



UNIVERSIDAD MICHOACANA DE SAN NICOLÁS DE HIDALGO

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

ANÁLISIS DE VEHÍCULOS EN UNA RED AD HOC
CON POSIBILIDAD DE REBASE

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

Licenciado en Ciencias Físico

Matemáticas

PRESENTA:

David Baez Magaña

TUTOR:

Dra. Karina Figueroa



Morelia, Michoacán, Febrero 2017

*A la Facultad de Ciencias Físico-Matemáticas y a la Universidad Michoacana De San
Nicolás De Hidalgo, por la formación que me han dado.
Es gracias a ustedes que es posible el presente trabajo.
En verdad, gracias.*

Reconocimientos

También quisiera reconocer en primer lugar y en un plano personal a mis padres todo los esfuerzos y sacrificios que han hecho para que pueda llegar ser la persona que soy hoy en día y para que pueda cumplir con mis ambiciones. Tambien agradecerles a mis hermanos, Marisol y Lupita, la comprensión y el apoyo recibido a lo largo de todos esos años. Es necesario mencionar a mis compañeros de universidad que son mis amigos y que me han acompañado todos esos años en los buenos y en los malos momentos. Quiero agradecer a mi tutor Hector Tejeda V. Ha hecho posible la realizacion de este proyecto, me ha apoyo para que pueda dar un paso más en lo académico y gracias a sus consejos y directivas me ha enseñado mucho.

Declaración de autenticidad

Por la presente declaro que, salvo cuando se haga referencia específica al trabajo de otras personas, el contenido de esta tesis es original y no se ha presentado total o parcialmente para su consideración para cualquier otro título o grado en esta o cualquier otra Universidad. Esta tesis es resultado de mi propio trabajo y no incluye nada que sea el resultado de algún trabajo realizado en colaboración, salvo que se indique específicamente en el texto.

David Baez Magaña. Morelia, Michoacán, 2016

Resumen

Palabras clave: redes,computo,plataforma NS3,ad-hoc.

Una preocupación actual es el incremento de la tasa de accidentes de tránsito en los últimos años, que ha llevado a la muerte de 1.5 millones de personas a nivel mundial, y la tendencia es que siga creciendo en los próximos años (OMS 2013). Para lograr la reducción de accidentes, se han implementado diversas tecnologías que han hecho a los vehículos más seguros, como los frenos antibloqueo. Como los automovilistas hacen uso de vías compartidas, o bien, tienen intersección con vías férreas, o pasos peatonales, entonces se requieren soluciones que tengan en cuenta la interacción con otros usuarios. Una manera de saber de los diferentes usuarios es mediante el uso de tecnologías de telecomunicación e informática. Lo anterior se conoce como un sistema de transporte inteligente, el cual incrementa la seguridad de los usuarios, además tiene otros beneficios como el reducir el tiempo de viaje, el consumo de combustible, y la contaminación ambiental.

En este trabajo se hace una contribución al sistema de transporte inteligente al estudiar una simulación de una red sin infraestructura, o ad-hoc, con un conjunto mínimo de tres vehículos en movimiento uniformemente rectilíneo acelerado y movimiento uniformemente rectilíneo. Para el establecimiento de la red de vehículos ad-hoc, los vehículos deben proporcionar información de su posición, mediante un dispositivo GPS y su velocidad promedio. El objetivo de la simulación es determinar si una comunicación de difusión amplia, o *broadcast* parcial en la red ad-hoc, es suficiente para que un vehículo tenga suficiente información para determinar si es seguro rebasar a otro vehículo.

Para establecer este modelo, se sabe que una colisión puede ocurrir entre tres vehículos o más, razón por el cual solo se analizan tres vehículos, considerando que la colisión puede suceder tanto en el carril izquierdo como en el derecho. Para evitar la pérdida de mensajes se utilizan las ecuaciones de movimiento en una recta, motivo por el que existe una gran ventaja debido a su manejo sencillo de sus variables como lo es la velocidad y aceleración constante, además con la comunicación inalámbrica de un salto, se evitan inundaciones. Todo esta investigación se hizo con la intención de simular de una forma más simple que compleja y que en base a éste puede ser alternativa para mejorarlo en

un futuro.

Usando una plataforma de simulación ns-3, la cual se puede instalar en cualquier equipo con Linux y en base a la programación en el lenguaje C++ y Python , se implementaron las bibliotecas adecuadas para configurar un esquema de nodos (vehículos), con el fin de analizar los parámetros de tiempo, posición y velocidad, el cual se pueden visualizar en la terminal y animador gráfico. Con lo cual se obtuvieron resultados de margen de distancia para rebasar y tiempo de encuentro entre los vehículos que circulan en sentido contrario. Este análisis contribuye al uso de nuevas tecnologías, aplicaciones, estándares asociados y el desarrollo de un proyecto específico, en nuevos sistemas de comunicación.

ABSTRACT

A current concern is the increase in the rate of traffic accidents in recent years, which has led to the death of 1.5 million people worldwide, and the trend is to continue growing in the coming years (WHO 2013). To achieve the reduction of accidents, they have implemented various technologies that have made vehicles safer, such as antilock brakes. As motorists make use of shared paths or have to railways intersection or crosswalk, they require solutions that take into account the interaction with other users. One way to know the different users is through the use of telecommunications and information technologies. This is known as an intelligent transportation system, which increases the safety of users, also has other benefits such as reducing travel time, fuel consumption and environmental pollution.

In this paper a contribution to intelligent transport system by studying a simulation of a network without infrastructure or ad-hoc, with a minimum set of three vehicles in uniformly accelerated rectilinear motion and rectilinear motion uniformly done. For the establishment of the network of ad-hoc vehicles, vehicles must provide position information via a GPS device and its average speed. The goal of the simulation is to determine whether a communication widely available, or partial broadcast in the ad-hoc network, is sufficient for a vehicle has enough information to determine whether it is safe to pass another vehicle.

To establish this model, it is known that a collision can occur between three vehicles or more, which is why only three vehicles are analyzed, considering that the collision can occur both in the left lane and right. To avoid losing messages equations of motion in a straight line, which is why there is a great advantage because of its simple handling of its variables such as the speed and constant acceleration, are used in addition to the wireless communication of a jump, floods are avoided. All this research was intended to simulate a simpler complex shape and that based on this may be an alternative to improve in the future.

Using a simulation platform ns-3, which can be installed on any computer running Linux and based on programming in C ++ and Python language, the right to set up a

scheme nodes (vehicles) libraries, to be implemented analyzing the parameters of time, position and velocity, which can be displayed on the terminal and graphic animator. Whereupon results were obtained clearance distance and time to move beyond meeting between vehicles traveling in the opposite direction. This analysis contributes to the use of new technologies, applications, associated standards and the development of a specific project, new communication systems.

Índice general

Índice de figuras	XIII
Índice de tablas	XV
1. Introducción	1
1.1. Presentación	1
1.2. Objetivo	2
1.2.1. Objetivo general	2
1.2.2. Objetivos específicos	2
1.3. Motivación	2
1.4. Planteamiento del problema	3
1.5. Contribuciones	4
1.6. Estructura de la tesis	4
2. INVESTIGACIÓN DOCUMENTADA	7
2.1. Redes Ad-Hoc Vehículos	7
2.1.1. Características de la redes VANETs:	8
2.1.2. Estándares inalámbricos	9
2.2. Protocolos de enrutamiento	11
2.2.1. Introducción	11
2.2.2. Protocolos Broadcast	14
2.2.3. Protocolos unicast	14
2.2.4. Protocolos multicast	18
3. Diseño del escenario	21
3.1. Justificación de escenarios a simular.	21
3.1.1. Cuestión cinemática	22
3.1.2. Características al problema broadcast	22
3.1.3. Análisis de la zona extra que puede beneficiarse de un mensaje broadcast.	24
3.1.4. Escenario 1: Comunicación Broadcast Con Movimiento Rectilíneo Uniformemente Acelerado(MRUA)	26

ÍNDICE GENERAL

3.1.5. Escenario 2: Comunicación Broadcast Con Movimiento Rectilineo Uniforme(MRU)	28
4. Plataforma de simulación	29
4.1. Componentes Software de la plataforma de simulación	29
4.2. Procesos De Simulación	31
4.3. Instalación de la plataforma	35
4.3.1. Sistema operativo y soporte del compilador	36
5. Resultados	41
5.1. Comunicación entre cuatro vehículos con (MRUA).	42
5.1.1. Resultados con velocidad mínima	42
5.1.2. Resultados con velocidad máxima	46
5.2. Comunicación entre cuatro vehículos con (MRU).	48
5.2.1. Resultados con velocidad mínima	48
5.2.2. Resultados con velocidad máxima	50
6. Conclusiones	53
6.1. Logros	53
6.2. Futuras líneas de trabajo	54
A. Código/Manuales/Publicaciones	55
A.1. Apéndice	55
Bibliografía	59

Índice de figuras

2.1. Diagrama 802.11b	10
2.2. MPR Y NES	14
2.3. Protocolo DSDV	15
2.4. Protocolo AODV	17
2.5. Protocolo OLSR	18
3.1. Conectividad broadcast entre nodos blancos grises y negros	23
3.2. Superposición de señales	24
3.3. Rango de cobertura	25
3.4. Proyección en un camino vial	27
3.5. Broadcast en línea recta con MRUA	28
4.1. Ns3 Módulos	32
5.1. Trayectorias de los vehículos	42
5.2. Comparación de datos de la Tabla 5.1, 5.2, 5.3 y 5.4	45
5.3. Paquetes Tx y Rx	45
5.4. Comparación de datos de la Tabla 5.3, 5.4, 5.5 y 5.6	47
5.5. Datos de los paquetes transmitidos y recibidos	48
5.6. Datos de los paquetes transmitidos y recibidos	50
5.7. Características del vehículo 2 respecto al vehículo 1.	51
5.8. Datos de los paquetes transmitidos y recibidos	52

Índice de tablas

2.1. Estándares inalámbricos.	11
2.2. Características de protocolos.	13
3.1. Símbolos del MRUA.	26
4.1. Módulos de ejecución.	32
5.1. Características escalares del vehículo V1 respecto al vehículo V2 en donde V1 manda una transmisión de paquetes en un tiempo de $t=0.5$ s a los vehículos V0, V2 y V3.	43
5.2. Características escalares del vehículo V2 respecto al vehículo V1 en donde V2 manda una transmisión de paquetes en un tiempo de $t=1$ s a los vehículos V0, V1 y V3.	43
5.3. Características escalares del vehículo V1 respecto al vehículo V2 en donde V1 manda una transmisión de paquetes en un tiempo de $t=0.3$ s a los vehículos V0, V2, y V3.	44
5.4. Características escalares del vehículo V2 respecto al vehículo V1 en donde V2 manda una transmisión de paquetes en un tiempo de $t=1$ s. a los vehículos V0, V1 y V3.	44
5.5. Características escalares del vehículo V1 respecto al vehículo V2 en donde V1 manda una transmisión de paquetes en un tiempo de $t=0.3$ s a los vehículos V0, V2 y V3	46
5.6. Características escalares del vehículo V2 respecto al vehículo V1 en donde V2 manda una transmisión de paquetes en un tiempo de $t= 0.6$ a los vehículos V0, V1 y V3.	47
5.7. Características escalares del vehículo V1 respecto al vehículo V2 en donde V1 manda una transmisión de paquetes en un tiempo de $t=0.3s$ a los vehículos V0, V2 y V3.	49
5.8. Características escalares del vehículo V2 respecto al vehículo V1 en donde V2 manda una transmisión de paquetes en un tiempo de $t= 0.6$ a los vehículos V0, V1 y V3.	49

ÍNDICE DE TABLAS

5.9. Características escalares del vehículo V1 respecto al vehículo V2 en donde V1 manda una transmisión de paquetes en un tiempo de $t= 0.3$ a los vehículos V0, V2 y V3.	50
5.10. Características escalares del vehículo V2 respecto al vehículo V1 en donde V2 manda una transmisión de paquetes en un tiempo de $t= 0.6$ a los vehículos V0, V1 y V3.	51

Introducción

1.1. Presentación

Las redes inalámbricas han revolucionado al mundo con los intercambios de datos y definiendo un nuevo paradigma. Dentro de este paradigma, el tráfico de vehículos abren un nuevo campo de investigación en el ámbito científico. Las redes de comunicaciones ad-hoc web(1), y en específico las redes vehiculares ad-hoc o VANETs (*Vehicular Ad-Hoc Network*) plantean nuevos retos. En este tipo de redes no existe infraestructura de red sino que se compone de los propios nodos móviles comunicándose entre sí por enlaces inalámbricos. Los nodos deben asumir ciertas responsabilidades de enrutamiento de datos y de mantenimiento de la red. Tomemos en cuenta que los nodos, toman un control en la red. Las características de VANETs son en general hostil al intercambio de tráfico.

Las redes ad-hoc en movimiento se pueden utilizar entre los vehículos que utilizan el medio inalámbrico para establecer la comunicación, es decir no requieren un punto de acceso centralizado. En su lugar, los dispositivos de la red inalámbrica se conectan directamente entre sí. Si configuramos tres nodos en el modo inalámbrico ad-hoc, éstos se podrán conectar correctamente sin necesidad de pasar o usar un router central. Estas redes son importantes debido a que existe un continuo intercambio de información entre los usuarios que se encuentran en sus vehículos en movimiento, así como el intercambio de información. El objetivo de este proyecto es estudiar y simular redes con tres nodos en movimiento, generando un escenario en particular que permita comprender su funcionamiento, principales aplicaciones para evitar una colisión entre vehículos.

1.2. Objetivo

1.2.1. Objetivo general

Este trabajo tiene por objetivo realizar un estudio y simulación de redes con un conjunto de tres vehículos en movimiento rectilíneo basándonos en la comunicación broadcast y con el protocolo de enrutamiento *User Datagram Protocol* (UDP) que permita comprender su funcionamiento fijando algunas variables (distancia, posición tiempo velocidad, número y tamaño de paquetes, transmitidos y recibidos) que den la información suficiente en dos escenarios uno con movimiento uniformemente rectilíneo y el otro con movimiento rectilíneo acelerado y conforme a la dependencia e independencia de unas variables respecto a otras, observar si entre los resultados a determinar existen posibilidades de haber o no colisiones entre los vehículos dentro del perímetro de simulación.

1.2.2. Objetivos específicos

Para llevar a cabo el objetivo general se seleccionará un simulador (ns-3), que permita realizar una exploración sobre la información existente de los módulos, librerías, clases y métodos más apropiadas que existen en el simulador con fin de establecer, la forma de trabajo de estas redes, sus características, requerimientos, estándares y aplicaciones.

Una vez que se haya estudiado a fondo los módulos del simulador ns-3, es necesario montar una tarjeta de red inalámbrica con el estándar 802.11b en modo ad-hoc, haciendo una difusión amplia de paquetes *broadcast*. Se requiere estar cambiando los datos de las variables en cada escenario a simular para poder analizar los resultados de forma gráfica y virtual.

Por último, con el modelo de simulación ya establecido, se analizará su eficiencia y así mismo determinar qué tan apropiado es para llevarlo a cabo en un caso más real, con el fin de mejorar la vialidad y accidentes de tráfico.

1.3. Motivación

A menudo en que día con día se ve la problemática que existe en el tráfico vehicular, se ha invertido tiempo y dedicación en la búsqueda de estrategias para darle solución a dicha problemática. En este caso se optó por entrar a la parte de la telecomunicación trabajando con redes ad-hoc, el cual se ha planteado un escenario en una forma simple, donde circulan un conjunto de vehículos en una línea recta. La razón por la que se planteó simular solo con tres vehículos fue por el simple hecho de proyectarlo de lo

más simple a lo más complejo, es decir, para que exista una colisión se requiere de tres vehículos, dos en un sentido y uno en sentido contrario.

Este surgimiento intrínseco resulta el punto de partida en la que esta idea toma forma, y va a condicionar el cómo se llevará a cabo a la sociedad, es decir, la estrategia de ejecución del modelo de simulación de tráfico vehículos(nodos) móvil hacia un modelo más amplio. Como ya hemos hablado, existen muchas formas de identificar oportunidades de lograr hacer este tipo de tecnología, igual que existen muchos tipos de personas investigando sobre este tipo de sistemas de comunicación, así que el propósito es llegar a cierto punto de investigación y que en un tiempo a futuro se le de continuidad y sea más sofisticado.

1.4. Planteamiento del problema

En este apartado se describen los requisitos mínimos a los cuales se deberá ajustar la solución propuesta para resolver el problema previamente planteado. Dada la propuesta en cuanto al programa para llevar a cabo la simulación de dos escenarios posibles, se tendrá que definir de manera concreta los requisitos de la solución contemplada para poder analizar si se puede evitar algún porcentaje de riesgos de colisiones dentro del tráfico de vehículos. De un lado, se comentarán los requisitos de la plataforma de simulación que debe cumplir para proporcionar un entorno fiable para las simulaciones. Por otro lado, se definirán requisitos de la topología de red para llevar a cabo el escenario.

Para cumplir de manera satisfactoria los requisitos de este proyecto la plataforma de simulación debe presentar características específicas. Estos requisitos son:

- Los diferentes programas que componen la plataforma deben ser libres y de código abierto. Este requisito es fundamental por dos razones. Primero, por razones económicas obvias; el carácter libre del código nos libera de la necesidad de pagar licencias. Por otro lado, es necesario que sea de código abierto para temas de flexibilidad a la hora de modificar partes del código si fuese necesario para alcanzar nuestros objetivos de simulaciones.
- Es necesario tener implementado en el simulador el protocolo elegido para observar y analizar dicha simulación.
- Es necesario disponer de herramientas que permitan especificar de forma rápida los escenarios que se proponen simular. Esas herramientas deben proporcionar la posibilidad de definir patrones de movimiento de los vehículos utilizando el protocolo de enrutamiento que se plantea.
- Es necesario que el simulador proporcione estadísticas de la red, sobre el envío de los paquetes, como por ejemplo; número de paquetes(Rx, Tx), tamaño de paquetes (Rx, Tx), la ruta del paquete en otro contexto, quien envía a quien, velocidades, posiciones, tiempo distancia aceleración, etc. . . .

1. INTRODUCCIÓN

- Es útil que el simulador disponga de un animador gráfico. Esta herramienta muestra de forma gráfica y dinámica como se ha producido la simulación, los movimientos de los nodos, los paquetes transmitidos...etc.
- El objetivo final de este proyecto es obtener la simulación de uno de los protocolos de encaminamiento en entornos vehiculares, por lo cual el escenario a definir deben ser aproximaciones de la situaciones reales de este entorno. Por lo tanto se han elegido una línea recta asimilada con una (autopista).

Línea Recta

Al igual que en el caso del circuito urbano se debe considerar situaciones las más realistas posibles. Sin embargo, considerar una línea recta en su integralidad nos llevaría a un consumo de tiempo y de memoria considerable. Por lo tanto, consideraremos un tramo bastante significativo de las comunicaciones intercambiadas en ese entorno y aproximamos el comportamiento total considerando que corresponde a la suma de los comportamientos individuales simulados. Los requisitos son los siguientes:

- 300 mts de recorrido.
- Dos sentidos de circulación próximos entre sí.
- Cada sentido separadas por una línea de 20 cm de ancho.
- Velocidades absolutas de los vehículos entre 80, 100 120, 150 y 170 Km/h.
- Tramos muy uniformes, sin fuertes pendientes y radios de curvatura completamente nulos.

1.5. Contribuciones

La contribución principal de este trabajo es observar de alguna manera sí con la simulación de redes ad-hoc y comunicación broadcast entre vehículos que se realizó sea posible disminuir los accidentes de tráfico en una línea recta (autopista), más sin embargo se tiene en cuenta que este trabajo es solo una simulación y que faltaria complementarlo con montarlo físicamente. De otra forma este trabajo será útil para quienes requieran tomarlo dándole continuidad y tal vez lo logren mejorar con el fin de que su porcentaje de accidentes sea menor aun.

1.6. Estructura de la tesis

Este trabajo está dividido en 6 capítulos:
En el capítulo 2 se ocupará de describir algunos conceptos de las redes VANETs. Se describe las características principales de las redes VANETs, las tecnologías implicadas

en su desarrollo y los aspectos a investigar. Se detendrá con atención en describir los protocolos de encaminamiento que se han desarrollado para este tipo de redes. Y finalmente, describe el principal interés con el cual se pretende desarrollar sobre este tipo de redes con el propósito de utilizar la más conveniente para llevar a cabo la simulación.

En el tercer capítulo se definen los escenarios de comunicación a simular. Es decir estudiando qué pruebas queremos llevar a cabo.

El capítulo 4 se centra en la plataforma de simulación que se ha desarrollado para llevar a cabo este proyecto. Se describen los programas utilizados, sus instalaciones y el funcionamiento de la plataforma de simulación. Luego, se detallan las modificaciones efectuadas al código fuente original para la perfecta adecuación a nuestros objetivos.

En el capítulo 5, se presentan los resultados de las simulaciones realizadas. A partir de los escenarios definidos en el capítulo 3.

Finalmente, en un último capítulo se detendrá las conclusiones obtenidas en este proyecto.

INVESTIGACIÓN DOCUMENTADA

En este capítulo se da una visión general de los desafíos que plantean las redes VANETs (*Vehicular Ad-Hoc Network*). Se inicia definiendo este tipo de redes, las aplicaciones de las VANET, sus estándares y protocolos de enrutamiento etc. Siendo muy conscientes de la investigación actual sobre estos temas, y de las limitaciones de cantidad de información que podemos aportar, se hará una exposición muy por debajo de lo normal.

2.1. Redes Ad-Hoc Vehículos

Como definición más formal, se define un nodo, como el un punto de intersección, conexión o unión de varios elementos que confluyen en el mismo lugar, siendo componente principal de una red. En otras palabras, tanto si se trata de Internet, cada servidor u ordenador constituye un nodo y se encuentra conectado a otro u otros nodos. En segundo plano una red ad hoc (7) es un tipo de red formada por un grupo de nodos móviles que forman una red temporal sin la ayuda de ninguna infraestructura externa. Para que esto se pueda llevar a la práctica es necesario que los nodos se puedan ayudar mutuamente para conseguir un objetivo común: que cualquier paquete llegue a su destino aunque el destinatario no sea accesible directamente desde el origen.

Antes de pasar a las redes VANETs, se define una red móvil ad-hoc o MANET (*Mobile Ad hoc Network*) como una colección de nodos inalámbricos móviles que se comunican de manera espontánea y autoorganizada constituyendo una red temporal sin la ayuda de ninguna infraestructura preestablecida (como puntos de acceso WiFi o torres de estaciones base celulares) ni administración centralizada. Los equipos o nodos que forman parte de ella (*Notebooks, PDAs, Celulares*), se organizan por sí mismos para ayudarse los unos a los otros en el proceso de transportar paquetes de datos entre un origen y un destino. Por tanto las MANET dan un paso más en cuanto a movilidad (todos los nodos de la red pueden ser móviles) y flexibilidad (no se requiere inversión en infraestructura, y se minimiza la gestión de la red pues se auto-organiza ella misma).

Se sabe que el protocolo de enrutamiento es el responsable de descubrir las rutas entre los nodos para hacer posible la comunicación, sin embargo las Redes VANETs son un caso particular enfocadas a entornos vehiculares. Se trata de un conjunto de nodos que se comunican entre sí mediante enlaces inalámbricos sin la necesidad de una infraestructura de red fija. Cada nodo actúa como enrutador y tiene capacidades de encaminamiento para redirigir paquetes hacia su destino.

2.1.1. Características de la redes VANETs:

Movilidad: Este aspecto es la razón de ser de las redes Ad-hoc. Los nodos se pueden reubicar o simplemente ser móviles. Se pueden desplegar rápidamente sin la necesidad de descubrir la zona o formar grupos, es decir, cada nodo es individual y solvente.

Autonomía: Cada nodo es un nodo autónomo con capacidad de procesamiento de la información que se intercambia en la red. El control de la red no depende de una infraestructura externa sino que se distribuye en todos los nodos de la red siendo así más tolerante a fallos.

Topología de red variable: En una MANETs los nodos se pueden mover de forma arbitraria. Esa característica se debe matizar en el caso de las VANETs (10) ya que los vehículos suelen seguir un cierto patrón de movimiento, por ejemplo siguiendo las curvas de un circuito urbano. Aún así, los vehículos se mueven de forma más rápida que un terminal en una red móvil clásica. Debido a esa variabilidad de posición se pueden producir pérdidas importantes de paquetes. Serán necesarios mecanismos que detecten estas circunstancias y minimicen sus efectos.

Canales variables en tiempo y frecuencia: Debido a la velocidad de los vehículos, a los ambientes donde circulan y a los posibles obstáculos para la señal inalámbrica (edificios, árboles, etc.), la comunicación puede sufrir desvanecimientos en tiempo o frecuencia con mayor intensidad que otras redes móviles.

Nodos limitados: En la mayoría de los casos los nodos de este tipo de redes, serán embarcados en vehículos con capacidades limitadas de procesamiento, comunicación y alimentación por lo que es primordial que los algoritmos utilizados optimicen estos tres recursos.

Seguridad: Para poder acceder a una red cableada, un usuario tiene que tener acceso físico al cable. Sin embargo, las comunicaciones inalámbricas, ya que usan el aire como canal de comunicación, son débiles en cuanto a consideraciones de seguridad.

2.1.2. Estándares inalámbricos

En este apartado se detallan tecnologías inalámbricas susceptibles de dar soporte a las redes VANETs. Entre una de ellas son los IEEE, organización de líderes en la creación de estándares en el mundo. IEEE realiza sus estándares y mantiene las funciones a través de la Asociación de estándares IEEE. Estándares (3) IEEE afectan a una amplia gama de industrias, incluyendo: el poder y la energía, biomedicina y salud, tecnología de la información, las telecomunicaciones, el transporte, la nanotecnología, la seguridad de la información, y muchos más. En 2013, la IEEE tenía más de 900 estándares activos, con más de 500 normas en elaboración. Uno de los más notables estándares IEEE es la IEEE 802 LAN/MAN grupo de normas que incluye el estándar IEEE 802.3 *Ethernet* y el estándar IEEE 802.11 de red inalámbrica.

Estándares IEEE 802.11

Más conocida como WiFi, se basa en el estándar IEEE 802.11. Opera en bandas libres, es decir, aquellas que no requieren permiso especial de las autoridades o el regulador de telecomunicaciones para poder usarse. Las versiones b y g se han extendido mucho hasta el punto de que la mayoría de los equipos portátiles y PDAs la traen incorporada de serie. Tiene un alcance de unas centenas de metros y un ancho de banda de hasta 54 Mbps, dependiendo de la versión del estándar. La nueva versión, 802.11n pretende aumentar las tasas de transferencia hasta un 500Mbps. La seguridad forma parte de los protocolos desde el principio y fue mejorada en la revisión 802.11i

802.11p

Conocida como *Wireless Access for the Vehicular Environment* (WAVE), está en proceso de estandarización y será la encargada en un futuro de soportar las comunicaciones vehiculares. WAVE es una evolución del estándar IEEE 802.11a con modificaciones a nivel físico y MAC para mejorar su comportamiento en el entorno vehicular y dar soporte a sistemas de transporte inteligente (*Intelligent Transportation Systems* (ITS)). Asimismo, WAVE será la base sobre la que se desarrollará el DSRC (*Dedicated Short Range Communications*), otro proyecto de estandarización impulsado por el ministerio de transporte de EE UU y por un número importante de fabricantes de la industria automovil, cuyo objetivo es crear una red nacional de comunicaciones vehiculares. El propósito del proyecto es definir un estándar para las comunicaciones V2V y las comunicaciones con la infraestructura vial (V2I) que se puede instalar en semáforos o paneles de información, por ejemplo. La fecha de publicación de la primera versión del estándar está prevista para Abril 2009.

WAVE pretende aumentar las tasas de transferencia a corto alcance, típicamente entre 100 y 500m. La técnica de modulación se basa en IEEE802.11a, utilizando OFDM pero con tasas de transmisión de 3, 4.5, 6, 9, 12, 18, 24 y 27 Mbps en canales de 10MHz. En cuanto a la canalización, la norma define 7 canales no solapados de 10MHz en la banda de 5.9 GHz: 6 canales de servicio (SCH) y uno de control (CCH). El CCH está utilizado como canal de referencia para realizar una primera detección de los vehículos

2. INVESTIGACIÓN DOCUMENTADA

cercanos como paso previo al establecimiento de las comunicaciones. Al mismo tiempo, dicho canal se usa para anunciar los servicios disponibles en canales SCH (acceso a Internet, descarga de contenidos..etc.) El canal CCH se usa para la transmisión en modo broadcast de mensajes de seguridad vial. En la capa MAC, WAVE se basa en las definiciones del IEEE802.11 usando una técnica de acceso basada en CSMA/CA (*Carrier Sense Multiple with Collision Avoidance*). Sin embargo, CSMA/CA no logra solucionar el problema del terminal escondido. Los tráficos se transmiten en el canal CCH con una tasa de datos de 6Mbps, correspondiente a una modulación QPSK con un ratio de codificación de 1/2. En la capa MAC, WAVE se basa en las definiciones del IEEE802.11 usando una técnica de acceso basada en CSMA/CA (*Carrier Sense Multiple with Collision Avoidance*). Sin embargo, CSMA/CA no logra solucionar el problema del terminal escondido.

802.11b

El estándar original fue ratificado en 1999. 802.11b tiene una velocidad máxima de transmisión de 11 Mbps y utiliza el mismo método de acceso definido en el estándar original CSMA/CA. En la figura 2.1 el estándar 802.11b funciona en la banda de 2,4 GHz. Debido al espacio ocupado por la codificación del protocolo CSMA/CA, en la práctica, la velocidad máxima de transmisión con este estándar es de aproximadamente 5,9 Mbit/s sobre TCP y 7,1 Mbit/s sobre UDP.

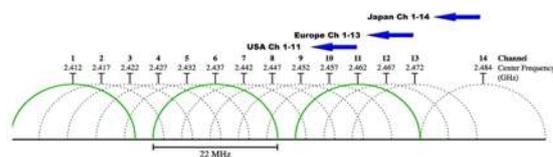


Figura 2.1: Diagrama 802.11b

Bluetooth

También conocido como 802.15.1. Es la tecnología más extendida en cuanto a comunicaciones inalámbricas personales (wPAN). Hay varias clases dependiendo de su alcance y consumo de potencia, alcanzando tasas de 2Mbps y rangos de hasta 100m. Opera en banda libre y sus mecanismos de seguridad son suficientemente robustos.

UWB

Ultra Wide Band es un estándar basado en 802.15.3 que funciona emitiendo a muy baja potencia en un espectro enorme. Su alcance es muy limitado (¡10m) pero proporciona tasas de transferencia muy elevadas llegando a los 480 Mbps. Su consumo de energía es muy reducido.

ZigBee

Es la tecnología más utilizada en redes de sensores ad hoc. Se basa en el estándar 802.15.4. Presenta anchos de banda muy pequeños y cobertura reducida (250 Kbps hasta 75m). Es de gran utilidad para enviar poca información en pequeñas distancias.

Tabla 2.1: Estándares inalámbricos.

Tecnología	Cobertura	Tasas	Consumo
802.11b	300m	11 Mbps	Alto
802.11g	500m	54 Mbps	Alto
Bluetooth (802.15.1)	20m	2 Mbps	Medio
UWB (802.15.3)	<10m	480 Mbps	Bajo
Zigbee(802.15.4)	75m	250 Kpbs	Muy bajo

la gran ventaja es que su consumo es extremadamente reducido. A continuación se muestra una tabla con las características más relevantes de cada tecnología:

NFC *Near Field Communication*, NFC es una nueva tecnología de comunicación inalámbrica que proviene de la combinación de varias tecnologías de identificación e interconexión. NFC provee una forma de comunicación entre dispositivos electrónicos, como pueden ser los teléfonos móviles, las PDAs...etc. La comunicación se realiza entre dos dispositivos de forma "peer-to-peer". Trabaja en la banda de los 13.56 MHz, banda que no necesita compra de licencia para su uso. El alcance de NFC es extremadamente corto, se usa para comunicaciones de dispositivos que se encuentran a menos de 4 cm de distancia. Dado ese rango muy corto de cobertura, las comunicaciones son de forma inherente totalmente seguras. Las tasas de transferencia son de hasta 424 Kbps. Según el entorno de los dispositivos se negocia las velocidades de transferencia y se puede reajustar ese parámetro en cualquier momento de la comunicación. En entorno vehicular, NFC ofrece muchas posibilidades de aplicación. Podemos citar por ejemplo, el pago de peajes, el uso del dispositivos como llave integrada, control de acceso a servicios de ocio...etc. NFC goza de una gran aceptación por parte de fabricantes e industrias del sector en general, por lo cual se espera una gran implementación en todo tipo de dispositivos de comunicación en un futuro.

2.2. Protocolos de enrutamiento

2.2.1. Introducción

La investigación en el campo de los protocolos de encaminamiento para redes ad-hoc se ha multiplicado estos últimos años. La movilidad de los nodos, la inestabilidad de las topologías, y la ausencia de una infraestructura de centralización hacen obsoletos los protocolos que se usan en redes fijas. En redes ad-hoc, los protocolos de encamina-

2. INVESTIGACIÓN DOCUMENTADA

miento deben ser capaces de funcionar de manera automática y distribuida.

A la hora de clasificar los protocolos de encaminamiento (2) existen varios criterios. Se puede considerar:

- El alcance:

unicast: El término unicast (8) hace referencia al envío de paquetes o información desde un único emisor a un único receptor. Ejemplos básicos de aplicaciones unicast son los protocolos http, smtp, ftp o telnet. Actualmente es la forma predominante de transmisión en Internet. En términos cotidianos, una comunicación unicast podría ser por ejemplo una llamada telefónica entre dos personas.

broadcast: Broadcast es un modo de transmisión de información donde un nodo emisor envía información a una multitud de nodos receptores de manera simultánea, sin necesidad de reproducir la misma transmisión nodo por nodo. En la vida cotidiana, un ejemplo de comunicación Broadcast es el de una emisora de radio, que emite señales sin saber quien la recibe, el receptor decide si recibirla o no, al igual que la señal de la televisión, que se envía a todos los receptores.

multicast: Multicast (multidifusión) es el envío de información en una red a múltiples receptores de forma simultánea, un emisor envía un mensaje y son varios los receptores que reciben el mismo. Si antes hablábamos de que una comunicación unicast era una llamada telefónica entre dos personas, podemos decir que una comunicación multicast podría ser una conferencia, en la que son varias las personas que se comunican entre sí. Un ejemplo claro de comunicación multicast en Internet es un IRC (*Internet Relay Chat*).

- El modo de descubrimiento de rutas : proactivo, reactivo, híbrido.
- Tipo de algoritmo que implementan : vector de distancias, estado de enlace.

Los protocolos que serán descritos son:

- Broadcast: Flooding, MPR, NES, CDS
- Unicast: DSDV, DSR, AODV, LAR, TORA, ZRP, OLSR y FSR
- Multicast: MAODV

Se han elegido esos protocolos por ser representativos de su grupo, por ser los más usados a nivel de simulación.

Tabla 2.2: Características de protocolos.

Protocolo	Alcance	Esquema	Información geográfica
Blind Flooding	Broadcast	-	No
MPR	Broadcast	-	No
NES	Broadcast	-	No
CDS	Broadcast	-	No
DSDV	Unicast	Proactivo	No
DSR	Unicast	Reactivo	No
AODV	Unicast	Reactivo	No
LAR	Unicast	Proactivo	Sí
TORA	Unicast	Reactivo	No
ZRP	Unicast	Híbrido	No
FSR	Unicast	Proactivo	No
OLSR	Unicast	Proactivo	No
MAODV	Multicast	Reactivo	No

2.2.2. Protocolos Broadcast

Blind-Flooding

Blind Flooding es el protocolo más simple. A la recepción de un mensaje, un nodo lo reenvía a todos sus vecinos. La única optimización que presenta este protocolo es que cada nodo recuerda los paquetes flooding que ha recibido y si le vuelven a llegar no los retransmite evitando así duplicidades. Aunque sea muy simple de implementar, el *"Blind Flooding"* introduce mensajes redundantes y colisiones a nivel MAC que empeoran el rendimiento de la red.

Multi-Point Relay Flooding (MPR)

MPR consiste en elegir un conjunto de nodos vecinos que cubre el acceso a los nodos distantes de 2 saltos. Los nodos de ese conjunto reenvían el tráfico, los demás no. Esta mejora permite dividir por 2 el número de mensajes de control. Véase en la fig. 2.2.

Neighbor Elimination Scheme (NES)

Un nodo que recibe un mensaje de broadcast no retransmite directamente sino que espera un tiempo aleatorio para ver si otro nodo manda la información. Los nodos escuchan los mensajes y apuntan que nodos ha mandado información a cual otro. Después del tiempo de espera, el nodo manda el tráfico a sus vecinos que no han sido informados por otros nodos. Véase en la fig. 2.2.

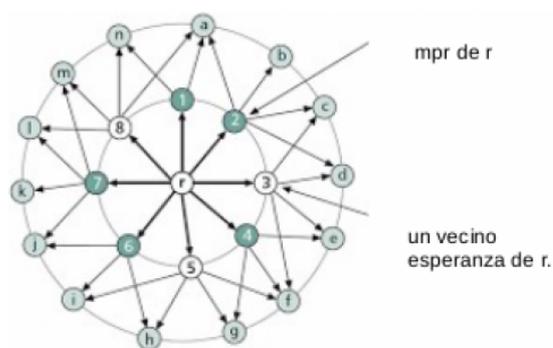


Figura 2.2: MPR Y NES

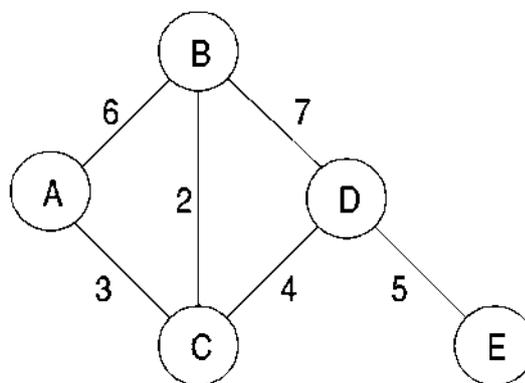
2.2.3. Protocolos unicast

DSDV

Destination-sequenced Distance-Vector Routing (DSDV) es un protocolo unicast proactivo adaptado del tradicional RIP (*Routing Information Protocol*). Su principal objetivo es evitar los problemas de bucles en la actualización de las tablas de encaminamiento. Por lo cual añade un nuevo campo a las tablas RIP, el número de secuencia

que permite distinguir entre una tabla antigua y una más reciente. Como su nombre lo indica, DSDV implementa un algoritmo basado en el vector de distancias. Eso significa que mantiene tablas con todos sus destinos accesibles junto con el siguiente salto, la métrica, y un número de secuencia de la entrada en la tabla generado por el nodo destino. Las tablas se mandan en modo broadcast de forma periódica o cuando ocurre un cambio significativo de la topología de red. Una ruta es considerada mejor que otra si tiene un número de secuencia mayor o, en caso de empate, si la distancia al destino es menor.

De la fig. 2.3, cuando un nodo B detecta que la ruta hacia cierto destino D se ha roto, inunda la red con una actualización de esa entrada en la que se ha incrementado el número de secuencia en uno y la distancia se marca como infinita. Cuando A recibe este mensaje incorpora a su tabla la actualización de la entrada hacia D a través de B siempre que no tuviera una entrada mejor para alcanzar D. Para conseguir una cierta consistencia en las tablas de encaminamiento de cada nodo al cambiar la topología de la red, las actualizaciones deben ser frecuentes y suficientemente rápidas para que cada nodo pueda tener una visión realista de la red en un momento dado. El problema fundamental de DSDV es la elevada sobrecarga de control que genera. Al no haber una especificación estándar, no hay productos comerciales basados en este protocolo. Sin embargo, es la base sobre cual se han desarrollado otros protocolos como por ejemplo AODV.



(a) The network

Figura 2.3: Protocolo DSDV

DSR

Dynamic Source Routing es un protocolo reactivo unicast. El protocolo se compone de dos mecanismos: el descubrimiento y el mantenimiento de rutas que permiten a un nodo origen descubrir y mantener las rutas hacia un nodo destino cuando se necesita mandar tráfico en la red ad-hoc. Se basa en una técnica de “*Source Routing*”. La idea de esta técnica es determinar la mejor ruta completa hacia un destino. El nodo origen

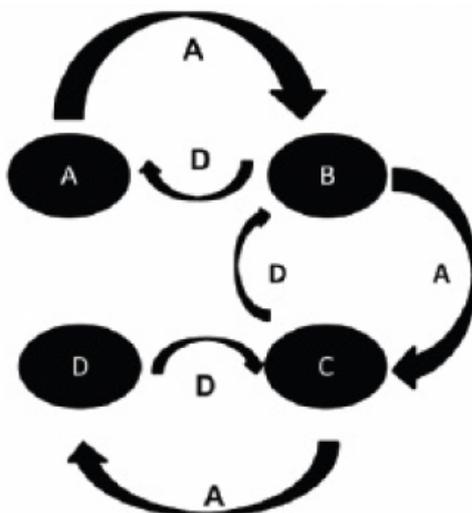
inunda la red con una trama de exploración. Al recibir una replica de la trama exploradora, cada nodo se agrega explícitamente en la cabecera de la trama, y actualiza sus tablas con la información contenida en la cabecera de dicha trama.

AODV

Ad-hoc On-Demand Distance-Vector Routing es un protocolo reactivo unicast. Se construye sobre el protocolo DSDV analizado previamente. La idea es mejorar DSDV minimizando el número de paquetes broadcast requeridos para crear rutas, ya que al ser bajo demanda, los nodos que están en el camino no tienen que participar en el intercambio de tablas ni que mantener la ruta. A pesar de ser un protocolo reactivo, AODV tiene la peculiaridad de emitir mensajes alertando sobre su presencia de forma periódica mediante una técnica llamada *Link Layer Feedback*. Esa técnica permite que los nodos tengan conocimiento de sus vecinos más cercanos y mantengan sus tablas actualizadas reflejando los cambios en la topología cercana. Estas tablas se mantienen actualizadas a lo largo del tiempo, eliminando las entradas innecesarias.

AODV no mantiene rutas para cada nodo de la red. Estas rutas son descubiertas según se vayan necesitando bien sea que se activen o desactiven nodos en la red. AODV es capaz de proveer transmisión unicast, multicast y broadcast. La transmisión unicast consiste en enviar datos de un nodo a otro, la transmisión multicast consiste en enviar información de un nodo a un grupo de nodos y la transmisión broadcast consiste en enviar datos de un nodo a los demás nodos de la red. Los descubrimientos de rutas son siempre bajo demanda y siguen un ciclo de petición/respuesta de ruta. Las peticiones son enviadas usando un paquete especial denominado RREQ (*Route Request*). A su vez, las respuestas son enviadas en un paquete denominado RREP (*Route Reply*).

Por ejemplo, observando la fig. 2.4 supongamos que el nodo A quiere descubrir una ruta hacia el nodo D:

**Figura 2.4:** Protocolo AODV

Para iniciar un descubrimiento de ruta el nodo A transmite un RREQ enviando un único paquete en modo broadcast, el cual es recibido por todos los nodos que están en el rango de transmisión de A (en nuestro ejemplo, incluiría el nodo B). Cada *Route Request* incluye el origen y el destino del descubrimiento de ruta, además del identificador único (en nuestro ejemplo el 1) otorgado por el iniciador de la petición. Cada *Route Request* contiene, además, datos del origen para que los nodos intermedios puedan actualizar sus tablas con esta información. Por último también se añade un campo con información del número de saltos que da el paquete. Cuando otro nodo reciba esta petición (el nodo B en el ejemplo), si él fuera el destinatario del descubrimiento de ruta, devolvería al origen un RREP. Cuando el origen recibiera esta *Route Reply*, almacenaría en su caché este camino para los futuros envíos al mismo destino. En nuestro ejemplo, el nodo B que recibe el *Route Request*, comprueba que no le ha llegado con anterioridad otra petición con mismo origen y mismo identificador. Después de esto, verifica que no es el destinatario del RREQ y tampoco dispone de una ruta hacia el nodo D. A continuación, reenvía por broadcast la petición incrementando en una unidad el número de saltos. Como también viaja la información del nodo que originó la petición, podría añadir una ruta en su tabla para llegar hasta dicho nodo. El vecino por el que le ha llegado el RREQ, sería el nodo escogido para enrutar los paquetes hacia el nodo origen del RREQ. Esta última petición es recibida por el nodo C, que hace lo mismo, y por el nodo A, que descarta el paquete debido a que él fue quién lo inició. Por último, la petición llega al nodo D que es el destino. Este último, mandará el *Route Reply* correspondiente al nodo A con la ruta obtenida por *Route Request*. Para enviar este paquete al nodo A, mirará en su caché para obtener algún camino o iniciará otro *Route Discovery* si fuera necesario. No hay que olvidar que debido a la omnidireccionalidad de los envíos, el nodo B también obtendría una copia del último mensaje, descartándolo

por haberlo reenviado con anterioridad.

OLSR

“*Optimized Link State Routing*” es un protocolo proactivo basado en el estado de enlace. OLSR es una optimización directa del algoritmo de estados de enlace adaptado a los requisitos específicos de una *WLAN* con alta movilidad. La optimización consiste principalmente en la reducción del tamaño de las tablas de enlaces intercambiadas así como del número de retransmisiones necesarias durante los periodos de inundación. La clave del algoritmo reside en el uso de retransmisiones multipunto (MPR).

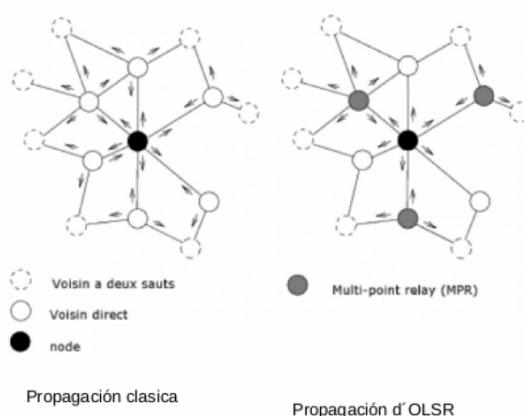


Figura 2.5: Protocolo OLSR

2.2.4. Protocolos multicast

MAODV

Es la extensión multicast de AODV conocida también como *Multicast AODV*. Lo que se pretende construir son árboles multicast bidireccionales compartidos que conecten múltiples fuentes y destinos para cada grupo multicast. Estos árboles se mantienen mientras hay miembros del grupo conectados por parte del árbol. Cada grupo multicast tiene un nodo líder, responsable de mantener el valor del número de secuencia. gracias a ese número de secuencia se consigue que el grupo multicast use siempre rutas actualizadas. El nodo líder es la raíz del árbol multicast. MAODV comparte muchas similitudes con el protocolo unicast AODV como son los paquetes de *Route Request* (RREQ) y *Route Reply* (RREP) así como la tabla de encaminamiento. Además se usan mensajes de *Multicast Activations* (MACT) y *Group Hello* (GRPH). El rango de disseminación de los RREQ lo indica el campo TTL de la cabecera. Los nodos líderes de grupo inundan la red periódicamente anunciando su dirección y su situación de líder de grupo así como el número de secuencia del grupo. Cuando un nodo quiere enviar mensajes a dicho grupo multicast para el cual no conoce el líder, primero intenta hacerse líder del grupo. Si no recibe respuesta él mismo se convierte en líder y comienza a emitir. Si ya

conocía la identidad del líder por haber recibido previamente un mensaje de anuncio, envía los mensajes de datos directamente al líder del grupo para que este los distribuya por el árbol multicast. Cuando un nodo desea unirse a un grupo como receptor, envía una petición inundando la red. Estas peticiones pueden ser contestadas por cualquier miembro del grupo multicast. Las respuestas son enviadas al origen de modo que los nodos por los que pasa se convierten en nuevos miembros del árbol multicast.

Diseño del escenario

En este capítulo, se presenta la introducción al desarrollo de la tesis en donde se detallan los estudios de simulación que se desean realizar. Se entiende por estudio de simulación los escenarios aplicados a una cierta topología. Este escenario particularmente definido se obtiene datos para una línea recta (autopista) con los perfiles de tráfico de cuatro vehículos en los que solo se toman en cuenta tres vehículos con dos tipos de movimiento (MRUA y MRU) y sus diferentes velocidades. En este caso se harán una prueba: una usando conexión *User Datagram Protocol* (UDP) que son las únicas que están disponibles para broadcast. La conexión *Transmission Control Protocol* (TCP) o Protocolo de Control de Transmisión no es necesario simular ya que no está permitida en broadcast según sus características. Son los dos protocolos de comunicación más usados por los servicios corrientes en la actualidad.

3.1. Justificación de escenarios a simular.

La manera en la que se han presentado los escenarios con tres nodos y una línea recta es debido a cuestiones de cinemática y los problemas de colisión de paquetes, tormenta y de inundación broadcast. En este trabajo es fácil partir con un conjunto de tres nodos móviles que pueden comunicarse entre sí y que comparten un único canal común, pero no detectan colisiones de paquetes debido a que los nodos en movimiento son comunicados directamente con otros nodos en un solo salto. En este caso, si se produce un escenario de multisalto que ocurre cuando los paquetes enviados por el nodo de origen se retransmiten por varios nodos intermedios antes de llegar al nodo destino, podría existir la colisión ya que muchas retransmisiones son repetitivas o porque los nodos están muy cerca unos de otros y a la falta del mecanismo (*Request to Send / Clear to Send*) RTS/CTS. La solución que citan algunos autores (11) para este problema es reducir la probabilidad de transmisión repetitiva y diferenciar el periodo de transmisión. Siguiendo estas instrucciones, han desarrollado varios escenarios los cuales son: probabilidad o *probabilistic*, basado en un contador *counter-based*, basado en distancia *distance-based*, basado en la localización *location-based*, esquemas para facilitar

la radio transmisión.

3.1.1. Cuestión cinemática

Al tener un esquema de tráfico como se muestra en la figura 3.4. El vehículo cero (**V0**) como **V1** debe tener un vehículo de testigo, para **V0** es **V2** mientras que el de **V1** es **V3**, en este caso el vehículo **V3** se toma como extra. Objetivo por el cual se va a determinar si **V2** pueda a rebasar a **V0** ó **V3** a **V1**, tomando en cuenta que la colisión puede ocurrir **V2** con **V1** y **V3** o **V0** y **V1**. Con esto cabe destacar que simular con tres vehículos es suficiente y con cuatro o cinco es extra. Para la parte cinemática se trabajó con las ecuaciones de Movimiento Rectilíneo Uniformemente Acelerado(MRUA) y Movimiento Rectilíneo Uniforme(MRU) el cual son fáciles de manipular en una línea recta teniendo resultado reales a diferencia si se tienen más variables, por ejemplo; las curvas cerradas, el trabajo se vuelve más complejo.

3.1.2. Características al problema broadcast

- El broadcast es espontáneo: Cualquier nodo móvil puede emitir una operación de emisión en cualquier momento. Por razones tales como la movilidad del nodo y la falta de sincronización.

- El broadcast es puro y fiable: El intento debe hacerse para distribuir un mensaje broadcast a tantos nodos como sea posible. (i) Un nodo puede perder un mensaje broadcast si está apagado, fuera del alcance de la red o por colisiones.(ii) Los reconocimientos puede causar graves problemas de contención y por lo tanto, otra tormenta. (iii) En muchas aplicaciones, por ejemplo, en el descubrimiento de una ruta transmisión 100 % fiable no es necesaria.

- Suponemos que un nodo puede detectar el duplicado de un mensaje broadcast. Esto es esencial para evitar el indudamiento de un mensaje. Una forma de como hacer esto es asociar a cada mensaje broadcast una ID de origen o un número de secuencia. Finalmente, comentamos que nos limitamos a la retransmisión del mismo mensaje. El fenómeno donde la transmisión de un paquete desencadena a otros el mismo paquete. Se va a demostrar que si las inundaciones se utilizan a ciegas, se enviaran muchos mensajes repetitivos y colisiones.

- Una aproximación directa hacia adelante para llevar a cabo el broadcast es por las inundaciones. Un nodo al recibir un mensaje broadcast, para un primer tiempo tiene la obligación de retransmitir un mensaje.

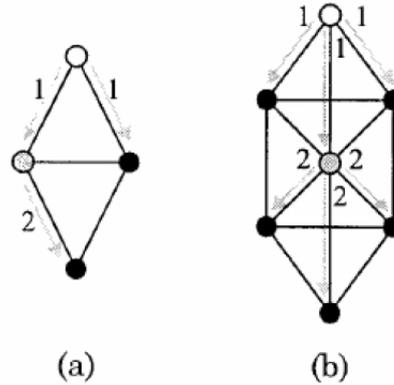


Figura 3.1: Conectividad broadcast entre nodos blancos grises y negros

En una red *Carrier Sense Multiple Access with Collision Avoidance* o Acceso Múltiple por Detección de Portadora y Prevención de Colisiones (CSMA/CA), inconvenientes de inundaciones incluyen:

- Retransmisiones Repetitivas: Se produce cuando un nodo decide retransmitir un mensaje a sus vecinos, sin embargo, todos los vecinos ya han recibido el mensaje. Así, la transmisión es redundante e inútil.
- Contención(detención): Se produce cuando los nodos vecinos reciben un mensaje broadcast y deciden volver a retransmitir el mensaje. Estos nodos deben enfrentarse entre sí por el medio broadcast es decir pueden competir seriamente con los demás.
- Colisión: Debido a la carencia de un mecanismo *back-off* de diálogo (RTS/CTS), y la ausencia de *Carrier Sense Multiple Access with Collision Detection* o Acceso Múltiple por Detección de Portadora con Detección de Colisiones (CSMA/CD), las colisiones son más propensas a ocurrir y causar más daño.

En la figura 3.1 se utiliza dos casos para demostrar la cantidad de repeticiones que se podrían generar. En la figura (a) el nodo blanco inicia un mensaje broadcast el cual es recibido por los nodos gris y negro, uno de estos dos decide transmitir el mensaje si el último nodo (negro) no lo ha recibido. Se intenta reducir la redundancia con dos transmisiones en lugar de cuatro transmisiones. En la figura (b), solo dos transmisiones son suficientes para completar el broadcast en oposición a 7 transmisiones causadas por las inundaciones.

La razón principal de tal redundancia es que las señales de radio desde diferentes antenas son muy probables de que se solapen entre sí. Suponiendo que el área que puede

ser cubierta por una antena forma un círculo, mostrado en la figura 3.2 el problema de solapamiento de la señal correspondiente al escenario en la figura 3.1 (b). Los niveles de gris de la figura indican los niveles de superposición de señales. Como puede verse, muchas zonas están cubiertas por el mismo paquete de difusión más de una vez. En el peor de los casos, una zona puede ser cubierto por el paquete de siete veces.

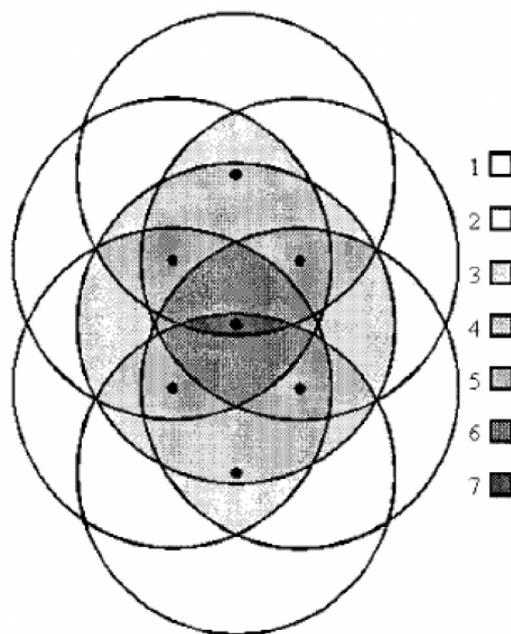


Figura 3.2: Superposición de señales

3.1.3. Análisis de la zona extra que puede beneficiarse de un mensaje broadcast.

Primero se considera un escenario en la figura 3.3. **A** envía un paquete broadcast y **B** decide retransmitir el paquete. Sea S_A y S_B las áreas cubiertas por las transmisiones de **A** y **B** respectivamente. El área beneficiada por la retransmisión de **B** es la región sombreada, denominada como $S_B - A$, sea r el radio de S_A y S_B y d la distancia entre **A** y **B**. Se puede deducir que

$$|S_B - A| = |S_B| - |S_A \cap B| = \pi * r^2 - Intercesion(d) \quad (3.1)$$

Donde

$$y^2 + x^2 = r^2 \Rightarrow y = \sqrt{r^2 - x^2} \quad (3.2)$$

$$Intercesion(d) = 4 \int_{d/2}^r \sqrt{r^2 - x^2} dx \quad (3.3)$$

en intervalos muy cortos unos con respecto de sus nodos vecinos. A decir también que existe la probabilidad de obstruirse la señal y el paquete no llegue a su nodo destino si hay un obstáculo físico de por medio motivo por el cual se toma en cuenta simular en línea recta.

3.1.4. Escenario 1: Comunicación Broadcast Con Movimiento Rectilíneo Uniformemente Acelerado(MRUA)

Este escenario debe cumplir los requisitos de comunicación broadcast y estándar apropiado disponibles en la actualidad. El cual se simula una situación en la que todos los nodos están dentro del rango de transmisión de 300 m, es decir una comunicación inalámbrica entre tres vehículos (nodos) y uno más extra con Movimiento Rectilíneo Uniformemente Acelerado(MRUA). A este conjunto se le implementaron tarjetas wifi con su capa física y su capa mac configuradas con estándares de 802.11b, en modo ad-hoc respectivamente. El envío de paquetes se llevó a cabo con una intensidad de señal de -80 dBm (unidad de medida de potencia expresada en decibelios), y un tamaño de 1000 bytes en tiempos de transmisión cada 0.3 s y 0.5 s. Dada la comunicación esto genera un tráfico de paquetes, para lo cual, los vehículos se están moviendo en direcciones determinadas y en cada envío de paquetes cada vehículo estará registrando la posición, velocidad, número de paquetes transmitidos, número de bytes transmitidos etc. Con la tecnología fijada, este escenario permite tener una comparación de los efectos de esquemas de movilidad en las comunicaciones, viendo cómo influyen las siguientes ecuaciones de (MRUA) (9) con las variables de distancia, tiempo y velocidad.

Término	Símbolo
Velocidad Inicial	v_i
Velocidad Final	v_f
Incremento de Posición	$\Delta\vec{x}$
Aceleración	a
Tiempo	t

Tabla 3.1: Símbolos del MRUA.

$$\vec{a} = \frac{\vec{v}_f - \vec{v}_i}{t_f - t_i} = cte \quad (3.8)$$

$$\vec{v}_f = \vec{v}_i + \vec{a}t \quad (3.9)$$

$$\Delta\vec{x} = \frac{\vec{v}_f + \vec{v}_i}{2}t \quad (3.10)$$

$$\Delta\vec{x} = \vec{v}_i t + \frac{1}{2}\vec{a}t^2 \quad (3.11)$$

$$\vec{v}_f^2 = \vec{v}_i + 2\vec{a}\Delta\vec{x} \quad (3.12)$$

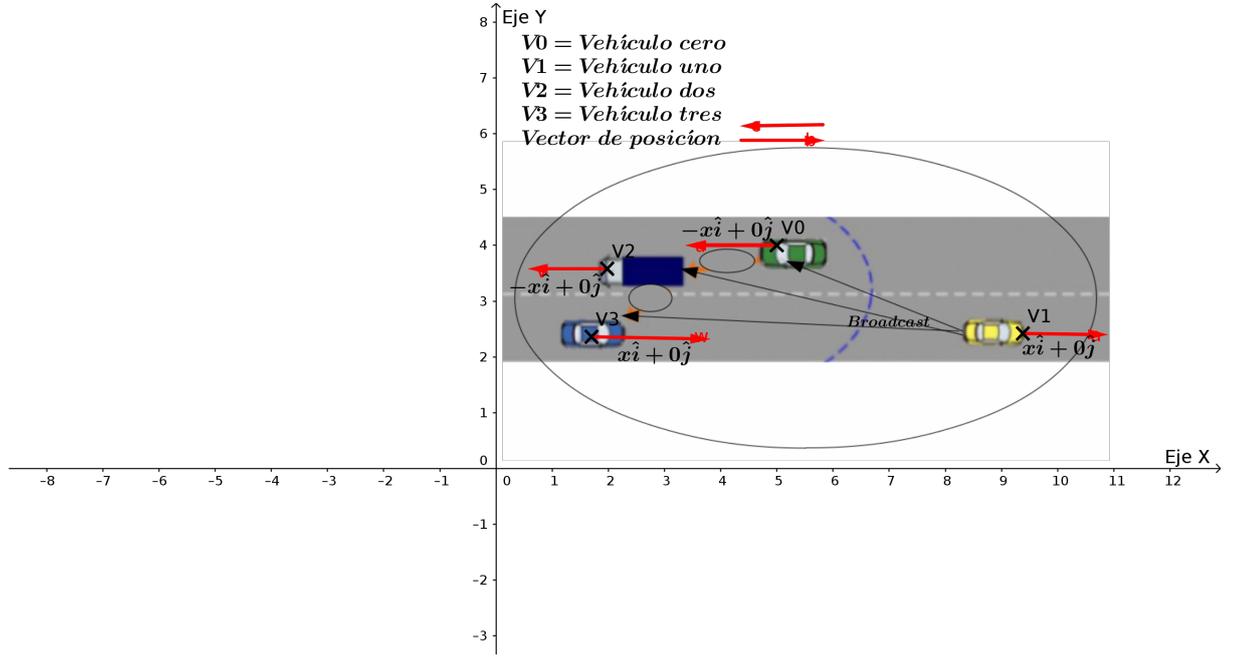


Figura 3.4: Proyección en un camino vial

El escenario es el siguiente: Se establece una comunicación broadcast desde un vehículo origen cuando sus destinos se encuentra a su alcance, mientras que si la distancia de separación es mayor que dicho radio, la comunicación se cortará. Se hace énfasis a que cualquier vehículo que esté dentro de la comunicación broadcast escuche a los demás vehículo en un solo salto y después de un corto intervalo de tiempo emita el mensaje (posición y velocidad), siendo esto consecuentemente a evitar la inundación de mensajes.

En la figura 3.5. El vehículo V2 el cual presentará una velocidad máxima de 47.2 m/s (170 km/h) y una mínima de 33 m/s (120 km/h) en periodos de transmisión de mensajes de $t=0.3$ s, 0.5, s 0.6 s y 1 s, hace un broadcast con respecto a los demás vehículos y así sucesivamente los vehículos V0, V1 y V3 en intervalos de tiempos cortos y sucesivos, tomando en cuenta que los vehículos que estén dentro del área de cobertura podrán escuchar a V2. Los resultados serán presentados en forma escalar y vectorial

3. DISEÑO DEL ESCENARIO

siendo así el vector de dirección de cada vehículo establecido de la siguiente manera: para V_0 moviéndose en la dirección $-x\hat{i}+0\hat{j}$, $V_1=x\hat{i}+0\hat{j}$, $V_2=-x\hat{i}+0\hat{j}$ y $V_3=x\hat{i}+0\hat{j}$ a partir de estos vectores se tomarán medidas de la posición y velocidad en su primer transmisión broadcast, como también la aceleración, tiempo de encuentro con el vehículo, margen de separación en su primer transmisión, tanto del nodo emisor como el receptor.

En la figura 3.6 muestra un ejemplo de como se ve la conexión en el simulador ns-3.

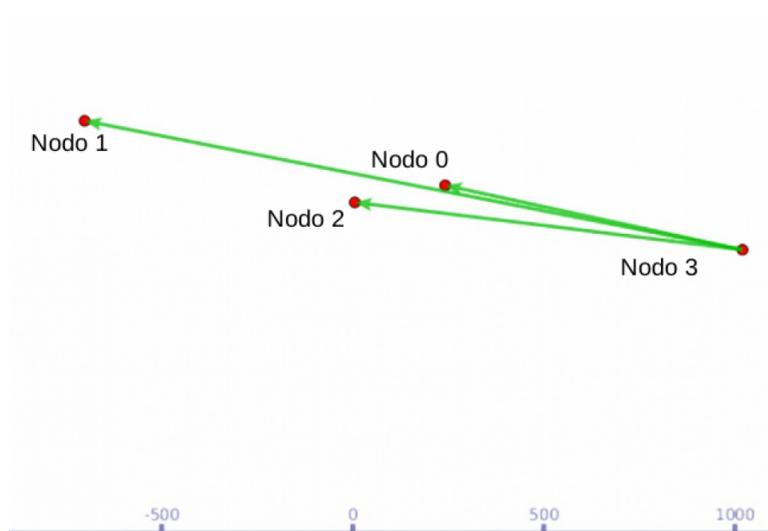


Figura 3.5: Broadcast en línea recta con MRUA

3.1.5. Escenario 2: Comunicación Broadcast Con Movimiento Rectilíneo Uniforme(MRU)

De la misma forma en que se llevó a cabo la simulación del escenario anterior se tomarán en cuenta las mismas características a diferencia que en este escenario solo cambia las ecuaciones de movimiento. En este caso se utiliza la ecuación $v=d/t$.

A partir de los resultados de la simulación con Movimiento Rectilíneo Uniforme Acelerado (MRUA) y Movimiento Rectilíneo Uniforme (MRU) se permite obtener una comparación que de la información del comportamiento de los vehículos de tal forma que se pueda llegar a la conclusión de cómo varían los parámetros de la posición y velocidad en su primer transmisión broadcast, como también la aceleración, tiempo de encuentro con el vehículo, margen de separación en su primer transmisión.

Existen muchos estudios que se proponen en comparar protocolos de encaminamiento cuando el número de conexiones entre los vehículos se multiplica es decir en multisalto. Recordar que este trabajo se centra en la influencia de la topología de red en conexiones de un solo salto.

Plataforma de simulación

Siguiendo los requisitos de la plataforma de simulación se ha elegido un conjunto de programas de libre distribución y de código abierto. La base de la plataforma es el simulador ns-3 (*Network Simulator*). Esta elección se ha hecho considerando la amplia aceptación que recibe entre los grupos de trabajo para desarrollo de protocolos en redes MANETs. El hecho de que un programa sea muy utilizado es una ventaja porque dispondrá de numerosas implementaciones de protocolos, documentación y foros de usuarios. ns-3 genera trazas de una simulación muy completas que nos permiten extraer la totalidad de las medidas de los indicadores elegidos para la comparativa. Sin embargo, es interesante visualizar de manera más eficaz los flujos de datos y los movimientos de vehículos, para poder detectar por ejemplo las pérdidas de paquetes.

En este capítulo describimos los programas utilizados, la instalación y el funcionamiento de la plataforma de simulación.

4.1. Componentes Software de la plataforma de simulación

Como lo se ha descrito antes, la plataforma de simulación esta en software de código abierto. A continuación, se describe en detalle sus acontecimientos (?):

La plataforma de simulación de redes ns, se basa en eventos discretos, dando uso en ambientes educativos y de investigación. Permite simular tanto protocolos unicast como multicast y se utiliza intensamente en la investigación de redes móviles ad-hoc. Implementa una amplia gama de protocolos tanto de redes cableadas como de redes inalámbricas.

La primera versión del ns, conocidos como ns-1, fue desarrollado en *Lawrence Berkeley National Laboratory (LBNL)* en el marco de tiempo 1995-1997 por *Steve McCanne, Sally Floyd, Kevin Fall*, y otros colaboradores. Esto se conoce como el LBNL simulador

4. PLATAFORMA DE SIMULACIÓN

de red, y deriva de un simulador anteriormente conocida como real por *S. Keshav*. El núcleo del simulador fue escrito en C++ y en Tcl secuencias de comandos basada en simulación de escenarios.

ns-2 fue desarrollado en C++ y provee una interfaz de simulación a través de OTcl, una variante orientada a objetos de Tcl. El usuario describe una topología de red por medio de scripts OTcl, y luego el programa principal de ns-2 simula dicha topología utilizando los parámetros definidos. ns-2 está diseñado para sistemas operativos Linux, FreeBSD, Solaris, Mac OS X y puede ejecutarse bajo Windows utilizando Cygwin. La última versión, 2.34, se presentó al público el 17 de junio de 2009.

La variante ns-3 surge en el año 2005, a partir del impulso de Tom Henderson, según la lista de correo del grupo de realizadores de ns, se decide realizar una nueva versión desde cero, utilizando el lenguaje de programación C++. La base de desarrollo fue el paquete yans (*Yet Another Network Simulator*). El desarrollo de ns-3, se proyectó para un periodo de tiempo de cuatro años. Principalmente fue desarrollado por investigadores de las instituciones: Universidad de Washington, Instituto Tecnológico de Georgia y el grupo de investigación Planète en INRIA. La primera liberación de ns-3.1 fue hecha en junio de 2008. En el año 2011 ns-3 llegó a la versión 3.11. La infraestructura de ns-3 permite el desarrollo de modelos de simulación de alto desempeño, lo que habilita el uso de la herramienta como emulador. ns-3 soporta simulación de redes IP, no IP; así como redes inalámbricas tales como Wi-Fi, WiMAX, o LTE, además de un diferentes protocolos de ruteo entre los que se destacan OLSR y AODV.

Funcionamiento

ns-3 simula las diferentes capas del modelo OSI (Open Systems Interconnection). Con el modelo OSI se describen y uniforman las funciones de comunicación de sistemas de cómputo o telecomunicaciones sin importar su estructura interna y tecnología subyacente.

ns-3 registra cada paquete de datos que atraviesa la red junto a sus características principales, por ejemplo, el instante de recepción, el número de secuencia, el tipo de paquete, etc. Estos parámetros forman la traza como resultado del procedimiento de ns-3.

Secuencia de ejecución

A la hora de abordar la simulación de un protocolo en ns-3, es necesario seguir los siguientes pasos:

- Implementación del protocolo a analizar mediante la incorporación de código C++ del núcleo de ns-3. Este paso no es necesario si se desea usar protocolos ya propuesto por ns-3. ns-3 cuenta con muchos protocolos ya implementados en su versión descargable (AODV, TORA, DSDV, DSR).

- Ejecución de la simulación. Se lanza el simulador proporcionándole el fichero de descripción de simulación previamente definido.
- Existen múltiples formas de analizar los resultados de la simulación. En primer lugar, para visualizar la ejecución se puede usar el animador gráfico de ns-3 con el comando `./waf -run ejemplo -vis`. Es posible extraer medidas cuantitativas a partir de los ficheros de traza que aporta la simulación. Se realiza un posprocesado que permite extraer medidas de indicadores de rendimiento de red, mediante un programa de tratamiento.

4.2. Procesos De Simulación

El esquema de la simulación mostrado en la figura 4.1 está programado en el que consta de cuatro nodos, con un enlace de comunicación, el cual se crea con un objeto wifi del módulo ayudante o *helper*. Del objeto wifi se le adapta el estándar de comunicación 802.11b, como también se le añade una capa física, una capa de mac address y los nodos correspondientes para ser instalados en un dispositivo contenedor de redes. A partir del protocolo UDP, cada nodo hace una comunicación cliente servidor con el objeto socket mismo que llama al método que establece y permite la difusión (broadcast) de paquetes. Finalmente con el módulo de movimiento de nodos se permite obtener la posición y velocidad de cada nodo.

Tabla 4.1: Módulos de ejecución.

1.-Núcleo
2.-Red
3.-Movilidad
4.-Ayudante
5.-Wifi
6.-Udp

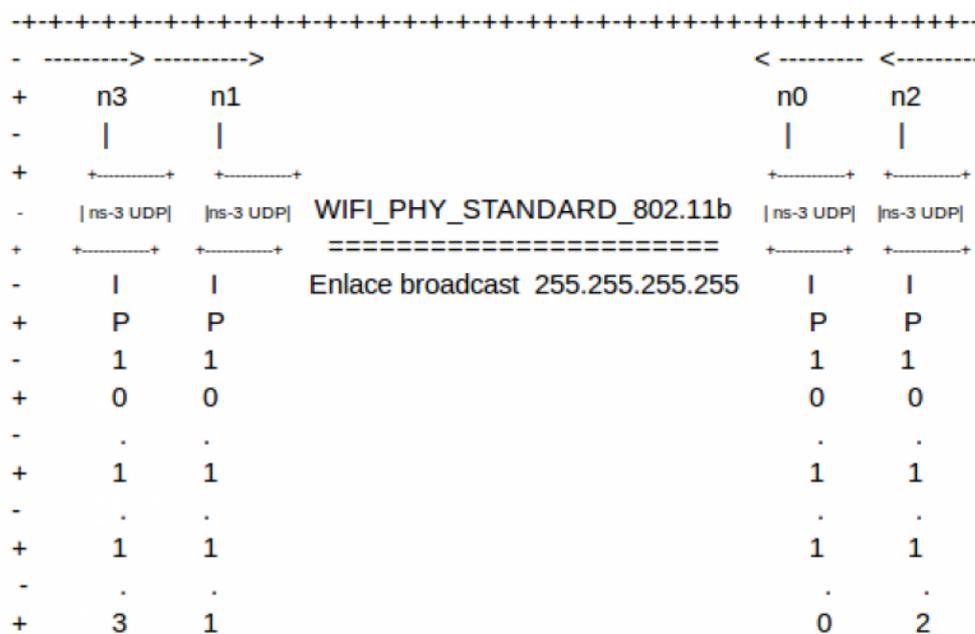


Figura 4.1: Ns3 Módulos

En esta sección se dará a conocer alguna lista de algunos de los módulos (6) que se llevaron a cabo para la ejecución de este trabajo.

Los módulos a utilizarse en está simulación son:

- **módulo núcleo:** Los nodos virtuales de espacios de nombres de red de núcleo puede tener interfaces TAP (redes virtuales para crear un puente virtual) Este módulo contiene una clase simulador utilizada para crear, programar y cancelar eventos, en tiempo real. Además una clase-puntero inteligente *NS3::Ptr* diseñado para trabajar en conjunto con *NS3::Object* En este tipo de configuración, los es-

pacios de nombres del núcleo se utilizan para proporcionar paquetes de datos a ns-3 dispositivos y canales. Esto permite, por ejemplo, que los modelos inalámbricos desarrollados para ns-3 puedan ser utilizados en un contexto de emulación. Los usuarios simulan redes con ns-3 escribiendo programas en C++ o Python que importan la biblioteca ns-3. Los modelos de simulación son objetos instanciados en estos programas.

■ **módulo red:** Este módulo hereda las siguientes clases:

1. Socket; Crea la comunicación entre cliente y servidor para el envío de paquetes.
2. Paquetes; Cada paquete de red contiene una memoria intermedia de bytes, un conjunto de etiquetas de bytes, un conjunto de etiquetas de paquetes, y los metadatos.
3. Canal de transmisión; Establece un canal de comunicación por puertos determinados.
4. Dirección de comunicación; Cada nodo debe contar con una dirección IP de comunicación.
5. Velocidad de datos; La velocidad de datos enviados y recibidos varía dependiendo de la cobertura.
6. Dispositivos de red; Instala una capa física, capa de enlace de datos y nodos a la red.

■ **módulo movilidad:**

El soporte de movilidad incluye:

1. un conjunto de métodos de movilidad que se utilizan para realizar el seguimiento y mantener la posición cartesiana actual y la velocidad de un objeto.
2. un seguimiento que se utiliza para registrar a los oyentes y a los cambios de rumbo de un modelo de movilidad.
3. una serie de clases de ayuda que se utilizan para colocar nodos y configurarlos de distintas maneras a según lo que se pretenda emplear.

- **Ayudante:**

1. El módulo ayudante hace un uso intensivo de los contenedores de objetos .
2. El módulo ayudante no es genérica es decir que no se esfuerza para maximizar la reutilización de código.
3. La API ayudante normalmente trabaja con objetos. Para algunos programas, los usuarios no tendrán que preocuparse por ningún objeto bajo el nivel de crear o manipular los punteros Ptr y que pueden conformarse con contenedores de objetos y ayudantes asignados.

- **Módulo wifi:**

Los nodos pueden contener una colección de objetos en dispositivos de red, al igual que un equipo real contiene las tarjetas de interfaz separados para Ethernet, WiFi, Bluetooth, etc. El módulo wifi describe la *WifiNetDevice* y modelos relacionados. Mediante la adición de objetos *WifiNetDevice* uno puede crear modelos de infraestructura basada y redes ad hoc.

- **udp echo cliente:**

Este módulo establece un protocolo del nivel de transporte basado en el intercambio de datagramas a través de una interfaz entre la capa de red y la capa de aplicación.

Este código de simulación se conforma de varios método y son los siguientes:

- **Vector GetPosition(Ipv4Address adr):** devuelve una dirección de la posición que ocupa el nodo, el cual sirve para ver la identidad de los nodos, en este caso de los vehículos.
- **static void SetPosition (Ptr< Node > node, Vector3D position):** se asigna de la clase Nodo un objeto nodo y un objeto posición de la clase Vector, es decir asignamos la posición a cada vehículo

- **static Vector3D GetPosition (Ptr< Node > node):** Devuelve el objeto node del puntero inteligente, con el fin de que regrese el vector de la posición de cada vehículo
- **static void AdvancePosition 3(Ptr< Node > node):** se define las posiciones iniciales de cada nodo utilizando los métodos *GetPosition* y *SetPosition* e imprime conforme se van moviendo.
- **static void SetVelocity (Ptr < Node > node, Vector3D vel):**se asigna de la clase Node un objeto node y un objeto velocidad de la clase Vector.
- **static Vector3D GetVelocity (Ptr < Node > node):** devuelve el objeto node del puntero inteligente. En este caso seria con el fin de que regrese el vector de velocidad de cada vehículo
- **AdvancePosition(Ptr < Node > node):**se define las velocidades iniciales en forma vectorial de cada nodo utilizando los métodos *GetVelocity* y *SetVelocity* e imprime conforme se van moviendo.
- **void ReceivePackete (Ptr < Socket > socket):** sirve para crear la comunicación cliente servidor entre los vehículos por medio de un socket y que adentro de este método se crea un paquete.
- **static void GenerateTraffic (Ptr < Socket > socket, uint32_t pktSize, uint32_t pktCount, Time pktInterval):** manda los datos que se generan a través de la comunicación entre los vehículos, como el tamaño del paquete, el contador de paquetes y el tiempo de paquetes transferidos .
- **Metodo main():**en este método principal se define la parte más importante de la simulación, como lo que es la capa de red los mulos de movilidad de los nodos la creación de sockets en broadcasts mandando llamar métodos mencionados anteriormente y finalmente correr el programa.

4.3. Instalación de la plataforma

ns-3 es bastante complejo y tienen múltiples dependencias. Algunas de las herramientas que proporcionan apoyo a ns-3 (por ejemplo, “GNU cadena de herramientas”, Mercurial es un editor para el ajuste), son necesarias para asegurar que

las diversas bibliotecas están en el sistema. El ns-3 (4) ofrece un wiki un documento y una página html con varios consejos sobre el sistema, además se desarrolla principalmente en las plataformas GNU / Linux, y los requisitos mínimos para ejecutar simulaciones básicas son un gcc y Python.

4.3.1. Sistema operativo y soporte del compilador

ns-3 es compatible con las siguientes plataformas principales:

- Linux x86 y x86_64: versiones de gcc 4.2 a 4.8
- FreeBSD x86 y x86_64: tañido versión 3.3, versión 4.2 gcc
- Mac OS X Intel: tañido500.2.79, basado en LLVM 3.3svn (OS X Mavericks y Xcode 5.0.1), y gcc4.2 (disponible con Xcode versión 4 o anterior) Algunos aspectos de ns-3 dependen de soporte en Unix (Linux o específicamente), tales como las características de emulación o TapBridge, y esos componentes no están habilitadas en las versiones de Windows.

Ubuntu fue la plataforma de simulación en la cual se trabajó por condiciones favorables.

Ubuntu

La siguiente lista de paquetes debe ser precisa para Ubuntu 14.10 versión.

- **Requisitos mínimos para C++ (liberación):** Este es el conjunto mínimo de paquetes necesarios para ejecutar ns-3 desde la terminal ubuntu.
- Instalar el compilador gcc y g++.
\$ sudo apt-get install gcc g++ python
- Instalar Python.
\$sudo apt-get install gcc g++ python python-dev
- qt marco de trabajo.
\$ sudo apt-get install qt4-dev-tools
- Mercurial repositorio.
\$sudo apt-get install mercurial

- Control revisión Python.
\$ sudo apt-get install

- Compilador paquetes.
\$ sudo apt-get install cmake libc6-dev libc6-dev-i386 g++-multilib

- Herramientas de depuración.
\$ sudo apt-get install gdb valgrind

- Biblioteca para la función matemática.
\$ sudo apt-get install gsl-bin libgsl0-dev libgsl0ldbl

- Generador de análisis en el proyecto GNU.
\$ sudo apt-get install flex bison libfl-dev

- Herramientas de captura de paquetes de red, generalmente instalan como por defecto.
\$ sudo apt-get install tcpdump

- Para base de datos.
\$ sudo apt-get install sqlite sqlite3 libsqlite3-dev

- Soporte de la biblioteca XML.
\$ sudo apt-get install libxml2 libxml2-dev

- Librerías GTK que es utilizar para la creación de la interfaz gráfica de usuario.
\$ sudo apt-get install libgtk2.0-0 libgtk2.0-dev

- Contenedor de Linux.

\$ sudo apt-get install vtun lxc

- El uso para la modificación del código fuente.

\$ sudo apt-get install uncrustify

4. PLATAFORMA DE SIMULACIÓN

- Edición de imágenes y texlive para la documentación.

```
$ sudo apt-get install doxygen graphviz imagemagick
```

```
$ sudo apt-get install texlive texlive-extra-utils texlive-latex-extra texlive-font-  
utils dvipng
```

```
$ sudo apt-get install python-sphinx dia
```

- Bibliotecas de Python para el gráfico.

```
$ sudo apt-get install python-pygraphviz python-kiwi python-pygoocanvas libgoocanvas-  
dev
```

- C++ bibliotecas.

```
$ sudo apt-get install libboost-signals-dev libboost-filesystem-dev
```

- Paso de mensajes para la computación de alto rendimiento.

```
$ sudo apt-get install openmpi-bin openmpi-common openmpi-doc libopenmpi-  
dev
```

- **Instalar NS3**

1. Abrir una terminal.

```
Teclear CTRL-ALT-T
```

2. Descargar la versión completa de ns-3.

```
$ wget http://www.nsnam.org/releases/ns-allinone-3.13.tar.bz2
```

3. Introducir.

```
$ cd ns-allinone
```

```
$ ./download.py
```

4. Cambiarse de directorio.

```
$cd ns-allinone.
```

5. Para instalar Python.

```
./build.py
```

6. Configurando NS3.
\$ cd ns-3.13
\$./waf --shell
7. Ejecutar Waf shell
\$./waf
8. Cambiarse de directorio
\$ cd ns-3.13
9. Ejecutar la carpeta.
\$./waf shell
10. Ejecutar el simulador en phytón
\$ python ./examples/*.py

Finalmente se dara un ejemplo de crear un archivo para después ejecutarlo

11. Como crear un primer ejemplo y ejecutarlo.
Crear un archivo con el nombre primero.cc y guardarlo en la carpeta scracht.
12. Para la ejecución se teclea lo siguiente:
\$.waf/ --run scracht/primero
13. Para visualizar el gráfico en phyton.
\$.waf/ --run primero --vis

Resultados

Los resultados de las simulaciones se obtuvieron bajo redes ad-hoc con modo broadcast y conexiones UDP en comunicación de un solo salto para evitar la colisión e inundación de mensajes broadcast. Estas características se implementa en una trayectoria recta (autopista) entre tres vehículos. Siguiendo los escenarios con MRUA y MRU ya descritos en el capítulo 3, para cada escenario se obtienen resultados en base a la velocidad y posición, determinando el rango de distancia y tiempo que hay entre cada vehículo para llevar a cabo un a rebase antes de coincidir en el un punto de intersección. Los datos tomados es en base a la cobertura de transmisión dependiente del estándar 802.11b, el cual son considerados los vehículos que pueden ocasionar la colisión, en este caso el vehículo V2 y el V1 en los dos escenarios. Las velocidad establecidas en estos escenarios es en base a los límites de velocidad que establecen las autopista en la realidad.

En la figura 5.1 (5) se muestra las direcciones vectoriales en la cual los vehículos se están moviendo en un tramo de 300 m de línea recta, por lo que sus componente en el eje Y son nulas. Definiendo las posiciones $\vec{r}_0 = x_0 \hat{i}$, $\vec{r}_1 = x_1 \hat{i}$, $\vec{r}_2 = x_2 \hat{i}$, $\vec{r}_3 = x_3 \hat{i}$ y velocidades $V_{x0} \hat{i}$, $V_{x1} \hat{i}$, $V_{x2} \hat{i}$, $V_{x3} \hat{i}$, para los vehículos V0, V1, V2, V3 respectivamente.

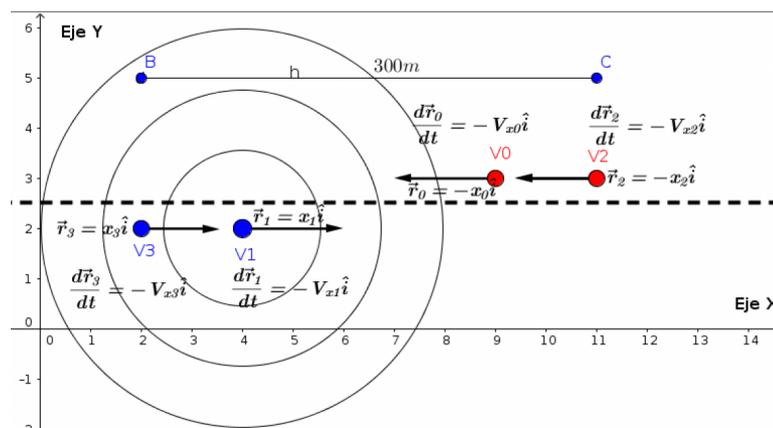


Figura 5.1: Trayectorias de los vehículos

5.1. Comunicación entre cuatro vehículos con (MRUA).

Se simula con la comunicación broadcast de tres vehículos y un vehículo extra (no afectan sus datos a los otros vehículos) con movimiento rectilíneo uniformemente acelerado, cada uno presentan cambios de velocidad, incrementada en 1, con una área circular de radio de 150 m y transmisiones de paquetes broadcast de intervalos de $t=0.5$ s inicializando en $t=0$ s, es decir, en $t=0$ el vehículo V0 manda su primer transmisión, en $t=1$, $t=2$ y $t=3$ para V1 V2 y V3 respectivamente en cual se comparan con envíos de paquetes broadcast de $t=0.3$ s misma forma inicializando en $t=0$ s. Para las velocidades de los vehículos tomaremos en cuenta una velocidad máxima de 47.2 m/s equivalente a 170 km/h y mínima de 33 m/s equivalente a 120 km/h. El momento en que el vehículo V1 manda sus transmisión de paquetes en tiempos de $t=0.5$ s y $t=0.3$ s, el vehículo V2 las manda en tiempos de $t=1$ s y $t=0.6$ seg.

5.1.1. Resultados con velocidad mínima

Se determinan los siguientes parámetros mostrados en la Tablas 5.1, 5.2, 5.3, 5.4 y sus respectivas gráficas.

Resultados del vehículo V1 con respecto al vehículo V2				
MRUA [vehículos V0,V1,V2,V3 y su velocidades(22,27,33,8 m/s) respectivamente]				
Tiempo en que hace su transmisión de paquetes t= 0.5 s				
Velocidad en su primer transmisión de paquetes	Posición en su primer transmisión de paquetes	Aceleración	Tiempo de encuentro con el vehículo V2	Margen de separación en su primer transmisión para rebasar
$\vec{V}_x=33$ m/s	$\vec{X}_1=269$ m	$\vec{a}_x= 10$ m/s ²	t = 2.67 s	d = 246 m

Tabla 5.1: Características escalares del vehículo V1 respecto al vehículo V2 en donde V1 manda una transmisión de paquetes en un tiempo de t=0.5 s a los vehículo V0, V2 y V3.

Resultados del vehículo V2 con respecto al vehículo V1				
MRUA [vehículos V0,V1,V2,V3 y su velocidades(22,27,33,8 m/s) respectivamente]				
Tiempo en que hace su transmisión de paquetes t = 1s				
Velocidad en su primer transmisión de paquetes	Posición en su primer transmisión de paquetes	Aceleración	Tiempo de encuentro con el vehículo V2	Margen de separación en su primer transmisión para rebasar
$\vec{V}_x=44$ m/s	$\vec{X}_1=-48$ m	$\vec{a}_x= 10$ m/s ²	t = 2.67 s	d = 198 m

Tabla 5.2: Características escalares del vehículo V2 respecto al vehículo V1 en donde V2 manda una transmisión de paquetes en un tiempo de t=1 s a los vehículos V0, V1 y V3.

5. RESULTADOS

Resultados del vehículo V1 con respecto al vehículo V2				
MRUA [vehículos V0,V1,V2,V3 y su velocidades(22,27,33,8 m/s) respectivamente]				
Tiempo en que hace su transmisión de paquetes t= 0.3 s				
Velocidad en su primer transmisión de paquetes	Posición en su primer transmisión de paquetes	Aceleración	Tiempo de encuentro con el vehículo V2	Margen de separación en su primer transmisión para rebasar
$\vec{V}_x=31$ m/s	$\vec{X}_1=277.3$ m	$\vec{a}_x= 10$ m/s ²	t = 2.67 s	d = 263.8 m

Tabla 5.3: Características escalares del vehículo V1 respecto al vehículo V2 en donde V1 manda una transmisión de paquetes en un tiempo de t=0.3 s a los vehículos V0, V2, y V3.

Resultados del vehículo V2 con respecto al vehículo V1				
MRUA [vehículos V0,V1,V2,V3 y su velocidades(22,27,33,8 m/s) respectivamente]				
Tiempo en que hace su transmisión de paquetes t = 0.6s				
Velocidad en su primer transmisión de paquetes	Posición en su primer transmisión de paquetes	Aceleración	Tiempo de encuentro con el vehículo V2	Margen de separación en su primer transmisión para rebasar
$\vec{V}_x=40$ m/s	$\vec{X}_1=-27.9$ m	$\vec{a}_x= 10$ m/s ²	t = 2.67 s	d = 218.8 m

Tabla 5.4: Características escalares del vehículo V2 respecto al vehículo V1 en donde V2 manda una transmisión de paquetes en un tiempo de t=1 s. a los vehículos V0, V1 y V3.

5.1 Comunicación entre cuatro vehículos con (MRUA).

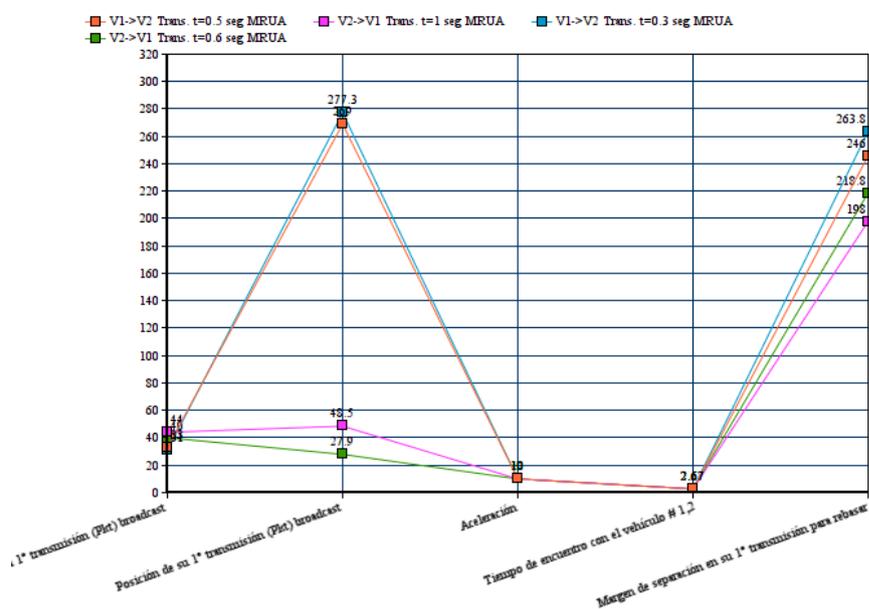


Figura 5.2: Comparación de datos de la Tabla 5.1, 5.2, 5.3 y 5.4

A continuación se muestran las características de recepción y transmisión de paquetes broadcast en donde anteriormente se tomaron en cuenta las variables en el momento que los vehículos hacen su transmisión de paquetes. Observando el número de paquete, y de bytes enviados (Tx) y recibidos (Rx) en un tiempo de $t=1s$.

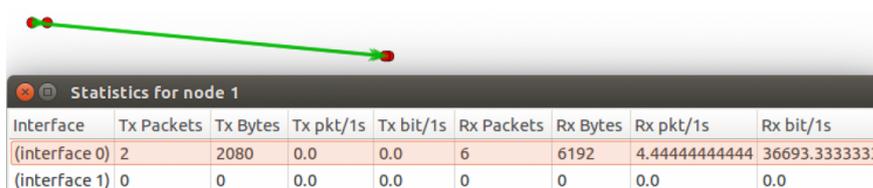


Figura 5.3: Paquetes Tx y Rx

Se observa que solo se emitieron los paquetes que estaban dentro del rango de cobertura en donde por cada nodo el total de paquetes recibidos fueron de 6 y 0 perdidos.

Se justifica que entre más corto se el periodo de transmisión de paquetes, mejor margen de distancia tienen para hacer el rebase, es decir, para la Tablas 5.1 donde el vehículo V1 transmite con un tiempo $t=0.5s$ y su distancia para el rebase con respecto al vehículo V2 es de 246 m mientras que para la Tabla 5.3 con un periodo de transmisión de $t=0.3s$ y su distancia de rebase de 263.8 m. De la misma forma para las Tablas 5.2 y 5.4.

5. RESULTADOS

5.1.2. Resultados con velocidad máxima

De igual forma se considera los vehículos V1 y V2. Analizando los resultados con velocidad mínima se determina que es más conveniente simular en periodos más cortos de transmisión de paquetes. Se determinan los siguientes parámetros mostrados en la Tablas 5.5, 5.6, 5.7, 5.8 y sus respectivas gráficas.

Resultados del vehículo V1 con respecto al vehículo V2				
MRUA [vehículos V0,V1,V2,V3 y su velocidades(22,27,47.2,8 m/s) respectivamente]				
Tiempo en que hace su transmisión de paquetes t= 0.3 s				
Velocidad en su primer transmisión de paquetes	Posición en su primer transmisión de paquetes	Aceleración	Tiempo de encuentro con el vehículo V2	Margen de separación en su primer transmisión para rebasar
$\vec{V}_x=31 \text{ m/s}$	$\vec{X}_1=277.3 \text{ m}$	$\vec{a}_x= 10 \text{ m/s}^2$	$t = 2.42 \text{ s}$	$d = 259.54 \text{ m}$

Tabla 5.5: Características escalares del vehículo V1 respecto al vehículo V2 en donde V1 manda una transmisión de paquetes en un tiempo de t=0.3 s a los vehículos V0, V2 y V3

Resultados del vehículo V2 con respecto al vehículo V1				
MRUA [vehículos V0,V1,V2,V3 y su velocidades(22,27,47.2,8 m/s) respectivamente]				
Tiempo en que hace su transmisión de paquetes t= 0.6 s				
Velocidad en su primer transmisión de paquetes	Posición en su primer transmisión de paquetes	Aceleración	Tiempo de encuentro con el vehículo V2	Margen de separación en su primer transmisión para rebasar
$\vec{V}_x=54.2$ m/s	$\vec{X}_1=-36.42$ m	$\vec{a}_x= 10$ m/s ²	t = 2.42 s	d = 228.28 m

Tabla 5.6: Características escalares del vehículo V2 respecto al vehículo V1 en donde V2 manda una transmisión de paquetes en un tiempo de t= 0.6 a los vehículos V0, V1 y V3.

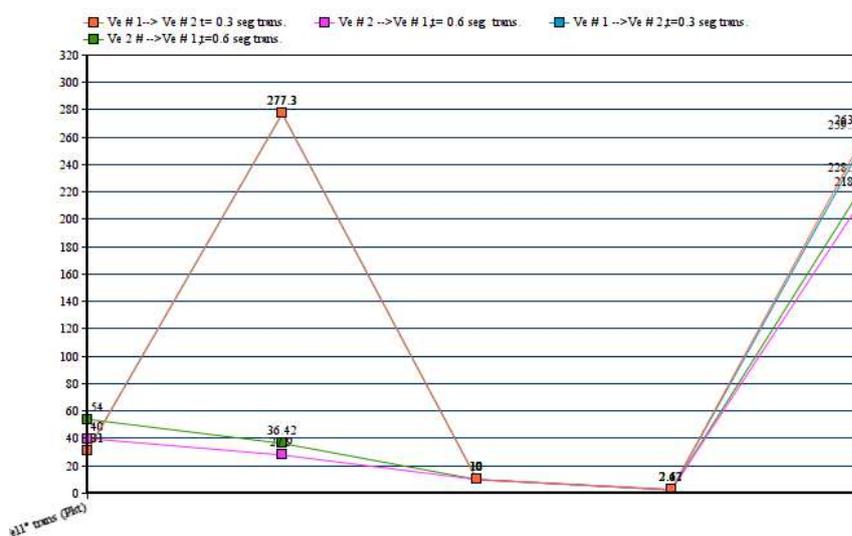


Figura 5.4: Comparación de datos de la Tabla 5.3, 5.4, 5.5 y 5.6

Las características de recepción y transmisión de paquetes broadcast en donde anteriormente se tomaron en cuenta las variables en el momento que los vehículos hacen su transmisión de paquetes. Observando el número de paquete, y de bytes enviados(Tx) y recibidos (Rx) en un tiempo de t=0.3 s.

5. RESULTADOS

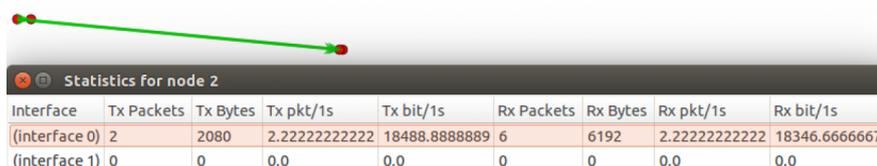


Figura 5.5: Datos de los paquetes transmitidos y recibidos

Se concluye en esta sección que entre más corto se el periodo de transmisión de paquetes, mejor margen de distancia tienen para hacer el rebase, y la otra conclusión es que entre menor velocidad lleve el vehículo V2, más margen de distancia tendrán para obtener un mayor rebase, es decir, el caso en que el vehículo V2 lleva una velocidad de 33 m/s en un tiempo $t = 0.3$ seg de transmisión.

5.2. Comunicación entre tres vehículos con (MRU).

Se simula con la comunicación broadcast de tres vehículos y movimiento rectilíneo uniformemente, cada uno presentan velocidad constante, con una área circular de radio de 150 mts y transmisiones de paquetes broadcast de intervalos de $t = 0.3$ seg con velocidades iguales de 22 m/s (80 km/hr) para los vehículos V0, V1, V3 excepto para el vehículo V2 que presenta una velocidad mínima de 120 km/hr (33 m/s) y máxima de 170 km/hr (47.2 m/s). Finalmente se hace la comparación de resultados entre estas dos velocidades. De los 4 vehículos, solo se centra en los vehículos donde pueda ver una colisión que es el objetivo principal de estos resultados. Los vehículos a seguir son el vehículo 1 y el vehículos 2.

5.2.1. Resultados con velocidad mínima

Determinando los parámetros siguientes mostrados en las Tablas 5.7, 5.8 y 5.9 y sus gráficas correspondientes.

Resultados del vehículo V1 con respecto al vehículo V2 MRU [vehículos V0,V1,V2,V3 y su velocidades(22,22,33,22 m/s) respectivamente] Tiempo en que hace su transmisión de paquetes t= 0.3 s				
Velocidad en su primer transmisión de paquetes	Posición en su primer transmisión de paquetes	Aceleración	Tiempo de encuentro con el vehículo V2	Margen de separación en su primer transmisión para rebasar
$\vec{V}_x=22$ m/s	$\vec{X}_1=279.4$ m	$\vec{a}_x= 0$ m/s ²	t = 3.88 s	d = 266.5 m

Tabla 5.7: Características escalares del vehículo V1 respecto al vehículo V2 en donde V1 manda una transmisión de paquetes en un tiempo de t=0.3s a los vehículos V0, V2 y V3.

Resultados del vehículo V2 con respecto al vehículo V1 MRU [vehículos V0,V1,V2,V3 y su velocidades(22,22,33,22 m/s) respectivamente] Tiempo en que hace su transmisión de paquetes t= 0.6 s				
Velocidad en su primer transmisión de paquetes	Posición en su primer transmisión de paquetes	Aceleración	Tiempo de encuentro con el vehículo V2	Margen de separación en su primer transmisión para rebasar
$\vec{V}_x=33$ m/s	$\vec{X}_1=-25.8$ m	$\vec{a}_x= 0$ m/s ²	t = 3.88 s	d = 244 m

Tabla 5.8: Características escalares del vehículo V2 respecto al vehículo V1 en donde V2 manda una transmisión de paquetes en un tiempo de t= 0.6 a los vehículos V0, V1 y V3.

5. RESULTADOS

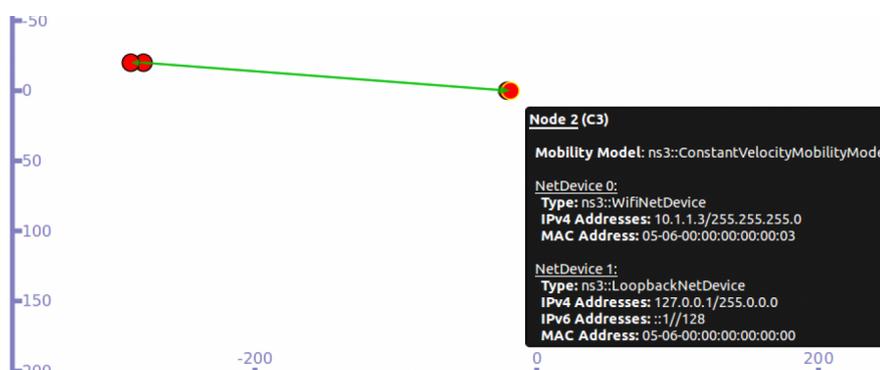


Figura 5.6: Datos de los paquetes transmitidos y recibidos

En la figura 5.6 muestra el animador de la plataforma ns-3 donde existe la transmisión de paquetes por parte del vehículo V2 y son recibidos de forma correcta por los demás vehículos.

5.2.2. Resultados con velocidad máxima

Resultados del vehículo V1 con respecto al vehículo V2				
MRU [vehículos V0,V1,V2,V3 y su velocidades(22,22,47.2,22 m/s) respectivamente]				
Tiempo en que hace su transmisión de paquetes t= 0.3 s				
Velocidad en su primer transmisión de paquetes	Posición en su primer transmisión de paquetes	Aceleración	Tiempo de encuentro con el vehículo V2	Margen de separación en su primer transmisión para rebasar
$\vec{V}_x=22$ m/s	$\vec{X}_1=279.4$ m	$\vec{a}_x= 0$ m/s ²	t = 3.33 s	d = 262.24 m

Tabla 5.9: Características escalares del vehículo V1 respecto al vehículo V2 en donde V1 manda una transmisión de paquetes en un tiempo de t= 0.3 a los vehículos V0, V2 y V3.

Se hace una comparación entre las dos figuras mediante una gráfica, observando que hay más margen de separación en la 1ª transmisión para poder rebasar en la fig. 5.14. con 266.5 m. Se observa que la mejor velocidad para tener mayor distancia para rebasar es de 33 m/s.

Resultados del vehículo V2 con respecto al vehículo V1				
MRU [vehículos V0,V1,V2,V3 y su velocidades(22,22,47.2,22 m/s) respectivamente]				
Tiempo en que hace su transmisión de paquetes t= 0.6 s				
Velocidad en su primer transmisión de paquetes	Posición en su primer transmisión de paquetes	Aceleración	Tiempo de encuentro con el vehículo V2	Margen de separación en su primer transmisión para rebasar
$\vec{V}_x=47.2$ m/s	$\vec{X}_1=-34.32$ m	$\vec{a}_x= 0$ m/s ²	t = 3.33 s	d = 235.48 m

Tabla 5.10: Características escalares del vehículo V2 respecto al vehículo V1 en donde V2 manda una transmisión de paquetes en un tiempo de t= 0.6 a los vehículos V0, V1 y V3.

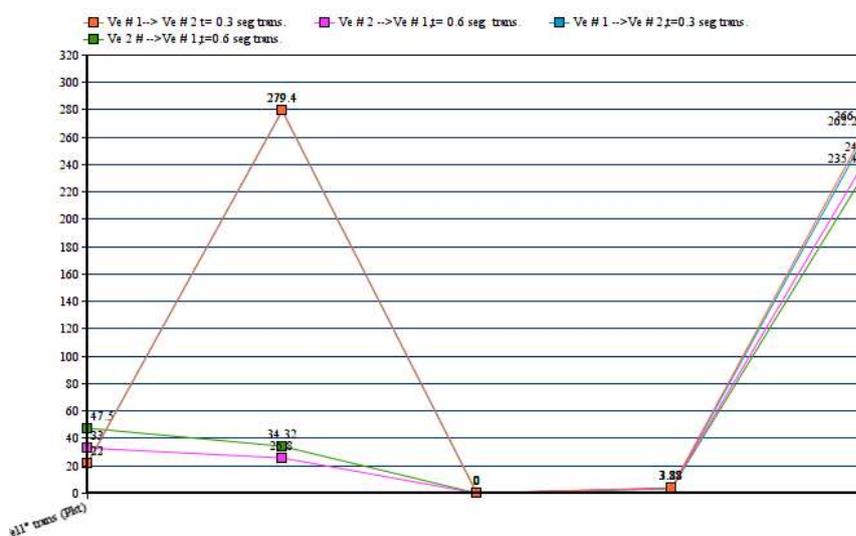


Figura 5.7: Características del vehículo 2 respecto al vehículo 1.

Características de recepción y transmisión de paquetes broadcast en donde anteriormente se toma en cuenta las variables en el momento que los vehículos hacen su transmisión de paquetes.

5. RESULTADOS

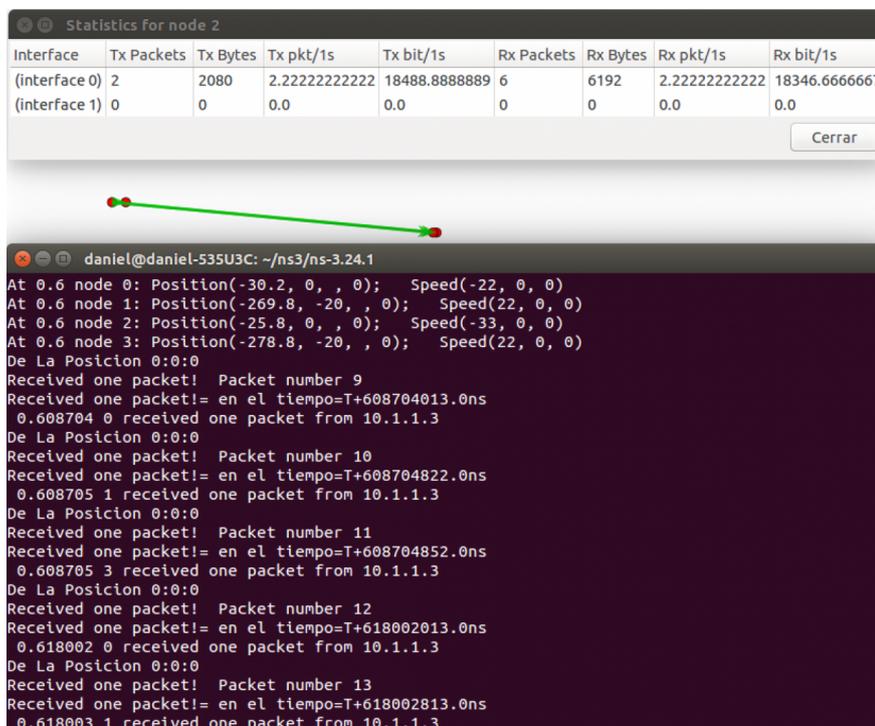


Figura 5.8: Datos de los paquetes transmitidos y recibidos

En conclusión de este último escenario simulado con velocidades de 27 m/s (100 km/hr), 33 m/s (150 km/hr) y 47.2 m/s (170 km/hr) para el vehículo 2, sucede que entre menos velocidad, mayor distancia de separación y tiempo de encuentro. Esto sucede en la primera retransmisión emitida por el vehículo V1 y escuchada por el vehículo V2. Existe más probabilidad de colisionar ya que las velocidades tiende acercarse a las velocidades de sus nodo vecinos.

Por otra parte si el vehículo inicia con una velocidad mayor, menor distancia de separación y tiempo de encuentro. Esto sucede en la primera retransmisión emitida por el vehículo V1 y escuchada por el vehículo V2. Existe menos probabilidad de colisionar ya que las velocidades tiende alejarse a las velocidades de vehículos vecinos, es decir entre más rapidez tenga el vehículo 2 respecto a los demás vehículos mas rapido rebasará a su vehículo sucesor y meterse a su carril.

Conclusiones

6.1. Logros

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

La plataforma permitirá la implementación de servicios de seguridad vial ayudando al conductor y pudiendo contribuir a disminuir la cifras de accidentes en la carretera. Gracias a este sistemas de conducción basados en la plataforma de comunicaciones broadcast se pudo ver las cosas más en un entorno real, claro que se puede esperar una gestión más eficiente del tráfico basados en servicios de valor añadido para los ocupantes de los vehículos y que más adelante mencionaremos una mejor alternativa.

La comparación de los escenarios que se llevaron a cabo en capítulos anteriores, ha permitido valorar la eficiencia de los protocolos UDP y comunicación broadcast. Se han obtenido resultados que se podrán reutilizar en otros proyectos.

Uno de los objetivos que se habían propuesto era la definición de una plataforma de simulación para el entorno móvil. Este objetivo se ha cumplido ya que ahora disponemos de una plataforma de simulación ns-3 eficiente que ha permitido comparar distintos escenarios de comunicaciones broadcast.

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

6.2. Futuras líneas de trabajo

Obviamente, aunque se ha avanzado en la investigación acerca de las redes vehiculares, queda mucho camino por recorrer y es necesario seguir haciendo esfuerzos de investigación en este campo. Primero, es suma importancia seguir investigando para proponer una plataforma de simulación más realista y más completa. Tener una implementación propia de un enlace de respaldo dando resultados más aproximados a lo real.

Se podría simular el siguiente escenario: De la misma manera en que se llevó a cabo la simulación de este trabajo y con la ayuda de estos resultados, se podrá mejorar, si los paquetes transmitidos y recibidos se van enlistando en comunicación multi salto, es decir, si se tienen cuatro vehículos (V1, V2, V3, V4) en donde V1 y V2 van en el mismo sentido a diferencia de V3 y V4 en sentido contrario. Se tendrían un mayor margen de distancia para rebasar si el V4 le envía la posición a V3, estando V4 más lejos de V2 y con V3 más cercas de V2, V3 le mandaría su posición y la de V4 a V2, así mismo V2 mandaría la posición de V4, V3 y su posición a su vehículo más próximo en su mismo sentido que sería V1. Esa solución, hoy en día no parece muy fiable, debido a la sobrecarga, los retardos de establecimientos de comunicación, y la complejidad que conlleva. Sin embargo, la comunidad científica está en continuos avances en estos campos, y pronto podría ofrecernos las garantías de éxito que necesitamos para desarrollar un proceso de encaminamiento polimorfo.

Código/Manuales/Publicaciones

A.1. Apéndice

GLOSARIO DE ACRÓNIMOS

AODV: Ad-hoc On-Demand Distance-Vector Routing

BS: Base Station

CSMA/CA: Carrier Sense Multiple Access With Collision Avoidance

CTS: Clear To Send

DSDV: Destination Sequenced Distance Vector

DSR: Dynamic Source Routing

DSRC: Dedicated Short Range Communications

FTP: File Transfer Protocol

GAMER: Geocast Adaptative Mesh Environment for Routing

GPS: Global Positioning System

IEEE: Institute of Electrical and Electronics Engineers

IP: Internet Protocol

MAC: Media Access Control

MANET: Mobile Ad-hoc Network

NS3: network Simulator 3

OFDM: Ortogonal Frequency Division Multiplexing

OLSR: Optimized Link State Routing

SID: Sistema de Detección de Intrusos

SUMO: Simulation for Urban Mobility

TCP: Transmission Control Protocol

TORA: Temporally Ordered Routing Algorithm

UDP: User Datagram Protocol

UMTS: Universal Mobile Telecommunications System

UWB: Ultra Wide Band

V2V: Vehicule-to-Vehicule Communications

VANET: Vehicular Ad-hoc Network

WAVE: Wireless Access in the Vehicular Environment

WPAN:Wireless Personal Area Network

MRU:Uniform Line Movement

MRUA:Uniform Line Movement Acceleration

Tx:Transmission Data

Rx:Receiver Data

V1:Vehicle1

V2:Vehicle2

V3:Vehicle3

V4:Vehicle4

Pkt:Package

Bibliografía

- [1] DOUMENTC, E. (Junio 2008). Estudio comparativo de protocolos de encaminamiento en redes vanet, @Vanet. 1
- [2] <http://bibing.us.es/proyectos/abreproy/12216/fichero/PFC>
Evaluacion+de+protocolos+de+encaminamientos+para+redes+vehiculares+28VANET29+-+Jose+Manuel+Garcia+Campos.pdf (Junio 2015). Redes vanets @ADHOC. 12
- [3] <http://es.ccm.net/contents/789-introduccion-a-wi-fi-802-11-o-wifi> (Junio 2015). Estándares inalámbricos, @IEEE. 9
- [4] <https://gnu-linux.org/installation-of-ns3-on-ubuntu.html> (Julio 2015). Instalación ns-3 en ubuntu @Install. 36
- [5] <https://www.geogebra.org/?lang=es> (Julio 2015). Software geogebra @GEOGEBRA. 41
- [6] <https://www.nsnam.org/doxygen/> (Agosto 2007). Módulos @MODULE. 32
- [7] <http://www.soporteparapc.com/2014/09/diferencia-modo-ad-hoc-y-infraestructura.html> (Junio 2015). Red adhoc @ADHOC. 7
- [8] Pinto, G. J., López, D., and Pedraza, L. F. (2012). Simulación y análisis de desempeño de protocolos unicast para redes vanet. *Tecnura*, 16(31):66–75. 12
- [9] Resnick, R., Halliday, D., and Krane, K. S. (1993). Física. 26
- [10] Torres Giraldo, M. A. (2013). Estudio y simulación de redes ad-hoc vehiculares vanets. 8
- [11] Tseng, Y.-C., Ni, S.-Y., Chen, Y.-S., and Sheu, J.-P. (2002). The broadcast storm problem in a mobile ad hoc network. *Wireless networks*, 8(2-3):153–167. 21