

**ANÁLISIS DEL USO DE NODOS
MÓVILES VIRTUALES
PARA PROCESOS DE ENCAMINAMIENTO
EN REDES VEHICULARES AD-HOC.**

JUAN PABLO HURTADO

Egresado de Ingeniería Electrónica
Universidad Politécnica Salesiana

EDGAR PATRICIO SIGUENZA

Egresado de Ingeniería Electrónica
Universidad Politécnica Salesiana

Dirigido por:

ING. JACK BRAVO TORRES, Mg.T.

Ingeniero Electrónico
Docente de la Universidad Politécnica Salesiana
Director del Área de Ciencias Exactas
Unidad de Posgrados
Universidad Politécnica Salesiana



CUENCA - ECUADOR

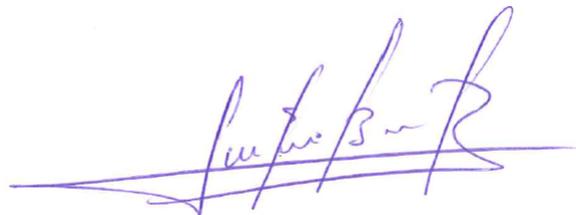
2013

Ing. Jack Bravo Torres, Mg.T. Msc

Certifica:

Que el trabajo intitulado “*Análisis del uso de Nodos Móviles Virtuales para Procesos de Encaminamiento en Redes Vehiculares Ad-Hoc*”, realizado por Juan Pablo Hurtado y Edgar Patricio Siguenza, cumple exitosamente con todos los objetivos trazados.

Cuenca, Julio 2013

A handwritten signature in blue ink, appearing to read 'Jack Bravo Torres', with a long horizontal flourish extending to the left.

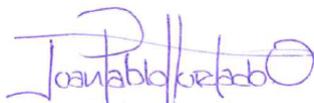
Ing. Jack Bravo Torres, Mg.T. Msc

DIRECTOR DE TESIS

Declaratoria de Responsabilidad

Las ideas, resultados y conclusiones contenidas en esta tesis: “*Análisis del uso de Nodos Móviles Virtuales para Procesos de Encaminamiento en Redes Vehiculares Ad-Hoc*”, son exclusiva responsabilidad de los autores y autorizamos el uso total a la Universidad Politécnica Salesiana para fines académicos.

Cuenca, Julio 2013



Juan Pablo Hurtado

AUTOR



Edgar Patricio Siguenza

AUTOR

Dedicatoria

Dedico a Dios, quien me ha permitido dar un paso más hacia mi objetivo. A mi Hermano por todas sus palabras que han cambiado mi vida.

“Los cambios son duros; pero hay que darles una oportunidad. Esperar que las cosas sigan igual, no funciona.” Remigio Hurtado

“Es tiempo de decidir, sí o no; las cosas a medias no funcionan.” Remigio Hurtado

Juan Pablo

Esta tesis está dedicada a mis padres, abuelos y tíos. Sus esfuerzos hicieron todo esto posible.

“Dos ratoncitos cayeron en un cubo de nata; el primer ratón enseguida se rindió y se ahogó, el segundo ratón decidió pelear, y se esforzó tanto que finalmente transformó la nata en mantequilla y consiguió escapar. Caballeros, desde este momento yo soy ese segundo ratón”

Patricio

Agradecimientos

Agradezco a Dios, El ha sido la inspiración y el motivo de cada esfuerzo. A mi familia y Amigos.

A mi profesor Jack Bravo por todo su apoyo.

Juan Pablo

Agradezco a mis padres, abuelos y tíos por el apoyo y los consejos brindados a lo largo de toda mi vida. A Jack Bravo por todo su dedicacion hacia el desarrollo de esta tesis

Patricio

Resumen

Los nodos virtuales son generados por una Capa Virtual añadida dentro del modelo TCP/IP. La Capa Virtual es un programa abstracto basado en regiones propuesto en[1] para redes móviles ad-hoc (MANET). Un nodo virtual es formado por nodos físicos dentro de una misma región e independiente del movimiento de los nodos físicos; por tanto un nodo virtual es simulado como un nodo fijo, en una ubicación conocida. Esta característica estable y predecible facilita la implementación de aplicaciones y elimina muchas de las dificultades de las redes móviles.

Esta tesis aprovecha las ventajas provistas por la Capa Virtual para el desarrollo de un protocolo de enrutamiento basado en nodos virtuales para Redes Vehiculares ad-hoc (VANET). Para la implementación de la Capa Virtual se tomó como referencia las propuestas planteadas en[2] y[3].

Durante el desarrollo de la Capa Virtual, se determinaron ciertas optimaciones para incrementar el rendimiento de la Capa Virtual. Después de establecer un modelo óptimo para la Capa Virtual, se desarrolló un protocolo de enrutamiento reactivo capaz de trabajar sobre nodos virtuales. Este fue construido manteniendo las principales características de AODV (un protocolo bien conocido para redes móviles), con el fin de contrastar su funcionamiento. Además se implementó una aplicación VANET para la evaluación del tráfico vehicular.

Esta tesis validó las ventajas de la Capa Virtual mediante simulaciones y comparaciones, en estas se obtuvo un mejor desempeño en cuanto a múltiples factores de rendimiento, frente a redes móviles sin la implementación de la Capa Virtual.

Por último se desarrolló un software basado en JAVA para la creación rápida de escenarios de simulación para redes vehiculares; desde esta aplicación se generó el tráfico vehicular y se estableció los parámetros de la red VANET para el desarrollo de las simulaciones.

Contenidos

Dedicatoria	iv
Agradecimientos	v
Resumen	vi
1. Introducción	1
1.1. Estructura de la tesis	2
2. Estado del Arte	3
2.1. Redes Móviles Ad-Hoc	3
2.2. Redes Vehiculares Ad-Hoc	4
2.3. Protocolos de Encaminamiento para MANETs	4
2.3.1. Protocolos Proactivos	4
2.3.2. Protocolos Reactivos	5
2.3.3. Protocolo <i>Ad hoc On-Demand Distance Vector</i> (AODV)	5
2.4. Capa de Nodo Virtual (<i>VNLayer: Virtual Node Layer</i>)	8
2.4.1. Benéficos y Limitaciones de la <i>VNLayer</i>	9
2.4.2. Ejemplo del uso de la <i>VNLayer</i>	10
3. Modelo de la Capa Virtual y Encaminamiento sobre Nodos Virtuales	12
3.1. Capa Virtual	12
3.1.1. Regiones	13
3.1.2. Localización de Nodos	16
3.1.3. Estados	18
3.1.4. Mensajes de la Capa Virtual	18
3.1.5. Elección del Líder	19
3.1.6. Elección de Backups	22
3.2. Encaminamiento VAODV	24
3.2.1. Mensajes <i>Hello</i>	25
3.2.2. Descubrimiento de Ruta	26
3.2.3. Mantenimiento de Rutas	27
3.2.4. Envío de Mensajes	31
3.2.5. Pérdida de Paquetes	31
3.2.6. Estado de Sincronización	32
3.2.7. Restricciones	32
3.2.8. Desventajas de VAODV	33
3.3. Optimizaciones a VAODV	35
3.3.1. <i>REQUEST</i> y <i>REPLY</i> No Descartados (RRND)	35

3.3.2.	<i>REPLY</i> unicast (RU)	35
3.3.3.	Corrección de Ruta por <i>REQUEST</i> (CRR)	36
3.3.4.	Sincronización por Canal Compartido (SCC)	37
3.4.	Optimizaciones de la Capa Virtual	37
3.5.	Aplicación VANET	38
4.	Simulaciones y Comparaciones	39
4.1.	Consideraciones de Simulación	39
4.2.	Evaluación del Rendimiento de la Capa Virtual	40
4.3.	Evaluación del Rendimiento de VAODV	41
4.3.1.	Sobrecarga de Tráfico	42
4.3.2.	Número de Saltos	44
4.3.3.	Descubrimientos de Ruta	45
4.3.4.	Paquetes Entregados	46
4.3.5.	Efecto del Incremento del Tráfico Vehicular	47
4.4.	Consideraciones Adicionales	47
5.	Conclusiones y Trabajos Futuros	51
5.1.	Sobre la Capa Virtual	51
5.2.	Sobre VAODV	52
5.3.	Sobre los Resultados de las Simulaciones	53
5.4.	Trabajos Futuros	53
5.4.1.	Aplicaciones para VANETs	53
5.4.2.	Líderes Sobrecargados	53
5.4.3.	Hibridación del Servicio	53
A.	Programación de la Capa Virtual y de VAODV en <i>ns-2</i>	55
B.	<i>Simulator of Urban Mobility</i> y <i>VANet Simulator modeler</i>	57
B.1.	SUMO	57
B.2.	VANS	57
	Bibliografía	59

1. Introducción

Durante los últimos años, el incremento y miniaturización de dispositivos móviles han inducido un cambio revolucionario en el mundo de las comunicaciones inalámbricas [4]. La naturaleza ubicua de los dispositivos móviles, hace que las redes inalámbricas sean una fácil solución al acceso de información.

Las redes inalámbricas se pueden dividir en: infraestructuradas y sin infraestructura. Las redes con infraestructura son aquellas que están supervisadas por dispositivos centrales; estos son los encargados de gestionar el tráfico entre los dispositivos finales. La ventaja de usar una infraestructura es el control y gestión centralizada de la información. Por otro lado, las redes inalámbricas sin infraestructura permiten la conexión de dispositivos finales sin la utilización de un dispositivo central de gestión. Es decir, un dispositivo final o nodo es al mismo tiempo, un transmisor, un receptor o un enrutador de datos. Las redes de este tipo son llamadas redes Ad-Hoc. La ventaja de usar una red inalámbrica ad-hoc son: bajo costo en implementación, tiempo reducido de configuración, y no necesita de gestión ni de control centralizado. La desventaja es el aumento de procesamiento de los nodos y la dependencia de nodos intermedios para la comunicación[5]. De igual forma, el consumo de energía es una característica limitante en redes ad-hoc. Los móviles cuentan con una fuente de energía limitada, por lo que es esencial el uso de protocolos de enrutamiento con poca sobrecarga de tráfico para la conservación de energía.

Los protocolos de enrutamiento existentes en redes cableadas son incompatibles en redes móviles ad-hoc (MANETs) debido al comportamiento dinámico de los nodos. Existen protocolos dedicados a este tipo de redes como AODV[6]. El problema de usar estos protocolos de encaminamiento, es que no funcionan correctamente en topologías con un gran número de nodos, inducen alto tráfico en la red y retardo. Brown, et al en [1] propone el uso de una capa Virtual (*VNLayer*) como una forma de aliviar problemas de encaminamiento en MANETs.

El desarrollo de esta tesis, toma como referencias propuestas realizadas en[1] y[2] para crear una capa virtual (*VNLayer*) para redes MANETs. Se desarrolla un protocolo de enrutamiento reactivo basado en el protocolo AODV con soporte para nodos Virtuales (VNs). Se implementa una aplicación para redes VANETs basada en nodos virtuales servidores, y una aplicaciones para comunicaciones entre usuarios. La aplicación de la capa virtual, y el protocolo de enrutamiento fueron simulados en *Network Simulator 2 (ns-2)* [7].

ns-2 es un simulador de redes de comunicaciones, soporta la simulación de protocolos TCP, UDP, así como de protocolos de enrutamiento como: RIP, OSPF, AODV,

OLSR. También simula escenarios cableados, inalámbricos y satelitales. Además se puede crear o modificar protocolos, tiene un ambiente de simulación visual llamado Nam¹. *ns-2* está desarrollado en C++, con un intérprete OTcl como su interfaz, está disponible tanto para Windows como para Linux. Además su principal ventaja es que cuenta con una capa de enlace óptima para redes inalámbricas (802.11). Esto disminuye sustancialmente el tiempo y la complejidad del desarrollo de la *VNLayer*.

Finalmente se realizó una aplicación gráfica basada en Java el cual facilita la creación de redes vehiculares. Esta aplicación llamada *VANet Simulador modeler* (VANS) tiene las características de generar todos los archivos necesarios para crear una topología de tránsito vehicular en Simulator of Urban MObility (SUMO²) y su posterior exportación a *ns-2*. Además, esta aplicación ejecuta automáticamente simulaciones, tanto en SUMO, *ns-2* y Nam

1.1. Estructura de la tesis

En el capítulo II se hará una revisión de las redes móviles ad-hoc (MANETs), las redes vehiculares ad-hoc (VANET) y los protocolos de enrutamiento para redes MANETs. Se estudia a fondo al protocolo *Ad hoc On-Demand Distance Vector* (AODV) y finalmente se define el concepto de la Capa de Nodo Virtual (VNLayer). En el capítulo III se presenta el diseño de la implementación de la capa Virtual y el protocolo de enrutamiento VAODV. En el capítulo IV se contrasta los dos protocolos: AODV con VAODV en escenarios VANETs. Finalmente las conclusiones y recomendaciones son abordadas en el capítulo V.

¹Network AniMator (Nam) es la interfaz visual de *ns-2*.

²Simulator of Urban MObility [8](SUMO) es un simulador microscópico de tránsito vehicular.

2. Estado del Arte

El desarrollo de dispositivos móviles personales y el incremento de las comunicaciones inalámbricas están generando una mayor demanda de aplicaciones en el campo de las redes de comunicaciones inalámbricas ad hoc (MANETs)[9]. Dentro de este ámbito, las redes vehiculares ad hoc (VANETs) son uno de los principales centros de interés por parte de la comunidad científica. Los nuevos Sistemas de Transporte Inteligente (ITS)[10] que buscan mejorar los procesos de transporte en las ciudades y disminuir los índices de accidentes, son una muestra del potencial de estas nuevas redes.

No obstante, las redes VANETs son parte de las redes MANETs, su características hace que los protocolos de encaminamiento empleadas en unas no sean adecuadas para las otras. Así, en este capítulo, se revisará las principales características de ambas redes. Se desarrolla un estudio de los protocolos de encaminamiento para redes MANETs, profundizando en el protocolo reactivo AODV. Finalmente, analizaremos la idea de la virtualización como un medio para hacer frente a las dificultades presentes en las redes MANETs en cuanto a su variabilidad en topología.

2.1. Redes Móviles Ad-Hoc

Las redes móviles Ad-Hoc son redes inalámbricas que no necesitan de infraestructura ni supervisión centralizada. Este tipo de redes están conformadas por un conjunto de nodos móviles que se auto-organizan de manera dinámica[4]. Su desarrollo se ha incrementado en los últimos años, debido al creciente uso de dispositivos móviles como Laptops, teléfonos inteligentes, sensores, entre otros. Su versatilidad en la implementación y su no necesidad de infraestructura son interesantes características que han impulsado los procesos de investigación en este campo[4]. Las redes móviles ad-hoc son utilizadas en comunicaciones vehiculares, redes personales, redes de sensores, para operaciones de emergencia y búsqueda, control de tránsito, aplicaciones militares, entre otras[11].

Si bien esto es una ventaja, dado que permite su rápido despliegue, también es la principal causa de los problemas existentes en la implementación de aplicaciones con calidad de servicio (QoS)[12]. En contraste con las redes cableadas, las redes MANETs presentan frecuentes cambios en su topología y un comportamiento dinámico debido al movimiento de sus dispositivos finales. Además, estas redes presentan restricción en su ancho de banda y en la capacidad de sus enlaces. De igual manera, los dispositivos móviles trabajan con limitaciones de energía y seguridad física[13].

Así, los protocolos de encaminamiento desarrollados para las redes cableadas no son aptos para redes MANETS[6, 14, 15].

2.2. Redes Vehiculares Ad-Hoc

Las redes vehiculares ad-hoc son una subclase de las redes móviles ad-hoc, estas son utilizadas para la comunicación inalámbrica entre vehículos. Al igual que las redes MANETS, estas redes no tienen infraestructura fija y en su lugar se basan en los propios vehículos para proporcionar funcionalidad de red[16]. La diferencia fundamental entre las VANETs y MANETS es el patrón de movilidad y el rápido cambio en la topología debido a las altas velocidades[17]. Por ello, los protocolos de encaminamiento usados en MANETS no son aptos para las redes VANETs[18]. Los servicios que brindan las VANETs son: advertencia de colisión, alerta de obstáculos en el camino, conducción cooperativa, advertencia de colisión en intersección, y asistencia en cambio de carril, entre otros[19].

2.3. Protocolos de Encaminamiento para MANETS

Los protocolos de encaminamiento para redes móviles Ad-hoc deben considerar ciertas limitantes como: ancho de banda, topología variable, pérdidas por propagación, consumo de energía, entre otros[16]. Los protocolos de encaminamiento de redes fijas no son compatibles con las redes móviles ad-hoc. Entre las razones para esto podemos citar las siguientes[2]:

- Los protocolos de enrutamiento cableado asumen una topología fija y estable,
- la arquitectura de red plana en MANET requiere que cada router tenga una entrada de ruta para cada destino. Este intercambio de información requiere una sobrecarga de tráfico en la red,
- y por último, el medio de transmisión es inalámbrico y compartido entre los nodos móviles adyacentes, lo que causa pérdidas de paquetes debido a colisiones de mensajes y congestiones.

Por lo tanto, nuevos protocolos deben ser usados para este tipo de redes. En[5] se clasifica a los protocolos de enrutamiento MANET en dos tipos:

- Protocolos proactivos
- Protocolos reactivos

2.3.1. Protocolos Proactivos

En este tipo de protocolos, las rutas son calculadas periódicamente sin la necesidad de que un paquete requiera transmitirse, la ventaja de este tipo de protocolos es la

rápida detección de fallas de enlace y enrutamiento. La desventaja es la introducción de tráfico constante e innecesario a la red[2]. Algunos protocolos proactivos son: DSDV[20], OLSR[21], DBF, GSR, WRP[22].

2.3.2. Protocolos Reactivos

Este tipo de protocolos calcula la ruta solo cuando es necesario realizar una transmisión. Cuando se produce una transmisión de origen a destino, inicia el procedimiento de descubrimiento de ruta, caso contrario no se realiza ninguna acción. La ruta expira cuando se produce una falla de enlace y debe iniciarse el proceso de descubrimiento. La desventaja de este tipo de protocolos es el tiempo de actualización de las rutas. Algunos protocolos proactivos son: DSR[23], AODV[6, 24], y TORA[25]

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

AODV fue propuesto por Charles E. Perkins[6] quien además desarrollo el protocolo DSDV. AODV usa tablas de enrutamiento en lugar de caches de rutas, esto significa que para cada destino solo una ruta es mantenida en su tabla de enrutamiento. Usa un número de secuencia de destino para garantizar la ausencia de bucles. Además, su funcionamiento se basa en dos estados: descubrimiento de la ruta y mantenimiento de la ruta.

Cuando se desea enviar un paquete a un nodo destino, el primer proceso es consultar su tabla. Cada ruta conocida es contenida en su tabla de enrutamiento con los siguientes datos: IP destino, número de secuencia, número de saltos, dirección del siguiente salto hacia el destino, estado de la ruta y el tiempo de expiración.

AODV tiene dos modos de operación que se diferencian por la manera en que detectan las fallas de enlace de las rutas: el primero detecta fallos cuando la capa de Enlace advierte una falla de enlace a la capa de red, mientras que en el segundo modo se da cuando expira una ruta al no escuchar mensajes *Hello*.

En el primer modo de operación, el fallo de enlace se detecta cuando la sub-capa MAC no puede realizar un mapeo entre la dirección MAC y la dirección IP del *next_hop* o cuando no recibe mensajes de confirmación del *next_hop*. Antes de enviar un mensaje de error se compara el número de *hops* hacia el destino y la fuente, si es más corto el camino al destino, se realiza una reparación local. Si esta reparación no tiene éxito, mensajes de error son difundidos en broadcast.

El segundo modo de operación envía mensajes *Hello* para refrescar la ruta. Los mensajes *Hello* no son utilizados para crear o actualizar rutas (*next_hop*, *hops*), sino para refrescar el tiempo de expiración de la ruta (*time_expire*). Cuando una ruta se ha establecido, una falla de enlace se detectará al no recibir mensajes *Hello* del *next_hop*, durante cierto tiempo. Para que una falla de enlace sea detectada rápidamente, el tiempo de envío de los *Hello* debe ser lo menor posible; sin embargo,

al disminuir el tiempo se genera más tráfico en la red. Cuando se detecta un fallo de enlace, la ruta se invalida y un mensaje de error es enviado en broadcast. Los que escuchen el mensaje comparan si el que envió el mensaje de error es el *next_hop* hacia algún destino; si lo es, invalidarán la ruta y enviarán réplicas del mensaje de error. El resultado de los mensajes de error, es invalidar la ruta hacia el destino –ahora inalcanzable– en todos los hops que son parte de la ruta.

Independiente del modo de operación, tres procedimientos son realizados: descubrimiento de ruta, mantenimiento de ruta y envío de mensajes.

2.3.3.1. Descubrimiento de Ruta

Cuando la capa de aplicación de un nodo desea transmitir mensajes hacia un nodo destino, el mensaje es recibido primero por la capa de transporte, los datos se dividen en segmentos o datagramas y se añade el encabezado del protocolo de la capa de transporte (TCP o UDP). La capa de red añade la dirección destino y origen. Antes de enviar el paquete se consulta la tabla de enrutamiento, si hay una coincidencia y la ruta esta activa, se añade al paquete el siguiente salto hacia al destino. Si no hay una ruta valida, un proceso de descubrimiento de ruta comenzará.

Para ello, el nodo origen envía un mensaje de solicitud (*REQUEST*). El mensaje lleva el número de secuencia de origen, el número de secuencia conocido del destino, el número de *hops*, un identificador de broadcast y la dirección IP destino. Todos los nodos que escuchan el *REQUEST* (si no tienen una ruta actual hacia el destino) replican el mensaje, esto aumenta el número de *hops* en un valor de 1. Cuando un nodo escucha un *REQUEST* añade a su tabla la ruta hacia el nodo origen. Si un nodo tiene la ruta hacia el destino, comparará el número de secuencia del destino contenido en el mensaje *REQUEST* con el contenido en su tabla; si el segundo es mayor, enviará un mensaje de respuesta (*REPLY*) hacia el origen. Los mensajes *REQUEST* son replicados a manera de broadcast en cada nodo; debido a esto, un nodo puede escuchar la misma solicitud varias veces. Cada *REQUEST* lleva un identificador que es almacenado en cada nodo por cierto tiempo. Cuando un nodo escucha varios *REQUEST* con el mismo identificador, procesará el primero y los demás mensajes los descartará.

Por otra parte, cuando ningún nodo intermedio, entre el origen y destino, posea una ruta actual, el mensaje *REQUEST* es retransmitido hacia el destino. De igual forma, varios *REQUEST* pueden recibirse en el destino. Por tanto, el primer *REQUEST* en llegar, será la ruta que tome el mensaje *REPLY* de vuelta hacia el origen. Los mensajes *REPLY* son enviados en unicast.

En la Figura 2.1 se muestra el proceso de descubrimiento¹, cuando el nodo 1 envía un *REQUEST* hacia el destino (nodo 6), el mensaje es escuchado por el nodo 3 y

¹Se supone un radio enlace igual para cada nodo. La tabla de enrutamiento en cada nodo está vacía en un principio. Mensajes *REQUEST* son transmitidos en broadcast (se muestran en líneas discontinuas), los mensajes *REPLY* son unicast.

4. Estos añadirán la ruta hacia el nodo 1 ($next_hop = 1$, $hops = 1$), guardarán el identificador de broadcast y replicarán el *REQUEST* aumentando en uno el número de hops. El nodo 2 y 5 escucharán los mensajes –solamente el primero en arribar será procesado–, ambos añadirán la ruta hacia 1 (en el nodo 2: $next_hop = 3$, $hops = 2$; en el nodo 5: $next_hop = 4$, $hops = 2$), aumentarán el número de hops y replicarán el *REQUEST*. El nodo destino al escuchar el *REQUEST* reenviado por el nodo 5, aprenderá la ruta hacia el origen ($next_hop = 5$, $hops = 3$) y enviará en *REPLY* hacia el nodo 1 utilizando la ruta recién aprendida.

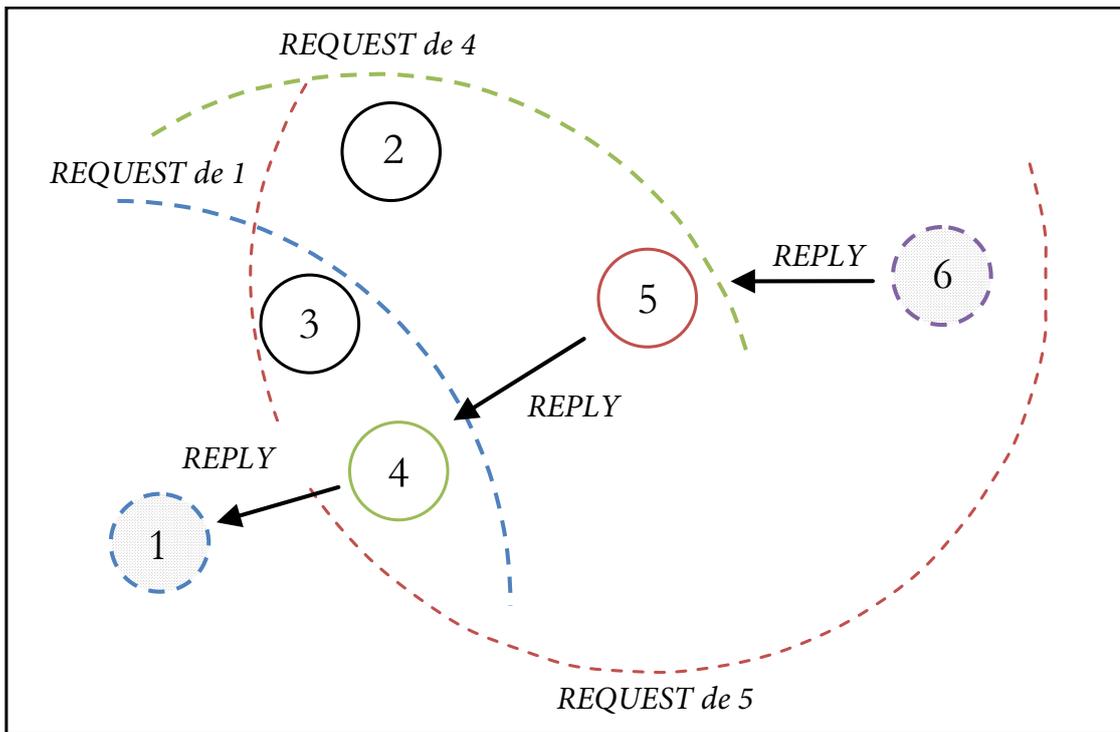


Figura 2.1.: Proceso de descubrimiento de ruta, desde el nodo origen 1, al nodo destino 6.

Cuando el nodo 5, 4 y 1 reciben el *REPLY* generado por el nodo 6, aprenden la ruta hacia éste nodo (en el nodo 5: $next_hop = 6$, $hops = 1$; en el nodo 4: $next_hop = 5$, $hops = 2$; en el nodo 1: $next_hop = 4$, $hops = 3$). Al conocer el nodo 1 la ruta hacia el nodo 6, enviará los paquetes almacenados en la cola previo al proceso de descubrimiento de ruta.

2.3.3.2. Mantenimiento de Ruta

El protocolo AODV establece un temporizador (*timer*) para depurar el estado de todas las rutas. El tiempo del este *timer* debe ser corto para que las rutas sean

actualmente validas, pero debe ser escogido de forma que no sobrecargue el procesamiento del protocolo.

Una ruta es válida, si su estado está activo (*RTF_UP*). Una ruta pasa a estado inactivo (*RTF_DOWN*) cuando el *time_expire* es menor al tiempo actual. Cuando AODV opera en el primer modo, la ruta puede tener un tercer estado (*RTF_REPAIR*), esto ocurre cuando se realiza una reparación local; si la reparación tiene éxito, la ruta será activa nuevamente; caso contrario, la ruta será inactiva.

En la Figura 2.1, si se reporta por la capa MAC una falla de enlace entre el nodo 5 y el nodo 6 al momento de transmitir un mensaje del nodo 1, se hará una reparación local, enviando un mensaje *REQUEST* con destino al nodo 6. Si la reparación no tiene éxito, el nodo 5 invalidará la ruta hacia 6 y enviará un mensaje de error (*ERROR*) en broadcast. En este caso, el nodo 4 invalidará la ruta y replicará el mensaje. El nodo 1 también inactivará la ruta hacia 6. Si la falla de enlace hubiese sido reportada en el nodo 4, al estar más cerca al nodo origen que al nodo destino (del mensaje), no hará una reparación local.

Operando en el modo 1, cada vez que se utiliza una ruta para enviar o reenviar mensajes y no se detecta fallas en el enlace (lo cual quiere decir que el mensaje llegó al destino), el *time_expire* es actualizado. En el modo 2, el *time_expire* se actualiza cada vez que se escucha en mensaje *Hello* del *next_hop* de la ruta.

2.3.3.3. Envío de Mensajes

Cuando un paquete debe ser enviado, se consulta la tabla de enrutamiento, si la ruta hacia el destino es conocida y además el estado de la ruta está activo, se envía el mensaje hacia el *next_hop* de la ruta. En la Figura 2.1, después del descubrimiento de la ruta, cuando el nodo 1 desea enviar un mensaje al nodo 6, el mensaje es enviado al nodo 4. El nodo 4 consulta la dirección IP destino del paquete; para ello, consulta en su tabla una ruta hacia el destino procediendo a enviar el paquete hacia el nodo 5, quien a su vez enviará al último *hop* de la ruta (el destino: nodo 6).

2.4. Capa de Nodo Virtual (*VNLayer: Virtual Node Layer*)

El uso de la Capa de Nodo Virtual fue propuesta por Brown, et al., en[1] como una forma de aliviar problemas de encaminamiento en MANETs. Brown define a *VNLayer* como una abstracción de programación con dos tipos de entidades: un programa de nodos virtuales predecibles (VNs: *Virtual Nodes*) que corresponden a los nodos virtuales, y los nodos clientes impredecibles (CNs: *Clients Nodes*), que corresponden a los nodos físicos en el sistema.

Según[1], las *VNLayers* pueden dividirse en dos categorías principales: la primera establece que la propagación es ideal, es decir, no existe pérdidas de mensajes, colisiones, retraso; asume que los mensajes siempre serán recibidos y que los nodos se comportan de una manera ideal. La segunda categoría en cambio, propone una mayor aproximación a la realidad, asume pérdidas de mensajes no uniformes, colisiones, retrasos de propagación, comportamiento de nodos no ideales, etc. Bajo estas condiciones, tanto en la primera como en la segunda categoría se realizan investigaciones en[15] de *VNLayer* fijas, con la desventaja de fallos de los nodos Virtuales debido a la ausencia de nodos físicos. En[15] propone una posible solución al problema, consiste en mover al nodo Virtual a través de trayectorias predefinidas conocidas por todos los nodos físicos. Sin embargo, la ausencia de nodos físicos en la trayectoria provoca fallos del Nodo Virtual. En[26], sugiere que los VNs re-calculen su ruta en tiempo real, evitando llegar a zonas despobladas de nodos físicos.

2.4.1. Benéficos y Limitaciones de la *VNLayer*

Wu en[2] destaca ciertos beneficios y limitaciones de la *VNLayer*. Primero, la *VNLayer* crea un sistema de agrupación de nodos físicos para establecer un nivel de jerarquía en la MANET. Esto reduce el número de nodos a ser manejados, y a su vez reduce la carga y el tráfico en la red. Además al situar a los VNs en posiciones fijas, el movimiento de los nodos sobre la red se vuelve predecible y estable.

El uso de *VNLayer* facilita la programación, ya que los programadores sólo tienen que hacer frente a la interfaz de usuario de la *VNLayer*, en lugar de tratar con un conjunto de nodos físicos altamente impredecibles. Por último, la agrupación de nodos basada en localización geográfica, hace que el trabajo de agrupamiento sea robusto. En la *VNLayer*, los nodos móviles verifican sus ubicaciones geográficas para determinar en qué región están.

Por el otro lado, la *VNLayer* tiene ciertas limitaciones, como el alto costo de implementación de dispositivos GPS en cada nodo para localizar su ubicación, aunque en[27] Lingxuan He et. al., diseña un método basado en aproximaciones Monte Carlo, para determinar la ubicación de un nodo móvil, de esta manera se reduce el costo de implantación.

La *VNLayer* es diseñada para mantener constante el estado de la aplicación, esto requiere un intercambio de mensajes de control entre los nodos Líderes y No-Líderes para la sincronización de su estado. Por tanto, la *VNLayer* necesita ser cuidadosamente diseñada a fin de mantener niveles bajos de tráfico de mensajes.

Aunque el mecanismo de sincronización de estado de la *VNLayer* puede garantizar un cierto nivel de uniformidad de estado entre el Líder y los nodos No-líderes, un nodo No-líder cuyo estado está fuera de sincronización, puede tener que asumir el control una región antes de que sea capaz de actualizar su estado, esto genera problemas como bucles de enrutamiento, etc.

La pérdida de mensajes es otra limitación de la *VNLayer*, en[27] transmiten los mensajes en modo de broadcast para asegurar que los mensajes puedan ser escuchados por todos los nodos físicos que emulan un nodo virtual, esto produce altas tasas de perdidas, interferencia a otros paquetes, colisión y sobrecarga en la red. En[2] usan un modo de capa de enlace más eficiente al combinar mensajes en broadcast y unicast.

2.4.2. Ejemplo del uso de la *VNLayer*.

La virtualización de nodos, pretende superar la desventaja del constante cambio de rutas, así como disminuir el tráfico en la red, generado por el protocolo de enrutamiento. Las redes estáticas por su parte, se comunican por medio de enlaces cableados. La Figura 2.2 muestra la ruta que toma un paquete desde el nodo 1 hacia el nodo 8. Cuando una red es estática, poco es el tráfico generado por el protocolo de enrutamiento y las rutas casi nunca cambiarán. La probabilidad que el nodo 4 (que forma parte de la ruta) se mueva hacia otra posición, de tal manera que inhabilite la ruta, es mínima. La ruta puede ser inhabilitada cuando el nodo 4 sea desconectado; bajo esta condición, se deberá tomar otra ruta.

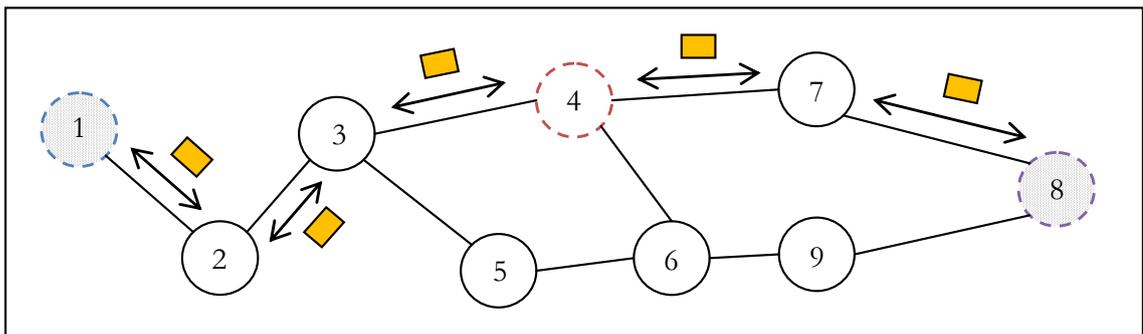


Figura 2.2.: Red inalámbrica; comunicación entre el nodo 1 y 8, después del proceso de descubrimiento de ruta.

Por otro lado, los nodos en una red inalámbrica se mueven; de hecho, ya que el movimiento de los nodos es constante, la red es inalámbrica. En la Figura 2.3, el nodo 1 se comunica con el nodo 8, por medio del nodo 3, 4 y 7. Cuando el nodo 4 se mueve hasta romper el enlace entre el nodo 3 o 7, se deberá iniciar un nuevo descubrimiento de ruta. Tras el nuevo proceso, habrá una sobrecarga de tráfico en la red. Al ser una topología de la red cambiante (debido al movimiento de los nodos) varios descubrimientos de ruta podrán ocurrir.

La Figura 2.4 muestra la misma red inalámbrica de la Figura 2.3, la diferencia es que se ha virtualizado los nodos. Una vez establecida la ruta desde 1 hacia 8, y dentro del proceso de envío de mensajes, el nodo 4 al moverse hacia otra posición, no afecta la ruta actual. Si además el nodo 5 se retira, la ruta aun permanecerá activa, debido

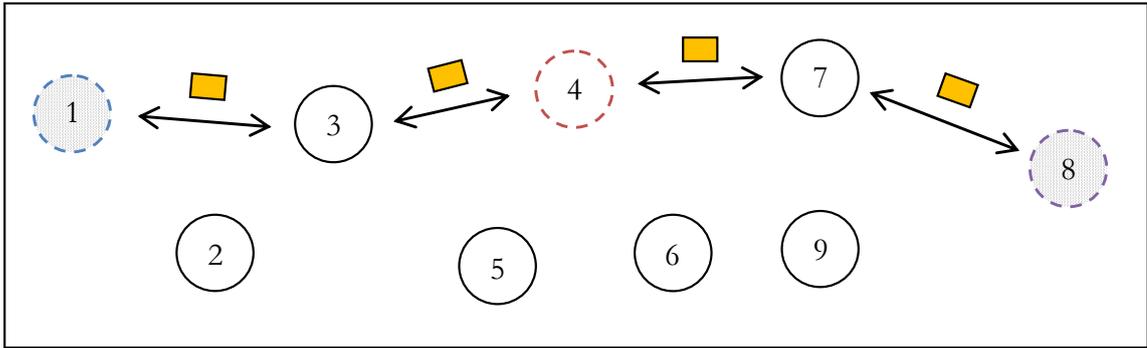


Figura 2.3.: Red inalámbrica; comunicación entre el nodo 1 y 8, después del proceso de descubrimiento de ruta.

a la presencia del nodo 6. Solamente cuando todos los nodos que emulan el nodo virtual abandonen la región, la ruta quedará inactiva.

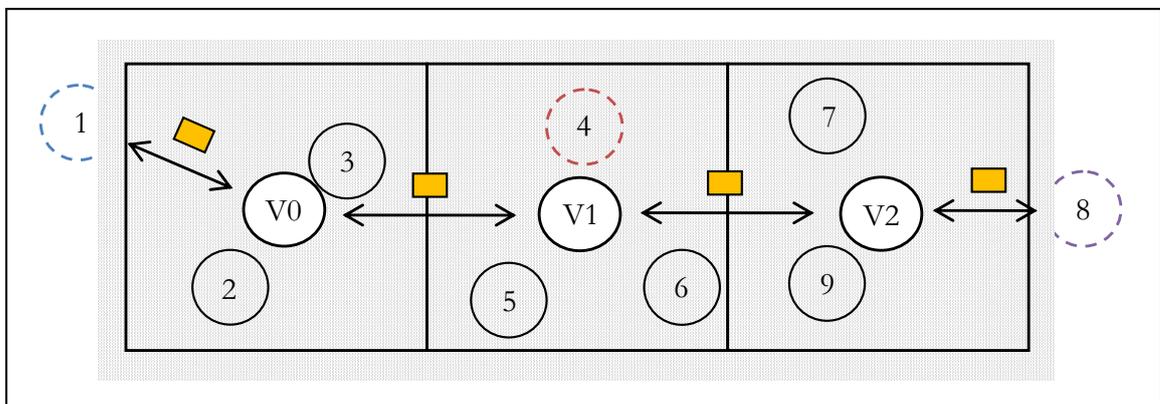


Figura 2.4.: Virtualización de nodos; el nodo 2 y 3 emula al nodo virtual V0, el nodo 4, 5 y 6 emula al nodo virtual V1, y el nodo 7 y 9 a V2. La comunicación entre 1 y 8 utiliza la ruta V0, V1 y V2.

La virtualización disminuye notablemente la sobrecarga de la red (debido al menor número de procesos de descubrimientos de ruta), los nodos no están limitados a permanecer fijos (ventaja importante de las redes inalámbricas), y las rutas permanecen constantes mayor tiempo.

El éxito de la virtualización consiste en: dividir la zona geográfica en regiones; conocer la posición, dirección de movimiento y velocidad del nodo; establecer un nodo líder en cada región; sincronizar las tablas de enrutamiento en todos o parte de los nodos de una región; y utilizar un protocolo de enrutamiento común en todos los nodos de la red.

3. Modelo de la Capa Virtual y Encaminamiento sobre Nodos Virtuales

La capa Virtual es una capa abstracta que permite mitigar muchos de los problemas de las redes MANET. En [28] la capa Virtual dentro del modelo TCP/IP está ubicada entre la capa de Internet y la capa de Acceso a la Red. Un nodo virtual –nodo abstracto– es creado por la capa Virtual y es emulado por varios nodos físicos. El nodo virtual permite desarrollar aplicaciones como si hubiese un servidor en una localización geográfica fija.

En el Capítulo 2 se dio a conocer algunas ventajas de los nodos virtuales, en este capítulo proponemos un modelo para la capa Virtual basado en los modelos definidos en [2] y [3]; planteamos ciertas modificaciones en el protocolo AODV para adaptarlo a la capa Virtual y creamos una aplicación para redes VANET.

3.1. Capa Virtual

A diferencia del modelo de capa Virtual propuesto en [28], nosotros proponemos que la capa Virtual sea añadida sobre la capa de Transporte y la capa de Internet (Figura 3.1). Así, la capa Virtual recibe los servicios de la capa de Internet; esta última añade el direccionamiento lógico (direcciones IP) a los mensajes enviados por la capa Virtual.

Los datos de la capa de Aplicación, son encapsulados en segmentos o datagramas en la capa de Transporte; la capa Virtual los encapsula en el PDU de esta capa, en esta tesis lo llamaremos “*datavirs*”, estos a su vez son encapsulados en paquetes en la capa de Internet, y la capa de Acceso a la Red los encapsula en tramas para ser enviadas por el medio. Los *datavirs* llevan la PDU de la capa de Transporte junto con el encabezado de la capa Virtual. El encabezado de un *datavirs* se muestra en la Figura 3.5.

Esta sección examina la arquitectura de la capa Virtual, referida a los elementos básicos que la conforman: división del espacio geográfico en regiones, localización de los nodos, mensajes entre nodos, elección del Líder, sincronización y elección de Backups.

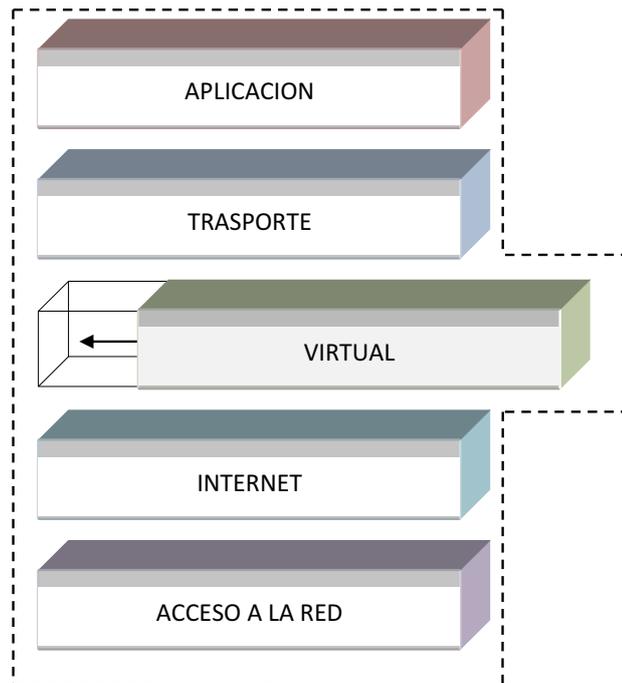


Figura 3.1.: Capa virtual dentro del modelo TCP/IP.

3.1.1. Regiones

Un nodo virtual es generado cuando se divide el espacio geográfico en regiones. La división es abstracta pero eficaz. Las redes celulares -haciendo una comparación- modelan la cobertura de cada nodo como una celda hexagonal, aunque en realidad presenten una forma amorfa. En los nodos virtuales la división de regiones es perfecta, la división no está limitada por la potencia de la señal emitida, sino por un algoritmo matemático basado en la posición geográfica del móvil[14].

Cuando la división de regiones no se basa en la limitación del alcance de la señal, sino por un algoritmo matemático, una buena consideración para determinar una forma geométrica adecuada, es la simplicidad del algoritmo. Otra consideración importante, es la eficiencia geométrica, esta se determina en función del desperdicio de la señal emitida de acuerdo a las condiciones de la Tabla 3.1 (después se analizará la necesidad de limitar la comunicación entre regiones vecinas).

El siguiente análisis determinará la eficiencia y la complejidad del algoritmo al dividir el espacio en regiones hexagonales y en regiones¹ rectangulares.

¹Se supone parámetros iguales (directividad, tipo de antena, potencia de la señal emitida, sensibilidad en el receptor, longitud de onda) en cada uno de los nodos.

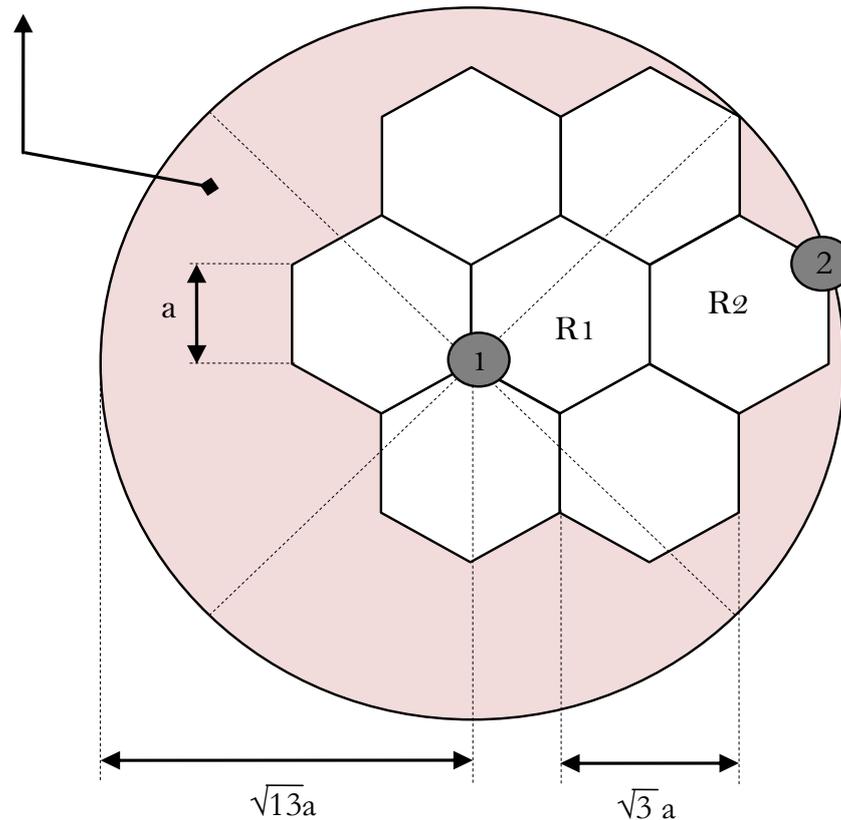
Cobertura Mínima \rightarrow R (hexagonal)

Figura 3.2.: División geográfica en regiones hexagonales regulares. El nodo 1 en la región R1, puede comunicarse con el nodo 2 de la región R2. La circunferencia muestra el alcance de la señal del nodo 1.

Condiciones:
Cada nodo en una región puede comunicarse con todas las regiones vecinas.
La antena en el móvil tiene una radiación omnidireccional.
Todos los móviles tienen el mismo alcance.

Cuadro 3.1.: Condiciones de la capa Virtual para la división de regiones.

Regiones hexagonales se muestran en la Figura 3.2, el nodo 1 en la región R1 tiene 6 regiones vecinas. Suponiendo el caso más crítico (el nodo 1 en el extremo de R1 y el nodo 2 en el extremo de R2) el nodo 1 debe ser capaz de comunicarse con el nodo 2. Cuando esto es así, y debido a la omnidireccionalidad, el nodo 1 emitirá una señal que llegará a regiones más lejanas, esto es considerado como el desperdicio de la señal. Tanto en regiones hexagonales como en rectangulares, los nodos deben tener un alcance mínimo para cumplir la primera condición de la Tabla 3.1.

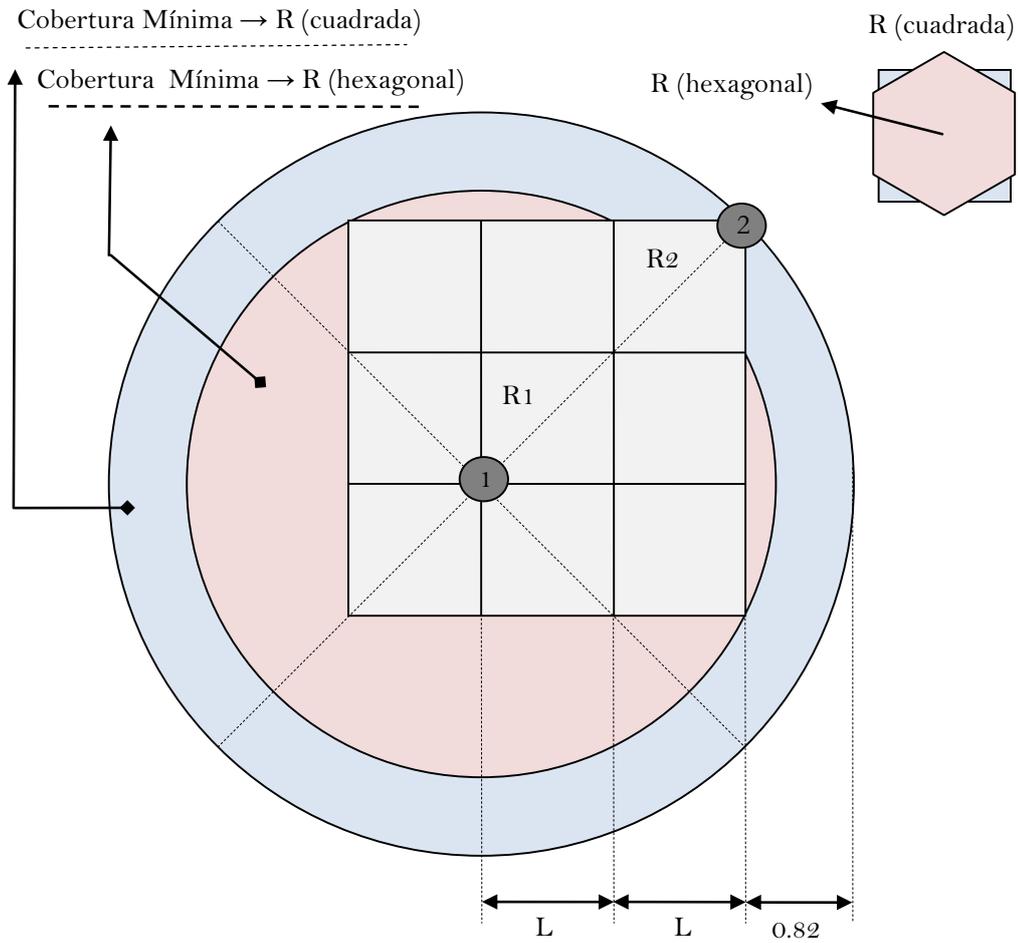


Figura 3.3.: El nodo 1 en la región R1 debe tener una cobertura mínima para alcanzar a su vecino en la región 2. La circunferencia mayor es la cobertura del nodo 1; la circunferencia menor representa la cobertura del nodo 1 de la Figura 3.2.

$$Alcance\ min_{R \rightarrow hexagonal} = \sqrt{13}a \quad (3.1)$$

$$\eta_{R \rightarrow hexagonal} = \frac{7(Area\ R)}{Area\ Alcance\ min} = 44\% \quad (3.2)$$

La Figura 3.3 compara la cobertura mínima requerida en regiones hexagonales y en rectangulares. Para que la relación tenga sentido, el área de R → hexagonal es igual

que $R \rightarrow$ rectangular (Ecuación 3.3)

$$\frac{3\sqrt{3}a^2}{2} = L^2 \quad (3.3)$$

$$Alcance\ min_{R \rightarrow rectangular} = 2.82L \quad (3.4)$$

$$\eta_{R \rightarrow rectangular} = \frac{9(Area\ R)}{Area\ Alcance\ min} = 36\% \quad (3.5)$$

De acuerdo a la Ecuación 3.3 y Ecuación 3.4 se desprende la Ecuación 3.6. Al utilizar regiones hexagonales, se necesita el 79% de la señal emitida (para cumplir las condiciones de la Tabla 3.1), que cuando se utiliza regiones rectangulares.

$$Alcance\ min_{R \rightarrow hexagonal} = 0.79 * Alcance\ min_{R \rightarrow rectangular} \quad (3.6)$$

La eficiencia para $R \rightarrow$ hexagonal es mayor (aunque no por mucho); sin embargo, la ventaja de las regiones rectángulas es la facilidad del algoritmo y por tanto el tiempo de procesamiento para que un nodo determine su región. En esta tesis utilizaremos regiones rectangulares.

3.1.2. Localización de Nodos

Un nodo determina la región en la que se encuentra conociendo su posición $P(x, y)$ dentro de una zona geográfica. Los parámetros de localización son: las dimensiones de la zona geográfica, el número de filas y número de columnas. Todos los nodos de la red, son configurados con parámetros iguales. Cada nodo dispone de un módulo de posicionamiento para obtener $P(x, y)$.

En la Figura 3.4, cuando el nodo 1 conoce $P(x, y)$ desarrolla un algoritmo rápido para conocer su región. Cada región es identificada en primera instancia como $R(xR, yR)$; mediante una transformación, cualquier región es determinada por un único identificador $R(id)$. El nodo 1 está ubicado en la región $R(1, 1) \rightarrow R(4)$.

$$xR = floor\left(\frac{x * columnas}{L_z}\right) \quad (3.7)$$

$$yR = \text{floor} \left(\frac{y}{A_z} \right) \quad (3.8)$$

$$id = xR + yR * \text{columns} \quad (3.9)$$

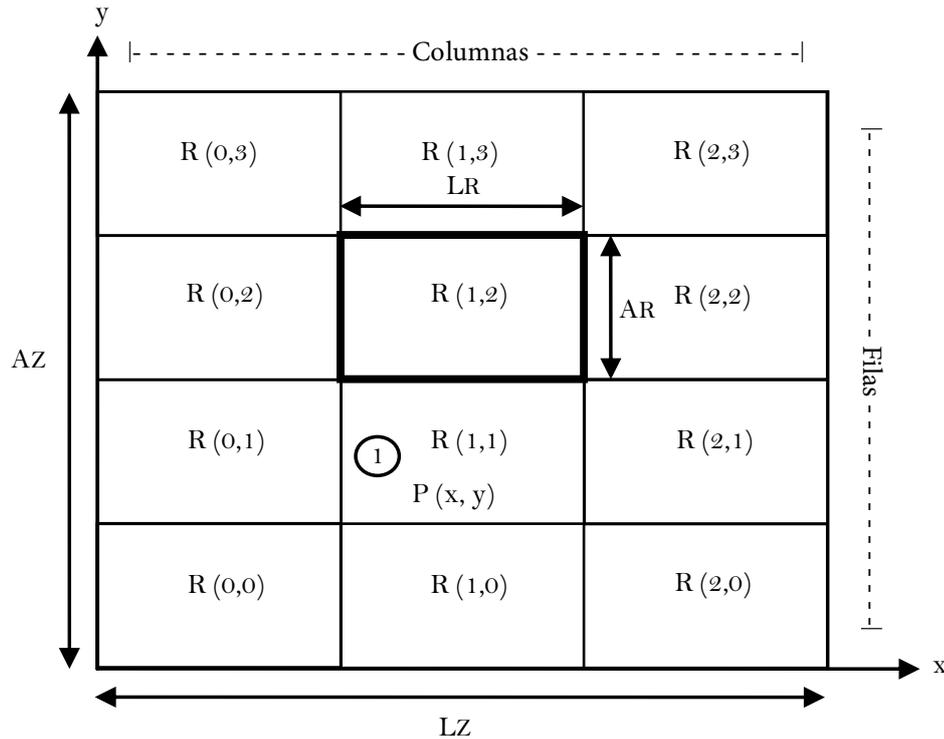


Figura 3.4.: Zona geográfica dividida en regiones rectangulares. El nodo 1 al conocer su posición $P(x,y)$ determinará que se encuentra en la región $R(1,1)$. El nodo 1 conoce su localización debido a los parámetros configurados.

Los nodos en una red VANET se mueven a velocidades entre 20km/h a 50km/h. Con la presencia de tráfico el movimiento puede reducir. Sin embargo, es necesario conocer la región actual del nodo en cada instante de tiempo. Utilizar el módulo de posicionamiento (MoPo) con alta frecuencia no es conveniente. Un algoritmo basado en la velocidad y dirección del móvil, reducen la frecuencia de utilización del MoPo. Con la velocidad y dirección, el móvil puede utilizar el MoPo, solo para verificar la entrada a una nueva región. Con un algoritmo robusto y parámetros precisos (velocidad y dirección), un móvil puede necesitar el uso del MoPo solo al momento de iniciar (encendido).

3.1.3. Estados

Varios o todos los nodos dentro de una región, simulan un nodo virtual. Uno de los objetivos principales de los nodos virtuales, es disminuir el tráfico generado por el protocolo de enrutamiento. De todos los nodos de la región, uno de ellos es elegido para formar parte de los procesos de descubrimiento de ruta y reenvío de mensajes, a este nodo se le denomina nodo “Líder”. Los nodos que no son Líderes, son denominados nodos “No-líder”.

Nodos No-lideres pueden ser elegidos Backup. Un Backup permite no depender del Líder; esto quiere decir, que si el Líder actual abandona la región, el Backup estará listo para reemplazar al Líder. Todo Backup debe estar sincronizado con el Líder. Solo unos pocos nodos No-Lideres serán elegidos Backup. En las siguientes secciones se describe el proceso para la elección del nodo Líder y los nodos Backup. Antes de conocer estos procesos, en la siguiente sección se detalla el tipo de mensajes de la capa Virtual.

3.1.4. Mensajes de la Capa Virtual

La capa Virtual además de encapsular segmentos o datagramas en datavirs, genera sus propios mensajes para interactuar con la capa Virtual de otros nodos. Cada mensaje² lleva los campos de la Figura 3.5.

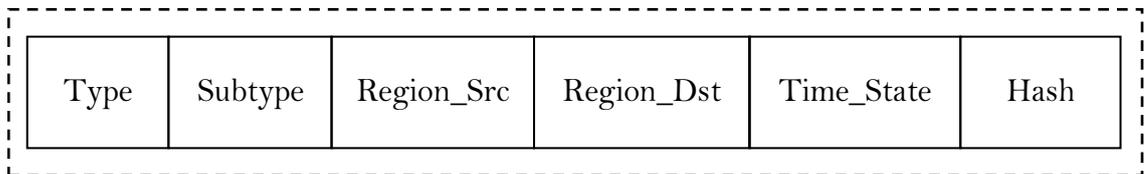


Figura 3.5.: Encabezado de los mensajes de la capa Virtual.

Los mensajes enviados entre nodos para la elección del Líder son:

- *LeaderRequest*
- *HeaderBeat*
- *LeaderLeft*
- *LeaderReply*

Los mensajes para la elección y sincronización de nodos Backup son:

- *SynRequest*
- *SynData*

²Antes del envío de cualquier tipo de mensaje, el nodo debe conocer su región.

- *SynAck*
- *BackupLeft*

Dos tipos de mensajes: *LeaderEleccion*, *Synchronization*; y siete subtipos.

3.1.5. Elección del Líder

Cuando un nodo inicia (comienza a funcionar) no conoce su región (estado *INIT*). Una vez determinada su región (estado *REQUEST*), cada nodo configura un timer (*TimerRequestWait*), establecido con un valor predeterminado junto con un valor aleatorio dentro de un rango. Cuando el *timer* expira (estado *LEADER*), el nodo envía un mensaje *Heartbeat*. En el campo “Region” coloca su región actual, todos los nodos que pertenezcan a la misma región escucharán el mensaje y los nodos de diferente región lo desecharán. En el campo *Time_State* coloca el tiempo en el que cambio su estado a Líder. Cuando un nodo en estado *REQUEST* escucha un *Heartbeat* (estado *NONLEADER*), lee el campo “Source” para determinar su Líder.

Un nodo No-líder espera escuchar mensajes *Heartbeat* dentro de un intervalo de tiempo, si no escucha dentro de este tiempo asume que el Líder abandonó la región (pasa a estado *REQUEST*). El Líder de la región, periódicamente envía mensajes *Heartbeat* para confirmar su liderazgo, el periodo siempre es menor que el intervalo de espera de un No-Líder. Cada vez que un No-Líder escucha un *Heartbeat* re-establece el tiempo de espera (*TimerHeartbeatWait*).

Cuando un mensaje unicast es enviado, la capa MAC del nodo destino confirma su recepción, si el nodo origen no recibe una confirmación, reenvía el mensaje. Cuando un mensaje broadcast es enviado, la capa MAC del nodo origen no espera recibir una confirmación. Todos los mensajes de tipo *LeaderEleccion* son enviados a manera de broadcast. En presencia de colisiones, los mensajes broadcast no serán reenviados, esto ocasiona pérdida de mensajes. Cuando hay pérdida de mensajes, el algoritmo expuesto es poco eficiente.

Enviar mensajes unicast supera este inconveniente, pero esto aumentaría el tráfico en la red (un nodo Líder debería enviar un *Heartbeat* a cada nodo No-Líder). Mensajes broadcast son necesarios en el proceso de elección de líder. Las consideraciones tomadas para mejorar el algoritmo en presencia de colisiones se detallan a continuación:

- Cuando un nodo cambia de estado, almacena el tiempo en el que se da la transición. El campo *Time_State* lleva este tiempo. Cuando ha expirado el temporizador (*TimerRequestWait*) un nodo en estado *REQUEST* envía un *Heartbeat*. Al enviar este mensaje, el nodo se considera Líder de la región y enviará *Heartbeats* periódicamente. Si un nodo en estado *REQUEST* en la misma región no escucha el *Heartbeat*, su *timer* expirará, se considerará Líder y enviará un *Heartbeat*. Dos líderes estarán presentes en una misma región. El

campo *Time_State* eliminada este problema. Cuando el Líder duplicado escuche un *Heartbeat* enviado por el primer líder, examinará el campo *Time_State*, como este es menor a su valor almacenado de transición a Líder, cambiará de estado a No-Líder.

- Un nodo en estado No-líder debido a una colisión, puede no escuchar el mensaje *Heartbeat* del Líder. Si esto sucede, el nodo No-líder podría pasar al estado *REQUEST* a pesar de que el nodo Líder está presente en la región. Para evitar esto, cuando no se escucha un *Heartbeat* el nodo No-líder cambia de estado a *UNSTABLE*³. En este estado, el nodo permanece a la espera de escuchar un *Heartbeat*, si lo escucha cambia de estado a No-Líder nuevamente; un máximo de 2 *Heartbeats* perdidos se admite antes de cambiar de estado a *REQUEST*.

Cuando el Líder abandona la región, los nodos No-lideres pueden estar a la espera de un *Heartbeat*, al no escucharlo debido a la ausencia del Líder, esperarán hasta tres intervalos del temporizador *TimerHeartbeatWait*. Al estar todo este tiempo en espera, gran cantidad de paquetes pueden ser perdidos o demorados debido a la ausencia de un Líder. Cuando hay tráfico en la red y movimiento constante de nodos, tiempos grandes de latencias en el envío de mensajes puede suceder. Una región debe elegir a su Líder en corto tiempo para mejorar el rendimiento del nodo Virtual. Consideraciones al respecto se detallan a continuación:

- Cuando el Líder abandona la región, envía un mensaje *LeaderLeft*. Un *LeaderLeft* alerta a los nodos de la región para comenzar un nuevo proceso de elección de Líder. Al escuchar un *LeaderLeft* los nodos No-Líderes cambian de estado a *REQUEST*. Si no todos los nodos de la región escucharon el *LeaderLeft* (debido a una colisión), los que hayan escuchado competirán por el liderato, el que ganó enviará un *Heartbeat*. Todos los nodos (incluso los que no escucharon el *LeaderLeft*) al escuchar el *Heartbeat* enviado por el nuevo líder, leerán el campo “*Source*” y cambiarán de Líder. De esta manera el nodo virtual se levantará rápidamente.
- Cada vez que un nodo cambia de región, cambia de estado a *REQUEST*. Al momento de entrar, enviará un mensaje *LeaderRequest*; si la región está liderada, el líder responderá con un mensaje *LeaderReply*. De esta forma, el nodo entrante cambiará de *REQUEST* a No-líder inmediatamente. Así mismo, si la región está vacía, el nodo entrante en corto tiempo tomará el liderato de la región.

En la Figura 3.6 se resume el proceso de elección de Líder en forma gráfica. El diagrama es un proceso condicional: dada una entrada se detalla la acción tomada por el nodo. En el diagrama, un nodo puede tomar cinco estados (*INIT*, *REQUEST*, *NON-LEADER*, *LEADER*, *UNSTABLE*); cambia de estado dependiendo de los mensajes recibidos, los temporizadores expirados o por los cambios de región. La Figura 3.6 toma en cuenta las consideraciones para mejorar la eficiencia del proceso de elección de Líder.

³En este estado, aún sigue considerando a su Líder actual como Líder

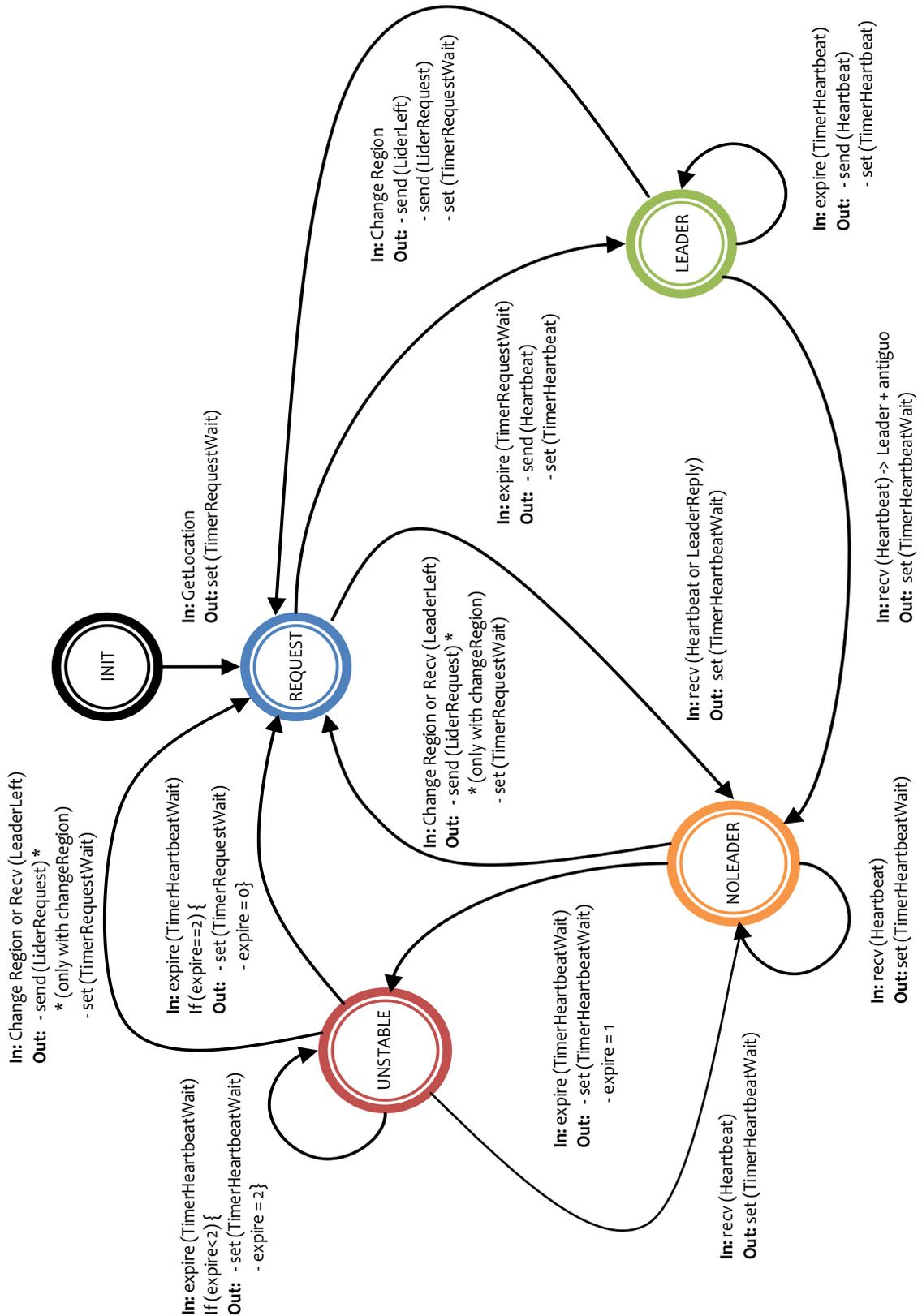


Figura 3.6.: Algoritmo para la elección del Líder. Se ha tomado en cuenta la posible pérdida de mensajes, así como la rapidez de conformación del nodo Virtual.

En [2] se propone un algoritmo basado en seis estados. El problema de un sexto estado (*UNKNOWN*) es el tiempo en que un nodo virtual llega a levantarse. Cuando un nodo cambia de región o recibe un *LeaderLeft*, cambia su estado a *UNKNOWN* y configura un temporizador para enviar un *LeaderRequest*, solo después de expirado el temporizador, el nodo cambia a estado *REQUEST*. En este estado configurará otro temporizador para convertirse en el líder de la región.

3.1.6. Elección de Backups

Cuando el Líder abandona la región, los nodos No-líderes elegirán un nuevo Líder para levantar el nodo virtual. Sin embargo, el Líder elegido debe estar sincronizado con el Líder anterior para continuar con el servicio; si esto no es así, el servicio del nodo virtual caerá y muchos paquetes se perderán. Es necesario entonces que cada región tenga nodos de respaldos⁴ para levantar el servicio tras la salida del nodo Líder. Si en una región hay varios nodos, no es necesario que todos sean Backups, pero es importante que haya un número apropiado.

En [2] se propone un algoritmo basado en probabilidad para que un nodo se considere Backup; si se desea que la mitad de los nodos de la región sean Backups, se establece una probabilidad del 50 %. Una gran cantidad de Backups puede aumentar el tráfico en la red, debido a mensajes de sincronización. Cuando este algoritmo se utiliza, una probabilidad del 30 % puede ser bueno, como demasiado malo (en el caso de que el 30 % constituya un número grande de Backups).

El algoritmo propuesto en esta tesis, considera un número máximo de Backups, así como un proceso de elección de Backups. Además, el nodo Líder conoce el número de nodos que lo están respaldando. El proceso de elección de Backups es el siguiente:

Después de la elección de Líder, la elección de Backups comenzará. Cada mensaje enviado por el Líder, lleva el número actual de Backups de la región. Un nodo No-líder compara este campo con el número máximo de Backups que debe tener una región, si el resultado es menor, establece un temporizador (*TimerRequestBackup*). Cuando este temporizador expira envía un mensaje *SynRequest* para solicitar ser Backup; cuando el Líder escucha el mensaje, aumenta en uno el número de Backups y responde con un *SynData*. Un *SynData* lleva los datos de sincronización.

Dentro del proceso de sincronización pueden colisionar mensajes, esto podría causar que un mensaje *SynData* se pierda. Al suceder esto, el nodo Líder supondrá que tiene un cierto número de Backup sincronizados, cuando en realidad no lo tiene; además ningún otro nodo solicitará ser Backup⁵. Para evitar este problema mensajes *SynAck* y una tabla de Backups es utilizada.

⁴Nodos de respaldo son llamados nodos Backup

⁵Si en los mensajes Heartbeats el campo que lleva el número de Backup es mayor al número máximo de Backups

La tabla Backups (*B_table*) está presente en todos los nodos, esta contiene la dirección IP del nodo Backup y su estado (sincronizado o no-sincronizado). Cuando el Líder de la región recibe un *SynRequest* coloca en su tabla de Backups al nodo origen del mensaje y lo establece como no-sincronizado, aumenta el número de Backups, envía un *SynData* y configura un temporizador (*TimerPurgeBackup*). Cuando el nodo que envió el *SynRequest* recibe el *SynData* se sincroniza y responde con un *SynAck* hacia el Líder. Al recibir el Líder un *SynAck*, recorre su tabla de Backups hasta encontrar una coincidencia con la fuente del mensaje, y coloca el estado del nodo Backup como sincronizado. Cuando el temporizador *TimerPurgeBackup* expira, el nodo Líder comprueba la sincronización de los nodos Backups; el número de Backups sincronizados será el número de Backups que envía en su próximo *Heartbeat*.

En [2] se da mayor prioridad de ser elegido Líder a un nodo Backup en estado sincronizado con respecto a nodos Backups no-sincronizados y frente a nodos clientes (no Backups); sin embargo no es posible dar prioridad entre Backups sincronizados. Es decir, los nodos sincronizados (después que el Líder abandono la región) establecerán un temporizador con un tiempo menor (más un valor aleatorio pequeño) que otros nodos de la región, para enviar un mensaje *LeaderRequest*; otros nodos establecerán un tiempo mayor.

Cuando hay un proceso de elección de Backups, es posible dar prioridad entre Backups sincronizados. Cuando un nodo No-líder escucha un mensaje *Heartbeat* con el campo “número de Backups” en cero, supondrá que es el primer Backup y tomará la prioridad entre todos los futuros Backups; así mismo, si un nodo lee el campo “numero de Backups” en uno, tendrá la siguiente prioridad. Por lo tanto, cuando el Líder abandone la región y envíe un *LeaderLeft*, el nodo Backup con la más alta prioridad, reclamará su liderazgo enviando un *Heartbeat*. El levantamiento del nodo virtual, tras la salida del Líder, con la presencia de nodos Backups sincronizados, es inmediata.

Un mensaje *BackupLeft* es enviado por un nodo Backup cuando abandona la región. Cuando este mensaje es recibido por el líder de la región, lo elimina de la tabla de Backups y disminuye el número de Backups en uno. Otros nodos Backup al escuchar el mensaje aumentan su prioridad⁶.

Cuando el Líder abandona la región, el Backup con más alta prioridad será elegido Líder. Cuando el nuevo líder envíe un *Heartbeat*, el proceso de elección y sincronización de Backups comenzará nuevamente, a pesar de que haya Backups del anterior líder completamente sincronizados. Esto podría aumentar el tráfico de la red innecesariamente. Para evitar esta sobrecarga, los mensajes *SynRequest* y *SynAck* son enviados en broadcast, para que todos los nodos de la región configuren (al igual que un líder) la tabla de Backups. Cuando esto se hace, al ser elegido un nodo Backup como Líder, tendrá el mismo número de Backups que el anterior Líder menos uno.

⁶Esto solo si el nodo Backup saliente, tenía más alta prioridad.

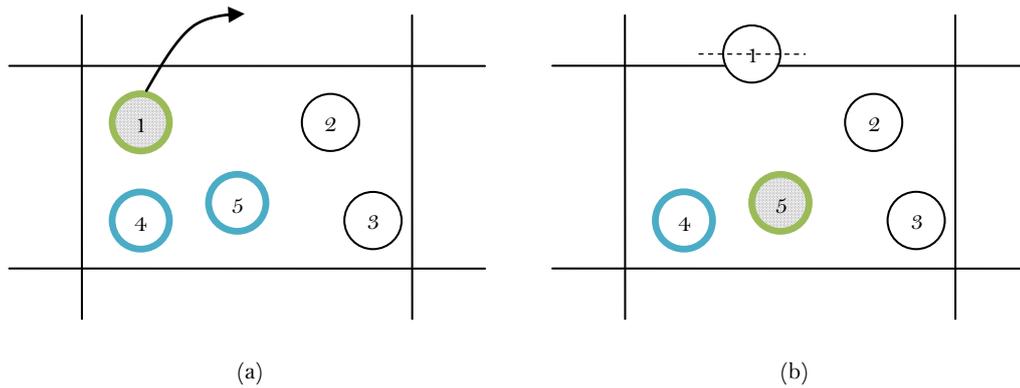


Figura 3.7.: Ilustración del proceso de elección de Backups. (a) El líder (nodo 1) está pronto a salir de la región, con dos Backups: el nodo 4 y 5. (b) El nodo 1 ya ha salido, el nodo virtual se ha levantado inmediatamente, estableciendo al nodo 5 como Líder, mientras que el nodo 4 permanece siendo Backup.

Los mensajes *SynData* son unicast, dirigidos al nodo fuente del mensaje *SynRequest*. Los mensajes unicast son más confiables (tienen menor probabilidad de ser perdidos) que los mensajes broadcast. Los mensajes *SynData* llevan información importante y puede ser crítica para mantener el nodo virtual levantado.

3.2. Encaminamiento VAODV

Los protocolos de enrutamiento trabajan sobre la capa de Internet del modelo TCP/IP o sobre la capa 3 de modelo OSI. En la Figura 3.1, la capa de Internet está por debajo de la capa Virtual; esto quiere decir, que el enrutamiento utilizado debe entender los datos enviados por la capa Virtual. Así como la capa de Enlace conoce el direccionamiento de la capa de Internet para realizar el mapeo con su propio direccionamiento; la capa de Internet debe saber interactuar con la capa Virtual. Para esto, ciertos cambios deben ser realizados en el protocolo de enrutamiento.

Para esta tesis, utilizaremos el núcleo del protocolo de enrutamiento AODV. Al igual que en [2] AODV fue modificado en ciertos aspectos, pero mantiene sus principales características. En el resto de esta tesis, el protocolo AODV modificado, será llamado VAODV.

Como se explicó en la Subsección 2.3.3, AODV es un protocolo de enrutamiento reactivo; esto quiere decir, que un proceso de descubrimiento de ruta se ejecutará solamente cuando se desee transmitir paquetes. Al ser un protocolo reactivo, poco tráfico se genera en la red.

La tabla de enrutamiento (R_table) contiene los mismos campos que AODV, pero se ha añadido el campo *next_region*. La capa Virtual informa a la capa de Red la región y el estado del nodo. La región es adicionada en los mensajes *REQUEST*,

REPLY, *ERROR* y *HELLO*. VAODV tiene tres procesos básicos: descubrimiento de ruta, mantenimiento de ruta y envío de mensajes.

Una tabla de líderes (*L_table*) es mantenida en cada nodo. Cada entrada en la *L_table* contiene: dirección del Líder, región y *time_expire*⁷. La *L_table* es actualizada cada vez que se escucha un *Heartbeat*. La depuración de la *L_table* es hecha cada cierto tiempo, para eliminar líderes antiguos (*time_expire* menor al tiempo actual). Cuando un nodo escucha un *Heartbeat*, primero busca en su *L_table* una región coincidente, si lo hay, actualiza su tabla, caso contrario, crea una nueva entrada.

VAODV trabaja sobre 802.11. Para detectar fallos de enlace, VAODV recibe mensajes de la capa MAC. VAODV envía mensajes Hello periódicamente, pero no tienen el mismo propósito que en AODV (su propósito se explicará en la siguiente sección).

3.2.1. Mensajes Hello

Los mensajes *Hello* al igual que en AODV no son utilizados para crear rutas⁸. En la Subsección 2.3.3 se explicó que los mensajes *Hello*, son una alternativa para detectar fallos de enlace, cuando no es posible recibir alerta de fallos de la capa MAC. VAODV recibe alertas de la capa de Acceso a la Red y además utiliza mensajes *Hello*.

Los mensajes Hello son enviados para que los nodos de una misma región se conozcan. En la Figura 3.8 el nodo 1 conocerá que su nodo vecino es 2, y 2 conocerá sobre 1; así mismo, entre los nodos 3 y 4, 5 y 6, 7 y 8. Los mensajes *Hello* son enviados periódicamente; el tiempo de envío no es crítico, puesto que estos no son utilizados para detectar fallos de enlace. De esta manera, la desventaja principal de los mensajes *Hello* es eliminada.

Los mensajes Hello evitan el envío de *REQUEST* innecesarios. Si un mensaje *REQUEST* tiene como destino un nodo de la misma región, el Líder no replicará el mensaje; porque sabe que si él escucha, el nodo destino también lo hará⁹. Si un Líder replicara el mensaje, el tráfico y el número de hops podrían incrementar.

Evitar *REQUEST* innecesarios a costa de enviar mensajes *Hello* periódicos no parece ser tan útil; sin embargo, esta no es la función específica de los mensajes Hello, solo es una forma de aprovecharlos. Una aplicación para redes VANET será analizada posteriormente; los mensajes *Hello* son la base de esta aplicación.

⁷El *time_expire* de una *L_table* es establecido a $2.5 * (\text{tiempo de envío de } \textit{Heartbeat})$, esto permite como máximo un *Heartbeat* perdido.

⁸Si los mensajes *Hello* crearan rutas, AODV no sería un protocolo reactivo.

⁹El nodo Líder considera que todos los mensajes recibidos en broadcast, también será escuchado por todos los nodos de la región; esta consideración es válida, porque se ha limitado la comunicación entre regiones vecinas.

3.2.2. Descubrimiento de Ruta

Cuando un nodo desea transmitir un mensaje hacia un destino (suponiendo que su R_table está vacía), inicia un descubrimiento de ruta. El nodo origen enviará un mensaje *REQUEST* en broadcast; los nodos que reciban el mensaje, aprenderán la ruta hacia el origen (incluida su región). Únicamente los nodos Líderes (en regiones distintas a la región del origen) replicarán el mensaje. Al igual que en AODV, un identificador de broadcast es utilizado para evitar el envío repetido de un mismo *REQUEST*.

Cuando en nodo Líder reenvía un *REQUEST*, modifica la región origen y añade los datos de la capa de Red (dirección IP origen). Cuando el mensaje *REQUEST* llega al destino, este crea una ruta hacia el origen de la misma forma que en AODV, pero además completa el campo *next_region* con la región del *next_hop*.

Los mensajes *REPLY* son enviados en broadcast; se generan cuando un mensaje *REQUEST* llega al destino o cuando un nodo intermedio tiene una ruta actual hacia el destino. Al enviar un *REPLY*, el nodo consulta la ruta hacia el origen del *REQUEST* y envía hacia el *next_hop* de la ruta. Un *REPLY* será escuchado por los nodos dentro de la región del *next_hop*, pero solo el Líder reenviará el *REPLY*. Debido a que los mensajes *REQUEST* y *REPLY* son enviados en broadcast, los nodos Backups aprenderán tanto la ruta inversa como la ruta directa.

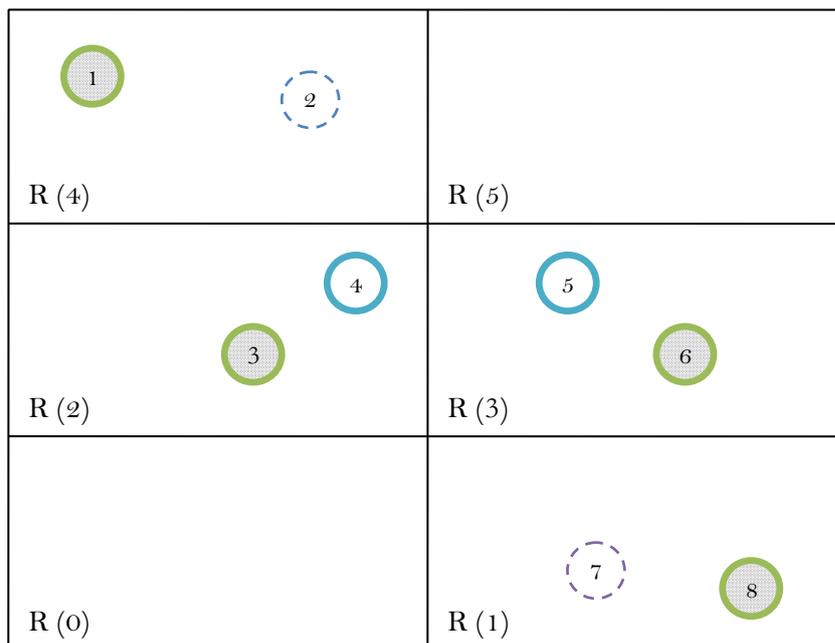


Figura 3.8.: Descubrimiento de ruta para el envío de mensajes desde el nodo 2 hacia el nodo 7. Las regiones 1, 2, 3 y 4 han elegido su líder (nodo 8, 3, 6, 1 respectivamente). El nodo 4 y 5 son Backups de su región.

En la Figura 3.8, cuando el nodo 2 desea enviar un mensaje hacia el nodo 7, envía un mensaje *REQUEST*. Las regiones vecinas al nodo 7, escucharán el *REQUEST*; los nodos 3, 4, 5 y 6 aprenderán la ruta inversa, pero solo los nodos 3, 6 replicarán el mensaje. El nodo 1 no reenviará el mensaje puesto que está en la misma región que el nodo solicitante. Después que el mensaje *REQUEST* sea replicado por los nodos Líderes de la región 2 y 3, el mensaje será escuchado por los nodos 7 y 8. El nodo 8 debido a que está en la misma región que el nodo 7, no responderá con un *REQUEST* ni con un *REPLY*. El nodo 7 al ser el destino del *REQUEST* responderá con la ruta recién aprendida hacia el nodo 2.

El nodo 7 colocará en el campo *next_hop* y *next_region* la dirección IP y la región del nodo del cual escucho primero el *REQUEST*. Suponiendo que el *REQUEST* del nodo 6 llego primero hacia la región 1, el *next_hop* y *next_region* en 7 hacia 2, será el nodo 6 y región 3. El mensaje *REPLY* de 7, se podrá escuchar en todas las regiones vecinas, pero solamente el líder de la región 3 replicará el mensaje. Cuando el nodo 2 escuche el *REPLY* del nodo 6, aprenderá la ruta hacia 8, cuyo *next_hop* y *next_region* serán 6 y 3 respectivamente.

3.2.3. Mantenimiento de Rutas

Cada nodo realiza el mantenimiento de su *R_table* de manera independientemente. Toda ruta dentro de su tabla, contiene el campo “*state*” para conocer el estado de la ruta. Una ruta es válida cuando su estado es *RTF_UP*; no es válida cuando su estado es *RTF_DOWN*; y está en reparación en *RTF_REPAIR*. Cuando una ruta es recién aprendida, se establece el *time_expire* por primera vez y se levanta la ruta (*RTF_UP*). Dentro de cada nodo se realiza un proceso periódico de depuración de rutas; cuando el tiempo actual es mayor al *time_expire*, la ruta es invalidada (*RTF_DOWN*).

Cuando una ruta es utilizada para el envío de mensajes, el *time_expire* es actualizado. Si la capa MAC no recibe una confirmación del destinatario, reporta una falla de enlace¹⁰; el nodo deberá tomar la decisión de realizar una reparación local o enviar un mensaje de *ERROR*. Si el fallo de enlace es detectado en un nodo más cercano al destino que al nodo origen del mensaje, se realiza una reparación local y la ruta toma el estado *RTF_REPAIR*. Si un nodo recibe un mensaje que debe enviar hacia un destino, cuya ruta está en reparación, el mensaje es almacenado en cola hasta que termine la reparación. Si una reparación no tuvo éxito, un mensaje de *ERROR* es enviado en broadcast. Los mensajes de *ERROR* en VAODV son manejados igual que en AODV.

¹⁰Una falla de enlace puede ser detectada, cuando el nodo destino está fuera del área de cobertura o debido a varias colisiones al intentar enviar el mensaje.

3.2.3.1. Correcciones de Ruta

Cuando un nodo recibe un mensaje destinado hacia él, pero hacia a una región diferente a su región actual, envía un mensaje *REPLY* en broadcast sin ningún destinatario y con el campo TTL=1. Los nodos que escuchen el *REPLY* corregirán la ruta sin reenviar el mensaje. En la Figura 3.8, si el nodo 7 se mueve a la región 3, un mensaje *REPLY* será escuchado en las regiones vecinas incluyendo al nodo 2. Cuando el nodo 2 corrige la ruta, los nuevos mensajes serán enviados directamente hacia 7. En AODV, aunque el nodo 7 se mueva muy cerca de 2, el mismo camino generado en el descubrimiento de ruta, se seguirá utilizando. Una corrección de ruta debe tomar efecto en la trayectoria, inversa y directa; si esto no sucede, el envío de mensajes entre dos nodos podría tomar rutas diferentes. En el caso de la Figura 3.8, cuando el nodo 2 envíe el mensaje hacia 7, este aun tendrá dos saltos hacia 2. Para resolver este problema, el nodo 7 debe también corregir la ruta, cuando escuche el mensaje del nodo 2. La corrección de rutas eventualmente reduce el número de saltos; mientras menos saltos haya en una ruta, la latencia disminuirá y el envío de mensajes será mayor.

3.2.3.2. Soft Handoff

En las redes celulares, cuando un móvil se mueve entre diferentes celdas mientras una conversación está en proceso, la operación de *Handoff* (traspaso) es necesaria. *Soft Handoff* (traspaso suave) es la habilidad de seleccionar entre varias señales recibidas de distintas estaciones base[29]. En[30] el proceso de *Handoff* es definido como un elemento clave para proveer Calidad de Servicio (QoS) y suportar la movilidad de los usuarios.

El concepto de *Soft Handoff* es utilizado para el mantenimiento¹¹ de rutas. VAODV utiliza el campo *next_region* para mantener la ruta hacia los nodos virtuales y no hacia los nodos físicos; esta característica hace a VAODV independiente del movimiento de los nodos Líderes. Bajo esta condición, si el *next_hop* de la ruta se mueve de región, el nodo seguirá enviando el mensaje a la misma región especificada por el campo *next_region* de su tabla. Como los mensajes son enviados en unicast, el campo *next_hop* debe cambiar; para este proceso se utiliza la operación de *Soft Handoff*.

En el proceso de descubrimiento de ruta, puede haber varios saltos entre el origen y el destino. La capa Virtual asegura que todos los nodos intermedios en una ruta, serán nodos Líderes. Dentro de un proceso de envío de mensajes, un nodo intermedio en la ruta puede abandonar la región; si en dicha región existe un nodo Backup sincronizado, un *Soft Handoff* entre este último y el nodo saliente tomará efecto.

El *Soft Handoff* permite el cambio de *next_hop* sin pérdida de paquetes, sin el envío mensajes de *ERROR* y sin reparaciones locales. Los mensajes de *ERROR* son

¹¹*Soft Handoff* mantiene las rutas sobre nodos virtuales, no sobre nodos físicos

generados, o si el nodo Backup no tiene su R_table sincronizada o si su tabla ARP (ARP_table) no está actualizada. La pérdida de paquetes es consecuencia directa de la generación de los mensajes de $ERROR$.

La Figura 3.9 detalla el envío de mensajes en el proceso de *Soft Handoff*. En la Figura 3.8, suponiendo que la ruta formada desde el nodo 2 hacia el nodo 7, es 2, 6 y 7 (el nodo 6 es el $next_hop$ para 7 en 2, y 6 es el $next_hop$ para 2 en 7); el nodo 5 es Backup de la región 3 y está sincronizado. Se supone que los nodos 2 y 7 están en un proceso de transferencia de información. Cuando el nodo 6 abandone la región, el mensaje *LeaderLeft* hará que el nodo 5 tome el liderato de la región.

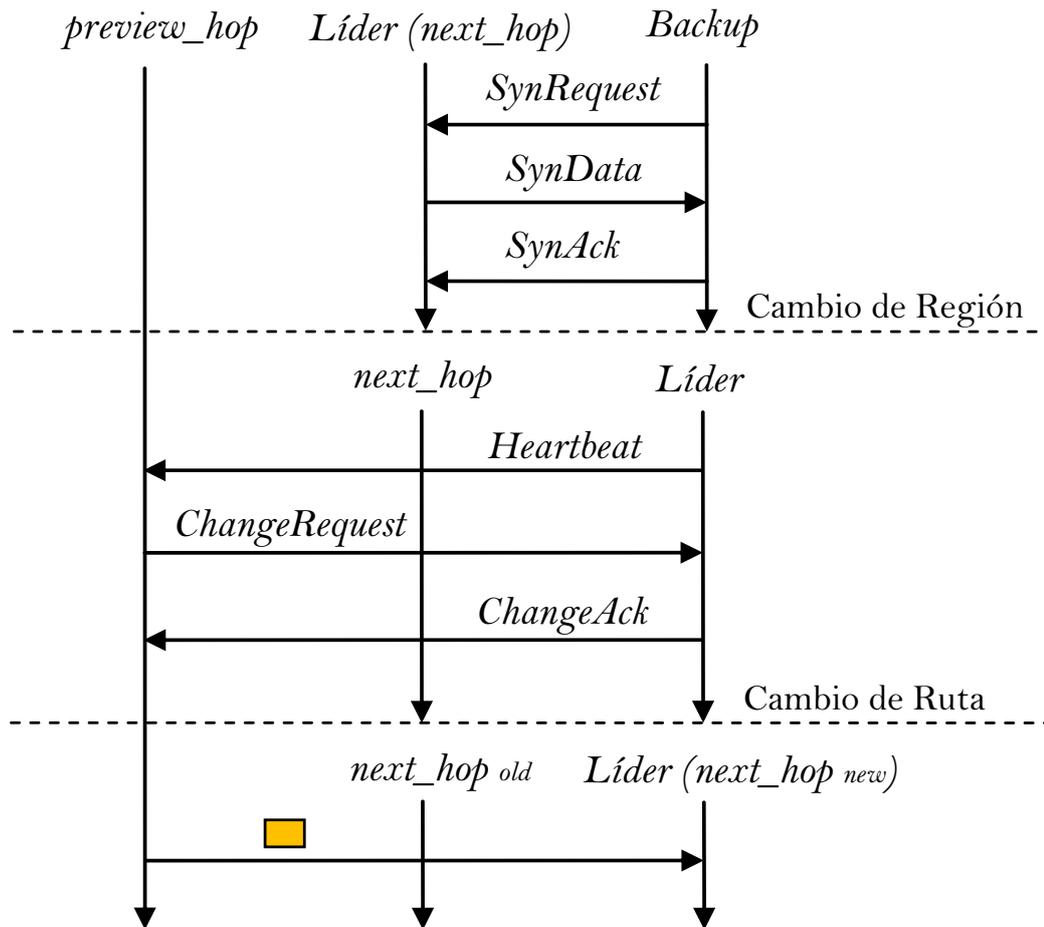


Figura 3.9.: Proceso de *Soft Handoff*. Antes del cambio de ruta el *preview_hop* seguirá enviando los mensajes hacia el *next_hop* antiguo. Solamente después del mensaje *ChangeAck* los paquetes se enviarán al *next_hop* nuevo. Se supone que el nodo Backup se sincronizó, antes del cambio de región del *next_hop* antiguo.

Todos los nodos al escuchar el *Heartbeat* del nodo 5, actualizarán su L_table . Cuando el nodo 2 o el nodo 7 envíen un mensaje hacia la región 3, en su R_table aun

tendrán como *next_hop* el nodo 6. Antes de enviar el mensaje, comprobarán en su *L_table* si el Líder de la región 3 es igual al *next_hop*; como no lo es, enviarán un mensaje dirigido al nodo 5 solicitando un cambio de ruta. El nodo 5 al escuchar el *ChangeRequest* comprobará si tiene rutas validas hacia los destinos solicitados. Si el *ChangeRequest* recibido es del nodo 2, el nodo 5 comprobará si tiene una ruta valida hacia 7; si lo tiene, enviará un *ChangeAck*. El mismo procedimiento se hará entre el nodo 5 y 7. Cuando el nodo 2 y 7 reciban el *ChangeAck* de 5, cambiarán el *next_hop* (de 6 a 5) de la *R_table*.

Durante el proceso de *Soft Handoff* los mensajes entre 2 y 7, seguirán enviándose al nodo 6 (este aún puede alcanzar a los nodos 2 y 7 cuando recién abandona la región). Debido a esto ningún paquete es perdido en el proceso. En AODV cuando un nodo intermedio se mueve hasta romper el enlace, se intentará hacer una reparación local o se enviarán mensajes de *ERROR*; consecuencia de ello, varios paquetes se perderán.

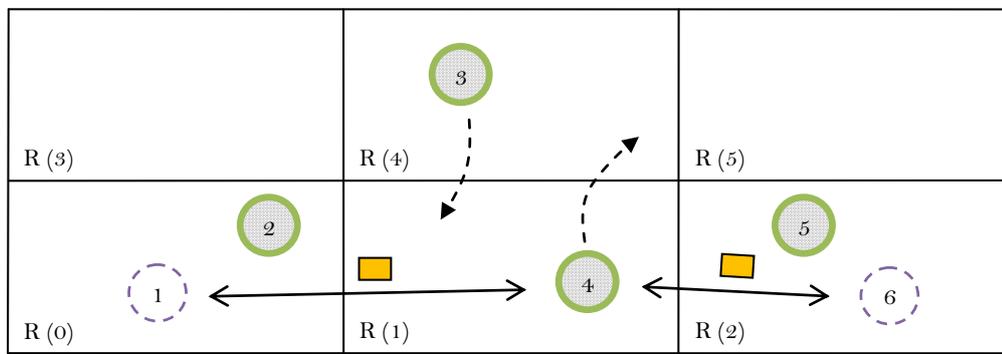


Figura 3.10.: Ilustración de la operación *Soft Handoff* sin *Backups* en la región. El proceso de descubrimiento de ruta ya se ha realizado. Los nodos 1 y 6 están comunicándose constantemente en forma bidireccional.

Un *ChangeRequest* lleva todas las rutas cuyos campos *next_region* y *next_hop* no coinciden con los campos región y dirección de origen del mensaje *Heartbeat* escuchado. El mensaje *ChangeAck* lleva todas las rutas activas referidas a las rutas solicitadas por el *ChangeRequest*. Si un nodo no recibe un *ChangeAck*, la misma ruta se mantendrá y no se volverá a enviar un *ChangeRequest*. En la Figura 3.10, se ha establecido la comunicación entre el nodo 1 y 6, en este momento el nodo 4 (líder de la región 1) abandona la región. El nodo virtual de la región 1 cae debido a la ausencia de nodos físicos; después de un momento el nodo 3 arribará a la región 1 y se convertirá en su nuevo Líder. Cuando se realice el proceso de *Soft Handoff* el nodo 3 no responderá con un *ChangeAck*, debido a que no tiene rutas hacia los destinos solicitados. El nodo 1 y el nodo 6 seguirán enviado paquetes por medio del nodo 4 hasta que el enlace se rompa¹².

¹²Si este mismo escenario sucediera en AODV, los resultados serían iguales.

Los mensajes *ChangeRequest* y *ChangeAck* tienen dos funciones importantes: la primera es asegurar que el nuevo *next_hop* tenga una ruta válida hacia el destino solicitado en el mensaje *ChangeRequest*; y la segunda es actualizar la *ARP_table*.

3.2.4. Envío de Mensajes

Después del proceso de descubrimiento de ruta, cuando en mensaje se desea enviar hacia un destino, se consulta el siguiente salto (*next_hop*) en la tabla de enrutamiento; los mensajes serán enviados hacia al *next_hop*. Cada vez que un nodo envía o reenvía un paquete, añade su región y su dirección IP (en el campo *preview_hop*), esto permite realizar correcciones de ruta (así se resuelve el problema planteado en la Subsubsección 3.2.3.1). Cuando un nodo utiliza una ruta para enviar un mensaje, se ejecuta un mantenimiento de ruta.

3.2.5. Pérdida de Paquetes

Tanto en AODV como en VAODV se pierden paquetes en tres casos:

- Cuando un nodo recibe un mensaje hacia un destino, y si al consultar en su *R_table* no tiene una ruta, enviará un mensaje de *ERROR* y eliminará el paquete
- Cuando un nodo dispone en su *R_table* una ruta válida, pero su *ARP_table* no está actualizada, el nodo enviará una solicitud ARP y colocará el paquete en un buffer. El buffer ARP solo puede almacenar un paquete por cada destino; esto quiere decir, que si otro paquete debe ser enviado al mismo destino, el anterior paquete puesto en el buffer será eliminado. En la presencia de alto tráfico, varios paquetes serán eliminados debido a este problema.
- Cada nodo posee una cola para el envío de mensajes; los mensajes son puestos en cola después del mapeo entre las direcciones lógicas y físicas. Cuando un nodo detecta una ruptura de enlace tras el envío de un paquete, el nodo consulta la dirección destino de este último para eliminar todos los mensajes de su cola, dirigidos hacia ese mismo destino. En la Figura 3.11 (después de haberse formado la ruta entre 1 y 4), el nodo 1 ha enviado varios paquetes hacia el *next_hop* (nodo 2), este ha procesado los paquetes y ha establecido a 3 como el *next_hop*; 2 ya ha enviado algunos paquetes hacia 3 pero otros aún están en cola. El nodo 3 ha enviado unos pocos mensajes hacia el destino 4; cuando se mueve hacia la región 10, aún tiene paquetes en cola hacia 4. Cuando se rompe el enlace entre 2 y 3, y entre 3 y 4, todos los mensajes en cola son eliminados, puesto que estos ya habían sido procesados para ser enviados a un nodo ahora inalcanzable. Si entre el nodo 2 y 1, hubiese más nodos intermedios, más paquetes serían eliminados. Si los paquetes son transportados por TCP, el nodo 1 deberá volver a enviar todos los paquetes perdidos nuevamente.

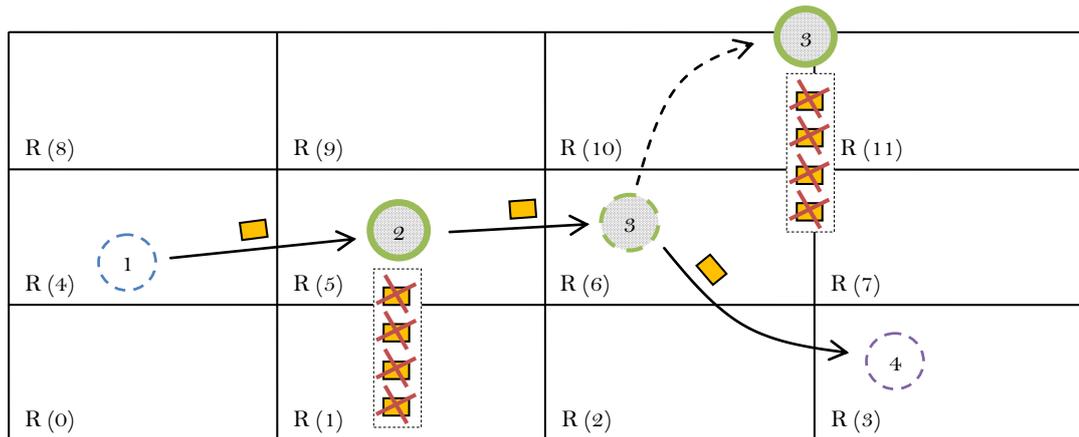


Figura 3.11.: Eliminación de paquetes debido a una ruptura de enlace. La falla de enlace se da entre el nodo 2 y 3 y entre 3 y 4. El nodo 1 ha enviado varios paquetes destinados hacia 4.

VAODV mediante la sincronización de rutas, evita el primer problema. El segundo problema podría ocurrir en una corrección de ruta. El último y más grave problema, podría suceder por la ausencia de Backups. *Soft Handoff* con nodos Backup sincronizados, evita todos estos problemas. AODV está expuesto al tercer problema todo el tiempo.

3.2.6. Estado de Sincronización

La sincronización de un nodo Virtual es fundamental. En un proceso de descubrimiento de ruta, si una región tiene nodos *Backups*, estos también aprenderán la ruta. En un nodo Backup, una ruta puede expirar incluso si aún es una ruta válida en su nodo virtual local; esto se debe a que los mensajes son enviados en unicast. Un Backup enviará una solicitud de sincronización cada vez que una ruta activa está próxima a expirar; el líder responderá enviando las últimas rutas utilizadas.

3.2.7. Restricciones

AODV restringe la comunicación entre regiones lejanas. Todos los nodos pueden determinar si un mensaje recibido es de una región vecina o una lejana. Para evitar que nodos de regiones lejanas se comuniquen entre sí, AODV descarta todos paquetes (*REQUEST* y *REPLY*), cuya región origen no sea una región vecina. Las razones para esta limitación son las siguientes:

- La comunicación se garantiza entre regiones vecinas; en regiones lejanas¹³ existe la posibilidad de ruptura del enlace. Por ejemplo, si el *next_hop* de una ruta,

¹³Región lejana es referida a una región no vecina

es un Líder de una región lejana, si esta abandona la región, un *Soft Handoff* entre el nuevo Líder no podrá garantizar (el nodo *Backup* del líder saliente, puede estar fuera del alcance del *preview_hop*).

- Cuando un nodo cliente envía un *REQUEST*, su nodo virtual local no replicará el mensaje, porque lo considera innecesario. Así mismo, si un nodo emulando al Líder de una región, escucha un *REQUEST* para un nodo de su misma región y conoce su ruta, no responderá con un *REPLY*. Esto evita *REQUEST* y *REPLY* innecesarios, pues al hacer la limitación entre regiones vecinas, todos los nodos tienen el mismo alcance.

3.2.8. Desventajas de VAODV

VAODV tiene ventajas significativas sobre AODV (serán analizadas en el siguiente capítulo); sin embargo, también presenta desventajas.

En la sección 3.1.1 se analizó el alcance mínimo de un nodo, para garantizar la comunicación entre sus regiones vecinas; para obtener esta distancia se consideró el caso más crítico. La distancia obtenida cumple la condición 1 de la tabla 3.1, pero con dicho alcance, la señal de un nodo puede llegar a regiones más lejanas. VAODV limita la comunicación entre regiones vecinas; esto quiere decir, que si un nodo escucha un *REQUEST* de una región lejana, descartará el mensaje. En la Figura 3.10, si el nodo 6 puede escuchar el *REQUEST* del nodo 1, no responderá con un *REPLY*; desechará el mensaje.

Una corrección de ruta puede en ciertos casos, involucrar pérdida de paquetes¹⁴. Cuando un nodo envía un *REPLY* para corregir la ruta, un nodo que no conozca la dirección MAC del nodo generador del *REPLY*, puede escuchar el mensaje y corregir la ruta. Cuando un mensaje se desee enviar por la ruta recién corregida, la capa MAC deberá enviar una solicitud ARP en broadcast, para conocer la dirección física del siguiente salto. Antes de recibir respuesta sobre la dirección MAC, varios paquetes pueden requerir ser enviados por la misma ruta; la capa MAC reportará errores de envío por cada uno de esos paquetes. Cuando VAODV reciba errores de la capa MAC, descartará todos los paquetes en su cola, inhabilitará la ruta, y enviará mensajes de *ERROR*. Los nodos que reciban el mensaje de *ERROR*, verificarán en su cola si tienen mensajes para el destino ahora inalcanzable y descartarán todos los paquetes; tras este proceso, un descubrimiento de ruta comenzará nuevamente.

En la Figura 3.12 la ruta desde el nodo 1 hacia el nodo 4 ha sido formada a través de dos saltos (nodo 2 y 3). Mientras el nodo 4 aún está en la región 3, múltiples mensajes hacia el nodo 2 (con destino 4) han sido enviados por el nodo 1; pocos de estos mensajes han sido enviados por el nodo 2 hacia 3 y varios aún están en cola. Cuando el nodo 4 se mueve a la región 2, tanto el nodo 2 como el nodo 3 escucharán el mensaje *REPLY*; los dos nodos corregirán la ruta hacia 4. Cuando el nodo 2 envíe

¹⁴En una comunicación con reducida tasa de bits (bps) esto no sucedería.

el próximo mensaje de su cola (con destino 4), enviará directamente al destino, pero la capa MAC desconoce la dirección física de 4, por lo que enviará una solicitud ARP. Antes que el nodo 4 escuche la solicitud, el nodo 2 nuevamente deseará enviar más paquetes hacia el nodo 4; sin embargo, aún desconoce la dirección MAC de 4. Llegará un momento en que el protocolo de enrutamiento considere inalcanzable al nodo 4 y descartará todos los paquetes de su cola; también desplegará un mensaje de *ERROR*. Este descarte de paquetes no sucedería si el tiempo en que el nodo 4 respondiese a la solicitud ARP, fuese menor al tiempo, en que el nodo 2 desea enviar los siguientes paquetes de su cola (con destino 4).

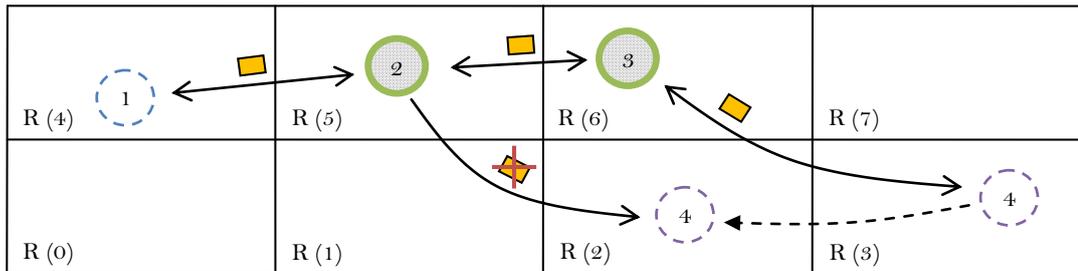


Figura 3.12.: Descarte de paquetes en un proceso de corrección ruta, debido a que el nodo 2 tiene varios paquetes en su cola hacia el nodo 4 y no conoce su dirección física.

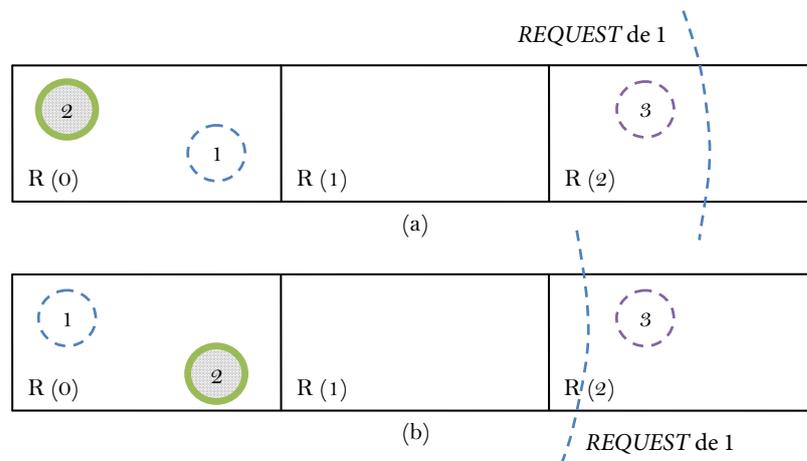


Figura 3.13.: Desventajas de VAODV debido a la limitación de comunicación entre regiones vecinas. (a) El nodo 3 escucha el *REQUEST* de 1 pero lo descarta. (b) El *REQUEST* de 1 no llega al nodo 3; el nodo 2 al replicar al mensaje podría alcanzar a 3, pero 2 descarta el mensaje.

Limitar la comunicación entre regiones vecinas conlleva ciertos problemas. En la Figura 3.13 (a), pese a que el nodo 3 escucha el *REQUEST* del nodo 1, no responde con un *REPLY*, pues examina la región de origen del *REQUEST* y comprueba que

no es una región vecina. En la Figura 3.13 (b), el *REQUEST* enviado por 1 no alcanza al nodo 3; el nodo 2 escucha la solicitud, pero considera innecesario replicar el mensaje, pues supone que tendrá el mismo alcance que el nodo 1. Si el nodo 2 replicara el *REQUEST*, alcanzaría al destino; pero aun así, el nodo 3 descartaría el mensaje. En ambos casos, AODV sí permitiría la comunicación.

En VAODV el mantenimiento de las rutas es complejo e involucra un aumento de tráfico en la red; sin embargo, las ventajas de este complejo mantenimiento de rutas, hacen despreciable el incremento del tráfico generado (en el capítulo 4 se analizan las ventajas de VAODV).

En la siguiente sección, se eliminan las desventajas descritas en este apartado, para mejorar el protocolo VAODV.

3.3. Optimizaciones a VAODV

A continuación se presentan algunas optimizaciones, para mejorar el rendimiento de AODV.

3.3.1. *REQUEST* y *REPLY* No Descartados (RRND)

Los nodos no descartarán ningún mensaje *REQUEST* y *REPLY* escuchados. Solamente los Líderes replicarán los *REQUEST*. Con RRND, la comunicación no es limitada a regiones vecinas. Esto permite aprovechar al máximo la potencia de la señal y disminuir el número de saltos en los procesos de descubrimiento de ruta. Los nodos Líderes no descartarán mensajes *REQUEST* de los nodos de su misma región, y si podrán reportar rutas de nodos de su misma región. RRND elimina los problemas de la Figura 3.13 (a) y (b).

3.3.2. *REPLY* unicast (RU)

AODV utiliza *REPLY* unicast haciendo uso de las rutas aprendidas en el envío de los mensajes *REQUEST*. Los mensajes *REQUEST* eventualmente pueden ser perdidos más fácilmente que los mensajes *REPLY*, debido a que son enviados en broadcast¹⁵. Los mensajes *REPLY* en broadcast facilitan la sincronización de los nodos Backup; sin embargo, son propensos a ser perdidos. La mayor desventaja de los mensajes *REPLY* en broadcast es que la tabla ARP no se actualiza previo al envío de mensajes¹⁶. RU garantiza que cuando un nodo desea utilizar una ruta, su tabla ARP estará actualizada.

¹⁵Siempre el envío de mensajes broadcast, involucra una mayor probabilidad de pérdida.

¹⁶La tabla ARP se actualiza únicamente después del envío de mensajes unicast.

3.3.3. Corrección de Ruta por *REQUEST* (CRR)

En la sección 3.2.8 se mencionó la desventaja de las correcciones de ruta por mensajes *REPLY* en broadcast, debido a la *ARP_table* no actualizada; CRR corrige este problema.

Cuando un nodo recibe un mensaje destinado hacia él, pero con una región diferente a su región actual, realiza un proceso CRR. El nodo enviará un mensaje *CRRequest* destinado hacia el origen del mensaje generador del CRR. Un *CRRequest* se basa en un mensaje *REQUEST*, pero lleva el campo TTL=1. Al recibir un nodo un *CRRequest*, no actualiza su *R_table* con la ruta inversa como lo haría con un mensaje *REQUEST*, solamente actualiza la región de su *R_table* y de todos los mensajes puestos en cola. Si el nodo tiene una ruta válida hacia el destino solicitado, responde en unicast con un *CRReply*; si no dispone de una ruta, descarta el mensaje. Un *CRReply* es igual que un *REPLY* con su campo TTL establecido en uno. Cuando el nodo que envió el *CRRequest* recibe un *CRReply* como respuesta, actualiza su *R_table* y envía un mensaje *REPLY* para corregir su ruta en los nodos intermedios.

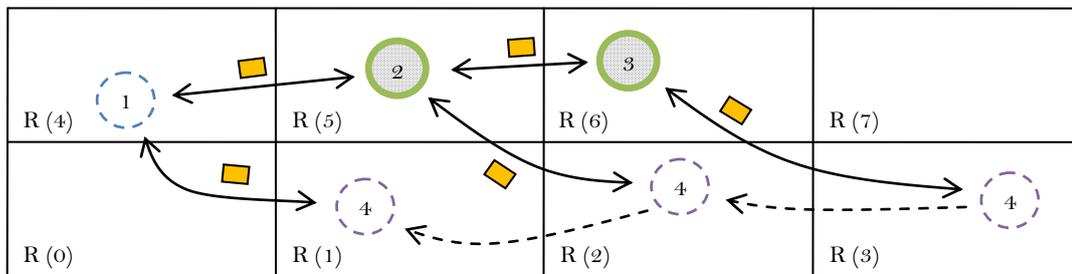


Figura 3.14.: Operación CRR; la ruta se corrige cada vez que el nodo 4 recibe mensajes hacia una región equivocada; de tal forma que, cuando 4 llega a la región 1, la comunicación entre el nodo 1 y 4 es directa.

La Figura 3.14 ilustra la operación CRR. Al abandonar el nodo 4 la región 3, recibe mensajes desde 3 dirigidos hacia una región diferente, el nodo 4 alertará este error enviando un *CRRequest* en broadcast destinado al nodo 1. Los nodos 2 y 3 al escuchar el *CRRequest* (suponiendo que el nodo 1 no lo puede escuchar) comprobarán la ruta hacia 4, el nodo 3 que está a un salto de 4 cambiará la región en su *R_table* y consultará si tiene paquetes en cola listos para ser enviados a 4 para cambiar la región destino del mensaje. Los nodos 2 y 3 responderán a 4, enviando un mensaje *CRReply* en unicast, en este proceso de envío, el nodo 2 aprenderá la dirección MAC de 4. Los paquetes en la cola del nodo 2 seguirán enviándose al nodo 3, y 3 enviará los nuevos mensajes destinados a la región correcta. Cuando el nodo 3 escuche el *CRReply* de 2 y 3, aceptará la ruta de 2, por tener un menor número de saltos hacia 1, y responderá con un mensaje *REPLY* hacia 2 para que corrija su ruta. Cuando 2 corrija la ruta hacia 4, los nuevos mensajes serán enviados directamente

hacia el destino¹⁷; la capa MAC hará el mapeo IP-MAC correctamente. Al seguirse moviendo el nodo 4, llegará a la región 1, aquí nuevamente se hará una corrección de ruta; se comunicarán los nodos 1 y 4 directamente.

CRR reduce el número de saltos, garantiza el mapeo IP-MAC después del cambio de ruta, y elimina la pérdida de paquetes. En AODV incluso cuando el nodo 4 este muy cercano al nodo 1, se mantendrá la misma ruta aprendida desde un inicio.

3.3.4. Sincronización por Canal Compartido (SCC)

El envío de mensajes de sincronización desde el líder hacia los nodos Backup incrementa el tráfico; aprovechar el medio compartido, elimina los constantes mensajes de sincronización. En el proceso de elección de Backups, el Líder de la región sincroniza la *R_table* de los Backups, de ahí en adelante los Backups escucharán todos los mensajes dirigidos a la región para sincronizar su *R_table*.

Para aprovechar el canal compartido la capa de Enlace, no elimina los mensajes que no sean dirigidos hacia este nodo, sino que pasa todos los mensajes a las capas superiores. Solamente los nodos establecidos como Backups procesarán los mensajes escuchados para añadir nuevas entradas en su *R_table* por medio de los mensaje *REPLY*, y actualizarán su tabla cuando escuchen mensajes de datos dirigidos hacia su región. Los nodos clientes rechazarán todos los mensajes SCC, para no incrementar su procesamiento.

3.4. Optimizaciones de la Capa Virtual

La pérdida de mensajes y las colisiones debido al envío de mensajes broadcast, puede incrementar el tiempo de establecimiento de un nodo Virtual, y aumentar el número de Líderes duplicados en una región. Un mensaje *LeaderLeft* perdido provoca nodos inestables, y por tanto en gran tiempo de re-establecimiento del nodo Virtual. Un mensaje *LeaderRequest* perdido provoca duplicado de Líderes, así como la demora en el proceso de elección de Backups en el caso de que la región no esté respaldada. *Heartbeat* perdidos provoca nodos inestable, esto no es crítico debido a que el Líder aún está presente en la región, y todos aun lo consideran su Líder; sin embargo, reduce en cierta forma el rendimiento de la capa Virtual.

Aprovechando la *B_table*, la *L_table* y la modificación a la capa de Enlace para aprovechar el canal compartido, se mejora el rendimiento de la capa Virtual. Cuando un nodo Líder abandona la región, examina en su *B_table* el Backup con la mayor prioridad para enviar un *LeaderLeft* unicast¹⁸. Los mensajes *LeaderRequest* también pueden ser enviados en unicast gracias a la *L_table*; cuando un nodo cambia de

¹⁷Los paquetes en cola, aun seguirán enviándose hacia el nodo 3.

¹⁸Un mensaje unicast perdido, es retransmitido por la capa MAC.

región, examina en su tabla la dirección IP del Líder de la nueva región para enviar el mensaje de solicitud. Un líder para el envío de *Heartbeats* consulta su *B_table* para enviar un mensaje dirigido, sino tuviese nodos Backups, envía el mensaje en broadcast. Mediante SCC los nodos clientes y los nodos Backup con menor orden de prioridad también escucharán los *Heartbeats*.

3.5. Aplicación VANET

Un nodo Virtual basado en Aplicaciones, permite la creación de servidores virtuales fijos. Nuevamente la sincronización es importante para mantener el servicio levantado, después que el líder abandone la región. Una simple Aplicación VANET es desarrollada en esta sección, para aprovechar las ventajas de la capa Virtual y del protocolo VAODV.

La Aplicación desarrollada permite conocer el Tráfico (número de vehículos) en una región. La Aplicación de Tráfico Vehicular (ATV) utiliza los mensajes *Hello*. Un nodo virtual escucha todos los mensajes Hello de los nodos de su misma región, de esta manera conoce cuál es el número actual de vehículos en la región. Los mensajes Hello son enviados periódicamente por todos los nodos; el tiempo de envío de un Hello es determinado de acuerdo al tamaño de la región y la velocidad promedio de los vehículos.

La ATV crea una tabla para mantener y gestionar el número de vehículos en su región. La tabla ATV contiene la dirección IP del originador del mensaje Hello, su región y el *time_expire*. Cuando un nodo recibe un mensaje Hello, crea una entrada en su tabla; si la entrada para dicho nodo ya existe, aumenta el *time_expire*. Cuando el *time_expire* es menor que el tiempo actual, se elimina de su tabla. Los mensajes Hello son enviados en broadcast, por lo que los nodos Backup solamente deberán sincronizar una sola vez su tabla ATV.

Cualquier nodo puede solicitar la ATV de cualquier nodo Virtual; el nodo que solicita el servicio es llamado “nodo cliente” (NC), el nodo Virtual que responde a la solicitud es llamado “nodo servidor” (NS). Cuando un nodo NC desea conocer el tráfico vehicular de una región R, enviará un mensaje *ApplicationClient* en broadcast con el campo *region_dest = R*. Los nodos Virtuales en diferentes regiones replicarán el mensaje hasta que el NS de la región R escuche la solicitud. El NS responderá a la solicitud con un *ApplicationServer* en unicast utilizando la ruta recién aprendida.

Este capítulo propuso el modelo de la capa Virtual para implementar un protocolo de enrutamiento basado en AODV y para la creación de Aplicaciones basada en nodos Virtuales.

4. Simulaciones y Comparaciones

En el capítulo III se presentó el modelo de la capa Virtual y el diseño de VAODV, este capítulo presenta los resultados de las simulaciones. En cada simulación se describe los detalles, parámetros y el escenario de simulación, se muestran graficas comparativas, y por último se analizan los resultados. Los resultados de las simulaciones mostrarán el aumento del rendimiento de la capa Virtual después de las optimizaciones descritas en el capítulo III; y las ventajas de VAODV sobre AODV.

4.1. Consideraciones de Simulación

La capa Virtual y VAODV fueron implementados en el simulador de redes *ns-2* (Apéndice A). Mediante el software VANS descrito en el capítulo I (Apéndice B), se generó la topología de la red vial y el tránsito vehicular; la misma aplicación permitió establecer los parámetros de la capa Virtual y de VAODV como: el número de regiones, numero de comunicaciones entre nodos, velocidad de transmisión, y la selección entre VAODV (con y sin sus optimizaciones) y AODV.

El escenario vehicular se importó desde un mapa urbano real. La velocidad de los móviles depende del tráfico y de la señalización (semáforos); la velocidad máxima fue establecida en 50Km/h. Los móviles toman una ruta dentro del mapa urbano de acuerdo a probabilidades establecidas en cada cruce; el tráfico se ha distribuido uniformemente en cada región. Al comenzar la simulación el tráfico se concentra en ciertas regiones, por esta razón en todas las simulaciones, las mediciones son tomadas desde el tiempo en que se distribuye el tráfico. El tiempo de simulación es de 2000seg, y se estabiliza el tráfico a los 300seg.

La comunicación entre nodos es constante, comienza desde el tiempo de estabilización, hasta el final de la simulación. La comunicación es restringida a una comunicación por nodo. La velocidad de transmisión es constante a 8Kbps mediante CBR (*Constant Bit Rate*). Cada sesión CBR es mantenida por una pareja de nodos. Las gráficas comparativas son medidas incrementando el número de sesiones CBR. Cada punto medido en una gráfica, representa un promedio de tres simulaciones hechas en tres computadores con diferentes características.

4.2. Evaluación del Rendimiento de la Capa Virtual

Esta sección evalúa el rendimiento de la capa Virtual con tráfico de datos en la red. Se establece el tiempo de envío de los *Heartbeat*, que determina en gran parte la sobrecarga de tráfico en la red. Se evalúa el tiempo de establecimiento de una región, los Líderes duplicados y los nodos inestables con el aumento de nodos. Se consideran las optimizaciones propuestas para comparar su rendimiento.

Los mensajes perdidos causan líderes duplicados. Para prever la pérdida de *Heartbeats*, se propuso en[2] la espera de dos *Heartbeats* más, antes de considerar la ausencia del Líder; cuando un *Heartbeat* se pierde, los nodos cambian de estado a inestables, y el nodo virtual permanece levantado. El problema de la inestabilidad, es la suposición de la presencia del Líder cuando un *LeaderLeft* es perdido, la inestabilidad en este caso ocasiona que el nodo virtual permanezca caído por tres periodos de *Heartbeat*, esta es la inestabilidad crítica.

Una región con Líderes duplicados se corrige casi de inmediato, debido al campo *Time_State*. Un duplicado puede ser causado, cuando el tiempo aleatorio de dos nodos entrando en una región al mismo tiempo coincide. También cuando un *LeaderRequest* no es escuchado por el Líder, el nodo solicitante al no recibir respuesta enviará un *Heartbeat*, a lo que el Líder actual responderá para corregir el problema. Una región sin Backup al salir el Líder, al no tener prioridad, puede coincidir su temporizador de solicitud.

El periodo de *Heartbeat* no debe ser muy pequeño como para incrementar el tráfico excesivamente, ni tan grande como para levantar al nodo virtual demasiado tarde, en el caso de que el Líder sufra una caída. El tiempo de *Heartbeat* establecido en las simulaciones es de 5seg, los nodos clientes esperan como máximo 7seg; en Líder o un Backup al abandonar la región, primero envían un mensaje alertando su salida, después de 0.2seg envían el mensaje de solicitud a la nueva región.

Una región con nodos Backups, tras la salida del Líder, levantan el nodo virtual de inmediato; siempre el Backup con mayor prioridad obtiene el liderato. Un nodo arribando a una región vacía, levanta el nodo virtual en máximo 0.5seg.

Para comparar y evaluar el rendimiento de la capa virtual, se ha obtenido el número de nodos inestables y el número de duplicado de Líderes. Se ha incrementado progresivamente el número de nodos dentro de una zona de 500x500 con 16 regiones, el tiempo de simulación es de 1000seg; además se ha expuesto a tráfico de datos (10 sesiones CBR sobre TCP) para incrementar la probabilidad de colisiones y pérdida de paquetes.

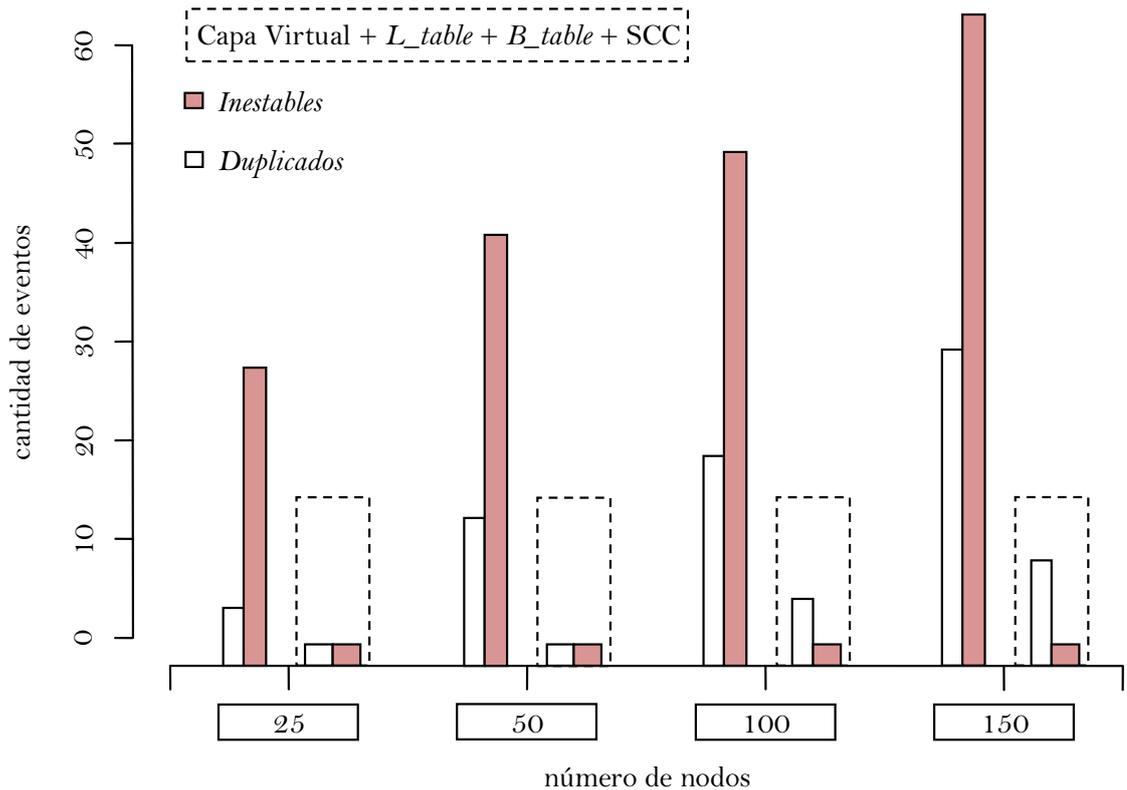


Figura 4.1.: Comparación de eventos registrados (nodos inestables y duplicado de Líderes) entre la capa Virtual, y la capa Virtual con optimizaciones (L_table , B_table y el aprovechamiento del Canal Compartido SCC).

La Figura 4.1 muestra los resultados de añadir las optimizaciones a la capa Virtual. La cantidad de nodos inestables y de Líderes duplicados en la capa Virtual sin las mejoras, crece a medida que el número de nodos en la simulación aumenta, esto es porque al haber más nodos las colisiones de paquetes aumentan. En la capa Virtual con las mejoras, no hay estados de inestabilidad debido al envío de *Heartbeat* en Unicast y a SCC. Los líderes duplicados se eliminan en gran parte por la L_table ; los pocos Líderes duplicados es por el arribo de nodos a una región vacía con una coincidencia en los temporizadores de solicitud.

4.3. Evaluación del Rendimiento de VAODV

En esta sección se presentan varias simulaciones para evaluar el rendimiento de VAODV. Se compara AODV con VAODV y se analiza el efecto de añadir las mejoras propuestas en la Sección 3.3

El escenario utilizado en esta sección, es de 800x800m dividido en 64 regiones de 100x100m. El número de nodos es 320, lo que significa un alto tráfico vehicular; los

nodos circulan dentro del escenario durante toda la simulación. El radio alcance de cada móvil es de 282m (como se analizó en la Subsección 3.1.1), esto garantiza que un mensaje enviado en una región, será escuchado por todas sus regiones vecinas.

El número de nodos se mantiene fijo, mientras que las sesiones CBR son incrementadas. CBR ahora es transportado sobre UDP, para evitar el reenvío de mensajes perdidos, con el fin de obtener una métrica adecuada de los mensajes entregados al destino exitosamente. Las simulaciones presentadas a continuación evaluarán los siguientes aspectos: paquetes entregados, el tráfico de control generado, número de descubrimientos de ruta y el número promedio de saltos entre el origen y destino de los paquetes.

4.3.1. Sobrecarga de Tráfico

En las redes MANETs el tráfico de datos siempre conlleva el tráfico de control; tráfico generado por la capa de Enlace para la seguridad y para el acceso al medio, tráfico generado por la capa de Red para el descubrimiento de ruta, y el tráfico de capa de la capa de Transporte para la retransmisión de datos perdidos. En esta sección evaluaremos el tráfico de control generado únicamente por la capa de Red y la capa Virtual. En AODV el tráfico de control está formado por los mensajes *REQUEST*, *REPLY* y los mensajes de *ERROR*; para VAODV además del tráfico anterior, se considerará todos los mensajes de capa Virtual y los mensajes de sincronización.

La Figura 4.2 muestra el incremento de tráfico producido por AODV a medida que aumenta el número de sesiones CBR; en las simulaciones más del 70 % del tráfico fue producido por los mensajes *REQUEST*. VAODV genera mayor tráfico con un número reducido de sesiones CBR, debido a los mensajes de elección de Líder generado por la capa Virtual; a medida que el número de sesiones aumenta el tráfico se compensa. Con el aumento de sesiones CBR, el tráfico para establecer los nodos Virtuales se mantiene, mientras que el tráfico de sincronización y de descubrimientos de ruta aumenta controladamente.

La Figura 4.3 muestra como el número de mensajes *REQUEST* en AODV sobrepasan por mucho a los generados por VAODV. En AODV solamente los nodos Líderes replican los mensajes *REQUEST*, en AODV todos los nodos que puedan escuchar un *REQUEST* lo replicarán; esta es la causa por lo que al aumentar el número de sesiones CBR el tráfico en VAODV es menor.

El efecto de añadir SCC a VAODV también se muestra en la Figura 4.2. SCC disminuye el tráfico de sincronización; un nodo Virtual solo sincronizará a sus nodos Backup una sola vez, después serán ellos los que escucharán el medio compartido como el fin de actualizar las rutas¹ y aprender nuevas. VAODV sin SCC, debe

¹Un Backup actualiza una ruta cuando escucha mensajes de datos y mensajes de Soft Handoff procesados por su Líder.

mantener sincronizado a todos los nodos Backup que conforman el nodo Virtual, incrementado el tráfico de sincronización y de mantenimiento de rutas.

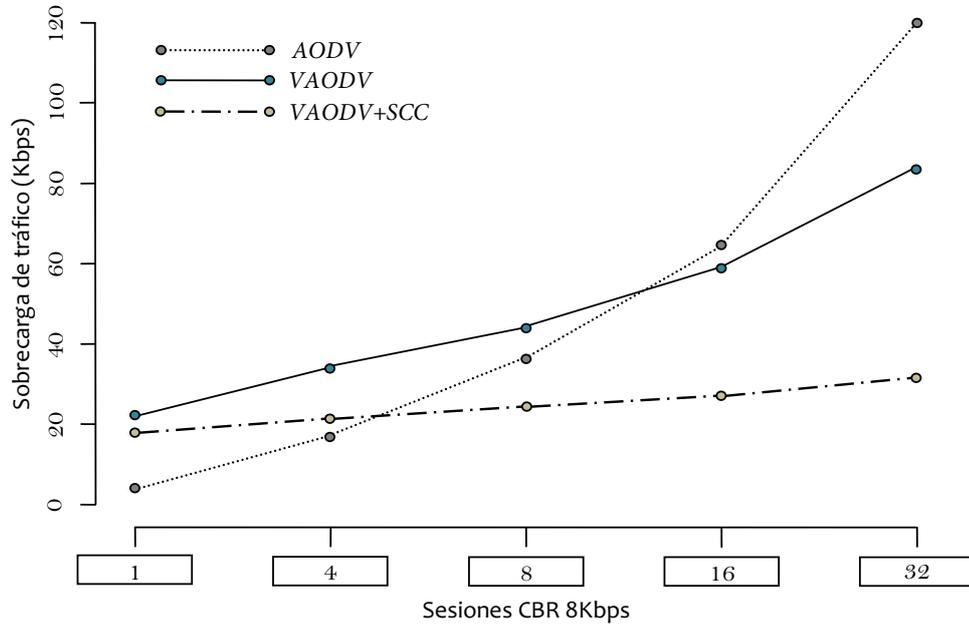


Figura 4.2.: Sobrecarga de tráfico para mantener el incremento de sesiones CBR en la red. La sobrecarga se ha medido en Kbps ignorando el encabezado de la capa de Enlace.

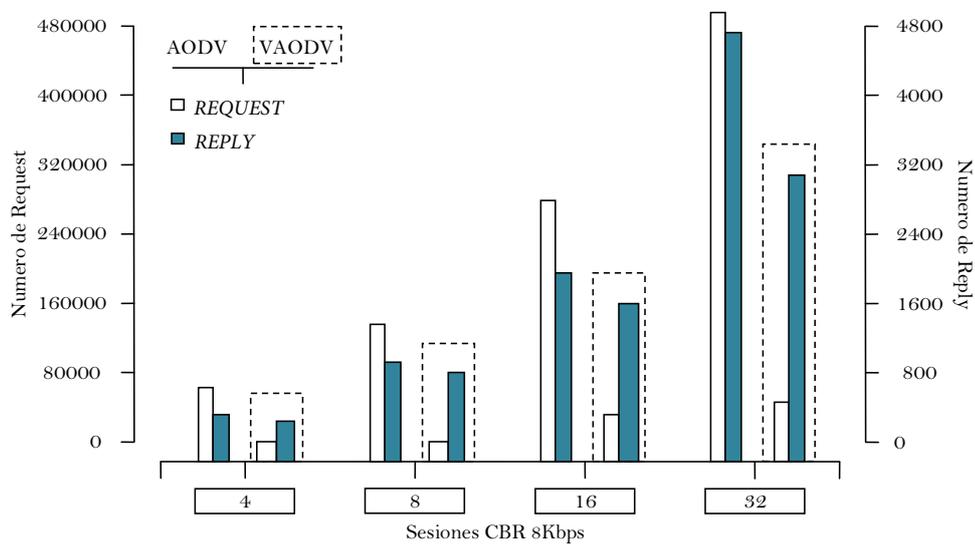


Figura 4.3.: Cantidad de mensajes *REQUEST* y *REPLY* incrementando el número de sesiones CBR.

4.3.2. Número de Saltos

El número saltos, es el número de nodos intermedios entre los dispositivos finales. El número de saltos determina también la latencia de los mensajes desde el origen al destino. Mientras mayor sea el número de nodos intermedios mayor es la latencia y más vulnerable la comunicación. Un nodo intermedio puede salir del área de cobertura y dejar inhabilitada la ruta; en este caso una ruta válida deberá levantarse para continuar con el envío de mensajes. Eventualmente un nuevo descubrimiento de ruta genera pérdida de paquetes por dos razones: un nodo que formaba parte de la ruta, puede aún tener paquetes en cola que no pudo enviar hacia el destino, y segundo, el tráfico aumenta debido al envío de mensajes *REQUEST* lo que puede ocasionar colisiones con otras comunicaciones.

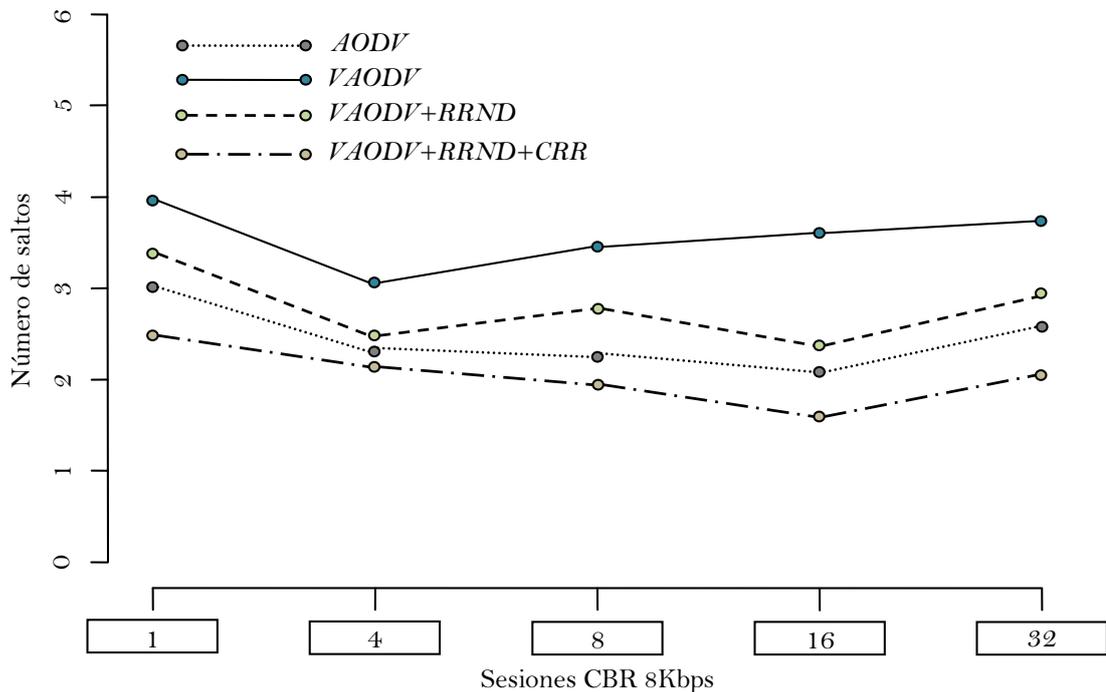


Figura 4.4.: Numero de saltos entre los dispositivos finales. El número de saltos a medida que aumentan las sesiones CBR, representan un promedio general entre todas las comunicaciones.

En VAODV el número de saltos no es tan crítico como en AODV, puesto que una ruta no depende de los físicos intermedios, sino de los nodos virtuales; además el tráfico generado en un descubrimiento de ruta es menor. Sin embargo, un menor número de saltos puede llevar a un mayor número de paquetes recibidos, puesto que disminuye la latencia, lo cual significa que, tras la finalización de la simulación un mayor número de paquetes habrán llegado al destino.

La Figura 4.4 muestra que VAODV presenta el mayor número de saltos; esto es

porque la comunicación está limitada a pasar por nodos virtuales vecinos. Al adicionar RRND, el número de saltos disminuye (como era de suponer), pues busca el camino más corto, pero aún está limitado a establecer su ruta a través de los nodos Líderes. AODV por su parte no tiene ninguna limitación, por lo que presenta un menor número de saltos que los anteriores. Al adicionar RRND y CRR, los saltos disminuyen aún más; CRR realiza una corrección de ruta cada vez que los nodos finales cambian de región con el fin de reducir el número de los saltos.

4.3.3. Descubrimientos de Ruta

Los nodos Virtuales pretenden simular un dispositivo enrutador ubicado en una posición fija, independiente del movimiento de los nodos. Cuando un descubrimiento de ruta es realizado por un nodo, la ruta formada se mantendrá constante mientras que los nodos virtuales que forman parte del trayecto de los paquetes, se mantengan levantados. Una ruta en AODV puede cambiar por dos razones: el movimiento de los nodos intermedios o por el movimiento de los nodos finales (*Endpoints*). En VOADV una ruta cambiará por la segunda razón mencionada y por la caída de un nodo virtual intermedio.

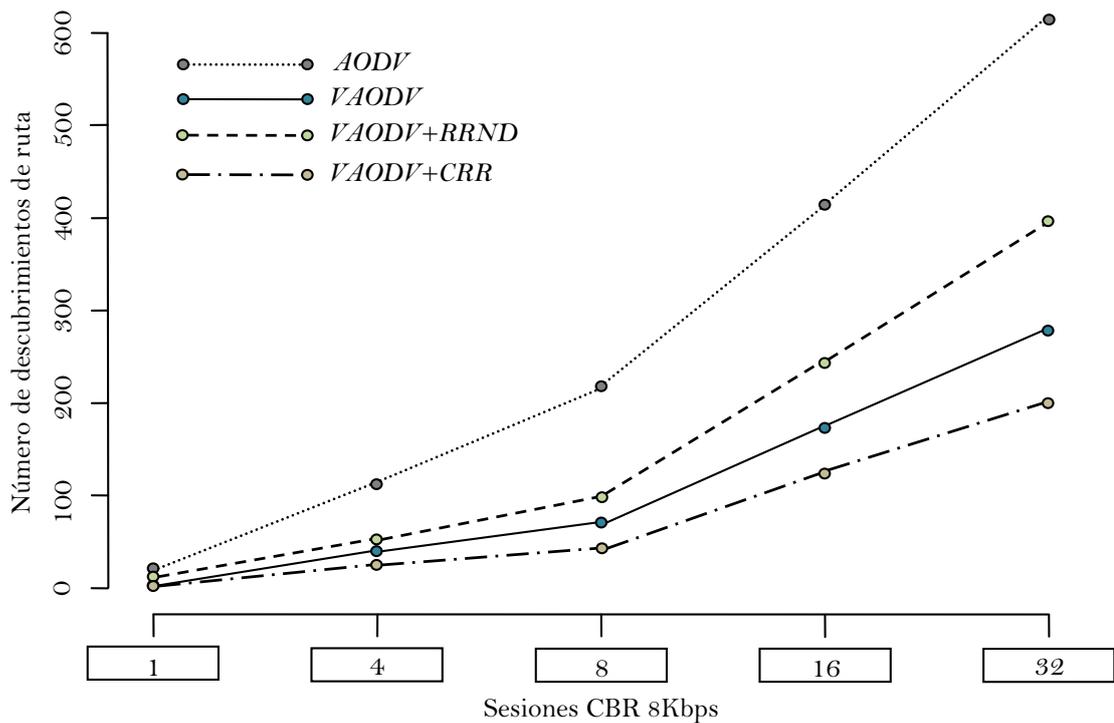


Figura 4.5.: Numero de descubrimientos de ruta.

Según la Figura 4.5, AODV genera más descubrimientos de ruta que VAODV. Cuando a VAODV se la adicionó RRND, las rutas son más vulnerables a cambiar; sin

RRND un nodo virtual intermedio (que no caiga debido a la presencia de Backups), siempre será una ruta válida; no así con RRND, donde un cambio de Líder en una región intermedia, puede poner en riesgo la ruta². CRR por su lado, disminuye los descubrimientos de ruta, al hacer reparaciones. Las reparaciones hechas por CRR, no generan tráfico significativo como un descubrimiento de ruta lo haría. CRR solo tiene efecto cuando los dispositivos finales cambian de posición; cuando estos se mantienen fijos CRR no tiene forma de actuar.

4.3.4. Paquetes Entregados

CBR transmite a una tasa de bits constante; sin embargo, no todos los paquetes llegan exitosamente al destino debido a la pérdida de paquetes. Los mensajes se pueden perder por colisiones, por la congestión en la red y por fallas de enlace debido al movimiento de los nodos. VAODV evita el incremento de tráfico en un descubrimiento de ruta y evita fallas de enlace al utilizar un enrutamiento por nodos Virtuales y no por nodos físicos. El reducido número de descubrimiento de rutas y la menor sobrecarga de tráfico insertada en la red, permiten que VAODV tenga mejor desempeño en la entrega de paquetes que AODV.

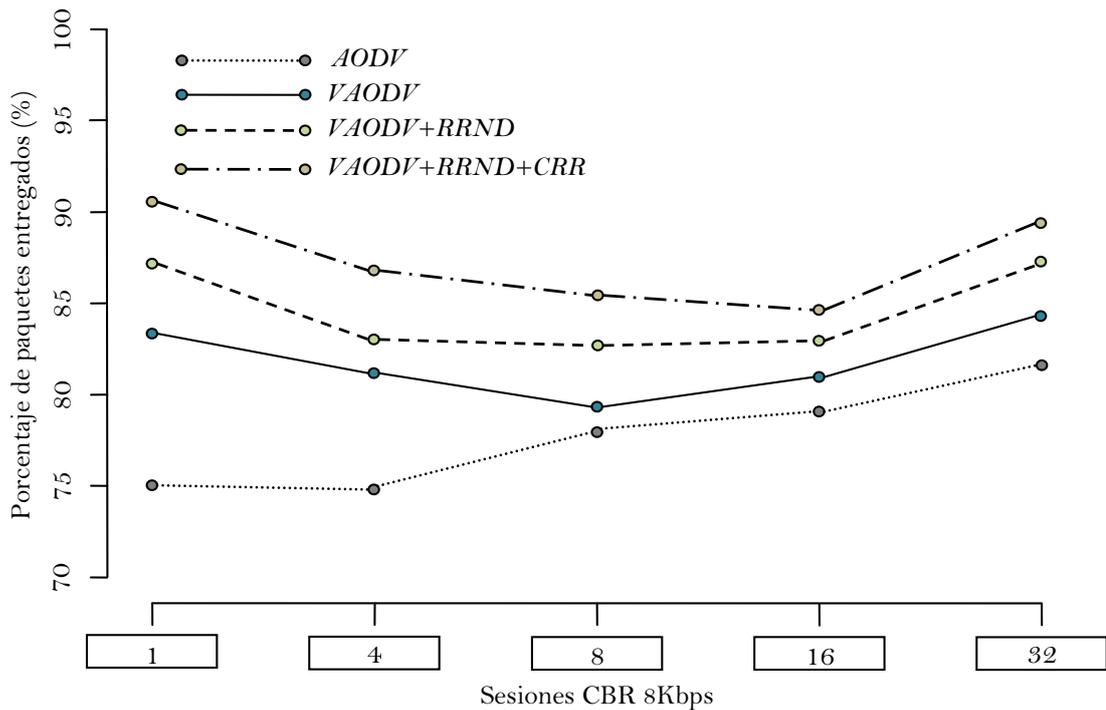


Figura 4.6.: Porcentaje de paquetes entregados exitosamente al destino.

²Con RRND el enlace con el Líder de una región, no garantiza el enlace con los futuros Líderes, estos podrían estar en una posición más alejada de la misma región

La Figura 4.6 muestra que VAODV en general entrega más paquetes al destino que AODV. Con RRND la entrega es mayor debido al menor número de saltos entre el destino y el origen, la misma razón explica el mejor rendimiento al añadir CRR. VAODV (solo) es más eficiente en el envío de mensajes, debido al menor número de descubrimientos de ruta. En AODV cada vez que ocurre una falla de enlace, se genera gran tráfico para descubrir una nueva ruta hacia el destino, provocando pérdida de paquetes. Un mensaje *REPLY* de vuelta hacia el origen, es muy probable que colisione debido al alto tráfico generado por los varios nodos que participan en el descubrimiento de ruta. En VAODV por el contrario, el número de agentes de enrutamiento es menor. Por otro lado, VAODV mantiene una ruta por mayor tiempo, ya que el camino formado depende de nodos virtuales y no de nodos físicos.

4.3.5. Efecto del Incremento del Tráfico Vehicular

En condiciones de alto tráfico vehicular las ventajas de la capa Virtual son más notables. A continuación se ha incrementado el tráfico vehicular con el fin de evitar la ausencia de nodos físicos en una región, esto garantizará que un nodo virtual permanecerá levantado; además se ha colocado a los *Endpoints* en posiciones fijas, con el objeto de poner a prueba la sincronización y elección de Backups en los nodos virtuales. Dicha simulación permitirá verificar el reducido número de descubrimientos de ruta, y por tanto una disminución del tráfico de control. Nuevamente se ha utilizado 64 regiones, con los *Endpoints* en ubicaciones alejadas y fijas. El número de nodos se ha incrementado a 500.

La Figura 4.7, Figura 4.8 y Figura 4.9 muestra el rendimiento de la capa virtual al incrementar el tráfico vehicular. Cuando los nodos virtuales permanecen en condiciones estables (presencia de nodos físicos y sincronización) el número de descubrimientos de ruta es extremadamente pequeño; la sobrecarga de tráfico es únicamente la generada por la capa Virtual y el porcentaje de envío de paquetes incrementa considerablemente.

4.4. Consideraciones Adicionales

La capa Virtual añade procesamiento a los móviles, este en gran parte se refleja en el tiempo real de simulación. Esto quiere decir que a más tiempo de simulación, el procesamiento es mayor. Todas las simulaciones realizadas en este capítulo, fueron de 2000seg, pero este no es el tiempo real de simulación. La Tabla 4.1 muestra el tiempo real de simulación de AODV y VAODV de acuerdo al número de sesiones CBR. Es claro que la capa Virtual con una sesión CBR necesita más tiempo de procesamiento, pero a medida que aumentan las sesiones se comprueba la disminución del procesamiento, debido a que la capa Virtual evita el incremento de procesamiento por descubrimientos de ruta excesivos, el envío de mensajes de ERROR y los cambios de ruta constantes.

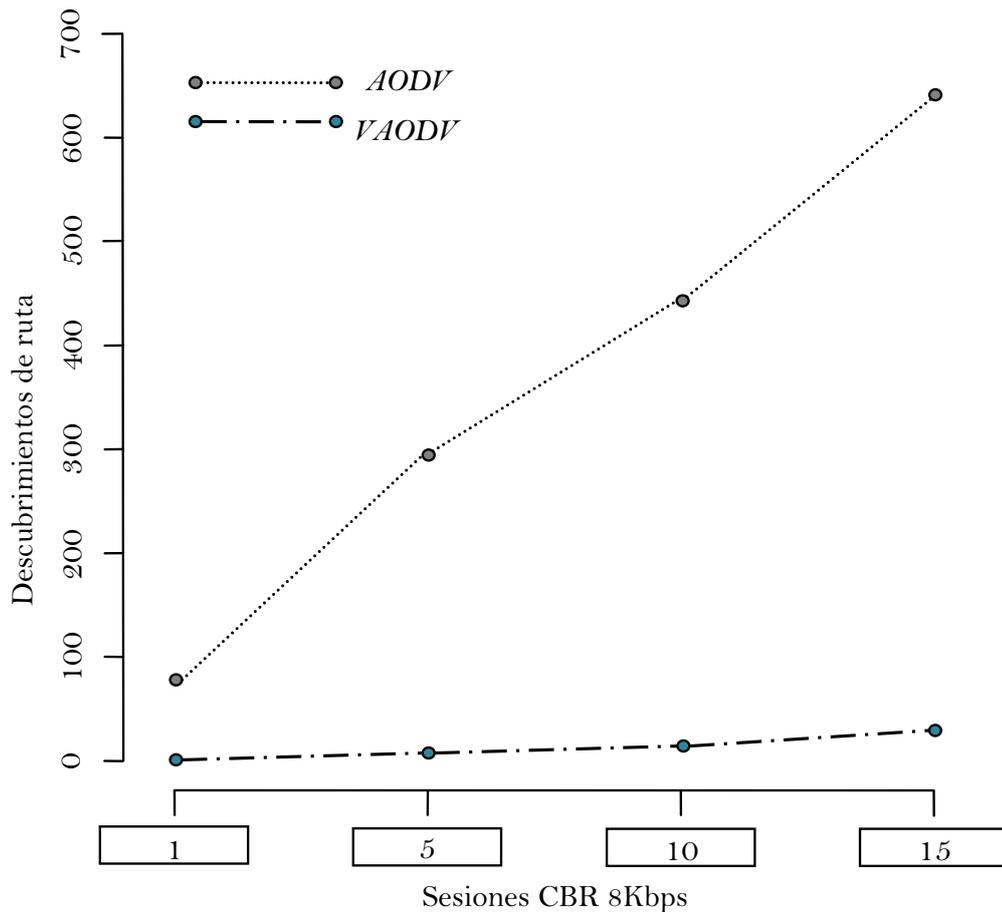


Figura 4.7.: Número de descubrimientos de ruta con nodos *Endpoints* fijos, e incrementando el número de móviles (alto tráfico vehicular).

Cuadro 4.1.: Tiempo real de simulación para 64 regiones de 8x8, 320 nodos, y 2000seg de simulación.

	1 CBR	4CBR	8CBR	16CBR	32CBR
AODV	1:24min	5:36min	9:30min	16:50min	30:34min
VAODV	5:50min	6:14min	6:22min	7:01min	8:12min

En las simulaciones de VOADV se estableció tres Backups por nodo Virtual; incrementar el número de Backups no incrementa el tráfico proporcionalmente. Una región con varios nodos físicos elegirá un Líder y el número de Backups establecidos, un nuevo Líder no deberá sincronizar nuevamente a los Backups, sino solamente a los nuevos. En presencia de alto tráfico vehicular un Líder a lo mucho deberá sincronizar a 2 o tres Backups; esto porque todos los Backups escogidos con prioridad menor seguirán siéndolo a pesar de haber muchos cambios de liderato. Con SCC los Backups se sincronizan sin generar tráfico adicional.

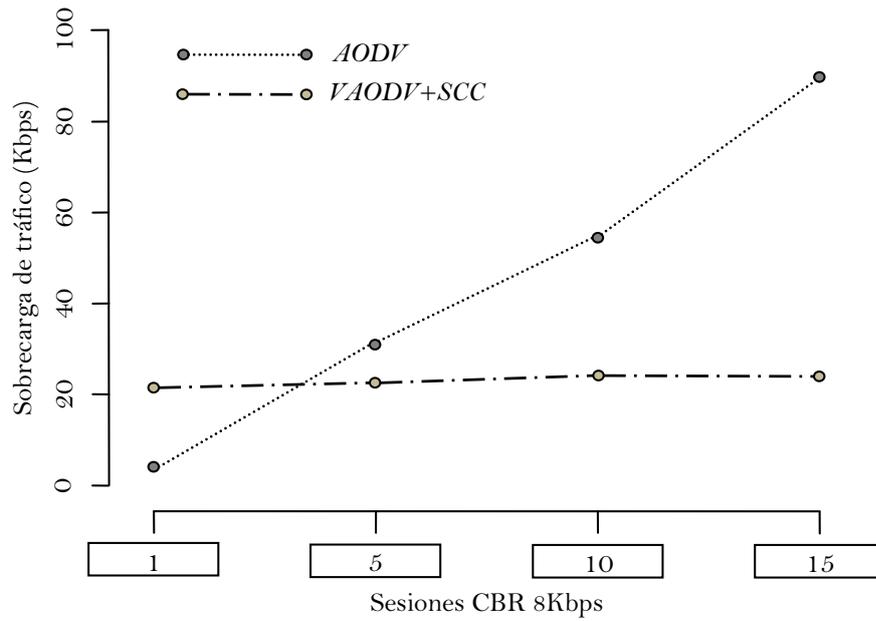


Figura 4.8.: Sobrecarga de tráfico, con *Endpoints* fijos y alto tráfico vehicular.

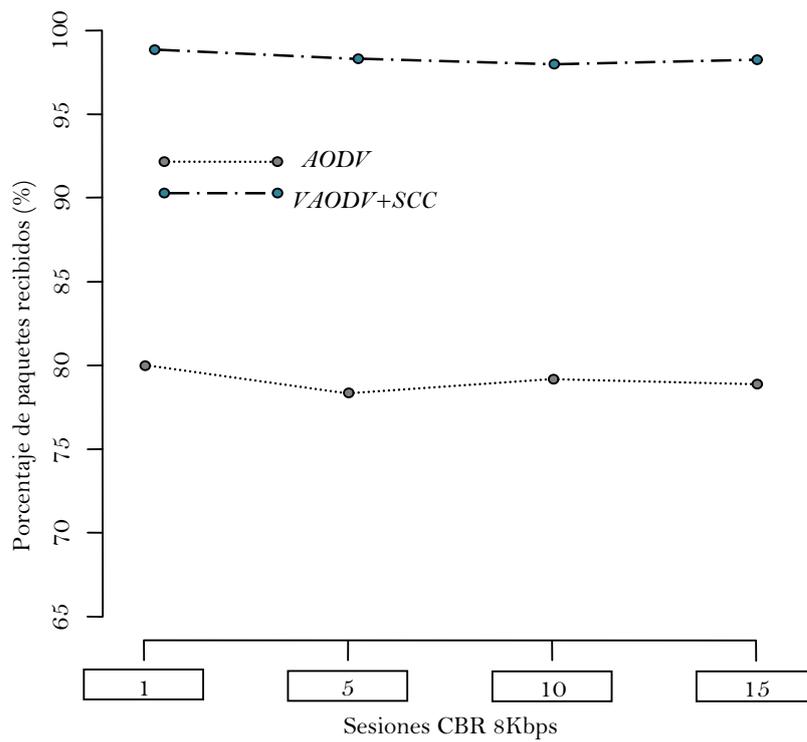


Figura 4.9.: Porcentaje de paquetes recibidos con nodos *Endpoints* fijos y alto tráfico vehicular.

Este capítulo evaluó la capa Virtual y el protocolo VAODV. Se verificó el mejor desempeño de la capa Virtual aprovechando el canal compartido. Realizando modificaciones en la MAC, los paquetes escuchados no son descartados, sino aprovechados por la capa Virtual. Utilizando SCC, los mensajes pueden ser envidos en Unicast, esto redujo grandemente los estados de inestabilidad y la duplicación de Líderes.

VAODV resultó de mayor eficiencia que AODV en muchos aspectos. VAODV crea rutas más confiables y estables. Aprovechando las ventajas de la virtualización VAODV redujo la sobrecarga de tráfico, evitando la congestión en la red, y generando más paquetes entregados al destino.

En presencia de mayor tráfico vehicular, las ventajas de la capa Virtual fueron más notorias. El mayor número de móviles incrementó la estabilidad de los nodos virtuales, y por tanto la estabilidad de las rutas generadas. La sobrecarga de tráfico se mantuvo casi constante y mucho menor que la generada por AODV. El rendimiento en la entrega de paquetes también incrementó.

5. Conclusiones y Trabajos Futuros

Este trabajo de tesis propuso un modelo para la capa Virtual y un protocolo de enrutamiento basado en AODV. La capa Virtual y VAODV se implementó en *ns-2*, y se simuló con la ayuda de Nam para verificar la creación de los nodos Virtuales, la elección del Líder, la elección de Backups, la sincronización, el algoritmo de ubicación y determinación de la región actual de cada nodo, así como el envío de mensajes y la visualización de las rutas estables a través de los nodos virtuales.

Se desarrolló un programa sobre JAVA (VANS) para la generación de movilidad de los nodos, y la creación de archivos tcl. El archivo tcl establece los parámetros de simulación, el número de nodos, número de sesiones CBR, tamaño y número de regiones, y otras opciones de simulación. VANS es una interfaz gráfica que facilitó la creación de todos los escenarios de simulación descritos en el capítulo IV.

La aplicación VANS, ofrece realizar simulaciones de VANETs de una forma rápida y eficiente sin tener mayores conocimientos en el simulador de red *ns-2* ni en el simulador de movilidad SUMO. El uso de software libre para la implementación y simulación de la capa Virtual, hace que la implementación a gran escala sea realizable, portable y compatible con los sistemas operativos más usados.

Este capítulo presenta algunas conclusiones importantes obtenidas en el desarrollo de esta tesis, y por último se mencionan futuros trabajos.

5.1. Sobre la Capa Virtual

La capa Virtual fue implementada por encima de la capa de Internet. Los mensajes generados por la capa virtual son significativamente pequeños (pocos bytes). Todos estos mensajes permiten mantener a los nodos virtuales en estados adecuados y de sincronización. El tiempo de envío de los mensajes *Heartbeat* fue establecido de acuerdo a ciertos parámetros como, la velocidad medio de los móviles y las consideraciones de aumento de tráfico. Para poner a prueba los tiempos establecidos, se desarrollaron simulaciones con el fin de no comprometer el tráfico en la red, ni el comportamiento de la capa Virtual.

En primera instancia todos los mensajes de la capa Virtual eran enviados en broadcast, lo cual disminuyó el rendimiento de la capa Virtual en presencia de colisiones y pérdida de paquetes. Después se añadieron las optimizaciones propuestas en el

capítulo III, para enviar todos los mensajes en unicast. Los mensajes *Heartbeat* unicast fueron posibles, gracias a las modificaciones realizadas en la MAC para evitar el descarte de paquetes, y gracias al establecimiento de la *B_table*. Los mensajes Unicast aumentaron el rendimiento de la capa Virtual enormemente.

A pesar de que los mensajes Unicast son más confiables de llegar al destino, esto no significa que el mensaje llegará de manera íntegra. La integridad de un mensaje, significa que este llegará al destino, exactamente como fue enviado; sin embargo, la integridad en un medio compartido e inalámbrico no siempre está garantizada. El hash en el encabezado de la capa Virtual, comprobaría si un mensaje ha sido alterado en la transmisión; en esta tesis el campo hash fue ignorado, pues se garantizó la integridad de los mensajes en las simulaciones, pero en casos más reales debería ser tomado en cuenta.

Los nodos utilizan un algoritmo sencillo para determinar la región en la que se encuentran. Un nodo utiliza este algoritmo después de conocer su posición geográfica; para esto cada móvil debe disponer un dispositivo GPS. Un móvil no necesita consultar su posición constantemente, otro algoritmo permite calcular el tiempo en que un nodo abandonará la región. Sin embargo, aprovechando las ventajas de los vehículos, un móvil solamente debería consultar su dispositivo GPS una sola vez, después, mediante su velocidad y el ángulo de cada giro del volante, podría conocer su posición en cada instante, para determinar su región actual.

5.2. Sobre VAODV

VAODV funciona sobre el núcleo de AODV, su tabla de enrutamiento tiene los mismos campos a excepción del campo *next_region*. El campo *next_region* permite utilizar rutas basadas en nodos virtuales y no en nodos físicos. VAODV además disminuye el número de agentes de enrutamiento, pues únicamente los nodos Líderes replican los mensajes *REQUEST*, los nodos No-líderes escuchan los mensajes para actualizar las rutas, pero no replican los mensajes.

Cuando un nodo Líder abandona su región, un proceso de *Soft Handoff* mantiene la ruta hacia la misma región pero hacia el nuevo móvil elegido como Líder. El *Soft Handoff* tiene éxito cuando el nodo virtual tiene Backups en estado de sincronización. La sincronización es fundamental para mantener los nodos virtuales; la sincronización por canal compartido disminuye la sobrecarga de tráfico, de esta forma los nodos Backups aprenderán todas las rutas conocidas por el nodo virtual, y después las actualizarán y aprenderán nuevas, escuchando todos los mensajes dirigidos a su región.

5.3. Sobre los Resultados de las Simulaciones

La implementación de la capa virtual en *ns-2* permitió simular escenarios VANETs, y desarrollar un protocolo de enrutamiento basado en nodos virtuales. El protocolo de enrutamiento implementado, permitió desarrollar una simple aplicación VANET y una aplicación para el envío de mensajes entre usuarios.

Las simulaciones mostraron que VAODV aprovecha las ventajas generadas por la capa virtual, disminuyendo la sobrecarga de tráfico, manteniendo rutas más estables, y entregando más paquetes a los nodos destinos.

Al añadir a VAODV la sincronización por canal compartido, el tráfico de control generado fue constante, a pesar de incrementar las sesiones CBR, esto quiere decir que VAODV es perfectamente escalable.

Al incrementar el tráfico vehicular, el desempeño de la capa Virtual y de VAODV tuvo un mejor rendimiento. En ausencia de tráfico, las regiones pueden quedar vacías, inhabilitando la comunicación entre regiones distantes. Cuando el número de móviles es reducido, los nodos virtuales pueden ser vulnerables a caer, y las rutas ya no son estables.

5.4. Trabajos Futuros

5.4.1. Aplicaciones para VANETs

La capa virtual implementada junto el protocolo de enrutamiento basado en nodos virtuales, permite que nuevos trabajos desarrollen aplicaciones propias para redes VANETs, aprovechando las ventajas provistas por la capa Virtual.

5.4.2. Líderes Sobrecargados

Evidentemente un nodo Líder puede sobrecargar su procesamiento debido a múltiples peticiones, ya sea de otros nodos virtuales, de nodos clientes o de nodos Backup solicitando datos de sincronización. Trabajos futuros deberían establecer nodos virtuales, con la capacidad de dividir la sobrecarga de procesamiento entre los nodos Backups, como si ellos también fuesen Líderes de la región.

5.4.3. Hibridación del Servicio

Eventualmente un descubrimiento de ruta no podría tener éxito debido a la ausencia de nodos físicos en regiones intermedias que podrían ayudar al establecimiento de un servicio. Por esta razón, futuros trabajos deberían establecer una opción viable

para estos problemas. Una forma de hacerlo sería implementar una combinación de servicios, uno basado en nodos virtuales y otro basado en nodos físicos colocados en posiciones fijas, de tal forma que cuando una comunicación no tenga éxito a través de nodos virtuales, se acceda al otro servicio con mayor garantía pero con un coste económico de por medio (como la red celular).

A. Programación de la Capa Virtual y de VAODV en *ns-2*

Basado en el modelo de la Capa Virtual y en el modelo del protocolo de enrutamiento VAODV, se desarrolló un software de simulación implementado en *ns-2*. *ns-2* es un simulador de redes desarrollado en C++, esto permite que sea rápido, potente y escalable.

ns-2 dispone de modelos inalámbricos desarrollados para la implementación de redes móviles. La estructura de un nodo móvil en *ns-2* contiene una Capa de Enlace (Link Layer), una tabla ARP, una cola para el envío de paquetes, una capa MAC (802.11), una interface de Red para acceder al canal, un modelo de radio propagación para simular la presencia de edificios, casas, etc., una antena omnidireccional con parámetros de transmisión y recepción modificables, además dispone de una serie de Agentes que son adicionados opcionalmente al móvil (Agente UDP, TCP, CBR, FTP, AODV, DSR, etc.).

Para las simulaciones, se implementó la Capa Virtual sobre un nuevo agente. El protocolo VAODV fue implementado sobre el agente AODV; en este se realizaron las modificaciones presentadas en el capítulo III.

La Figura A.1 muestra la estructura interna de un nodo móvil dentro de *ns-2*. La capa de enlace es totalmente provista por el simulador, lo cual reduce el tiempo de implementación de la Capa Virtual. El protocolo VAODV es una modificación del agente AODV disponible en *ns-2*. La capa Virtual es un nuevo agente que se comunica con el agente VAODV y con los agentes de Transporte. La capa de Aplicación está conformada por un conjunto de aplicaciones, el primer grupo son las aplicaciones de usuario (agentes de aplicación disponibles en *ns-2*), el segundo grupo está conformado por las aplicaciones propias para VANETs, en este caso la aplicación ATV descrita en la Sección 3.5.

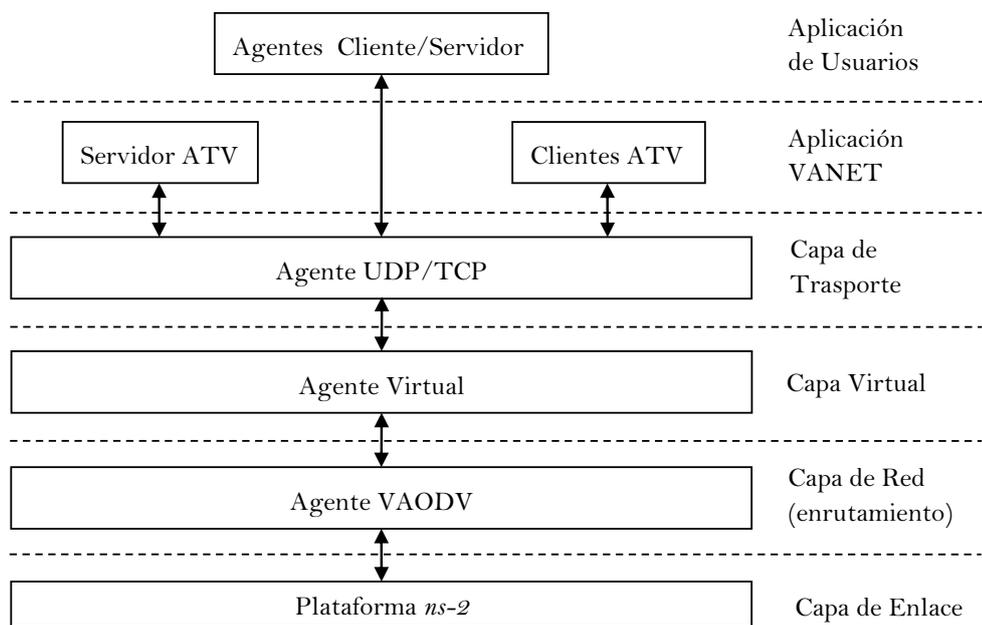


Figura A.1.: Arquitectura de la simulación en *ns-2*.

B. *Simulator of Urban Mobility y VANet Simulator modeler*

B.1. SUMO

Simulator of Urban Mobility (SUMO) es un simulador microscópico de tránsito vehicular que permite simular demanda y comportamiento de vehículos sobre redes viales. Su licencia es gratuita, y además es capaz de generar trazas de salida para ser utilizados en simuladores de red, como por ejemplo *ns-2*.

Para la creación de escenarios viales, se debe seguir los siguientes pasos:

1. Generación de la red vial en sí, la cual puede ser generada manualmente, aleatoriamente o importarla de otras fuentes como: *openstreetmap*, *visum*, *tiger*, entre otros. El resultado es un archivo con extensión *.net.xml*.
2. Definir cómo se van a desplazar los vehículos sobre la red vial ya creada. Un segundo archivo con extensión *.rou.xml* es generado, el cual contiene la información sobre la ruta de cada vehículo, origen, destino, así como cada tramo por el que atraviesa en la red. De igual manera, este archivo puede ser generado de forma manual o importado de otras fuentes como *Vissum* y *Vissim*.
3. El tercer paso es la compilación e integración de los dos archivos generados *.net.xml* y *.rou.xml* en uno global con extensión *.sumo.cfg*. Con este archivo ya se puede ejecutar y visualizar la simulación.
4. Opcionalmente, se puede exportar la simulación a trazas para ser utilizadas en simuladores de red como *ns-2*

B.2. VANS

Para facilitar la generación de modelos viales y escenarios de simulación VANETs, se desarrolló *VANet Simulator modeler* (VANS), está basado en MOVE pero con modificaciones para adaptar parámetros de la Capa Virtual realizados en el capítulo III. VANS está desarrollado sobre software libre (JAVA), integra el simulador de movilidad SUMO y de red *ns-2*, proporcionando un entorno de configuración gráfica.

VANS está dividida en dos secciones, la primera, genera los ficheros necesarios para crear una topología de red vehicular . La segunda configura parámetros de simulación de la Capa Virtual mostrado en la Figura B.1.

La primera sección tiene las siguientes características:

- Genera topologías de red y rutas vehiculares de forma manual usando ficheros de descriptores de nodos, calles, tipos y conectores. También puede generar automáticamente, o importa de otras fuentes como: openstreetmap y tiger.
- Genera modelos de demanda de tránsito vehicular manualmente mediante descriptores de viajes, flujos y probabilidad en cruces. También genera modelos de transito de forma automática o importar desde otras fuentes como Vissum y Vissim
- Compila la topología de red vial con el modelo de tránsito vehicular, y exporta el modelo de red vehicular hacia *ns-2*
- Ejecuta los programas sumo y sumo-gui

La segunda sección tiene las siguientes características:

- Configura los parámetros de simulación de una red VANET
- Ejecuta los programas de *ns-2* y Nam



Figura B.1.: Pantalla de modelo de movilidad, tiene las funciones de generación de una red vial, generación de rutas vehiculares, compilación, ejecución, y exportación.

Bibliografía

- [1] M. Brown, S. Gilbert, and N. Lynch, “The virtual node layer: A programming abstraction for wireless sensor networks.”
- [2] J. WU, “A simulation study on using the virtual node layer to implement efficient and reliable manet protocols,” Ph.D. dissertation, The City University of New York, 2011.
- [3] M. Spindel, “Simulation and evaluation of the reactive virtual node layer,” Master’s thesis, Massachusetts Institute of Technology, 2007.
- [4] M. Ilyas, *The Handbook of Ad Hoc Wireless Networks*, 2002.
- [5] S. Taneja and A. Kush, “A survey of routing protocols in mobile ad hoc networks.”
- [6] C. E. Perkins and E. M. Royer, “Ad-hoc on-demand distance vector routing,” *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*.
- [7] “The vint proyect. the network simulator - ns-2.” [Online]. Available: <http://www.isi.edu/nsnam/ns/>
- [8] “Simulation of urban mobility (sumo).” [Online]. Available: <http://sumo.sourceforge.net/>
- [9] J. Li, C. Blake, and S. Douglas, “Capacity of ad hoc wireless networks.”
- [10] G. Dimitrakopoulos and P. Demestichas, “Intelligent transportation systems,” *Vehicular Technology Magazine, IEEE*, pp. 77–84, March 2010.
- [11] B. Xu, S. Hischke, and B. Walke, “The role of ad hoc networking in future wireless communications,” *Communication Technology Proceedings, 2003. ICCT 2003. International Conference on (Volume:2)*, vol. 2, pp. 1353 – 1358, April 2003.
- [12] Y. Xiao and H. Li, “Local data control and admission control for qos support in wireless ad hoc networks,” *Vehicular Technology, IEEE Transactions on (Volume:53 , Issue: 5)*, pp. 1558 – 1572, Sept. 2004.
- [13] G. Anuj, H. Sadawarti, and A. Verma, “Performance analysis of aodv, dsr & tora routing protocols,” *IACSIT International Journal of Engineering and technology 2.2 (2010)*, pp. 226–231.

- [14] J. Wu, N. Lynch, and C. Newport, “Simulating fixed virtual nodes for adapting wireline protocols to manet,” *Network Computing and Applications, 2009. NCA 2009. Eighth IEEE International Symposium on*, pp. 12 – 19, July 2009.
- [15] S. Dolev, S. Gilbert, and N. A. Lynch, “Virtual mobile nodes for mobile ad hoc networks.”
- [16] S. Yousefi, Mousavi, and M. Fathy, “Vehicular ad hoc networks (vanets): Challenges and perspectives,” *ITS Telecommunications Proceedings, 2006 6th International Conference on*, pp. 761 – 766, June 2006.
- [17] M. Jerbi, S. Mohammed, and Y. Ghamri, *Vehicular Communications Networks: Current Trends and Challenges*.
- [18] Y. W. LIN, Y. S. CHEN, and S. L. LEE, “Routing protocols in vehicular ad hoc networks: A survey and future perspectives.”
- [19] X. Yang, J. Liu, and N. Vaidya, “A vehicle-to-vehicle communication protocol for cooperative collision warning,” *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, pp. 114 – 123, 22-26 Aug. 2004.
- [20] C. E. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers.”
- [21] P. Jacquet, P. Muhlethaler, and T. Clausen, “Optimized link state routing protocol for ad hoc networks,” *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*, pp. 62 – 68, 2001.
- [22] S. Murthy and J. J. Garcia, “An efficient routing protocol for wireless networks.”
- [23] D. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks.”
- [24] “Rfc: Ad hoc on-demand distance vector (aodv) routing.” [Online]. Available: <http://www.ietf.org/rfc/rfc3561.txt>
- [25] “Rfc: Temporally-ordered routing algorithm (tora) version 1.” [Online]. Available: <http://tools.ietf.org/html/draft-ietf-manet-tora-spec-00>
- [26] S. Dolev, S. Gilbert, and N. A. Lynch, “Autonomous virtual mobile nodes,” *R. Guerraoui, editor, 18th International Symposium on Distributed Computing (DISC 2004), Tripphenhuis, Amsterdam, the Netherlands, October, 2004, volume 3274 of Lecture Notes in Computer Science. Springer, December 2004.*, 2004.
- [27] L. Hu and D. Evans, “Localization for mobile sensor networks,” *Tenth Annual International Conference on Mobile Computing and Networking (Mobi-Com 2004). Philadelphia*, October 2004.
- [28] R. Branden, “Requirements for internet host–communication layer,” October 1989, rFC 1122 IETF.

- [29] T. S. Rappaport, *Wireless Communications: Principles and Practice (2nd Edition)*. Dorling Kindersley, 2009.
- [30] D. Zhao, X. Shen, and M. J. W., “Qos guarantee and power distribution for soft handoff connections in cellular cdma downlinks,” *Wireless Communications, IEEE Transactions on (Volume:5 , Issue: 4)*, pp. 910 – 919, Abril 2006.