 13 V2V, considerando que los archivos que se deben emplear son los de comunicación V2I.

Paso 14 V2I. Se sigue el mismo proceso que se detalla en el Paso 14 V2V.

Paso 15 V2I. Proceder a establecer los parámetros para la conexión de la red de datos básicamente igual al Paso 15 V2V con la excepción de que se modifica el archivo *.tcl* mediante lenguaje de programación para declarar las 5 infraestructuras (Fig. 33-2) y configurar la posición y nombre de cada una de las infraestructuras, como se muestra en la Fig. 34-2.



```
-----
# Define options
-----
set val(chan) Channel/WirelessChannel           ;# channel type
set val(prop) Propagation/TwoRayGround         ;# radio-propagation
set val(netif) Phy/WirelessPhyExt             ;# network interface type
set val(mac) Mac/802_11Ext                    ;# MAC type
set val(ifq) Queue/DropTail/PriQueue         ;# interface queue type
set val(ll) LL                                 ;# link layer type
set val(ant) Antenna/OmniAntenna             ;# antenna model
set val(ifqLen) 50                             ;# max packet in ifq
set val(nn) 125                                 ;# number of mobilenodes
set val(rp) AODV                               ;# routing protocol

set opt(x) 567                                 ;# x coordinate of topology
set opt(y) 587                                 ;# y coordinate of topology
set stopTime 588.88

#802.11p default parameters
Phy/WirelessPhyExt set CStresh_ 3.162e-12 ;# -85 dBm Wireless inte
Phy/WirelessPhyExt set Pt_ 0.001
Phy/WirelessPhyExt set freq_ 5.9e+9
Phy/WirelessPhyExt set noise_floor_ 1.26e-13 ;# -99 dBm for 10MHz ban
Phy/WirelessPhyExt set l_ 1.0 ;# default radio circuit
Phy/WirelessPhyExt set PowerMonitorThresh_ 6.31e-14 ;# -102dBm power monitor
Phy/WirelessPhyExt set HeaderDuration_ 0.000048 ;# 40 us
Phy/WirelessPhyExt set BasicModulationScheme_ B
Phy/WirelessPhyExt set PreambleCaptureSwitc_ 1
Phy/WirelessPhyExt set DataCaptureSwitch_ B
Phy/WirelessPhyExt set SINR_PreambleCapture_ 2.5118 ;# 4 dB
Phy/WirelessPhyExt set SINR_DataCapture_ 100.0 ;# 10 dB
Phy/WirelessPhyExt set trace_dist_ 106 ;# PHY trace until dist
```

Figura 33-2: Declaración de las 5 Infraestructuras adicionales a los 120 vehículos

Realizado por: Danni Tierra, 2018

```

120aomdv.tcl 120aomdv.tcl
$node_(121) setdest 457 265 1
# Infraestructura3
$node_(122) color "blue"
$node_(122) set X_ 283
$node_(122) set Y_ 470
$node_(122) set Z_ 0.0
$node_(122) setdest 283 470 1
# Infraestructura4
$node_(123) color "blue"
$node_(123) set X_ 273
$node_(123) set Y_ 81
$node_(123) set Z_ 0.0
$node_(123) setdest 273 81 1
# Infraestructura5
$node_(124) color "blue"
$node_(124) set X_ 284
$node_(124) set Y_ 271
$node_(124) set Z_ 0.0
$node_(124) setdest 284 271 1

# Now produce node movements
$ns_ at 0.0 "$node_(120) color blue"
$ns_ at 0.0 "$node_(120) setdest 100 275 1"
$ns_ at 0.0 "$node_(120) label RSU1"
$ns_ at 0.0 "$node_(121) color blue"
$ns_ at 0.0 "$node_(121) setdest 457 265 1"
$ns_ at 0.0 "$node_(121) label RSU2"
$ns_ at 0.0 "$node_(122) color blue"
$ns_ at 0.0 "$node_(122) setdest 283 470 1"
$ns_ at 0.0 "$node_(122) label RSU3"
$ns_ at 0.0 "$node_(123) color blue"
$ns_ at 0.0 "$node_(123) setdest 273 81 1"
$ns_ at 0.0 "$node_(123) label RSU4"
$ns_ at 0.0 "$node_(124) color blue"
$ns_ at 0.0 "$node_(124) setdest 284 271 1"
$ns_ at 0.0 "$node_(124) label RSU5"

```

Figura 34-2: Posicionamiento estratégico de las Infraestructuras en la topología

Realizado por: Danni Tierra,2018

Paso 16 V2I. Se sigue el mismo proceso que se detalla en el Paso 16 V2V, pero la ejecución del archivo `.tcl` compila 125 nodos (Fig. 35-2) ya que cuenta con 120 vehículos y 5 infraestructuras.

```

danny@danny-Dell-System-Inspiron-N4110:~/te$ ns 120aomdv.tcl
num_nodes is set 125
INITIALIZE THE LIST xListHead
Starting Simulation..
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 532.2
SORTING LISTS ...DONE!
NS EXITING...
danny@danny-Dell-System-Inspiron-N4110:~/te$ ns 120dsdv.tcl

```

Figura 35-2: Ejecución del archivo tcl con 120 vehículos y 5 infraestructuras

Realizado por: Danni Tierra,2018

Paso 17 V2I. Tanto en la Fig. 36-2 como en la Fig. 37-2 se aprecia la simulación de una red VANET con comunicación V2I.

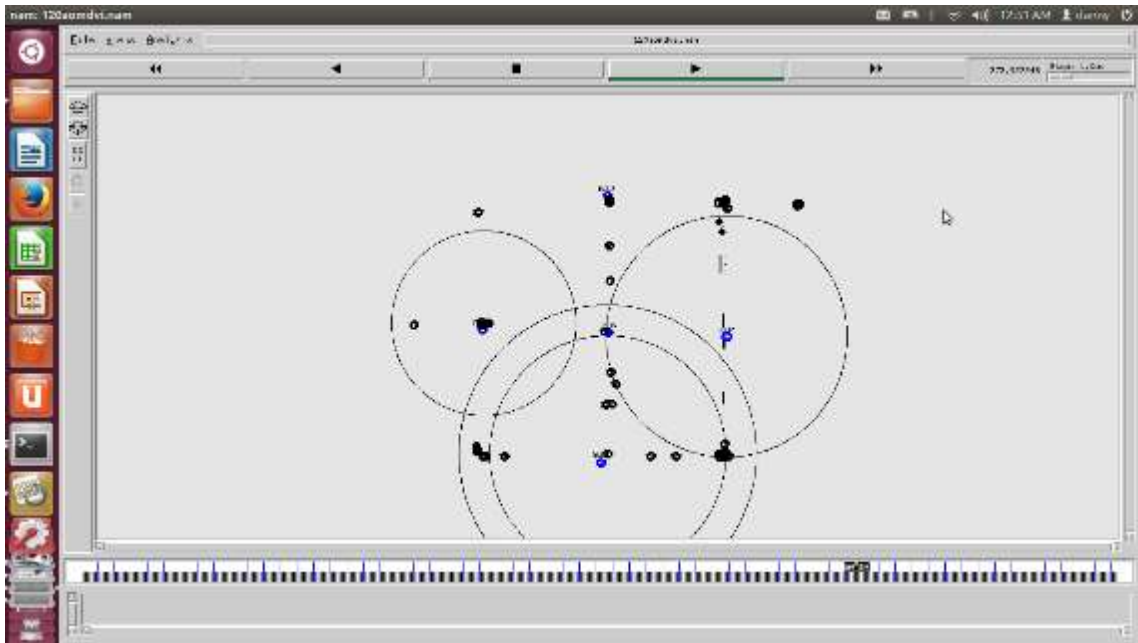


Figura 36-2: Comunicación V2I con 120 vehículos y 5 infraestructuras (AOMDV)

Realizado por: Danni Tierra, 2018

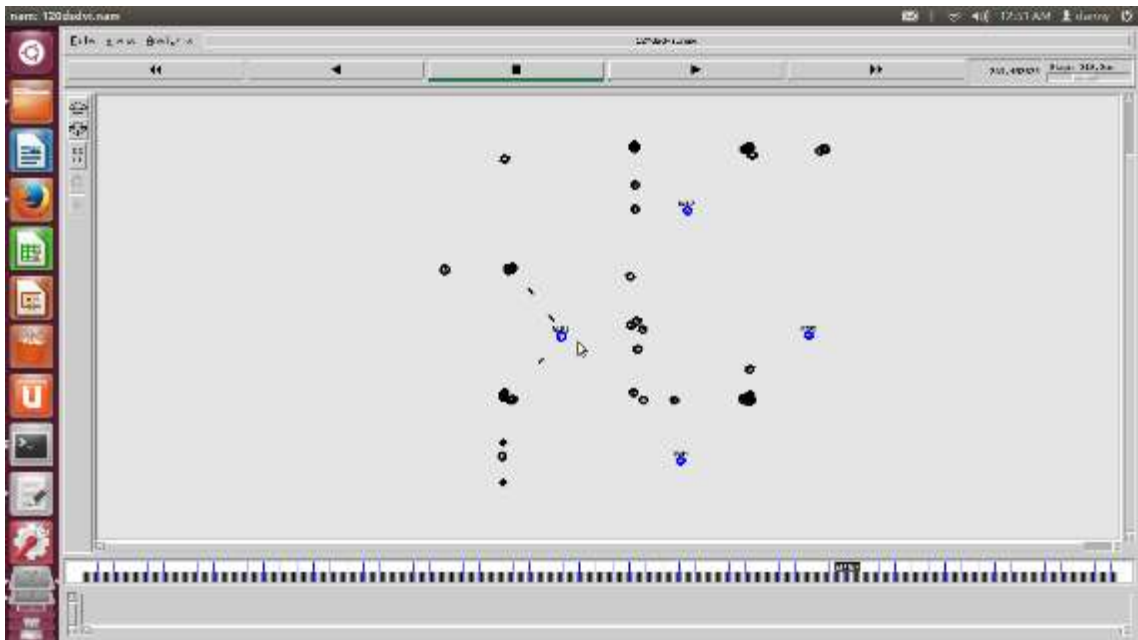


Figura 37-2: Comunicación V2I con 120 vehículos y 5 infraestructuras (DSDV)

Realizado por: Danni Tierra, 2018

CAPÍTULO III

3. MARCO DE RESULTADOS DE LAS SIMULACIONES

En el capítulo tres se analizan los resultados obtenidos en los distintos escenarios, luego de ejecutar las simulaciones y filtrar los resultados para evaluar que protocolo de enrutamiento (AOMDV o DSDV) tiene un mejor desempeño de acuerdo al modelo de movilidad empleado en el escenario, mismos que para este proyecto son el modelo sintético Manhattan y el modelo basado en un generador de tráfico IDM.

3.1. Análisis de resultados de las simulaciones

Los resultados se analizarán en base a los valores obtenidos para los parámetros de: Throughput (rendimiento), PDF (relación de entrega de paquetes), Average End to End Delay (retardo promedio) y NRL (carga de enrutamiento normalizada), al evaluar los resultados se tienen las pautas para determinar cuál es el protocolo de enrutamiento más eficiente para un entorno de simulación de red VANET.

- **Throughput:** Hace referencia al uso eficiente del ancho de banda es decir la cantidad de paquetes que se entregaron exitosamente durante un determinado tiempo. (García Campos 2014, p. 101)
- **PDF o PDR:** Este indicado resulta del cociente de paquetes enviado sobre los recibidos y ayuda a conocer la tasa de pérdida de paquetes. (García Campos 2014, p. 101)
- **Average end to end Delay:** Representa el tiempo de retardo que le toma a los paquetes llegar desde un nodo origen a un nodo destino. (García Campos 2014, p. 101)
- **NRL:** Carga de enrutamiento normalizada que es igual al cociente de paquetes enviado de la capa de enrutamiento sobre los paquetes recibidos en la capa de aplicación. (García Campos 2014, p. 101)

3.1.1. *Análisis de Comunicación V2V*

En esta sección del análisis de resultados se comparan los valores que se obtienen de las simulaciones de la comunicación vehículo a vehículo con distintos números de conexiones y tiempos de simulación.

3.1.1.1. *Simulación del Escenario 1*

El Escenario 1 está definido con 5 conexiones V2V durante un tiempo de simulación de 500 segundos en el que intervienen 10 vehículos y se emplean el protocolo reactivo AOMDV y el protocolo proactivo DSDV, de las simulaciones se obtienen los siguientes resultados.

Tabla 1-3: Datos del Escenario 1

<i>Comunicación V2V</i>				
Tiempo de simulación	500 segundos			
N° de Vehículos	10			
N° de conexiones	5			
Modelo de Movilidad	<u><i>IDM</i></u>		<u><i>MANHATTAN</i></u>	
Protocolo	<u><i>AOMDV</i></u>	<u><i>DSDV</i></u>	<u><i>AOMDV</i></u>	<u><i>DSDV</i></u>
PDR o PDF [%]	99,43	99,34	99,62	99,41
Throughput [kbps]	1797,59	1728,54	1972,38	1925,83
NRL	0,06	0,09	0,06	0,08
Delay End to End [ms]	53	24,37	18,04	9,31

Realizado por: Danni Tierra,2018

En los datos de la Tabla 1-3 se puede apreciar que al aplicar el modelo IDM el protocolo con mejor valor de throughput es AOMDV, con 3.84% superior en relación al throughput de DSDV. Mientras que con el modelo Manhattan el protocolo con mejor desempeño de throughput es AOMDV, superando con 2.36% al protocolo DSDV.

3.1.1.2. Simulación del Escenario 2

El Escenario 2 está definido con 30 conexiones V2V durante un tiempo de simulación de 500 segundos en el que intervienen 60 vehículos y se emplea el protocolo reactivo AOMDV y el protocolo proactivo DSDV, de las simulaciones se obtienen los siguientes resultados.

Tabla 2-3: Datos del Escenario 2

<i>Comunicación V2V</i>				
Tiempo de simulación	500 segundos			
N° de Vehículos	60			
N° de conexiones	30			
Modelo de Movilidad	<u><i>IDM</i></u>		<u><i>MANHATTAN</i></u>	
Protocolo	<u><i>AOMDV</i></u>	<u><i>DSDV</i></u>	<u><i>AOMDV</i></u>	<u><i>DSDV</i></u>
PDR o PDF [%]	99,21	98,56	99,31	99,23
Throughput [kbps]	1865,48	1686,55	2103,07	2002,54
NRL	0,4	2,77	0,38	2,46
Delay End to End [ms]	12,14	9,44	60,85	100,75

Realizado por: Danni Tierra,2018

En los datos de la Tabla 2-3 se puede apreciar que el protocolo con mejor valor de throughput es AOMDV, superando a DSDV con 9.59% en relación al throughput del modelo IDM y con 4.78% para el modelo Manhattan.

3.1.1.3. Simulación del Escenario 3

El Escenario 3 está definido con 60 conexiones V2V durante un tiempo de simulación de 500 segundos en el que intervienen 120 vehículos y se emplea el protocolo reactivo AOMDV y el protocolo proactivo DSDV, de las simulaciones se obtienen los siguientes resultados.

Tabla 3-3: Datos del Escenario 3

<i>Comunicación V2V</i>	
Tiempo de simulación	500 segundos
N° de Vehículos	120
N° de conexiones	60

Modelo de Movilidad	<i>IDM</i>		<i>MANHATTAN</i>	
	<i>AOMDV</i>	<i>DSDV</i>	<i>AOMDV</i>	<i>DSDV</i>
PDR o PDF [%]	99,37	99,07	99,33	99,16
Throughput [kbps]	1687,47	1560,89	1813,68	1701,49
NRL	0,87	16,25	0,79	16,58
Delay End to End [ms]	82,74	85,54	3,77	5,82

Realizado por: Danni Tierra,2018

En los datos de la Tabla 3-3 se puede apreciar que el protocolo con mejor valor de throughput es AOMDV, superando a DSDV con 7.50% en relación al throughput del modelo IDM y con 6.19% para el modelo Manhattan.

3.1.2. *Análisis de Comunicación V2I*

En este apartado se tiene el análisis de resultados y comparación de los valores que fueron obtenidos de las simulaciones de la comunicación V2I con distintos números de conexiones y tiempos de simulación iguales a los empleados en la comunicación V2V.

3.1.2.1. *Simulación del Escenario 4*

Tabla 4-3: Datos del Escenario 4

Comunicación V2I

Tiempo de simulación	500 segundos			
N° de Vehículos	10			
N° de conexiones	5			
Modelo de Movilidad	<i>IDM</i>		<i>MANHATTAN</i>	
	<i>AOMDV</i>	<i>DSDV</i>	<i>AOMDV</i>	<i>DSDV</i>
PDR o PDF [%]	95,03	94,15	99,69	99,8
Throughput [kbps]	632,96	534,07	1594,75	1557,31
NRL	0,27	0,34	0,1	0,11
Delay End to End [ms]	113,98	86,23	81,12	104,81

Realizado por: Danni Tierra,2018

El Escenario 4 está definido con 5 conexiones V2I durante un tiempo de simulación de 500 segundos en el que intervienen 10 vehículos y se emplea el protocolo reactivo AOMDV y el protocolo proactivo DSDV, de las simulaciones se obtienen los siguientes resultados.

En los datos de la Tabla 4-3 se puede apreciar que el protocolo con mejor valor de throughput es AOMDV, superando a DSDV con un 15.62% en relación al throughput del modelo IDM y con 2.35% para el modelo Manhattan.

3.1.2.2. Simulación del Escenario 5

El Escenario 5 está definido con 30 conexiones V2I durante un tiempo de simulación de 500 segundos en el que intervienen 60 vehículos y se emplea el protocolo reactivo AOMDV y el protocolo proactivo DSDV, de las simulaciones se obtienen los siguientes resultados.

Tabla 5-3: Datos del Escenario 5

Comunicación V2I

Tiempo de simulación	500 segundos			
N° de Vehículos	60			
N° de conexiones	30			
Modelo de Movilidad	<u><i>IDM</i></u>		<u><i>MANHATTAN</i></u>	
Protocolo	<u><i>AOMDV</i></u>	<u><i>DSDV</i></u>	<u><i>AOMDV</i></u>	<u><i>DSDV</i></u>
PDR o PDF [%]	98,98	99,26	97,74	98,47
Throughput [kbps]	1288,39	1199,77	321,42	388,34
NRL	0,69	3,98	2,79	14,91
Delay End to End [ms]	67,22	8,79	343,98	34,66

Realizado por: Danni Tierra, 2018

En los datos de la Tabla 5-3 se puede apreciar que el protocolo con mejor valor de throughput es AOMDV, superando a DSDV con un 6.88% en el modelo IDM. Mientras que al aplicar el modelo Manhattan el protocolo con mejor valor de throughput es DSDV que supera a AOMDV con un 17.23%.

3.1.2.3. Simulación del Escenario 6

El Escenario 6 está definido con 60 conexiones V2I durante un tiempo de simulación de 500 segundos en el que intervienen 120 vehículos y se emplea el protocolo reactivo AOMDV y el protocolo proactivo DSDV, de las simulaciones se obtienen los siguientes resultados.

Tabla 6-3: Datos del Escenario 6

<i>Comunicación V2I</i>				
Tiempo de simulación	500 segundos			
N° de Vehículos	120			
N° de conexiones	60			
Modelo de Movilidad	<i>IDM</i>		<i>MANHATTAN</i>	
Protocolo	<i>AOMDV</i>	<i>DSDV</i>	<i>AOMDV</i>	<i>DSDV</i>
PDR o PDF [%]	98,06	98,79	99	99,02
Throughput [kbps]	739,71	807,12	1065,73	1010,56
NRL	2,4	46	1,67	34,17
Delay End to End [ms]	10,37	107,94	113,92	125,97

Realizado por: Danni Tierra, 2018

En los datos de la Tabla 6-3 se puede apreciar que para el modelo IDM el protocolo con mejor valor de throughput es DSDV, superando al protocolo AOMDV con un 8.35%. Mientras que en el modelo Manhattan el protocolo con mejor valor de throughput es AOMDV superando a DSDV con un 5.18%.

3.2. Comparación de los resultados

Una vez que se recopilaron los datos de las simulaciones bajo las condiciones establecidas en el Capítulo 2, se comparan los valores de los parámetros mencionados en la sección 3.1 (Capítulo 3) para poder determinar el modelo de movilidad con mayor rendimiento en los escenarios de este proyecto de titulación.

3.2.1. Comparación de Comunicación V2V

3.2.1.1. PDR o PDF (Relación de entrega de paquetes)

En el Grafico 1-3 se representa el parámetro PDR de las simulaciones, con un valor superior al 98,5% tanto para los modelos de movilidad IDM como para Manhattan, independientemente del protocolo de ruteo y el número de conexiones. Este alto nivel de entrega exitosa de paquetes, se debe a la aplicación de la tecnología IEEE 802.11p en nuestros escenarios.

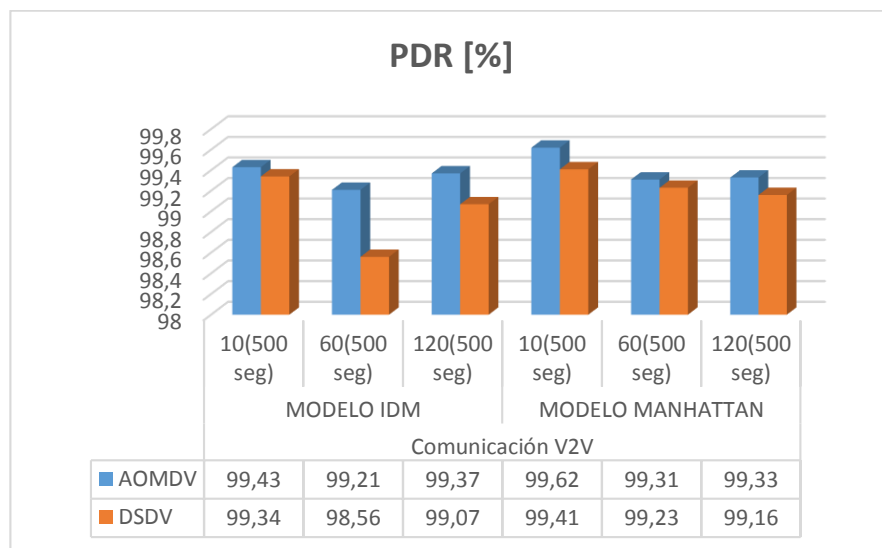


Gráfico 1-3: Parámetro PDR de 5, 10 y 30 conexiones V2V

Realizado por: Danni Tierra, 2018

3.2.1.2. Throughput (Rendimiento)

El Grafico 2-3 presenta el resumen de la métrica Throughput en las simulaciones, se evidencia que el aprovechamiento eficiente del ancho de banda del protocolo AOMDV supera en todos los casos al protocolo DSDV, independientemente del modelo de movilidad que se empleó. Lo que implica que la velocidad de envío de información por parte de AOMDV en todos los casos es superior a DSDV.

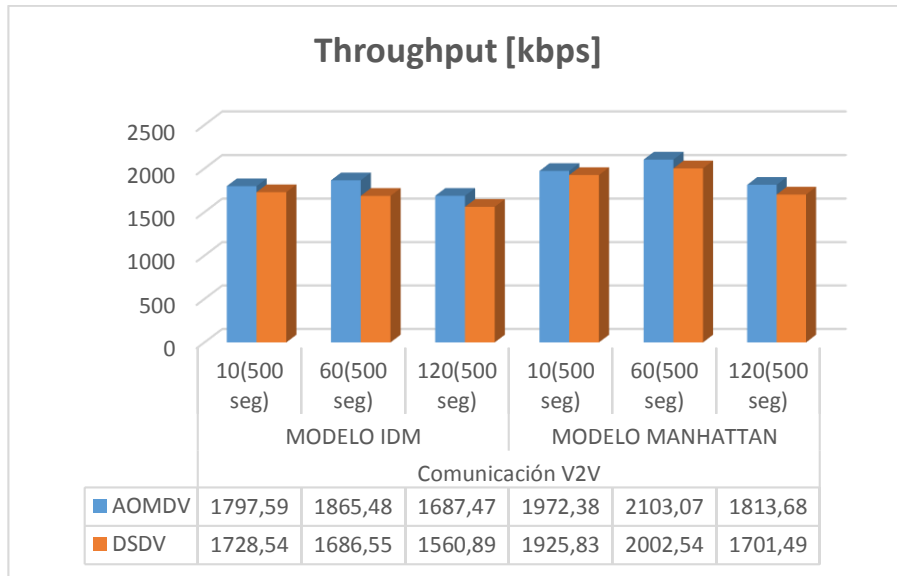


Gráfico 2-3: Parámetro Throughput de 5, 10 y 30 conexiones V2V.

Realizado por: Danni Tierra, 2018

3.2.1.3. *NRL (Carga de enrutamiento normalizada)*

El Gráfico 3-3 representa el parámetro Carga de Enrutamiento Normalizada, observándose que para esta métrica el protocolo AOMDV genera una menor sobrecarga de enrutamiento, lo que lo hace más rápido y eficiente que su par DSDV independientemente del modelo de movilidad que se aplique en el escenario de simulación.

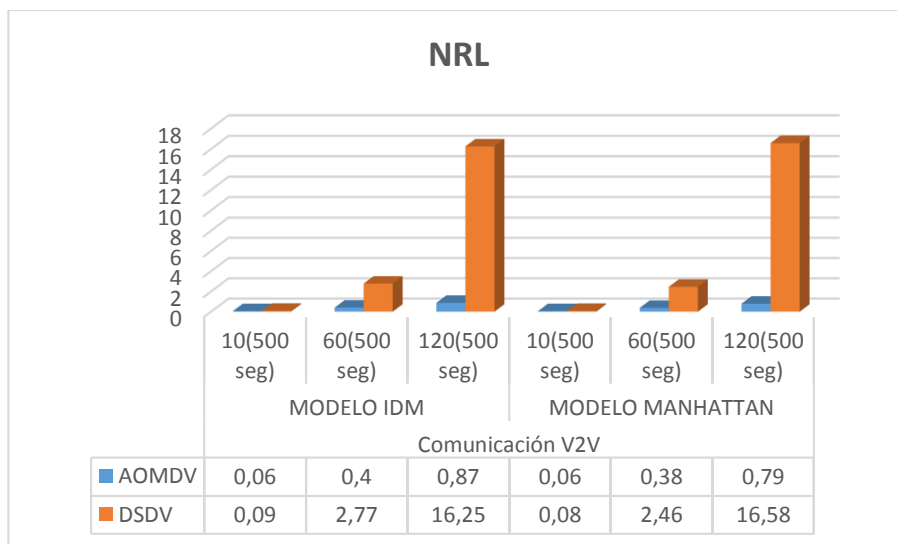


Gráfico 3-3: Parámetro NRL de 5, 10 y 30 conexiones V2V.

Realizado por: Danni Tierra, 2018

3.2.1.4. Average End to End Delay (Retardo promedio)

En el Grafico 4-3 se representa el parámetro Delay End to End, en él se evidencia que, para un número pequeño de conexiones y tiempos relativamente extensos de trabajo, el protocolo AOMDV registró mayores retardos que DSDV. En contrapartida, para un valor alto de conexiones, ocurrió lo contrario, es decir, que el protocolo AOMDV presentó menores retardos que DSDV.

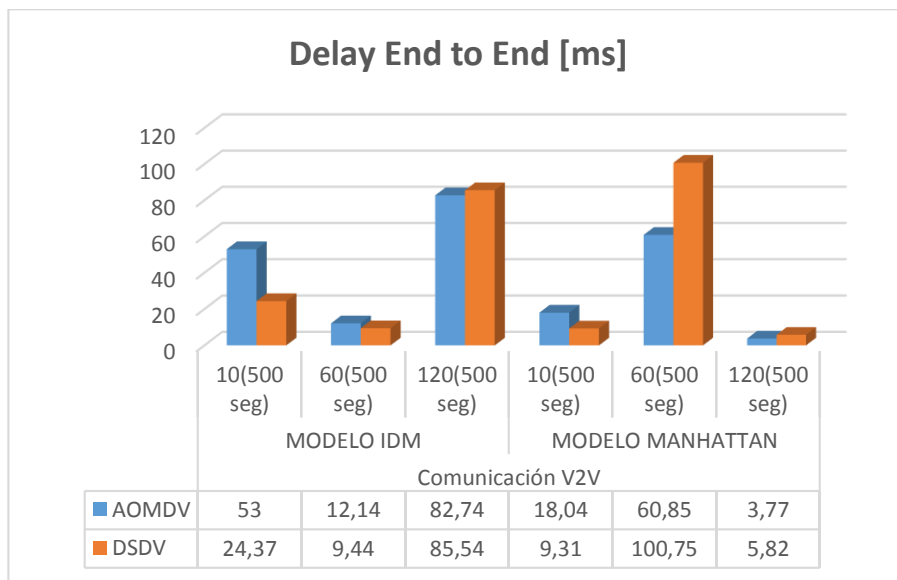


Gráfico 4-3: Parámetro Delay End to End de 5, 10 y 30 conexiones V2V

Realizado por: Danni Tierra, 2018

3.2.2. Comparación de Comunicación V2I

3.2.2.1. PDR o PDF (Relación de entrega de paquetes)

En el Grafico 5-3 se representa el parámetro PDR de las simulaciones, con un valor superior al 99.1% tanto para los modelos de movilidad IDM como para Manhattan, independientemente del protocolo de ruteo y el número de conexiones. Este alto nivel de entrega exitosa de paquetes, se debe a la aplicación de la tecnología IEEE 802.11p en nuestros escenarios.

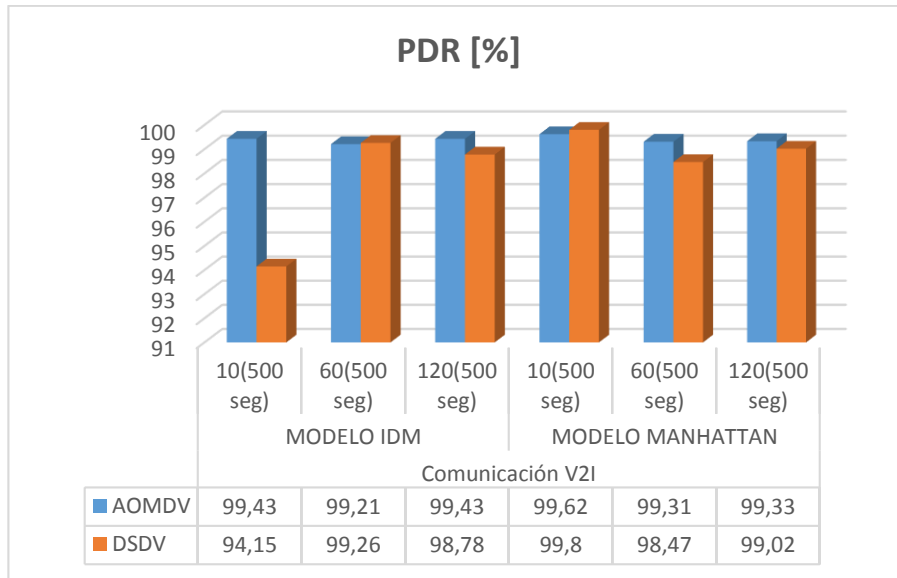


Gráfico 5-3: Parámetro PDR de 5, 10 y 30 conexiones V2I.

Realizado por: Danni Tierra, 2018

3.2.2.2. Throughput (Rendimiento)

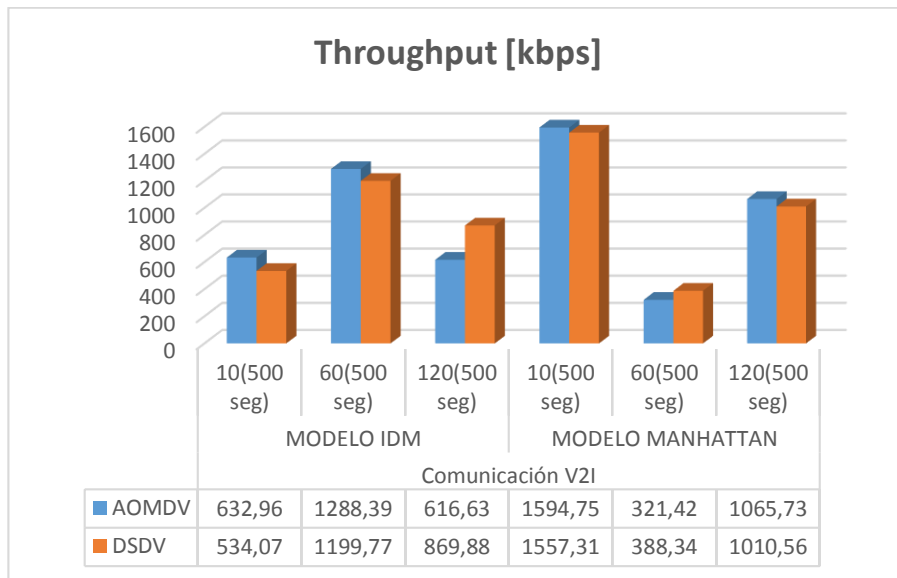


Gráfico 6-3: Parámetro Throughput de 5, 10 y 30 conexiones V2I.

Realizado por: Danni Tierra, 2018

El Gráfico 6-3 representa el resumen de la métrica Throughput en las simulaciones, en él se evidencia que, para un número pequeño de conexiones con tiempos extensos de trabajo, el protocolo AOMDV registró mayor throughput que DSDV independientemente del modelo de

movilidad empleado, mientras que con un valor alto de conexiones bajo el modelo IDM el Throughput del protocolo DSDV es superior al de AOMDV, en contraparte con el modelo Manhattan el protocolo con mejor valor de Throughput es AOMDV pero con un margen mínimo frente a DSDV.

3.2.2.3. *NRL (Carga de enrutamiento normalizada)*

En el Grafico 7-3 se representa el parámetro Carga de Enrutamiento Normalizada, observándose que la métrica NRL del protocolo AOMDV genera una sobrecarga de enrutamiento menor para todos los casos, para casos con un alto valor de conexiones se evidencia que el NRL generado por AOMDV es menor al de DSDV con un alto margen de diferencia.

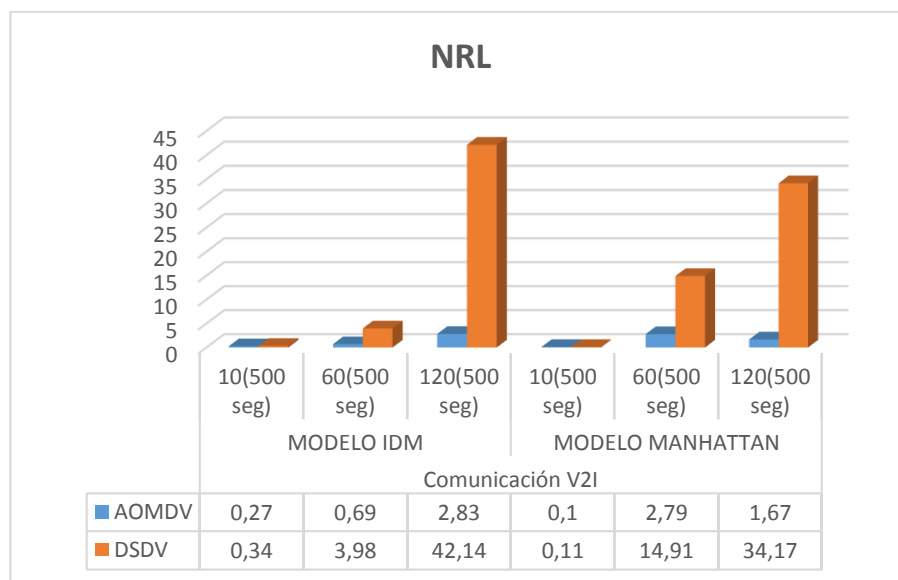


Gráfico 7-3: Parámetro NRL de 5, 10 y 30 conexiones V2I.

Realizado por: Danni Tierra, 2018

3.2.2.4. *Average End to End Delay (Retardo promedio)*

En el Grafico 8-3 se representa el parámetro Delay End to End, en él se evidencia que, el protocolo AOMDV registró mayores retardos que DSDV en todos los casos en los que se emplea el modelo de movilidad IDM. En contrapartida, para el modelo Manhattan, ocurrió lo contrario, es decir, que el protocolo AOMDV presentó menores retardos que DSDV en la mayoría de los casos.

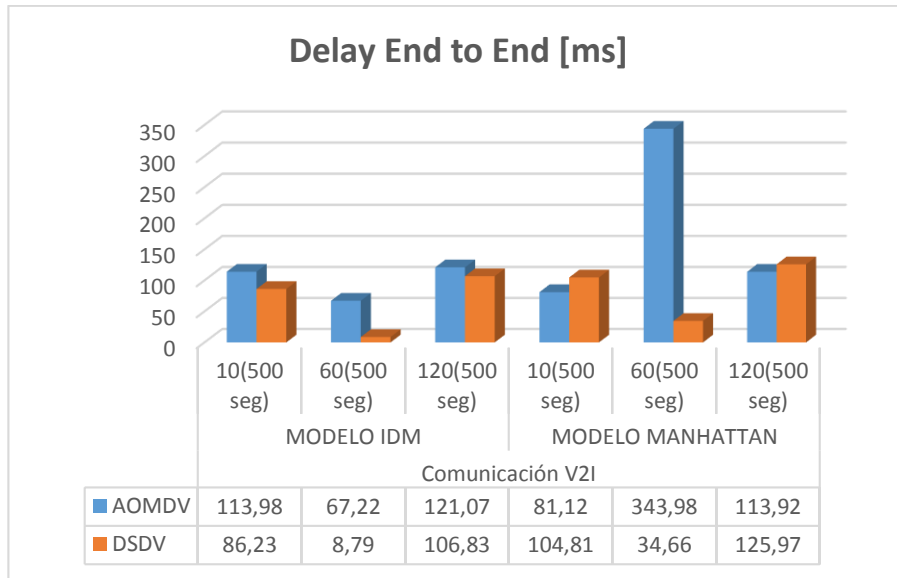


Gráfico 8-3: Parámetro Delay End to End de 5, 10 y 30 conexiones V2I

Realizado por: Danni Tierra, 2018

CONCLUSIONES

- Los protocolos más empelados en redes VANET son los reactivos y proactivos entre los que más destacan tanto en documentos de titulación como en investigaciones son los protocolos DSDV y AODV. En cuanto a modelos de movilidad los sintéticos son los menos confiables ya que no ofrecen datos cercanos a la realidad como es el caso de los modelos basados en generadores de tráfico.
- En base a la información recopilada, los protocolos más empleados para redes VANET son DSDV, protocolo que administra sus rutas en base a una tabla de distancias más cortas hacia los nodos de la red, dichas distancias deben ser actualizadas periódicamente. AODV, por su parte, realiza el cálculo de las rutas solo bajo petición; la evolución del protocolo AODV llamada AOMDV ofrece la mejora de contar con rutas de respaldo manteniendo además una tabla de enrutamiento que permite conocer toda la topología de la red y al igual que AODV calcula las rutas bajo petición. Esto hace que AOMDV reduzca la sobrecarga que implica actualizar las rutas de la forma en que lo hace DSDV.
- El modelo de movilidad sintético Manhattan, se acopla a la topología de nuestro escenario, mientras que se optó por IDM como modelo generador de tráfico ya que en este la movilidad no solo depende del automóvil predecesor, sino que también depende de sí mismo y de los autos que lo rodean.
- NS-2 y SUMO son simuladores que tienen una amplia variedad de funciones que permiten un mejor control de las variables que intervienen en la simulación de escenarios ayudándonos a obtener datos cercanos a la realidad, la herramienta MOVE permite acoplar estos simuladores y todos estos programas son libres por lo que no es necesario adquirir una licencia para su uso.
- Al comparar los valores de Throughput, PDF, Average End to End Delay y NRL se evidencia un mejor desempeño de ruteo del protocolo AOMDV frente a DSDV, pero el margen es mínimo (con una diferencia máxima de hasta el 9.59%), por lo que se concluye que para una

comunicación V2V independientemente del modelo de movilidad y número de conexiones se puede emplear tanto el protocolo reactivo AOMDV como el proactivo DSDV.

- Para comunicaciones V2I el margen de diferencia entre AOMDV y DSDV es más evidente (con una diferencia máxima de hasta el 15.62%) en la mayoría de los casos con superioridad por parte del protocolo reactivo AOMDV, por lo que se concluye que en escenarios de comunicación V2I, AOMDV tiene un mejor rendimiento en el encaminamiento de los paquetes.
- En general tanto para comunicaciones V2V como para V2I el protocolo que mayor eficiencia evidencio es AOMDV indistintamente del número de conexiones. Recalcando que en conexiones V2V, AOMDV es superior con un margen relativamente bajo mientras que en una comunicación V2I la superioridad de AOMDV sobre DSDV es más evidente. Este hecho se debe a que AOMDV además de realizar el cálculo de rutas bajo petición de los nodos también nos ofrece rutas de respaldo para dar una mayor garantía de la entrega de paquetes a su destino.

RECOMENDACIONES

- Emplear los protocolos proactivos y reactivos que cuenten con mayor documentación, además para obtener simulaciones cercanas a la realidad en entornos VANET es recomendable utilizar simuladores de movilidad microscópicos con modelos de movilidad basados en generadores de tráfico y no en modelos sintéticos.
- Revisar si el protocolo que se va a utilizar viene por defecto en su versión de NS-2 caso contrario instalar previamente los parches necesarios antes de realizar la simulación en cuanto a los modelos de movilidad se recomienda utilizar el modelo que mejor se adapte al entorno a simular.
- Revisar que la versión de Java JDK sea compatible con MOVE para que se ejecute la GUI, al usar MOVE se recomienda compilar el archivo .sumo.tr para poder exportar los datos a la herramienta de NS-2 sin ningún inconveniente. Además, al establecer las conexiones de los nodos en el archivo .tcl es mejor realizarlas por líneas de comando en lugar de hacerlo a través de la GUI para optimizar el tiempo de configuración del archivo.
- No cancelar la ejecución de los archivos .tcl durante la compilación, ya que de cancelar la ejecución el archivo quedará incompleto y no contendrá todas las trazas del movimiento de los nodos.
- Se recomienda emplear el protocolo proactivo DSDV siempre y cuando se cumplan estas condiciones; en comunicaciones V2I basadas en un modelo de movilidad sintético con un número medio de conexiones o cuando se tiene un alto número de conexiones, pero con un modelo de movilidad basado en simuladores de tráfico vehicular.

BIBLIOGRAFÍA

- 1 **Al-Doori, M.** *Directional Routing Techniques in VANET (tesis)* .[en línea]. Universidad de Montfort, Laboratorio de investigación de tecnología y software. Leicester, Reino Unido. 2011. p 87. Disponible en: <http://hdl.handle.net/2086/5504>
- 2 **Alaminos Chica, A. Y Castejón, J.L.** *Elaboración, Análisis E Interpretación De Encuestas, Cuestionarios Y Escalas De Opinión (tesis)* .[en línea]. Universidad de Alicante. Raspeig, España. 2006. pp. 98-100. Disponible en: <http://rua.ua.es/dspace/handle/10045/20331>
- 3 **Albero, T.** *Estudio y análisis del comportamiento de redes ad hoc móviles: Supervisión de procesos en entornos industriales*. Valencia, España: Editorial Académica Española, 2012, pp. 24-36.
- 4 **Anggoro, R.,et.al.** “Performance Comparison of Mobility Generator C4R and MOVE using Optimized Link State Routing (OLSR)”. *IOSR Journal of Engineering (IOSRJEN)*, vol. 06, no. 11 (2016), pp. 25-29.
- 5 **Auparay, E., Anggoro, R. Y Ijtihadie, R.M.** “Performance Study of Wireless Access in Vehicular Environment 802 . 11p in Ad-Hoc on Demand Distance Vector Routing (AODV) with Propagation Models Nakagami and Two Ray Ground”. *PTEK Journal of Proceedings Series*, no. 1 (2018), pp. 99-102.
- 6 **Bitam, S. Y Mellouk, A.** *Vehicular Ad Hoc Networks. Bio-inspired routing protocols for vehicular ad-hoc networks*. Londres, Inglaterra: ISTE-WILEY, 2014, pp. 1-23. ISBN 9781848216631. [Consulta: 23 enero 2018]. Disponible en: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119004967.ch1>
- 7 **Boukenadil, B.** “Importance Of Realistic Mobility Models For Vanet Network.” *International Journal of Computer Networks & Communications (IJCNC)*, vol. 6, no. 5 (2014), (Argelia) pp. 175-182.
- 8 **Boukenadil, B. Y Feham, M..** “Mobility Models for VANET simulation”. *2nd International Conference on New Technologies & Communication*. 2015. (Argelia) pp. 1-5.

- 9 **Caiza Changoluisa, M.D.** *Simulación de una VANET para un sector del centro histórico de la ciudad de Quito como apoyo a la movilidad vehicular (tesis)* .[en línea]. Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica. Quito, Ecuador. 2017. pp. 1-3, 11-13, 15-17, 24-27. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/17321>
- 10 **Calvo, A.P., et al.** *User Guide CityMob for Roadmaps C4R*. [en línea]. Teruel, España, 2011.[Consulta: 10 julio 2017]. Disponible en: <http://www.grc.upv.es/Software/c4r/usermanual.pdf>.
- 11 **Chalmeta Ugas, J.** *Estudio y análisis de prestaciones de redes móviles Ad Hoc mediante simulaciones NS-2 para validar modelos analíticos (tesis)* .[en línea]. Universidad Politécnica de Cataluña. Cataluña, España. 2009. pp. 31-34 . Disponible en: https://upcommons.upc.edu/bitstream/handle/2099.1/8374/PFC_Jordi_Chalmeta.pdf
- 12 **Chirinos Cadavid, S.** *Evaluación de los protocolos de enrutamiento AODV y OLSR en redes VANET (tesis)* .[en línea]. Pontificia Universidad Javeriana, Facultad de Ingeniería Electrónica. Bogotá, Colombia. 2013. pp. 4-30. Disponible en: <https://repository.javeriana.edu.co/bitstream/handle/10554/13616/ChirinosCadavidSantiago2013.pdf?sequence=1&isAllowed=y>
- 13 **Dirección General De Tránsito De España**, 2009. *Distracciones al volante*. [en línea]. Madrid, España. [Consulta: 3 julio 2017]. Disponible en: http://www.dgt.es/PEVI/documentos/catalogo_recursos/didacticos/did_adultas/Distracciones_al_volante.pdf.
- 14 **Orozco, A.M., Llano, G. Y Michoud, R.** “Redes vehiculares Ad-hoc: aplicaciones basadas en simulación”. *Ingenium, Ciencia & Tecnología*, vol. 6, no. 12 (2012), (Colombia) pp. 11-22.
- 15 **Orozco, O.A., Chavarro, D.F. y Calderón, O.J.** "Impact of the speed and mobility model in a data communication for a vehicular network". *Revista Entre Ciencia e Ingeniería*, no. 15 (2014), (Colombia) pp. 62-70.
- 16 **Prasad, R.** *OFDM for wireless communications systems*. Norwood, MA, USA: Artech House, 2004, pp. 11-13.

- 17 **Ripoll Cerezo, P.G.** *Estudio del simulador de redes vehiculares veins (tesis)* .[en línea]. Universidad Carlos III de Madrid, Ingeniería de Telecomunicaciones. Madrid, España. 2012. pp. 25-26, 36-39, 47. Disponible en: https://e-archivo.uc3m.es/bitstream/handle/10016/16704/PFC_PabloGonzalez-RipollCerezo.pdf?sequence=1
- 18 **Rodríguez Avalos, E.D. y Tucto Succhil, E.E.** *Diseño lógico y simulación de un esquema de red WLAN basada en el análisis de la propagación (tesis)* .[en línea]. Universidad de Trujillo, Facultad de Ciencias Físicas y Matemáticas, Escuela Académico Profesional de Informática. Trujillo, Perú. 2014. pp 24-25. Disponible en: [http://dspace.unitru.edu.pe/bitstream/handle/UNITRU/2420/Rodríguez Avalos Edgar D. - Tucto Succhil Elvis E..pdf?sequence=1&isAllowed=y](http://dspace.unitru.edu.pe/bitstream/handle/UNITRU/2420/Rodríguez_Avalos_Edgar_D._Tucto_Succhil_Elvis_E..pdf?sequence=1&isAllowed=y)
- 19 **Roy, R.R.** *Handbook of Mobile Ad Hoc Networks for Mobility Models*. New York, USA: Springer, 2011, pp. 4,24, 35-40, 65-68, 173-174.
- 20 **Ruiz, P.J.F., et al.** Deploying ITS Scenarios Providing Security and Mobility Services Based on IEEE 802.11p Technology. *Vehicular Technologies - Deployment and Applications*. Rijeka, Croacia: Intech, 2013, pp. 71-93. ISBN 9789537619343. [Consulta: 25 julio 2017]. Disponible en: <http://dx.doi.org/10.5772/55285>
- 21 **Stallings, W.** *Comunicaciones y Redes de Computadores*. 7ª ed. Madrid, España: Pearson Prentice Hall, 2004, pp. 286-300
- 22 **Tanenbaum, A.S. y Wetherall, D.J.** *Redes De Computadoras*. 5ª edición. Naucalpan de Juárez, México: Pearson Prentice Hall, 2012, pp. 67-79, 257-267, 291, 357, 360-370, 375-376.
- 23 **Valencia Ayala, S.M. y Risueño Benítez, E.D.** *Diseño E Implementación De Una Red Manet Con Dispositivos De Comunicación Móvil (tesis)* .[en línea]. Universidad de las Américas, Facultad de Ingenierías y Ciencias Agropecuarias. Quito, Ecuador. 2017. pp. 34-36, 40-41. Disponible en: <http://dspace.udla.edu.ec/bitstream/33000/7399/5/UDLA-EC-TIERI-2017-17.pdf>

- 24 **Werchau, J.P. Y Nazar, P.** *Estándar IEEE 802.11 X de las WLAN.* [en línea]. Tucumán, Argentina, 2013. [Consulta: 3 julio 2017]. Disponible en: http://www.edutecne.utn.edu.ar/monografias/standard_802_11.pdf
- 25 **Yunquera, J.** *Diseño de una red WI-FI para la E.S.I. (tesis)* .[en línea]. Universidad de Sevilla, Escuela Superior de Ingenieros. Sevilla, España. 2005. pp 36-51. Disponible en: <http://bibing.us.es/proyectos/abreproy/11138/fichero/memoria%252FCapítulo+3.pdf>

ANEXOS

ANEXO A. Instalación de herramientas

Como primer paso se debe instalar el sistema operativo Ubuntu en un equipo para proceder a la instalación de las herramientas seleccionadas que permitirán la simulación y configuración de los parámetros de la red VANET.

Instalación de Ubuntu

Se procede a la instalación de Ubuntu 12.04, se usa esta versión de Ubuntu ya que es la última versión compatible con MOVE y si se procede a instalar versiones más actuales esto ocasionará problemas de compatibilidad en el software y su funcionamiento no será correcto por lo que no se podrá hacer uso de la herramienta MOVE.

Como prerequisite para que el computador sea capaz de reconocer la unidad que contiene el sistema operativo es necesario realizar algunos ajustes en el *BIOS*, se ingresa en el BIOS presionando la tecla F2 o F12 según la marca de computador que sea, pero usualmente estas dos teclas permiten el ingreso.

Una vez que ya se ingresa en el *BIOS* se debe desplazar hasta la sección del *BOOT* y cambiar del modo *LEGACY* al modo *UEFI* para que permita reconocer la unidad y modificar la prioridad de arranque poniendo como primera opción la lectura de CDROM o USB según sea el caso. Estos cambios se pueden observar en la Fig. 1-A.

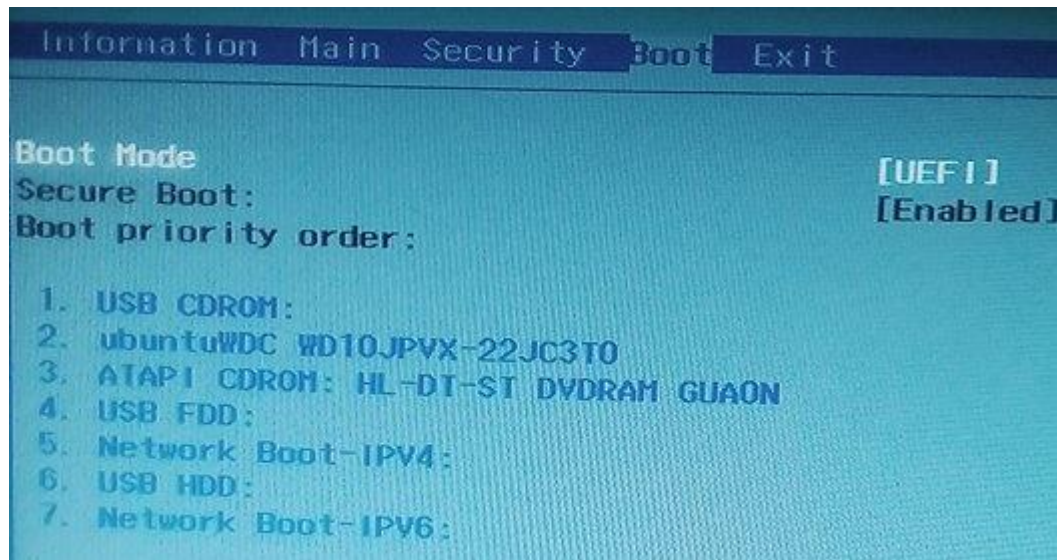


Figura 1-A: Interfaz BIOS

Realizado por: Danni Tierra,2018

Para iniciar con el proceso de instalación del sistema operativo (SO) se debe ingresar el *CD* en la unidad *CD/DVD* o flash en el puerto USB. En el proceso de arranque del *CD* la Fig. 2-A nos indica las opciones para proceder con la instalación.

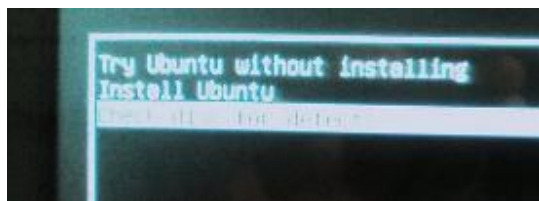


Figura 2-A: Arranque del *CD* que contiene el SO

Realizado por: Danni Tierra,2018

Seleccionar con la tecla *ENTER* la opción *Install Ubuntu* y seguir las indicaciones que nos indica el asistente gráfico de instalación. El asistente hace que el proceso sea intuitivo y fácil de instalar. En la Fig. 3-A se observa la interfaz de Ubuntu una vez instalado correctamente el SO.

libproj-dev libgdal-dev libxerces-c2-dev libfox-1.6-dev	
export CC=gcc-4.4 CXX=g++-4.4	Cambia la versión del compilador para C++
cp /Downloads/sumo-src-0.12.3.tar.gz /home/danny/	Copia el archivo <i>sumo-src-0.12.3.tar.gz</i> que se encuentra en descargas hacia el home
cd /home/danny	Ingresa al directorio de usuario
tar xzvf sumo-src-0.12.3.tar.gz	Descomprime el archivo
cd sumo-0.12.3/	Ingresa a la carpeta de sumo
./configure	Ejecuta el script <i>configure</i>
make	Permite la generación automática de las dependencias de la herramienta (Archivos binarios)
sudo make install	Pone los archivos binarios en rutas accesibles para el usuario
sumo-gui	Abre la interfaz gráfica de SUMO que se puede apreciar en la Fig. 4-A

Realizado por: Danni Tierra,2018

Fuente: <http://www.techerina.com/2015/03/sumo-0123-and-move-installation-in-ubuntu-1204-lts.html>

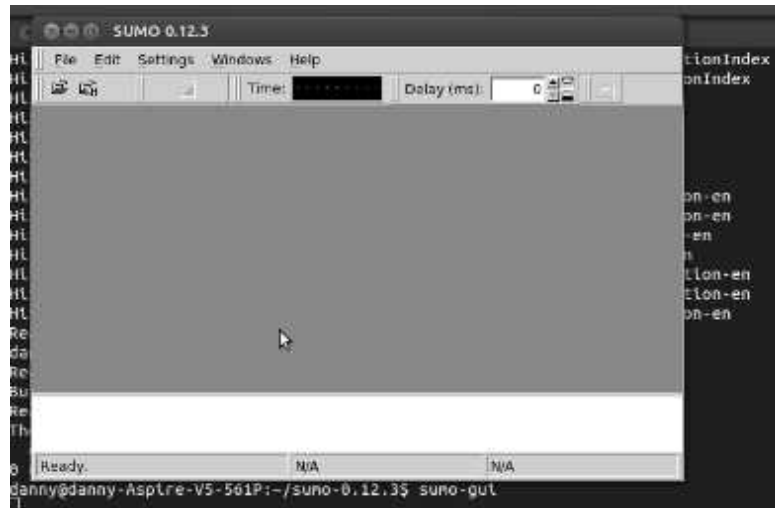


Figura 4-A: Interfaz de SUMO

Realizado por: Danni Tierra, 2018.

Instalación de NS-2

Proceder a descargar la herramienta NS-2 desde la siguiente dirección: <https://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.35/>, adicional a esto también se descargan los archivos del patch OLSR para adicionar este protocolo de enrutamiento a la herramienta. El patch se debe descargar desde <https://sourceforge.net/projects/um-olsr/files/um-olsr/1.0/> para luego continuar con la instalación.

En la tabla 2-A se observa el orden de los comandos a ingresar y una breve descripción. Para ingresar los comandos se debe ingresar a la consola de Ubuntu presionando las teclas Ctrl + Alt + T.

Tabla 2-A: Comparación de Simuladores de tráfico vehicular

Comandos	Descripción
sudo apt-get update	Actualiza la lista de paquetes disponibles de Ubuntu
sudo apt-get upgrade	Actualiza el sistemas y las aplicaciones que están instaladas

cp /Downloads/ns-allinone-2.35.tar.gz /home/danny/	Copia el archivo <i>sumo-src-0.12.3.tar.gz</i> desde la carpeta Descargas hacia el directorio del usuario.
cp /Downloads/um-olsr-1.0.tgz /home/danny/	Copia el archivo <i>um-olsr-1.0.tgz</i> desde la carpeta Descargas hacia el directorio del usuario.
cd /home/danny	Ingresa al directorio del usuario
tar xzvf ns-allinone-2.35.tar.gz	Descomprime el archivo que contiene el instalador de NS-2
tar xzvf um-olsr-1.0.tgz	Descomprime el archivo que contiene el patch de OLSR
mv /um-olsr /ns-allinone-2.35/ns-2.35/olsr	Mueve la carpeta <i>um-olsr</i> a <i>/ns-allinone-2.35/ns-2.35/</i> cambiándole el nombre por <i>olsr</i> .
sudo apt-get install build-essential autoconf automake gcc xorg-dev tcl-dev tk-dev perl xgraph libxt-dev libx11-dev libxmu-dev	Instala algunas librerías necesarias para que funcione correctamente la herramienta.
cd /ns-allinone-2.35/ns-2.35/	Ingresa a la carpeta <i>ns-2.35</i> .
patch -p1 < olsr/um-olsr_ns- 2.35_v1.0.patch	Ejecuta el patch para agregar OLSR a NS-2. En la Fig. 5-A se observa la instalación correcta del patch.
cd /linkstate	Ingresa a la carpeta <i>linkstate</i>
gedit ls.h	Modificar la línea 137 del archivo <i>ls.h</i> añadiendo <i>this -></i> antes de <i>{erase}</i> , guardar los cambios. En la Fig. 6-A se observa la modificación.
export CC=gcc-4.4 CXX=g++-4.4	Cambia la versión del compilador para C++
cd /ns-allinone-2.35/	Retroceder a la carpeta <i>ns-2.35</i> .

./install	ejecutar el archivo <i>install</i> .
gedit ~/.bashrc	Al final del archivo <i>bashrc</i> añadir las rutas para las librerías y aplicaciones necesarias para la herramienta NS-2. En la Fig. 7-A se observa la modificación y se guardan los cambios.
Ns	Ejecuta la herramienta NS-2. En la Fig. 8-A se observa el terminal si se instaló correctamente el programa.

Realizado por: Danni Tierra,2018

Fuente: <https://gist.github.com/slashesBin/a383559dc1f683c7cd502af9fd39c3ab>

```

danny@danny-Aspire-V5-561P: ~/ns-allinone-2.35/ns-2.35
Processing triggers for libc-bin ...
ldconfig deferred processing now taking place
danny@danny-Aspire-V5-561P:~$ cd ns-allinone-2.35/ns-2.35/linkstate/
danny@danny-Aspire-V5-561P:~/ns-allinone-2.35/ns-2.35/linkstate$ gedit ls.h
^C
danny@danny-Aspire-V5-561P:~/ns-allinone-2.35/ns-2.35/linkstate$ cd ..
danny@danny-Aspire-V5-561P:~/ns-allinone-2.35/ns-2.35$ patch -p1 < olsr/um-olsr_
ns-2.35_v1.0.patch
bash: olsr/um-olsr_ns-2.35_v1.0.patch: No such file or directory
danny@danny-Aspire-V5-561P:~/ns-allinone-2.35/ns-2.35$ patch -p1 < olsr/um-olsr_
ns-2.35_v1.0.patch
bash: olsr/um-olsr_ns-2.35_v1.0.patch: No such file or directory
danny@danny-Aspire-V5-561P:~/ns-allinone-2.35/ns-2.35$ patch -p1 < olsr/um-olsr_
ns-2.35_v1.0.patch
patching file Makefile.in
patching file common/packet.h
patching file queue/priqueue.cc
patching file tcl/lib/ns-agent.tcl
patching file tcl/lib/ns-default.tcl
patching file tcl/lib/ns-lib.tcl
patching file tcl/lib/ns-packet.tcl
patching file trace/cmu-trace.cc
patching file trace/cmu-trace.h
danny@danny-Aspire-V5-561P:~/ns-allinone-2.35/ns-2.35$ cd ..

```

Figura 5-A: Ejecución exitosa del patch de OLSR

Realizado por: Danni Tierra,2018.

```

iterator insert(const key & key, const i & item) {
    typename baseMap::value_type v(key, item);
    pair_iterator_bool ib = baseMap::insert(v);
    return ib.second ? ib.first : baseMap::end();
}

void eraseAll() { this->erase(baseMap::begin(), baseMap::end
()); }

T* findPtr(Key key) {
    iterator it = baseMap::find(key);
    return (it == baseMap::end()) ? (T *)NULL : &
(*it).second;
}
};

```

C/C++/ObjC Header Tab Width: 8 Ln 137, Col 27

Figura 6-A: Modificación del archivo ls.h

Realizado por: Danni Tierra,2018

```

nsake[1]: Leaving directory '/home/danny/ns-allinone-2.35/del8021nr-1.1.4'
ns-allinone package has been installed successfully.
Here are the installation places:
tcsh.5.19: /home/danny/ns-allinone-2.35/{bin,include,lib}
dksh.5.18: /home/danny/ns-allinone-2.35/{bin,include,lib}
otcl: /home/danny/ns-allinone-2.35/otcl-1.14
tcsh: /home/danny/ns-allinone-2.35/tcsh-1.28
ns: /home/danny/ns-allinone-2.35/ns-2.35/ns
nan: /home/danny/ns-allinone-2.35/nan-1.15/nan
xgraph: /home/danny/ns-allinone-2.35/xgraph-12.2
ot lib: /home/danny/ns-allinone-2.35/otlib, edriver, sqbzlib, sqbzns, sqbzccoms,
sqbzntems

-----
Please put /home/danny/ns-allinone-2.35/bin;/home/danny/ns-allinone-2.35/tcsh.5.19/untx;/home/danny/ns-allinone-2.35/kr.5.10/untx
into your PATH environment; so that you'll be able to run itr/tcsh/wish/xgraph.

IMPORTANT NOTICES:
(1) You MUST put /home/danny/ns-allinone-2.35/otcl-1.14, /home/danny/ns-allinone-2.35/lib,
into your LD_LIBRARY_PATH environment variable,
if it complains about X libraries, add path to your X libraries
into LD_LIBRARY_PATH.
if you are using csh, you can set it like:
setenv LD_LIBRARY_PATH $patho
if you are using sh, you can set it like:
export LD_LIBRARY_PATH=$patho
(2) You MUST put /home/danny/ns-allinone-2.35/tcsh.5.19/library into your TCL_LIBRARY
environmental
variable. Otherwise ns/nan will complain during startup.

After these steps, you can now run the ns validation suite with
cd ns-2.35; ./validata

```

```

# bashrc
if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
    . /etc/bash_completion
fi

# PATH
PATH="/home/danny/ns-allinone-2.35/bin:/home/danny/ns-allinone-2.35/tcsh.5.19/untx:/home/danny/ns-allinone-2.35/kr.5.10/untx:/home/danny/ns-allinone-2.35/bin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:/usr/lib:/usr/lib64:/usr/libexec:/usr/local/lib:/usr/local/sbin:/usr/sbin:/usr/local/bin:/usr/local/sbin"

# TCL_LIBRARY
TCL_LIBRARY="/home/danny/ns-allinone-2.35/tcsh.5.19/library"
export TCL_LIBRARY=$TCL_LIBRARY

# LD_LIBRARY_PATH
LD_LIBRARY_PATH="/home/danny/ns-allinone-2.35/lib:/usr/lib:/usr/lib64:/usr/local/lib:/usr/local/sbin:/usr/sbin:/usr/local/bin:/usr/local/sbin"
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH

```

Figura 7-A: Edición de las rutas en el archivo bashrc

Realizado por: Danni Tierra,2018.

```

danny@danny-Aspire-V5-561P:~/ns-allinone-2.35/ns-2.35$ ns
%

```

Figura 8-A: Prueba de instalación de NS-2 exitosa.

Realizado por: Danni Tierra,2018.

Instalación de MOVE

El archivo MOVE no requiere de una instalación propiamente dicha, ya que se trata de una herramienta ejecutable de JAVA. El archivo se debe descargar llenando los campos de la Fig. 9-A en la dirección <https://lens.csie.ncku.edu.tw/MOVE/download.php>.

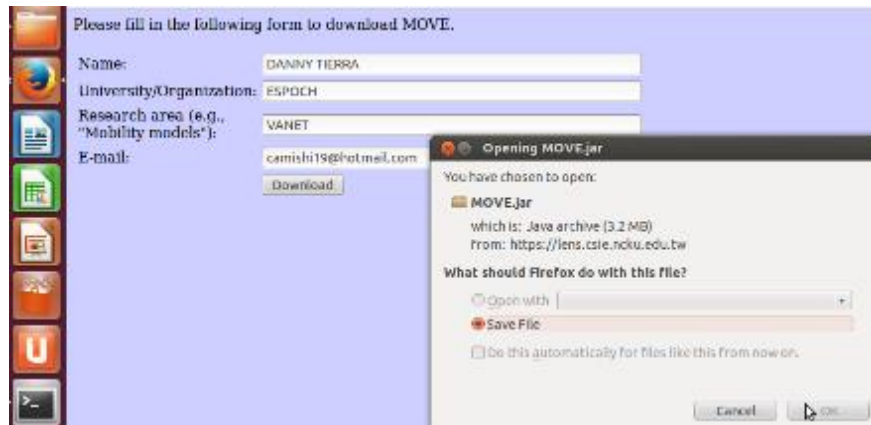


Figura 9-A: Descarga del archivo MOVE

Realizado por: Danni Tierra, 2018.

Para ejecutar el archivo MOVE ingresar la siguiente línea de comando, ingresar a la consola de Ubuntu presionando las teclas Ctrl + Alt + T.

```
java -jar MOVE.jar
```

En la Fig. 10-A se visualiza la GUI de MOVE, donde se identifican claramente dos secciones definidas. *Mobility Model* para modelar los parámetros de la red vehicular y *Traffic Model* para modelar los parámetros de la red de datos.



Figura 10-A: GUI de MOVE

Realizado por: Danni Tierra, 2018.

ANEXO B. Simulación del modelo de movilidad Manhattan

Simulación del entorno VANET basado en el modelo sintético Manhattan

Se procede a realizar la simulación de 120 vehículos con 60 conexiones bajo el modelo de movilidad Manhattan, para la simulación de 60 vehículos con 30 conexiones o 10 vehículos con 5 conexiones se sigue exactamente el mismo proceso, con la particularidad de asignar el valor de vehículos y conexiones que se desea simular.

En esta sección sufre ciertos cambios la parte de generación de rutas para los vehículos ya que el modelo Manhattan hace uso de un valor probabilístico típico del modelo para tomar la decisión de giro que van a tomar los vehículos en las intersecciones. Por lo que esta asignación de rutas se considera como automática ya que el programador no controla completamente todas las secciones de carretera por las que circulan los vehículos.

Comunicación V2V

Paso 1 V2V (manhattan). Para comenzar con la simulación se debe ejecutar el comando `java -jar MOVE.jar` desde el terminal de Ubuntu y nos aparecerá la interfaz de la Fig.1-B.

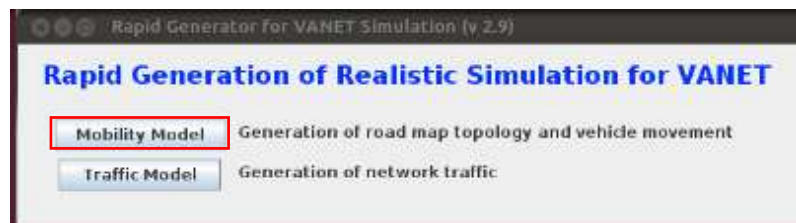


Figura 1-B: GUI de la herramienta MOVE

Realizado por: Danni Tierra, 2018.

Simulación de la sección de movilidad vehicular

Paso 2 V2V (manhattan). El siguiente paso es seleccionar la opción *Mobility Model* (Fig. 8-2) que es la sección para modelar el tráfico vehicular, en la Fig. 2-B se observa su interfaz con las subsecciones que la conforman.

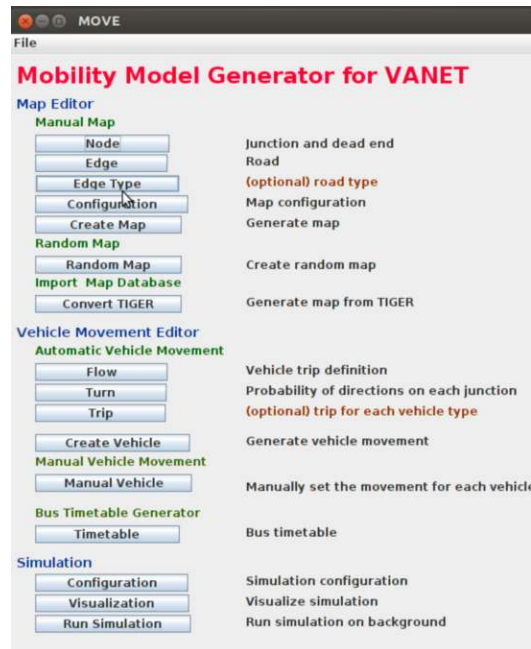


Figura 2-B: GUI Mobility Model para configurar los parámetros de red vehicular

Realizado por: Danni Tierra, 2018.

Paso 3 V2V (manhattan). Seleccionar la opción *Node* y configurar los nodos con coordenadas (x, y) y distancias que se pueden obtener desde Google Maps, estos nodos serán los puntos de intersección de las secciones de calle que conformarán el mapa de la zona de simulación. En la Fig. 3-B se observan los parámetros de configuración para las intersecciones.

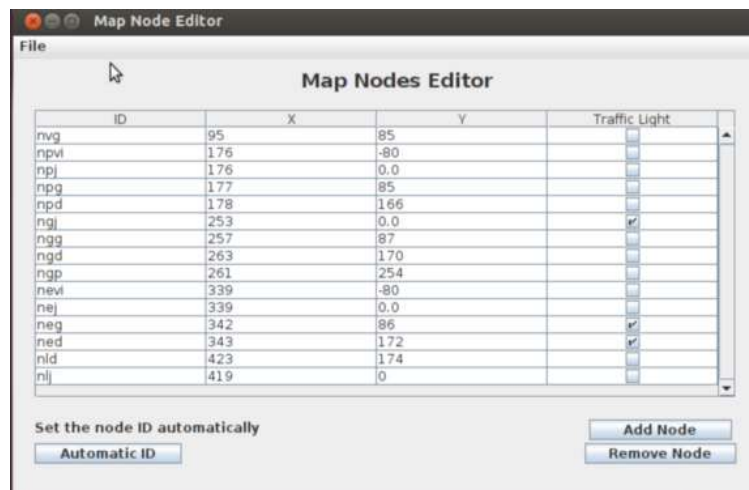


Figura 3-B: Configuración de nodos (intersecciones)

Realizado por: Danni Tierra,2018.

Paso 4 V2V (manhattan). En la Fig. 4-B se ve como configurar la opción *Edge Type* para definir los parámetros sobre el tipo de sección de carretera que se va a utilizar en la simulación. Para lo que se establecieron dos tipos de sección de calle que son primaria y secundaria con su respectivo id, nombre, número de carriles, prioridad, velocidad y capacidad.

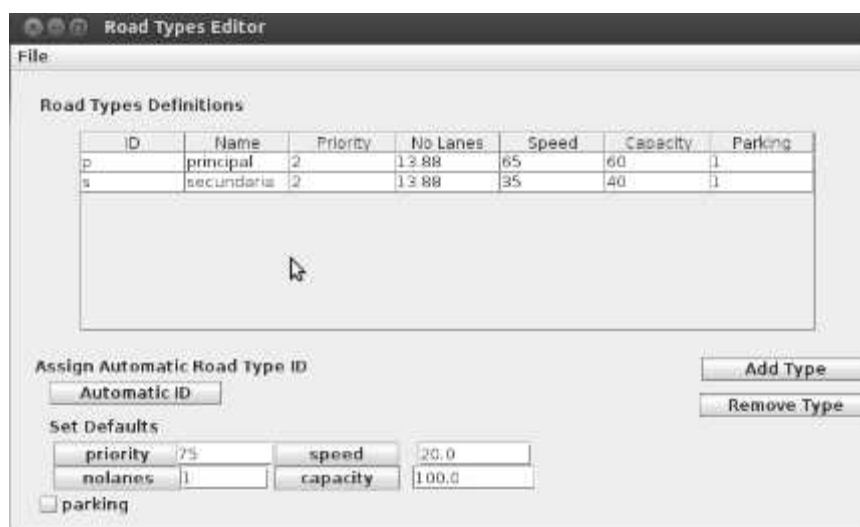


Figura 4-B: Configuración del tipo de sección de calle

Realizado por: Danni Tierra,2018.

Paso 5 V2V (manhattan). En la Fig. 5-B se observa la configuración de la opción *Edge*, colocando un id para cada sección de carretera, el id del tipo de sección (paso 4 V2V manhattan), el nodo inicio y fin que va a conectar la sección de carretera teniendo en cuenta que el orden en

que se coloque el nodo para una sección define la dirección que pueden tomar los vehículos, es decir el sentido de la vía.

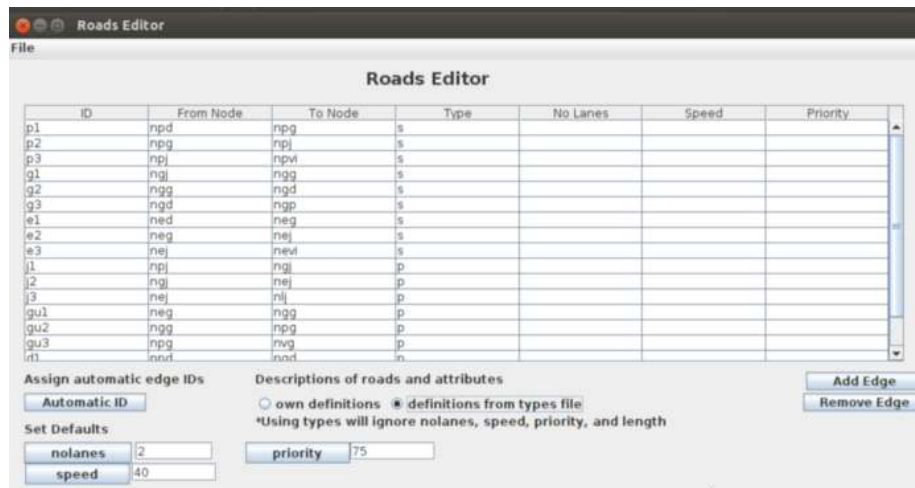


Figura 5-B: Configuración de las secciones que forman las calles

Realizado por: Danni Tierra,2018.

Paso 6 V2V (manhattan). En la Fig. 6-B se ven los parámetros de la subsección *Configuration* que permiten configurar el mapa que está basado en los archivos *.nod.xml*, *.edg.xml* y *.type.xml*. Establecer las rutas a los archivos ya mencionados para crear un archivo *.net.xml* que será el que contiene la información de las vías que forman el mapa. Al guardar estas configuraciones se crea un nuevo archivo *.netc.cfg* que nos será útil para el Paso 7 (manhattan).

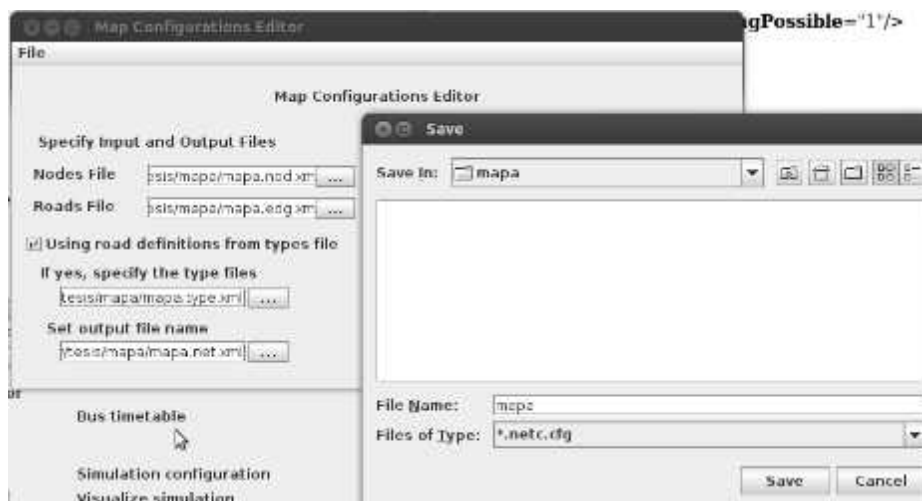


Figura 6-B: Rutas de los archivos para crear la topología de la zona de simulación

Realizado por: Danni Tierra,2018.

Paso 7 V2V (manhattan). En la subsección *Create Map* se procede a cargar el archivo de configuración *.netc.cfg* (Fig. 7-B) que se creó en el Paso 6 V2V (manhattan), mismo que contiene la ruta de los archivos necesarios para crear la topología del mapa. Luego click en *ok* para que comience el proceso de creación del mapa (Fig. 8-B).

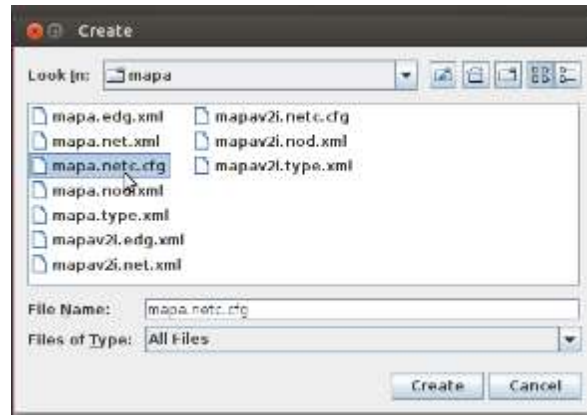


Figura 7-B: Subiendo archivo *.netc.cfg*

Realizado por: Danni Tierra,2018.



Figura 8-B: Generación exitosa del mapa

Realizado por: Danni Tierra,2018.

Paso 8 V2V (manhattan). En la Fig. 9-B se observa la sección *Flow* que nos permite controlar el flujo de vehículos que van a circular (*.flow.xml*), en el cual se indica el nodo inicio y el nodo fin pero no se controlan los nodos intermedios por lo que la movilidad se considera como automática por cómo se calculan las rutas en complemento con el archivo que nos indica la probabilidad de giro en las intersecciones (*.turn.xml*).

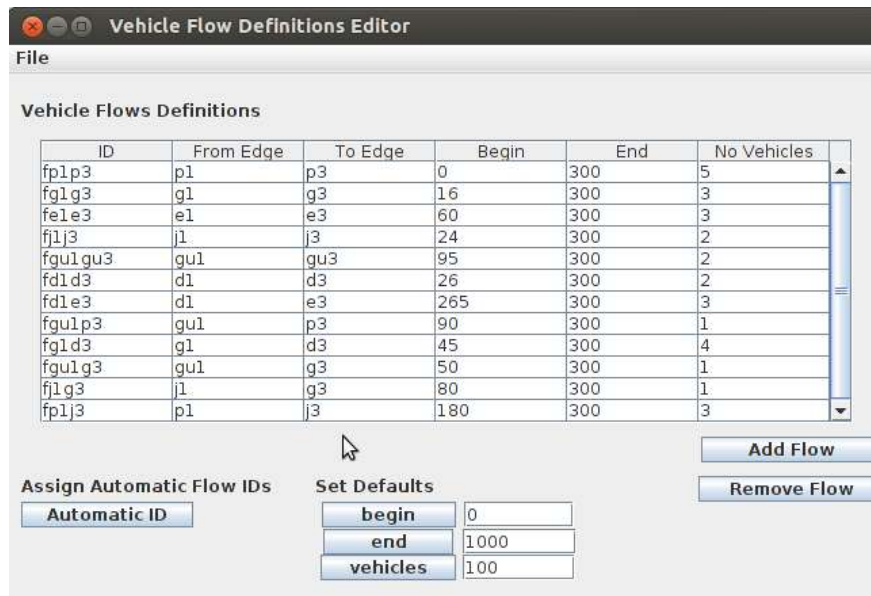


Figura 9-B: Parámetros para el flujo de vehículos en el modelo Manhattan

Realizado por: Danni Tierra,2018.

Paso 9 V2V (manhattan). En la Fig. 10-B se observa la sección *Turn* que nos permite configurar la probabilidad y el sentido de giro en las intersecciones, generando el archivo *.turn.xml*.

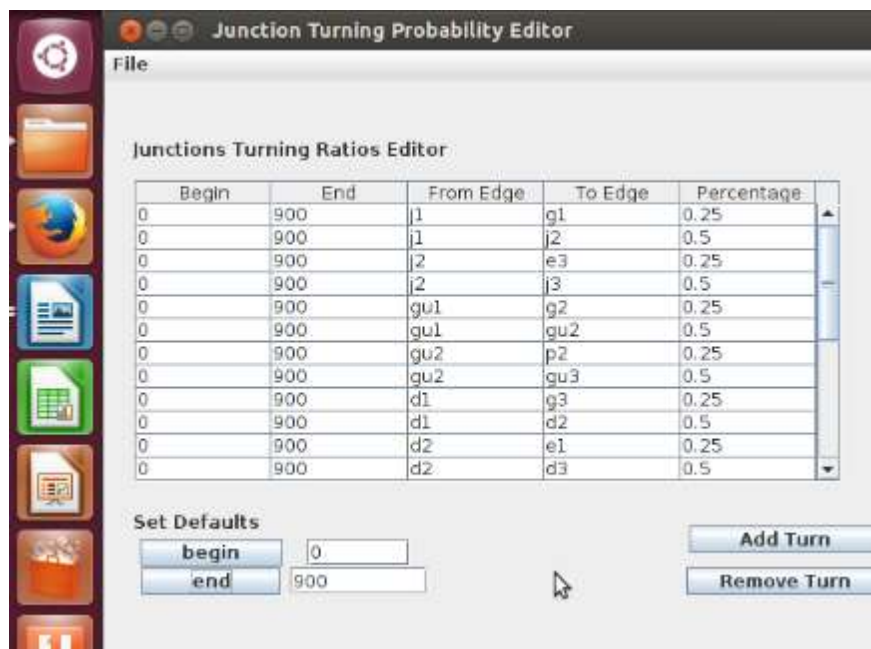


Figura 10-B: Probabilidad de giro en intersecciones (Modelo Manhattan)

Realizado por: Danni Tierra,2018.

Paso 10 V2V (manhattan). En la Fig. 11-B se observa la GUI de la sección *Create Vehicle*, aquí se carga el archivo *.flow.xml* del Paso 8 (manhattan) que contiene la información del flujo de los

vehículos y el archivo *.turn.xml* del Paso 9 (manhattan) para crear el archivo *.rou.xml* que contiene la información de las rutas que tomarán los vehículos en la simulación del modelo de movilidad Manhattan.

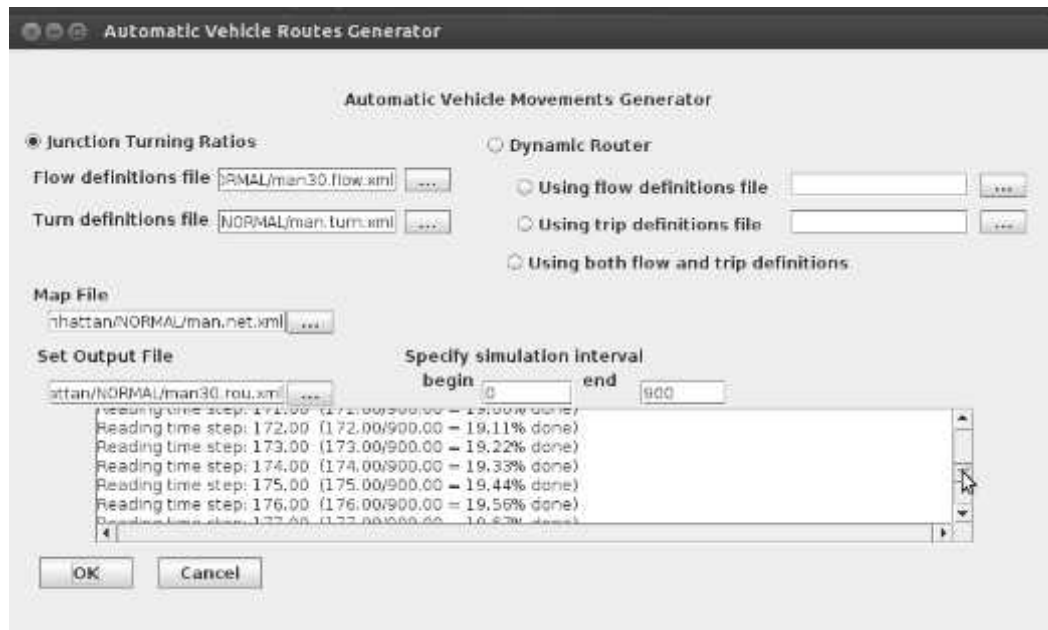


Figura 11-B: Creación del archivo *.rou.xml*

Realizado por: Danni Tierra, 2018.

Paso 11 V2V (manhattan). En la Fig. 12-B se observa la sección *Configuration*, aquí se sube el archivo *.rou.xml* del Paso 10 (manhattan) que contiene todas las rutas de los vehículos y el archivo *.net.xml* que contiene la información de la topología del mapa. Establecer el tiempo de simulación y se genera un archivo *.sumo.tr* que contiene las trazas a ser interpretadas por SUMO. Al guardar esta configuración se genera un archivo *.sumo.cfg*.



Figura 12-B: Creación de los archivo .sumo.tr y .sumo.cfg (Manhattan)

Realizado por: Danni Tierra,2018.

Paso 12 V2V (manhattan). En la Fig. 13-B se observa la sección *Visualization*, aquí nos direcciona hasta la GUI de SUMO y se carga el archivo *.sumo.cfg* del Paso 11 (manhattan) que contiene las rutas a los archivos necesarios para que la herramienta permita la visualización de la simulación como se aprecia en la Fig. 14-B.

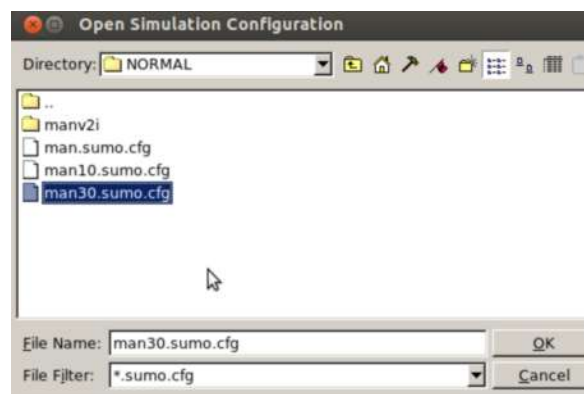


Figura 13-B: Cargando archivo .sumo.cfg (Manhattan) en la herramienta SUMO

Realizado por: Danni Tierra,2018

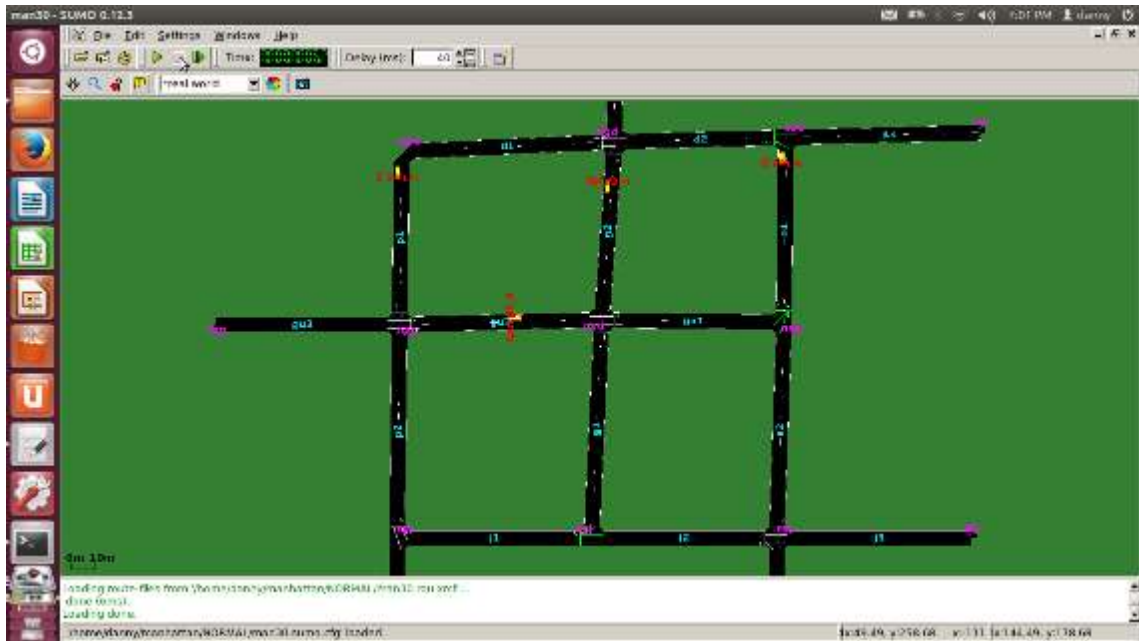


Figura 14-B: Simulación vehicular con 30 nodos (Modelo Manhattan)

Realizado por: Danni Tierra, 2018

Paso 13 V2V (manhattan). Seleccionar la sección *Run Simulation* en la que se sube el archivo *.sumo.cfg* (Fig. 15-B) del Paso 11 (manhattan), este paso se realiza para no tener inconvenientes al momento de exportar los datos hacia NS-2 ya que si no se realiza las trazas que se generan en SUMO presenta inconvenientes al momento de exportar los datos.

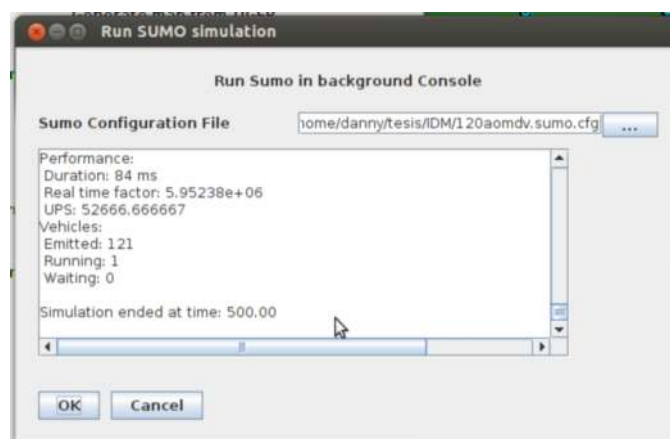


Figura 15-B: Cargando archivo *.sumo.cfg* para luego poder exportar los datos a NS-2

Realizado por: Danni Tierra, 2018

Simulación de la sección de red de datos

Esta parte de la simulación es exactamente igual a la del modelo de movilidad IDM (sección 2.10.1.2.) por lo que solamente se procede a indicar la visualización final de la simulación VANET basada en el modelo sintético Manhattan (Fig. 16-B).

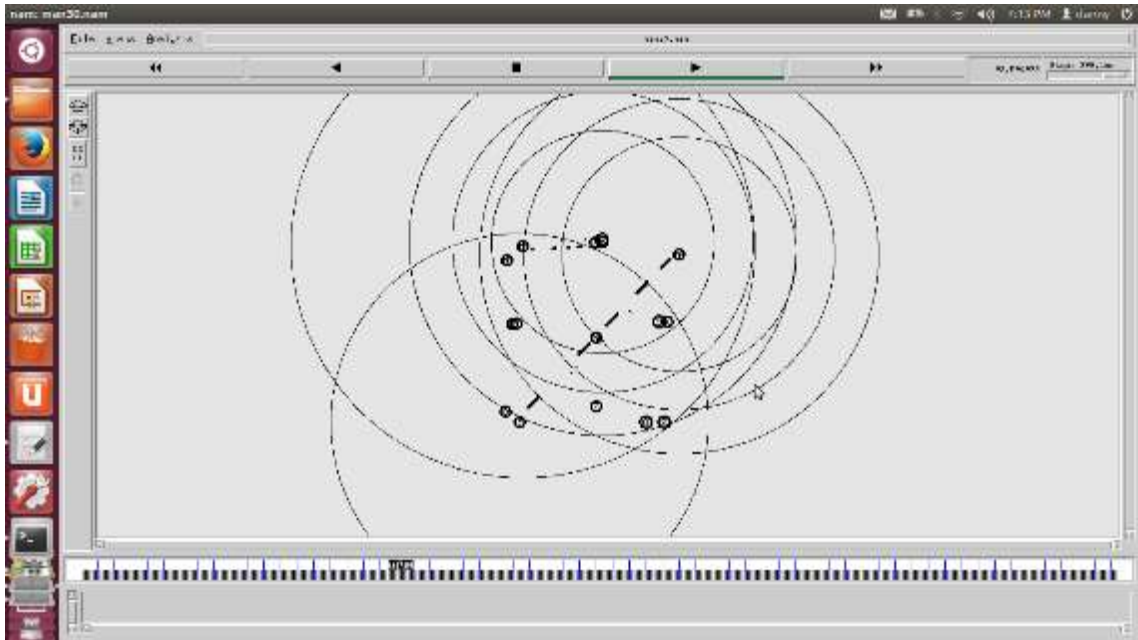


Figura 16-B: Simulación VANET completa (Modelo Manhattan), comunicación V2V

Realizado por: Danni Tierra, 2018

Comunicación V2I

Los pasos a seguir son idénticos, con cierta variación en la creación del tamaño del mapa de la zona de simulación para que exista la necesidad del uso de las infraestructuras.

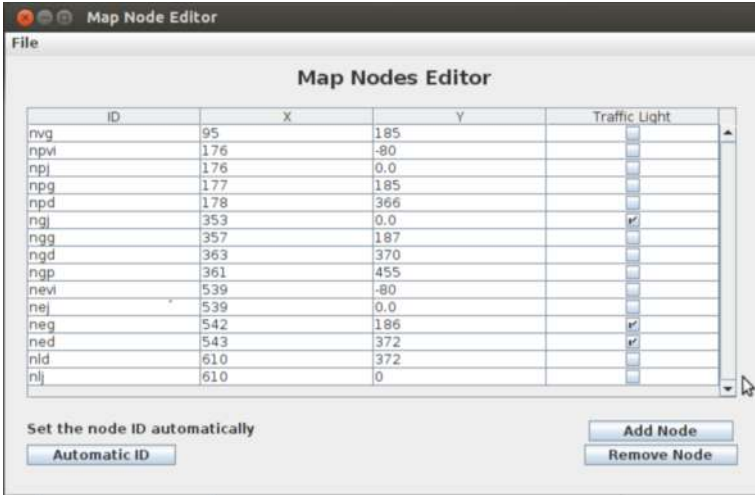
Simulación de la sección de movilidad vehicular

Paso 1 V2I (manhattan). Se sigue el mismo proceso que se detalla en el Paso 1 V2V (manhattan).

Paso 2 V2I (manhattan). Se sigue el mismo proceso que se detalla en el Paso 2 V2V (manhattan).

Paso 3 V2I (manhattan). Se sigue el mismo proceso que se detalla en el Paso 3 V2V (manhattan) con la modificación en las coordenadas de los nodos, se aumentó 100 metros a cada sección de carretera (Fig. 17-B) para que se vuelva necesario el uso de las infraestructuras al no tener dentro del rango de cobertura otro nodo que permita la comunicación.

Al utilizar el estándar IEEE 802.11p y no considerar las pérdidas por las edificaciones, la cobertura alcanza aproximadamente los 200 metros y las secciones de calle llegan a tener una longitud de entre los 75 a 90 metros por lo que se optó por aumentar la distancia entre los nodos.



The screenshot shows the 'Map Node Editor' window. It features a menu bar with 'File', a title bar 'Map Node Editor', and a main table with columns for 'ID', 'X', 'Y', and 'Traffic Light'. Below the table are buttons for 'Automatic ID', 'Add Node', and 'Remove Node'.

ID	X	Y	Traffic Light
nvg	95	185	<input type="checkbox"/>
npvi	176	-80	<input type="checkbox"/>
npj	176	0.0	<input type="checkbox"/>
npg	177	185	<input type="checkbox"/>
npd	178	366	<input type="checkbox"/>
ngj	353	0.0	<input checked="" type="checkbox"/>
ngg	357	187	<input type="checkbox"/>
ngd	363	370	<input type="checkbox"/>
ngp	361	455	<input type="checkbox"/>
nevi	539	-80	<input type="checkbox"/>
nej	539	0.0	<input type="checkbox"/>
neg	542	186	<input checked="" type="checkbox"/>
ned	543	372	<input checked="" type="checkbox"/>
nld	610	372	<input type="checkbox"/>
nlj	610	0	<input type="checkbox"/>

Figura 17-B: Configuración de intersecciones (V2I)

Realizado por: Danni Tierra, 2018

Paso 4 V2I (manhattan). Se sigue el mismo proceso que se detalla en el Paso 4 V2V (manhattan).

Paso 5 V2I (manhattan). Se sigue el mismo proceso que se detalla en el Paso 5 V2V (manhattan).

Paso 6 V2I (manhattan). Se sigue el mismo proceso que se detalla en el Paso 6 V2V (manhattan) teniendo presente que se debe cargar el nuevo archivo de nodos en el que se modificaron las coordenadas.

Paso 7 V2I (manhattan). Se sigue el mismo proceso que se detalla en el Paso 7 V2V (manhattan) y se guardan los nuevos archivos de la topología para un escenario de comunicación V2I.

Paso 8 V2I (manhattan). Se sigue el mismo proceso que se detalla en el Paso 8 V2V (manhattan).

Paso 9 V2I (manhattan). Se sigue el mismo proceso que se detalla en el Paso 9 V2V (manhattan), pero cargando los correspondientes archivos de la comunicación V2I.

Paso 10 V2I (manhattan). Se sigue el mismo proceso que se detalla en el Paso 10 V2V (manhattan), pero cargando los correspondientes archivos de la comunicación V2I. En la Fig. 18-B se observa la simulación de 120 vehículos en la nueva topología para comunicación V2I

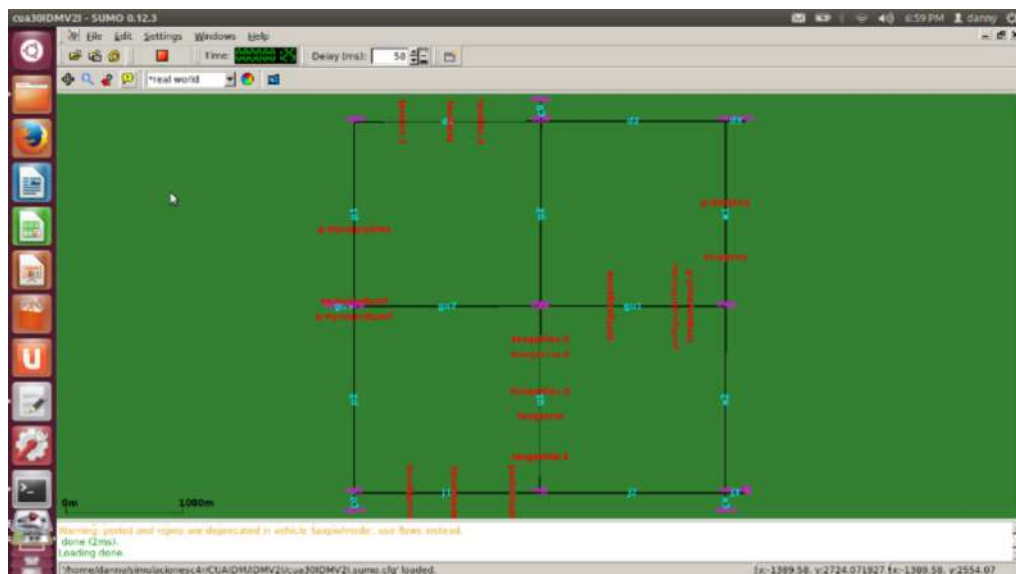


Figura 18-B: Topología para comunicación V2I

Realizado por: Danni Tierra, 2018

Paso 11 V2I (manhattan). Se sigue el mismo proceso que se detalla en el Paso 11 V2V (manhattan), pero ejecutando el correspondiente archivo de la comunicación V2I.

Paso 12 V2I (manhattan). Se sigue el mismo proceso que se detalla en el Paso 12 V2V (manhattan), pero ejecutando el correspondiente archivo de la comunicación V2I.

Paso 13 V2I (manhattan). Se sigue el mismo proceso que se detalla en el Paso 13 V2V (manhattan), pero ejecutando el correspondiente archivo de la comunicación V2I.

Simulación de la sección de red de datos

Esta parte de la simulación es exactamente igual a la del modelo de movilidad IDM (sección 2.10.2.2.) por lo que solamente se procede a indicar la visualización final de la simulación VANET basada en el modelo sintético Manhattan (Fig. 19-B).

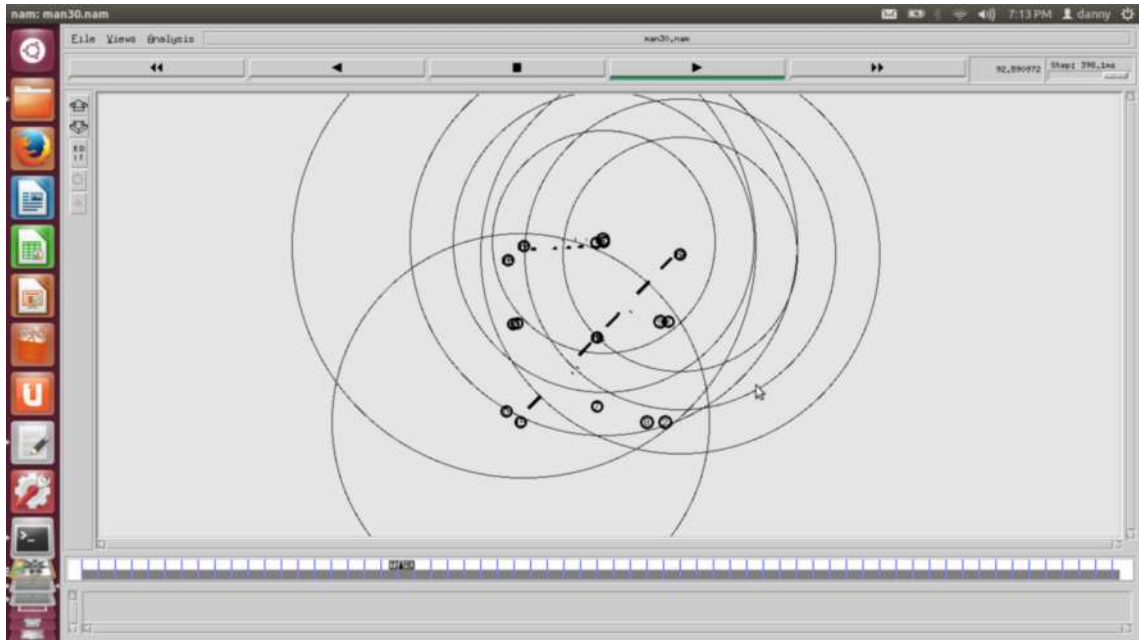


Figura 19-B: Simulación VANET completa (Modelo Manhattan), comunicación V2V

Realizado por: Danni Tierra, 2018

ANEXO C. Archivo tcl de 10 nodos con 5 conexiones V2V bajo el protocolo proactivo AOMDV

```
=====
# Define options
#
=====

set val(chan) Channel/WirelessChannel           ;# channel type
set val(prop) Propagation/TwoRayGround          ;# radio-propagation model
set val(netif) Phy/WirelessPhyExt              ;# network interface type
set val(mac) Mac/802_11Ext                     ;# MAC type
set val(ifq) Queue/DropTail/PriQueue          ;# interface queue type
set val(ll) LL                                 ;# link layer type
set val(ant) Antenna/OmniAntenna              ;# antenna model
set val(ifqlen) 50                             ;# max packet in ifq
set val(nn) 10                                 ;# number of mobilenodes
set val(rp) AOMDV                              ;# routing protocol

set opt(x) 380                                 ;# x coordinate of topology
set opt(y) 386                                 ;# y coordinate of topology
set stopTime 500.00

#802.11p default parameters
Phy/WirelessPhyExt set CStresh_ 3.162e-12 ;#-85 dBm Wireless interface
sensitivity (sensitivity defined in the standard)
Phy/WirelessPhyExt set Pt_ 0.001
Phy/WirelessPhyExt set freq_ 5.9e+9
Phy/WirelessPhyExt set noise_floor_ 1.26e-13 ;#-99 dBm for 10MHz
bandwidth
Phy/WirelessPhyExt set L_ 1.0 ;#default radio circuit gain/loss
Phy/WirelessPhyExt set PowerMonitorThresh_ 6.310e-14 ;#-102dBm power
monitor sensitivity
Phy/WirelessPhyExt set HeaderDuration_ 0.000040 ;#40 us
Phy/WirelessPhyExt set BasicModulationScheme_ 0
Phy/WirelessPhyExt set PreambleCaptureSwitch_ 1
Phy/WirelessPhyExt set DataCaptureSwitch_ 0
Phy/WirelessPhyExt set SINR_PreambleCapture_ 2.5118; ;# 4 dB
Phy/WirelessPhyExt set SINR_DataCapture_ 100.0; ;# 10 dB
Phy/WirelessPhyExt set trace_dist_ 1e6 ;# PHY trace until distance of 1
Mio. km ("infinty")
Phy/WirelessPhyExt set PHY_DBG_ 0

Mac/802_11Ext set CWMin_ 15
Mac/802_11Ext set CWMax_ 1023
```

```
Mac/802_11Ext set SlotTime_          0.000013
Mac/802_11Ext set SIFS_              0.000032
Mac/802_11Ext set ShortRetryLimit_   7
Mac/802_11Ext set LongRetryLimit_    4
Mac/802_11Ext set HeaderDuration_    0.000040
Mac/802_11Ext set SymbolDuration_    0.000008
Mac/802_11Ext set BasicModulationScheme_ 0
Mac/802_11Ext set use_802_11a_flag_  true
Mac/802_11Ext set RTSThreshold_      2346
Mac/802_11Ext set MAC_DBG            0
```

```
#
```

```
=====
```

```
=====
```

```
# Main Program
```

```
#
```

```
=====
```

```
=====
```

```
#
```

```
# Initialize Global Variables
```

```
#
```

```
set ns_ [new Simulator]
```

```
#EDITADO
```

```
# habilitando uso de nuevo formato de traza
```

```
$ns_ use-newtrace
```

```
#EDITADO
```

```
set tracefd [open /home/danny/te/10aomdv.tr w]
```

```
$ns_ trace-all $tracefd
```

```
set namtrace [open /home/danny/te/10aomdv.nam w]
```

```
$ns_ namtrace-all-wireless $namtrace $opt(x) $opt(y)
```

```
# set up topography object
```

```
set topo [new Topography]
```

```
$topo load_flatgrid $opt(x) $opt(y)
```

```
#
```

```
# Create God
```

```
#
```

```
create-god $val(nn)
```

```
# Configure node
```

```
set chan_1_ [new $val(chan)]
```

```
$ns_ node-config -adhocRouting $val(rp) \
```

```
    -llType $val(ll) \
```

```
    -macType $val(mac) \
```

```
    -ifqType $val(ifq) \
```

```
    -ifqLen $val(ifqlen) \
```

```
    -antType $val(ant) \
```

```
-propType $val(prop) \  
-phyType $val(netif) \  
-topoInstance $topo \  
-agentTrace ON \  
-routerTrace ON \  
-macTrace ON \  
-movementTrace ON \  
-channel $chan_1_
```

```
for {set i 0} {$i < $val(nn)} {incr i} {  
  set node_($i) [$ns_ node]  
  $node_($i) random-motion 0 ;# disable random motion  
}  
#  
# Provide initial (X,Y, for now Z=0) co-ordinates for mobilenodes  
#  
#predefine node in NAM  
# ID of SUMO: partpichincha  
$node_(0) set X_ 92.91553161917999  
$node_(0) set Y_ 250.20418346073663  
$node_(0) set Z_ 0.0  
$node_(0) setdest 92.91553161917999 250.20418346073663 1  
# ID of SUMO: taxesp  
$node_(1) set X_ 257.9151007644197  
$node_(4) setdest 100.55 90.05 1  
# ID of SUMO: partgarcia  
$node_(5) set X_ 178.1958790826963  
$node_(5) set Y_ 176.7190062543433  
$node_(5) set Z_ 0.0  
$node_(5) setdest 178.1958790826963 176.7190062543433 1  
# ID of SUMO: taxolmedo  
$node_(6) set X_ 100.55  
$node_(6) set Y_ 90.05  
$node_(6) set Z_ 0.0  
$node_(6) setdest 100.55 90.05 1  
# ID of SUMO: busgarcia  
$node_(7) set X_ 178.40991660875608  
$node_(7) set Y_ 172.059659485754  
$node_(7) set Z_ 0.0  
$node_(7) setdest 178.40991660875608 172.059659485754 1  
# ID of SUMO: taxgarcia  
$node_(8) set X_ 178.1958790826963  
$node_(8) set Y_ 176.7190062543433  
$node_(8) set Z_ 0.0  
$node_(8) setdest 178.1958790826963 176.7190062543433 1  
# ID of SUMO: partpiolmedo  
$node_(9) set X_ 92.91553161917999  
$node_(9) set Y_ 250.20418346073663  
$node_(9) set Z_ 0.0  
$node_(9) setdest 92.91553161917999 250.20418346073663 1
```


Now produce node movements

\$ns_ at 0.0 "\$node_(0) setdest 92.90619179986102 249.44660736622654 0.84"
\$ns_ at 1.0 "\$node_(0) setdest 92.88695621959694 247.886361362057 1.73"
\$ns_ at 2.0 "\$node_(0) setdest 92.85426685198054 245.23484503127173 2.93"
\$ns_ at 3.0 "\$node_(0) setdest 92.81212647671995 241.8167338429465 3.79"
\$ns_ at 4.0 "\$node_(0) setdest 92.76164697706741 237.72221542738012 4.54"
\$ns_ at 5.0 "\$node_(1) setdest 257.89967755385686 254.43289089645586 1.37"
\$ns_ at 5.0 "\$node_(0) setdest 92.70416261292564 233.0595149409312 5.17"
\$ns_ at 6.0 "\$node_(1) setdest 257.8718707435719 252.04184850590687 2.47"
\$ns_ at 6.0 "\$node_(0) setdest 92.63933981931898 227.8015760945101 5.83"
\$ns_ at 7.0 "\$node_(1) setdest 257.8358457261987 248.94414176511467 3.20"
\$ns_ at 7.0 "\$node_(0) setdest 95.865510771369 221.77311744266854 6.64"
\$ns_ at 8.0 "\$node_(1) setdest 257.79081445448224 245.07200833912438 4.00"
\$ns_ at 8.0 "\$node_(0) setdest 95.7858999305073 215.3156831132731 7.15"
\$ns_ at 9.0 "\$node_(1) setdest 257.7368895066018 240.43512856150105 4.79"
\$ns_ at 9.0 "\$node_(2) setdest 249.99293537178596 186.80249200833913 1.12"
\$ns_ at 9.0 "\$node_(0) setdest 95.70940236275192 209.1107741487144 6.89"
\$ns_ at 10.0 "\$node_(1) setdest 257.6753092425295 235.13998610145936 5.47"
\$ns_ at 10.0 "\$node_(2) setdest 247.75594440583737 186.7760361362057 2.35"
\$ns_ at 10.0 "\$node_(0) setdest 95.65113968033357 204.3849423210563 5.24"
\$ns_ at 11.0 "\$node_(1) setdest 257.60753717859626 229.31242529534398 6.02"
\$ns_ at 11.0 "\$node_(2) setdest 244.77646282140375 186.74079916608756 3.12"
\$ns_ at 11.0 "\$node_(0) setdest 95.60366226546213 200.53393050729673 4.27"
\$ns_ at 12.0 "\$node_(1) setdest 260.8353745656706 223.07221264767202 6.40"
\$ns_ at 12.0 "\$node_(2) setdest 240.97833773453786 186.69588047255039 3.99"
\$ns_ at 12.0 "\$node_(0) setdest 95.57419735927728 198.14395830437803 2.66"
\$ns_ at 13.0 "\$node_(1) setdest 260.7590465601112 216.5080041695622 6.78"
\$ns_ at 13.0 "\$node_(2) setdest 236.65666157053508 186.6447699791522 4.54"
\$ns_ at 13.0 "\$node_(0) setdest 95.5540722724114 196.51156219596942 1.81"
\$ns_ at 14.0 "\$node_(1) setdest 260.67810284920085 209.54684503127172 7.20"
\$ns_ at 14.0 "\$node_(2) setdest 231.62105211952746 186.58521612230714 5.29"
\$ns_ at 14.0 "\$node_(0) setdest 95.54017373175817 195.38421681723418 1.25"
\$ns_ at 15.0 "\$node_(0) setdest 95.38487977762335 187.8883488533704 2.43"
\$ns_ at 15.0 "\$node_(1) setdest 260.59220569840164 202.15969006254343 7.63"
\$ns_ at 15.0 "\$node_(2) setdest 226.1285253648367 186.52025851285615 5.77"
\$ns_ at 16.0 "\$node_(0) setdest 95.2223168867269 187.43488394718554 3.41"
\$ns_ at 16.0 "\$node_(3) setdest 103.73039054899236 259.7413217512161 0.81"
\$ns_ at 16.0 "\$node_(1) setdest 260.501467685893 194.3562209867964 8.06"
\$ns_ at 16.0 "\$node_(2) setdest 220.17908130646282 186.44989715079916 6.25"
\$ns_ at 17.0 "\$node_(1) setdest 260.1889923558026 188.37681028492008 8.42"
\$ns_ at 17.0 "\$node_(0) setdest 95.03170952050036 186.90318971507992 4.01"
\$ns_ at 17.0 "\$node_(3) setdest 106.04711605281446 259.63244614315494 2.04"
\$ns_ at 17.0 "\$node_(2) setdest 213.76320083391244 186.3740194579569 6.75"
\$ns_ at 18.0 "\$node_(3) setdest 109.51084781097985 259.4696664350243 3.05"
\$ns_ at 18.0 "\$node_(1) setdest 256.8524808895066 175.3504447533009 8.68"
\$ns_ at 18.0 "\$node_(2) setdest 206.86184572619874 186.2924002779708 7.24"
\$ns_ at 18.0 "\$node_(0) setdest 91.98436831132732 174.6313425990271 4.93"
\$ns_ at 19.0 "\$node_(3) setdest 114.08751633078528 259.2545837387074 4.03"
\$ns_ at 19.0 "\$node_(1) setdest 256.491744266852 165.03213342599025 8.95"
\$ns_ at 19.0 "\$node_(2) setdest 199.63684086170952 186.2069534398888 7.59"

\$ns_ at 19.0 "\$node_(0) setdest 91.91098401667824 168.3615719249479 5.50"
\$ns_ at 20.0 "\$node_(3) setdest 119.36828769979152 259.00641139680334 4.65"
\$ns_ at 20.0 "\$node_(1) setdest 256.1237526059764 154.50630298818623 9.14"
\$ns_ at 20.0 "\$node_(2) setdest 192.16433912439194 186.11857956914523 7.85"
\$ns_ at 20.0 "\$node_(0) setdest 91.82932731063238 161.38502710215428 6.12"
\$ns_ at 21.0 "\$node_(2) setdest 184.8707060458652 186.90613898540653 8.16"
\$ns_ at 21.0 "\$node_(3) setdest 125.37587491313413 258.72408200138983 5.29"
\$ns_ at 21.0 "\$node_(1) setdest 255.74931202223766 143.79601111883252 9.28"
\$ns_ at 21.0 "\$node_(0) setdest 91.73859763724809 153.6333106323836 6.80"
\$ns_ at 22.0 "\$node_(3) setdest 132.14434746351634 258.40599444058375 5.96"
\$ns_ at 22.0 "\$node_(1) setdest 255.3668102849201 132.8551424600417 9.50"
\$ns_ at 22.0 "\$node_(2) setdest 172.2680041695622 186.83101181375955 8.43"
\$ns_ at 22.0 "\$node_(0) setdest 91.64239749826268 145.41421125781793 7.21"
\$ns_ at 23.0 "\$node_(3) setdest 139.4806448922863 258.0612216817234 6.46"
\$ns_ at 23.0 "\$node_(1) setdest 254.9758443363447 121.67216817234191 9.70"
\$ns_ at 23.0 "\$node_(2) setdest 162.9315705350938 186.5977831827658 8.74"
\$ns_ at 23.0 "\$node_(0) setdest 91.53845865184155 136.533936066713 7.79"
\$ns_ at 24.0 "\$node_(3) setdest 147.38476719944407 257.6897637248089 6.96"
\$ns_ at 24.0 "\$node_(1) setdest 254.62075052119528 111.51526059763725 8.81"
\$ns_ at 24.0 "\$node_(2) setdest 153.2853467685893 186.35681584433635 9.03"
\$ns_ at 24.0 "\$node_(4) setdest 101.37 90.05 0.82"
\$ns_ at 24.0 "\$node_(0) setdest 91.42984989576094 127.25467546907575 8.13"
\$ns_ at 25.0 "\$node_(3) setdest 155.83400138985405 257.29268797776234 7.44"
\$ns_ at 25.0 "\$node_(1) setdest 254.33457956914523 103.32978457261989 7.10"
\$ns_ at 25.0 "\$node_(2) setdest 143.36138012508687 186.10891035441279 9.28"
\$ns_ at 25.0 "\$node_(4) setdest 103.3 90.05 1.93"
\$ns_ at 25.0 "\$node_(0) setdest 91.31430298818624 117.38263655316192 8.66"
\$ns_ at 26.0 "\$node_(3) setdest 164.7602084781098 256.87319666435025 7.86"
\$ns_ at 26.0 "\$node_(1) setdest 254.10765809589992 96.83904794996526 5.63"
\$ns_ at 26.0 "\$node_(5) setdest 178.26918693537178 175.12317998610146 1.37"
\$ns_ at 26.0 "\$node_(2) setdest 133.27717720639333 185.8570020847811 9.44"
\$ns_ at 26.0 "\$node_(4) setdest 106.39 90.05 3.09"
\$ns_ at 26.0 "\$node_(0) setdest 91.19261848505906 106.98621681723421 9.12"
\$ns_ at 27.0 "\$node_(3) setdest 173.99304100069492 256.4392953439889 8.12"
\$ns_ at 27.0 "\$node_(1) setdest 253.93917998610146 92.0199930507297 4.18"
\$ns_ at 27.0 "\$node_(5) setdest 178.3895830437804 172.50229742877 2.25"
\$ns_ at 27.0 "\$node_(2) setdest 123.03273801250869 185.60109103544127 9.59"
\$ns_ at 27.0 "\$node_(4) setdest 110.37 90.05 3.98"
\$ns_ at 27.0 "\$node_(0) setdest 91.06599722029186 96.16801250868659 9.49"
\$ns_ at 28.0 "\$node_(0) setdest 91.06996108408616 96.49139819318972 9.93"
\$ns_ at 28.0 "\$node_(3) setdest 183.58928144544825 255.98831549687284 8.45"
\$ns_ at 28.0 "\$node_(1) setdest 253.83398193189714 89.0109659485754 2.61"
\$ns_ at 28.0 "\$node_(5) setdest 178.55599722029186 168.87965531619182 3.11"
\$ns_ at 28.0 "\$node_(2) setdest 112.60669770674079 185.34064350243224 9.76"
\$ns_ at 28.0 "\$node_(4) setdest 115.16 90.05 4.79"
\$ns_ at 29.0 "\$node_(3) setdest 181.16113690062542 260.2452466990966 8.71"
\$ns_ at 29.0 "\$node_(1) setdest 253.76707435719248 87.09717859624742 1.66"
\$ns_ at 29.0 "\$node_(5) setdest 178.77806115357888 164.04558304378043 4.15"
\$ns_ at 29.0 "\$node_(2) setdest 102.05246838082002 185.07699374565672 9.88"
\$ns_ at 29.0 "\$node_(4) setdest 120.5 90.05 5.34"

\$ns_ at 30.0 "\$node_(3) setdest 182.41772202918693 260.2081792911744 8.89"
\$ns_ at 30.0 "\$node_(2) setdest 98.33200694927032 184.9890104239055 9.98"
\$ns_ at 30.0 "\$node_(1) setdest 253.7211257817929 85.7828908964559 1.14"
\$ns_ at 30.0 "\$node_(5) setdest 179.03544127866573 158.4427185545518 4.82"
\$ns_ at 30.0 "\$node_(4) setdest 126.53 90.05 6.03"
\$ns_ at 31.0 "\$node_(1) setdest 254.12529534398888 96.54473662265463 2.21"
\$ns_ at 31.0 "\$node_(3) setdest 199.0799430159833 261.6424378040306 9.08"
\$ns_ at 31.0 "\$node_(5) setdest 179.33562890896457 151.90798471160528 5.61"
\$ns_ at 31.0 "\$node_(4) setdest 133.2 90.05 6.67"
\$ns_ at 32.0 "\$node_(1) setdest 254.26984016678247 96.11466018068103 3.20"
\$ns_ at 32.0 "\$node_(3) setdest 207.29955941626127 261.43714107018764 9.29"
\$ns_ at 32.0 "\$node_(5) setdest 179.66738707435718 144.68599722029188 6.19"
\$ns_ at 32.0 "\$node_(4) setdest 140.35 90.05 7.15"
\$ns_ at 33.0 "\$node_(1) setdest 254.45639332870047 95.55959277275886 4.13"
\$ns_ at 33.0 "\$node_(3) setdest 215.65189298123698 261.2285295343989 9.44"
\$ns_ at 33.0 "\$node_(5) setdest 180.0376719944406 136.62532731063237 6.93"
\$ns_ at 33.0 "\$node_(4) setdest 147.96 90.05 7.61"
\$ns_ at 34.0 "\$node_(3) setdest 224.1280958999305 261.01682418346076 9.57"
\$ns_ at 34.0 "\$node_(5) setdest 180.4395274496178 127.87740375260597 7.51"
\$ns_ at 34.0 "\$node_(6) setdest 101.66 90.05 1.11"
\$ns_ at 34.0 "\$node_(4) setdest 156.19 90.05 8.23"
\$ns_ at 35.0 "\$node_(3) setdest 232.71932036136207 260.80224600416955 9.71"
\$ns_ at 35.0 "\$node_(5) setdest 180.8611813759555 118.69849061848507 7.87"
\$ns_ at 35.0 "\$node_(6) setdest 103.56 90.05 1.90"
\$ns_ at 35.0 "\$node_(4) setdest 164.95 90.05 8.76"
\$ns_ at 36.0 "\$node_(4) setdest 174.12 90.05 9.18"
\$ns_ at 36.0 "\$node_(3) setdest 241.3990229325921 260.5854579569145 9.82"
\$ns_ at 36.0 "\$node_(5) setdest 181.30958999305074 108.93715913829048 8.38"
\$ns_ at 36.0 "\$node_(6) setdest 106.28 90.05 2.73"
\$ns_ at 37.0 "\$node_(3) setdest 255.23006949270328 262.069892981237 9.94"
\$ns_ at 37.0 "\$node_(5) setdest 181.67719944405837 100.93473106323836 6.87"
\$ns_ at 37.0 "\$node_(6) setdest 109.89 90.05 3.61"
\$ns_ at 37.0 "\$node_(4) setdest 183.62 90.05 9.51"
\$ns_ at 38.0 "\$node_(5) setdest 181.95758860319665 94.83098679638637 5.24"
\$ns_ at 38.0 "\$node_(6) setdest 114.4 90.05 4.51"
\$ns_ at 38.0 "\$node_(4) setdest 193.45000000000002 90.05 9.83"
\$ns_ at 39.0 "\$node_(5) setdest 182.15771369006254 90.47449756775538 3.74"
\$ns_ at 39.0 "\$node_(6) setdest 119.4 90.05 5.00"
\$ns_ at 41.0 "\$node_(4) setdest 224.67000000000002 90.05 10.68"
\$ns_ at 42.0 "\$node_(7) setdest 178.76040305767896 164.42997915218902 3.03"
\$ns_ at 42.0 "\$node_(5) setdest 182.45308547602502 84.04459902710215 1.04"
\$ns_ at 42.0 "\$node_(6) setdest 137.65 90.05 6.71"
\$ns_ at 42.0 "\$node_(4) setdest 235.56 90.05 10.89"
\$ns_ at 43.0 "\$node_(5) setdest 181.83994440583737 190.16726893676164 2.33"
\$ns_ at 43.0 "\$node_(7) setdest 178.95517720639333 160.18997359277276 3.64"
\$ns_ at 43.0 "\$node_(6) setdest 144.79000000000002 90.05 7.14"
\$ns_ at 43.0 "\$node_(4) setdest 246.66000000000003 90.05 11.10"
\$ns_ at 44.0 "\$node_(4) setdest 257.93 90.05 11.27"
\$ns_ at 44.0 "\$node_(5) setdest 181.7288811674774 189.743787352328 3.05"
\$ns_ at 44.0 "\$node_(7) setdest 179.19168867268937 155.04139541348158 4.41"

\$ns_ at 44.0 "\$node_(6) setdest 152.41 90.05 7.62"
\$ns_ at 45.0 "\$node_(5) setdest 181.59596942321056 189.2369979152189 3.65"
\$ns_ at 45.0 "\$node_(7) setdest 179.45656011118834 149.27545378735235 4.95"
\$ns_ at 45.0 "\$node_(6) setdest 160.39 90.05 7.98"
\$ns_ at 46.0 "\$node_(7) setdest 179.74604586518416 142.9736872828353 5.41"
\$ns_ at 46.0 "\$node_(5) setdest 179.3787491313412 255.16507713690063 4.47"
\$ns_ at 46.0 "\$node_(6) setdest 168.64 90.05 8.24"
\$ns_ at 47.0 "\$node_(6) setdest 177.21 90.05 8.57"
\$ns_ at 47.0 "\$node_(7) setdest 180.06870743571926 135.94972202918694 6.03"
\$ns_ at 47.0 "\$node_(5) setdest 179.71631688672687 250.48906045865184 5.06"
\$ns_ at 48.0 "\$node_(7) setdest 180.41865879082695 128.33169006254343 6.54"
\$ns_ at 48.0 "\$node_(5) setdest 180.1085892981237 245.0552703266157 5.89"
\$ns_ at 48.0 "\$node_(6) setdest 186.04000000000002 90.05 8.85"
\$ns_ at 49.0 "\$node_(7) setdest 180.794829742877 120.14288811674774 7.03"
\$ns_ at 49.0 "\$node_(5) setdest 180.54489228630993 239.01156497567754 6.53"
\$ns_ at 49.0 "\$node_(6) setdest 195.08 90.05 9.04"
\$ns_ at 50.0 "\$node_(7) setdest 181.1913342599027 111.51144822793607 7.41"
\$ns_ at 50.0 "\$node_(5) setdest 181.02656011118833 232.33946212647672 7.22"
\$ns_ at 50.0 "\$node_(6) setdest 204.29000000000002 90.05 9.21"
\$ns_ at 51.0 "\$node_(7) setdest 181.58569840166783 102.92660180681028 7.38"
\$ns_ at 51.0 "\$node_(5) setdest 181.53758165392634 225.2607491313412 7.66"
\$ns_ at 51.0 "\$node_(6) setdest 213.73000000000002 90.05 9.44"
\$ns_ at 52.0 "\$node_(7) setdest 181.89605281445446 96.1705489923558 5.80"
\$ns_ at 52.0 "\$node_(5) setdest 182.08596247394024 217.66453231410702 8.22"
\$ns_ at 52.0 "\$node_(6) setdest 223.37 90.05 9.64"
\$ns_ at 53.0 "\$node_(7) setdest 182.12721334259902 91.13845448227936 4.32"
\$ns_ at 53.0 "\$node_(5) setdest 182.55295343988882 211.19573453787353 7.00"
\$ns_ at 53.0 "\$node_(6) setdest 233.17000000000002 90.05 9.81"
\$ns_ at 54.0 "\$node_(7) setdest 182.28560111188324 87.69053787352327 2.95"
\$ns_ at 54.0 "\$node_(5) setdest 182.91987491313412 206.11310771369006 5.50"
\$ns_ at 54.0 "\$node_(6) setdest 243.13 90.05 9.95"
\$ns_ at 55.0 "\$node_(6) setdest 253.22 90.05 10.09"
\$ns_ at 55.0 "\$node_(7) setdest 182.3717512161223 85.81515079916609 1.61"
\$ns_ at 55.0 "\$node_(5) setdest 183.212744961779 202.05624739402361 4.39"
\$ns_ at 56.0 "\$node_(7) setdest 182.4204447533009 84.75514940931203 0.92"
\$ns_ at 56.0 "\$node_(5) setdest 183.4195552466991 199.1914940931202 3.10"
\$ns_ at 57.0 "\$node_(7) setdest 181.8749020152884 190.30056150104238 1.97"
\$ns_ at 57.0 "\$node_(5) setdest 183.53029881862403 197.65746490618483 1.66"
\$ns_ at 58.0 "\$node_(7) setdest 181.77148575399582 189.90623766504515 2.84"
\$ns_ at 58.0 "\$node_(5) setdest 183.62236275191106 196.3821876302988 1.38"
\$ns_ at 59.0 "\$node_(5) setdest 184.13170257123002 266.39804308547605 2.62"
\$ns_ at 59.0 "\$node_(7) setdest 181.64257956914523 189.4147213342599 3.54"
\$ns_ at 60.0 "\$node_(5) setdest 184.31470465601112 265.9336455872134 3.42"
\$ns_ at 60.0 "\$node_(7) setdest 179.27400972897846 256.615936066713 4.18"
\$ns_ at 61.0 "\$node_(5) setdest 184.54211952744961 265.3565434329396 4.26"
\$ns_ at 61.0 "\$node_(7) setdest 179.60824183460736 251.98612508686588 5.01"
\$ns_ at 62.0 "\$node_(7) setdest 179.99384294649062 246.64474635163307 5.78"
\$ns_ at 63.0 "\$node_(7) setdest 180.4221403752606 240.71193467685893 6.42"
\$ns_ at 64.0 "\$node_(7) setdest 180.88312717164698 234.32630715774843 6.90"
\$ns_ at 65.0 "\$node_(7) setdest 181.36679638637943 227.6264808895066 7.25"

\$ns_ at 66.0 "\$node_(7) setdest 181.8724808895066 220.62169701181375 7.58"
\$ns_ at 67.0 "\$node_(7) setdest 182.39217512161224 213.4228492008339 7.79"
\$ns_ at 68.0 "\$node_(7) setdest 182.81380125086866 207.58244892286308 6.32"
\$ns_ at 69.0 "\$node_(7) setdest 183.11601111883252 203.39621264767197 4.53"
\$ns_ at 70.0 "\$node_(7) setdest 183.3481723419041 200.1802960389159 3.49"
\$ns_ at 71.0 "\$node_(7) setdest 183.5129534398888 197.89773453787353 2.47"
\$ns_ at 72.0 "\$node_(7) setdest 183.59501042390548 196.76107435719248 1.23"
\$ns_ at 73.0 "\$node_(7) setdest 184.10173731758167 266.47408478109804 2.47"
\$ns_ at 74.0 "\$node_(7) setdest 184.27724808895067 266.02869770674084 3.28"
\$ns_ at 75.0 "\$node_(7) setdest 184.50038220986798 265.4624586518416 4.17"
\$ns_ at 106.0 "\$node_(8) setdest 178.23975677553858 175.7638401667825 0.82"
\$ns_ at 107.0 "\$node_(8) setdest 178.32590687977762 173.8884530924253 1.61"
\$ns_ at 108.0 "\$node_(8) setdest 178.46984711605282 170.755042390549 2.69"
\$ns_ at 109.0 "\$node_(8) setdest 178.6474982626824 166.8877845726199 3.33"
\$ns_ at 110.0 "\$node_(8) setdest 178.87277275886032 161.98382209867964 4.21"
\$ns_ at 111.0 "\$node_(8) setdest 179.14085476025016 156.14799027102154 5.01"
\$ns_ at 112.0 "\$node_(8) setdest 179.44371785962474 149.5550145934677 5.66"
\$ns_ at 113.0 "\$node_(8) setdest 179.77868658790825 142.26313690062545 6.27"
\$ns_ at 114.0 "\$node_(8) setdest 180.14041000694928 134.38884086170953 6.76"
\$ns_ at 115.0 "\$node_(8) setdest 180.5272828353023 125.96707157748438 7.23"
\$ns_ at 116.0 "\$node_(8) setdest 180.93181375955524 117.1609061848506 7.56"
\$ns_ at 117.0 "\$node_(8) setdest 181.35186240444753 108.01693815149409 7.84"
\$ns_ at 118.0 "\$node_(8) setdest 181.71144544822792 100.18923558026408 6.73"
\$ns_ at 119.0 "\$node_(8) setdest 181.9784572619875 94.37670048644891 4.99"
\$ns_ at 120.0 "\$node_(8) setdest 182.19356497567756 89.69405698401667 4.02"
\$ns_ at 120.0 "\$node_(9) setdest 92.90252258512857 249.14898818624044 1.17"
\$ns_ at 121.0 "\$node_(8) setdest 182.32573314801945 86.81691035441278 2.47"
\$ns_ at 121.0 "\$node_(9) setdest 92.87661570535094 247.04761640027797 2.33"
\$ns_ at 122.0 "\$node_(8) setdest 182.42312022237664 84.69690757470467 1.82"
\$ns_ at 122.0 "\$node_(9) setdest 92.83803335649758 243.91810562890896 3.47"
\$ns_ at 123.0 "\$node_(8) setdest 181.8530535093815 190.21725364836692 2.52"
\$ns_ at 123.0 "\$node_(9) setdest 92.78799861014593 239.85966226546213 4.50"
\$ns_ at 124.0 "\$node_(8) setdest 181.72778874218207 189.73962195969423 3.44"
\$ns_ at 124.0 "\$node_(9) setdest 92.72829047949966 235.01658651841558 5.36"
\$ns_ at 125.0 "\$node_(8) setdest 181.56793050729672 189.13008617095204 4.39"
\$ns_ at 125.0 "\$node_(9) setdest 92.66035441278666 229.50612230715774 6.11"
\$ns_ at 126.0 "\$node_(8) setdest 179.4681445448228 253.92676441973592 5.04"
\$ns_ at 126.0 "\$node_(9) setdest 95.88674774148714 223.49570118137595 6.62"
\$ns_ at 127.0 "\$node_(8) setdest 179.8437387074357 248.72400277970814 5.63"
\$ns_ at 127.0 "\$node_(9) setdest 95.80713690062544 217.03826685198055 7.16"
\$ns_ at 128.0 "\$node_(8) setdest 180.2513551077137 243.07766643502433 6.11"
\$ns_ at 128.0 "\$node_(9) setdest 95.72596942321057 210.45456984016678 7.30"
\$ns_ at 129.0 "\$node_(8) setdest 180.6923280055594 236.96927310632384 6.61"
\$ns_ at 129.0 "\$node_(9) setdest 95.6633703961084 205.37700625434329 5.63"
\$ns_ at 130.0 "\$node_(8) setdest 181.16599027102154 230.408063933287 7.09"
\$ns_ at 130.0 "\$node_(9) setdest 95.61444753300904 201.40875052119526 4.40"
\$ns_ at 131.0 "\$node_(8) setdest 181.66833912439193 223.44948575399582 7.53"
\$ns_ at 131.0 "\$node_(9) setdest 95.58275886031967 198.83840305767893 2.85"
\$ns_ at 132.0 "\$node_(8) setdest 182.1980403057679 216.11202084781098 7.94"
\$ns_ at 132.0 "\$node_(9) setdest 95.55774148714386 196.80918137595552 2.25"

\$ns_ at 133.0 "\$node_(8) setdest 182.65235580264073 209.8188047255038 6.81"
\$ns_ at 133.0 "\$node_(9) setdest 95.5441765114663 195.70889228630995 1.22"
\$ns_ at 134.0 "\$node_(9) setdest 95.405318971508 187.9453634468381 2.36"
\$ns_ at 134.0 "\$node_(8) setdest 182.9959277275886 205.05961779013202 5.15"
\$ns_ at 135.0 "\$node_(9) setdest 95.25416400277972 187.52372063933285 3.18"
\$ns_ at 135.0 "\$node_(8) setdest 183.24943710910355 201.5479847116053 3.80"
\$ns_ at 136.0 "\$node_(9) setdest 95.07306323835998 187.01854482279361 3.81"
\$ns_ at 136.0 "\$node_(8) setdest 183.45024322446145 198.76640166782488 3.02"
\$ns_ at 137.0 "\$node_(8) setdest 183.5516469770674 197.3617484364142 1.52"
\$ns_ at 137.0 "\$node_(9) setdest 91.99744405837387 175.74850173731758 4.82"
\$ns_ at 138.0 "\$node_(8) setdest 183.59767894371092 196.72410979847115 0.69"
\$ns_ at 138.0 "\$node_(9) setdest 91.92285892981236 169.37613481584435 5.59"
\$ns_ at 139.0 "\$node_(8) setdest 183.70375260597638 195.25476858929812 1.58"
\$ns_ at 139.0 "\$node_(9) setdest 91.84093537178596 162.37679082696317 6.13"
\$ns_ at 140.0 "\$node_(8) setdest 184.2050104239055 266.21201250868666 2.77"
\$ns_ at 140.0 "\$node_(9) setdest 91.75247394023627 154.81886726893677 6.64"
\$ns_ at 141.0 "\$node_(8) setdest 184.39764419735928 265.72317303683116 3.61"
\$ns_ at 141.0 "\$node_(9) setdest 91.65667407922167 146.63396664350245 7.18"
\$ns_ at 142.0 "\$node_(9) setdest 91.55340236275191 137.81068936761642 7.73"
\$ns_ at 143.0 "\$node_(9) setdest 94.74399305072967 128.42303127171647 8.21"
\$ns_ at 144.0 "\$node_(9) setdest 94.62978040305768 118.66498818624046 8.56"
\$ns_ at 145.0 "\$node_(9) setdest 94.51036414176512 108.46236136205698 8.94"
\$ns_ at 146.0 "\$node_(9) setdest 94.38721195274496 97.94054621264767 9.23"
\$ns_ at 147.0 "\$node_(9) setdest 94.25845587213342 86.93994857539957 9.65"
\$ns_ at 148.0 "\$node_(9) setdest 97.76 90.05 9.91"
\$ns_ at 149.0 "\$node_(9) setdest 107.98 90.05 10.22"
\$ns_ at 150.0 "\$node_(9) setdest 118.49000000000001 90.05 10.52"
\$ns_ at 151.0 "\$node_(9) setdest 129.31 90.05 10.81"
\$ns_ at 152.0 "\$node_(9) setdest 140.3 90.05 11.00"
\$ns_ at 153.0 "\$node_(9) setdest 151.47 90.05 11.16"
\$ns_ at 154.0 "\$node_(9) setdest 162.82 90.05 11.35"
\$ns_ at 155.0 "\$node_(9) setdest 174.36 90.05 11.54"
\$ns_ at 156.0 "\$node_(9) setdest 186.10000000000002 90.05 11.74"
\$ns_ at 157.0 "\$node_(9) setdest 197.97000000000003 90.05 11.87"
\$ns_ at 158.0 "\$node_(9) setdest 209.95000000000002 90.05 11.99"
\$ns_ at 159.0 "\$node_(9) setdest 222.06 90.05 12.11"
\$ns_ at 160.0 "\$node_(9) setdest 234.31 90.05 12.24"
\$ns_ at 161.0 "\$node_(9) setdest 246.71 90.05 12.41"
\$ns_ at 162.0 "\$node_(9) setdest 259.24 90.05 12.53"

Setup traffic flow between nodes
Agent/TCP set window_ 20
Agent/TCP set packetSize_ 1500
Agent/TCP set maxburst_ 0
Agent/TCP set maxcwnd_ 0
Agent/UDP set packetSize_ 1000
Application/Traffic/CBR set rate_ 64Kb
Application/Traffic/CBR set random_ NO
Application/Traffic/CBR set maxpkts_ 2280000

```
set tcp0 [new Agent/TCP]
$tcp0 set class_2
set sink0 [new Agent/TCPSink]
$ns_ attach-agent $node_(2) $tcp0
$ns_ attach-agent $node_(9) $sink0
$ns_ connect $tcp0 $sink0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns_ at 0.0 "$ftp0 start"
```

```
$ns_ at 350.0 "$ftp0 stop"
```

```
set tcp1 [new Agent/TCP]
$tcp1 set class_2
set sink1 [new Agent/TCPSink]
$ns_ attach-agent $node_(5) $tcp1
$ns_ attach-agent $node_(0) $sink1
$ns_ connect $tcp1 $sink1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns_ at 40.0 "$ftp1 start"
```

```
$ns_ at 200.0 "$ftp1 stop"
```

```
set tcp2 [new Agent/TCP]
$tcp2 set class_2
set sink2 [new Agent/TCPSink]
$ns_ attach-agent $node_(3) $tcp2
$ns_ attach-agent $node_(6) $sink2
$ns_ connect $tcp2 $sink2
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
$ns_ at 200.0 "$ftp2 start"
```

```
$ns_ at 500.0 "$ftp2 stop"
```

```
set tcp3 [new Agent/TCP]
$tcp3 set class_2
set sink3 [new Agent/TCPSink]
$ns_ attach-agent $node_(1) $tcp3
$ns_ attach-agent $node_(8) $sink3
$ns_ connect $tcp3 $sink3
set ftp3 [new Application/FTP]
$ftp3 attach-agent $tcp3
$ns_ at 300.0 "$ftp3 start"
```

```
$ns_ at 400.0 "$ftp3 stop"
```

```

set tcp4 [new Agent/TCP]
$tcp4 set class_ 2
set sink4 [new Agent/TCPSink]
$ns_ attach-agent $node_(4) $tcp4
$ns_ attach-agent $node_(7) $sink4
$ns_ connect $tcp4 $sink4
set ftp4 [new Application/FTP]
$ftp4 attach-agent $tcp4
$ns_ at 300.0 "$ftp4 start"

$ns_ at 500.0 "$ftp4 stop"

#
# Tell nodes when the simulation ends
#
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ at $stopTime "$node_($i) reset";
}
$ns_ at $stopTime "stop"
$ns_ at $stopTime "puts \"NS EXITING...\" ; $ns_ halt"
proc stop {} {
    global ns_ tracefd namtrace
    $ns_ flush-trace
    close $tracefd
    close $namtrace
}

puts "Starting Simulation..."
$ns_ run

```