Universidad Carlos III de Madrid

Escuela Politécnica Superior

Ingeniería de Telecomunicación


# *OPTIMIZACIÓN DE RUTAS PARA REDES MÓVILES EN REDES AD-HOC VEHICULARES*

## PROYECTO FIN DE CARRERA

Autor: María Isabel Sánchez Bueno
Tutor: Carlos Jesús Bernardos Cano

Noviembre de 2010

Proyecto Fin de Carrera
OPTIMIZACIÓN DE RUTAS PARA REDES MÓVILES EN REDES AD-HOC VEHICULARES

Autor
María Isabel Sánchez Bueno

Tutor
Dr. Carlos Jesús Bernardos Cano

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 17 de Noviembre de 2010 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, el siguiente tribunal:

Presidente: María Calderón Pastor

Secretario: Iria Estévez Ayres

Vocal: Soledad Escolar Díaz

acuerda otorgarle la calificación de:

Calificación: 10. Matrícula de Honor

Leganés, a 17 de Noviembre de 2010

# Agradecimientos

Cuando empecé a pensar en escribir los agradecimientos, me di cuenta de que no doy las gracias tan a menudo como debería, o al menos tantas veces como pienso en dar las gracias. Me refiero a agradecer las cosas importantes, no a que te dejen pasar en un pasillo estrecho cuando hay mucha gente y tienes prisa, o cuando te pasan la sal en la mesa. Ahí suelo dar las gracias siempre. Por eso, quiero aprovechar este momento para dar las gracias a esas personas especiales, amables, o que "simplemente"me quieren, que han estado a mi lado en cada momento.

Es difícil empezar, no porque me cueste trabajo daros las gracias, eso lo haría un millón de veces, porque sin vosotros no habría llegado hasta aquí, sino porque no quiero olvidarme de nadie, y es algo que sale directamente del corazón. Esas son las cosas que más cuesta decir, pero allá voy.

Quiero dar las gracias, en primer lugar, a mi tutor, Carlos Jesús Bernardos, por su paciencia infinita, por compartir algo de su sabiduría conmigo y por su ayuda a lo largo de todo este tiempo, que ha sido bastante, por otro lado. Me has enseñado mucho, aunque no lo pareciera, y admiro mucho tu dedicación en todo lo que haces y tu compromiso. Muchas gracias por abrirme la puerta al mundo de la investigación, porque he descubierto algo que realmente me gusta. Espero poder trabajar contigo en el futuro, porque no creo que haya muchas personas con las que merezca tanto la pena.

A los profesores, porque aunque el camino ha sido duro, ha habido algunos que realmente aman la docencia e inculcan algo de esa pasión a los alumnos, lo que facilita bastante el trabajo a la hora de preparar un examen. No voy a dar nombres, pero volviendo la vista atrás si que me vienen algunos, incluso de los primeros años, así que muchas gracias también.

A vosotros, esas personas que han compartido mi sufrimiento, también mis alegrías, a los que habéis llegado (o estáis llegando) al final y a los que os quedasteis en el camino (o escapasteis a tiempo, no se sabe). No quiero dejarme ningún nombre, pero no puedo evitar nombrar a Carol, Laura B, Dani, Raúl, Héctor, María Maraya, María San Blas, Cristóbal, Grego, Gema, Joaquín. Nos hemos apoyado mucho, es natural, hemos pasado más tiempo en la universidad que en nuestra propia casa, y también hemos reído mucho. Pero por lo que de verdad os estoy agradecida es por vuestro apoyo en los malos momentos, que ha habido muchos también, pero es agradable sentir que hay alguien a tu lado, aunque a veces no te apetezca hablar con nadie.

A ti, compañero de fatigas, es cierto que me has dado mucha guerra, pero también lo es que siempre estás ahí (cuando hace falta y cuando no) y que te desvives por ayudarme, aun en las cosas en las que no tienes que hacerlo. Sé que te lo he dicho muchas veces, pero

debería decírtelo muchas más, así que ¡gracias!, porque sin ti esto hubiera sido mucho más aburrido y porque en más de una ocasión me sacas de apuros, y porque no importa los kilómetros que tengas que hacer para ayudarme, y porque siempre me compras algo para la merienda, y porque me das un abrazo cuando lo necesito, y porque podría seguir dando razones por las que agradecerte hasta el año 2024. Por ser un apoyo incondicional, un poco exigente eso sí, pero es un precio que estoy dispuesta a pagar. Muchas gracias, Pablo.

Merecen una mención especial Daniel Miranda, porque siempre tienes gestiones que hacer (eres un Miranda Puente) pero en los momentos importantes te has portado como un campeón. Grego, tú también eres un amigo de los de verdad, un poco pachón, pero eres grande. Raúl, últimamente no nos hemos visto mucho, pero siempre he sabido que puedo contar contigo. Y de forma inesperada, tengo también que agradecer a Juan Camilo su entrega y su afán por ayudarme en todo. Muchas gracias, chicos.

Por supuesto, tengo que agradecer a mi familia por su amor, su paciencia, porque siendo como sois me habéis hecho a mí como soy, y por haber creído en mí en todo momento, incluso más que yo misma. Nene, aunque me hagas bromas y te metas conmigo, sé que presumes de hermana por ahí, pero no te preocupes, seguirá siendo un secreto. Mamá, no sé que haría sin ti. No tengo palabras suficientes. Aunque no nos vemos mucho y me digas las cosas siete veces, sabes que te quiero mucho y que haría cualquier cosa por ti.

Por último, a ti, valiente, generoso, humilde, noble de corazón, ejemplo de lucha inagotable, de vitalidad, de amor incondicional. Por enseñarme a luchar, por darnos todo, por dedicarte a nosotros, a cuidarnos, a protegernos, por darme todas las oportunidades, porque te hubiera gustado darme todavía más, porque a pesar de lo que duele echarte de menos, sería más doloroso no notar tu ausencia. Por hacernos felices y enseñarnos lo importante en la vida, sólo puedo darte las gracias, papá. Siento mucho no haber terminado antes.

En fin, espero no haberme dejado nada sin decir. Seguro que podría haberlo hecho mejor, pero sólo espero poder demostraros con hechos lo que muchas veces no soy capaz de decir con palabras.

A todo el mundo que ha hecho posible de una forma u otra que hoy esté escribiendo esto, muchísimas gracias.

*Los que se enamoran de la práctica sin la teoría son como los pilotos sin timón ni brújula, que nunca podrán saber a dónde van.*
Leonardo Da Vinci (1452-1519)

*En lo tocante a la ciencia, la autoridad de un millar no es superior al humilde razonamiento de una sola persona.*
Galileo Galilei (1564-1642)

*Si buscas resultados distintos, no hagas siempre lo mismo.*
Albert Einstein (1879-1955)

*El individuo ha luchado siempre para no ser absorbido por la tribu. Si lo intentas, a menudo estarás solo, y a veces asustado. Pero ningún precio es demasiado alto por el privilegio de ser uno mismo.*
Friedrich Nietzsche (1844-1900)

# Resumen

Hoy en día, el acceso a Internet y las comunicaciones se producen en escenarios muy diversos y con una gran variedad de dispositivos. Sobre todo, existe una tendencia creciente a la demanda de movilidad, es decir, a poder realizar la comunicación no sólo en cualquier sitio, sino también en movimiento. Pero, ¿cómo cambiaría nuestra vida si los dispositivos de comunicaciones fueran nuestros propios vehículos? Aunque pueda parecer ciencia-ficción, en los últimos años se han producido grandes avances en la investigación en el campo de las redes vehiculares, existiendo incluso propuestas de estandarización del protocolo de acceso al medio en entornos vehiculares (DSRC/802.11p).

Por otro lado, el protocolo de soporte básico de movilidad de redes, NEMO BS, propone una extensión del protocolo de movilidad IP para gestionar el movimiento de redes completas, en lugar del movimiento de un terminal. NEMO BS hace posible que los nodos de la red móvil mantengan sus comunicaciones con el resto del mundo a través de las mismas direcciones IP(v6) que tienen en su *red hogar*. Sin embargo, presenta una desventaja ya que el enrutamiento del tráfico entre la red móvil y cualquier otro nodo pasa por la red hogar, independientemente de que exista o no una ruta más eficiente. A este respecto, NEMO BS no contempla ninguna optimización de rutas.

La utilización del protocolo de movilidad de redes encaja en el entorno de las redes vehiculares, ya que se espera que en un vehículo existan numerosos dispositivos con necesidad de conectividad, desplazándose simultáneamente. De esta forma, dado que en algún momento las redes vehiculares serán una realidad, no es descabellado proponer una optimización de rutas para el protocolo de movilidad de redes (NEMO) en una red vehicular (contemplada como una red inalámbrica ad-hoc). Además, por las características propias de este tipo de redes, se hace necesario que los mecanismos de enrutado, gestión y configuración de los dispositivos de la red, se realicen de la forma más eficiente posible, ya que se trata de un entorno que puede cambiar rápidamente.

En este proyecto se propone una optimización de rutas para NEMO en redes vehiculares. La idea principal es que dos vehículos presentes en la misma red puedan comunicarse entre sí directamente, a través de unos pocos saltos intermedios, en lugar de acceder a Internet para llegar a sus respectivas redes hogar, con el retardo que esto puede suponer. Para ello, se ha desarrollado, a partir de la optimización de rutas propuesta en VARON, una implementación en lenguaje C, que después ha sido puesta en práctica en un router comercial. Este router, que juega el papel de router móvil, es el elemento principal en este proyecto ya que se encarga de realizar todas las operaciones, tanto para la gestión de la movilidad de la red como para la creación de la ruta en la red vehicular.

**Palabras clave:** IPv6, Movilidad de Redes, Router Móvil, Redes Vehiculares, Optimización de Rutas, OpenWrt, Asus WL-500g Premium.

# Abstract

Nowadays, Internet access and communications occur in very different scenarios and with a wide variety of devices. Above all, there is a raising trend/fashion to demanding mobility, that is to say, to be able to communicate everywhere while in movement. How would our lives change if the communication device were our own vehicle? Although it could seem to be science fiction, there have been huge advances in research in the field of vehicular communications, arising standardization proposals for a medium access control layer in vehicular networks (DSRC/802.11p).

On the other hand, there is a network mobility basic support protocol, NEMO BS, which propose an extension to the IP mobility protocol in order to manage the movement of a whole network, instead of the movement of a single terminal. This protocol has great advantages and keeps mobile network connected to the rest of the world through the same IP(v6) address while moving. However, it has a drawback because every datagram exchange with any other node, has to travel in a tunnel to the home network, no matter what, independently of the existence of more efficient routes or at less cost. Related to this, NEMO BS does not consider any route optimization mechanism.

Right now, since is very likely that vehicular networks will become a reality not too far to come, it makes sense to propose a NEMO route optimization mechanism for ad-hoc vehicular networks. Moreover, due to the inherent characteristics of this kind of network, it is necessary to make routing mechanisms, manage and configuration as efficient as possible, because it is a frequently changing environment.

This master thesis analyzes a vehicular ad-hoc route optimization for NEMO. The main idea is to make two vehicles in the same network able to communicate directly with each other, by a few intermediate hops, instead of using the Internet to reach their home networks, with the subsequent delay that it could mean. A prototype has been developed, based on the procedure defined in VARON, that has been later deployed in a commercial router. This router acts as a mobile router, being the main device in this project as it manages the mobility of the network as well as the route optimization, without leaving aside the tasks a proper router does.

**Keywords:** IPv6, Network Mobility, Mobile Router, Vehicular Networks, Route Optimization, OpenWrt, Asus WL-500g Premium.

# Indice General

# Lista de Figuras

# Lista de Tablas

# Parte I

# Introducción

# Capítulo 1

# Introducción

## 1.1. Introducción

Este primer capítulo realiza una introducción a los objetivos de este proyecto y presenta brevemente las fases del trabajo realizado. También se detalla la estructura seguida en este documento, introduciendo las distintas partes y los capítulos en los que se puede encontrar la descripción de cada una de las tareas llevadas a cabo durante la realización de este proyecto fin de carrera.

## 1.2. Objetivos

Los principales objetivos de este proyecto se enumeran a continuación:

- Estudio del protocolo de Soporte Básico de Movilidad de Redes propuesto por el grupo de trabajo IETF (*Internet Engineering Task Force*) a través de la asimilación del documento donde se recoge su especificación [DWPT05].
- Análisis del *firmware* libre OpenWrt y de los procesos de cambio de *firmware* en los routers utilizados.
- Análisis de las herramientas proporcionadas por OpenWrt para el desarrollo de *software* y módulos del *Kernel* específicos para la arquitectura de los equipos utilizados.
- Estudio del proceso de optimización de rutas para NEMO propuesto en VARON [Ber06].
- Desarrollo de una aplicación para poder implementar VARON en un router móvil.
- Comprobar la compatibilidad de una implementación de NEMO BS en un router móvil con varias interfaces inalámbricas.
- Compatibilizar la solución implementada para VARON con la implementación del protocolo de movilidad de redes, NEMO BS.
- Despliegue y configuración de un escenario de red, para validar el funcionamiento de la solución desarrollada.

- Evaluación práctica de la implementación.
- Análisis del rendimiento y las prestaciones para la evaluación de la optimización de rutas en un dispositivo real.

## 1.3. Fases del desarrollo

El trabajo realizado en este proyecto fin de carrera ha sido dividido en las siguientes fases de desarrollo:

- **Documentación previa**. En primer lugar se realizó un estudio de los protocolos NEMO BS y VARON, entre otros, como los principales pilares en cuanto a protocolos de comunicaciones en este proyecto.
- **Análisis de las herramientas de desarrollo**. Al realizarse también una implementación física, no basta realizar un análisis teórico, sino que también es necesario conocer herramientas de desarrollo que permitirán llevar a cabo la instalación y configuración de los distintos dispositivos de la red.
- **Comparativa y análisis del rendimiento de los dispositivos a utilizar**. La realización de este proyecto con dispositivos comerciales requiere realizar una búsqueda en el mercado de unos dispositivos que reúnan ciertos requisitos. Una vez encontrados, se hizo un estudio comparativo de su rendimiento. Este es el caso de los adaptadores inalámbricos USB, necesarios para poder dotar al router de varias interfaces inalámbricas.
- **Desarrollo software**. En esta fase se realiza el desarrollo del software para la optimización de rutas propuesta en VARON.
- **Integración de los distintos módulos**. El software, por simplicidad, está dividido en módulos que fue necesario integrar para comprobar el funcionamiento del conjunto. Además, existe otro software, el de NEMO BS, que aunque no fue desarrollado en este proyecto constituye un elemento primordial para el desarrollo del mismo, y fue necesario comprobar su compatibilidad con todos los demás elementos.
- **Evaluación práctica del funcionamiento**. Una vez reunidas y terminadas todas las distintas partes es necesario verificar que el funcionamiento del conjunto es el adecuado.
- **Evaluación práctica de la eficiencia de la optimización de rutas propuesta**. Finalmente, se realiza una evaluación práctica de la optimización de rutas propuesta por VARON, para comprobar su rendimiento, eficiencia y medir parámetros de calidad, como por ejemplo el retardo introducido para crear la ruta.

## 1.4. Medios empleados

Para la realización de este proyecto, se han utilizado los siguientes dispositivos:

- Router ASUS WL500g-Premium: Este router es el elemento principal para el desarrollo de este proyecto. Para construir un escenario de pruebas se han utilizado 9 de estos routers.

- Adaptador inalámbrico USB Linksys WUSB54GC: Se han utilizado dos de estos adaptadores para conseguir una interfaz inalámbrica adicional en el router ASUS.
- Adaptador inalámbrico USB Pheenet WLU-803G: Se han utilizado también dos. El objetivo inicial era tener dos dispositivos diferentes para poder comparar el rendimiento de uno y otro.
- Tarjeta de red inalámbrica Alfa Networks AWPCI085S: Esta tarjeta de red se ha utilizado reemplazando la tarjeta inalámbrica original del router ASUS. El cambio se realizó ya que la nueva tarjeta, ya que la tarjeta *Atheros*, permite conseguir mayor flexibilidad de configuración y extender la funcionalidad del dispositivo que su tarjeta original, *Broadcom*.
- Varios ordenadores personales para ejecutar algunos módulos *software* y para realizar tareas de configuración y monitorización de los routers ASUS.

Además de estos medios físicos se han utilizado recursos *software*, procurando en todo momento que fueran de *software* libre, como el *firmware* OpenWrt y otras herramientas presentes en distribuciones Linux.

## 1.5. Estructura de la memoria

La presente memoria se compone de varias partes, cada una de ellas divida en capítulos como se detalla a continuación:

1. **Primera parte: Introducción**. Presentación de las motivaciones y objetivos del trabajo realizado, así como de la estructura de esta memoria. Consta de un sólo capítulo (Capítulo 1: Introducción) en el que se detallan los objetivos del presente proyecto, las fases de desarrollo y los medios utilizados, además se describe la estructura seguida en la memoria.
2. **Segunda parte: Estado del arte**: Se analiza la situación de la movilidad de redes, así como su aplicación en redes vehiculares, y de ahí la presentación del protocolo de Optimización de Rutas para Redes Móviles en Redes Ad-hoc Vehiculares, VARON. Se compone de los siguientes capítulos:
   2.1 **Capítulo 2**: Movilidad de redes. NEMO BS.
   2.2 **Capítulo 3**: Redes vehiculares.
   2.3 **Capítulo 4**: VARON.
3. **Tercera parte: Descripción del trabajo realizado**. En esta parte se detallan de forma exhaustiva los pasos seguidos en el desarrollo de la aplicación realizada y las decisiones de diseño adoptadas, así como las pruebas llevadas a cabo para evaluar el trabajo realizado. Está dividida en los siguientes capítulos:
   3.1 **Capítulo 5**: Extensión de NEMO BS para varias interfaces inalámbricas.
   3.2 **Capítulo 6**: Desarrollo y estructura del *software*.
   3.3 **Capítulo 7**: Escenario de funcionamiento desplegado.
   3.4 **Capítulo 8**: Evaluación del prototipo desarrollado.

4. **Cuarta parte: Conclusiones**. En esta parte se presentan las principales conclusiones extraídas del trabajo realizado, así como las líneas de trabajo para futuras ampliaciones. Consta de un sólo capítulo:

   4.1 **Capítulo 9**: Conclusiones y líneas de trabajo futuro.

5. **Quinta parte: Apéndices**. Esta última parte consta de varios apéndices en los que se explican con mayor nivel de detalle algunas de las partes del trabajo realizado, que por considerarse demasiado específicas ha preferido separarse del documento principal:

   5.1 **Apéndice A: Presupuesto y plan de proyecto**. Se presenta un análisis de las tareas realizadas y los costes derivados de este proyecto.

   5.2 **Apéndice B: Mensajes del protocolo de Soporte Básico de Movilidad de Redes - NEMO BS** En este apéndice se detalla la estructura de la cabecera de movilidad y de los mensajes de señalización del protocolo NEMO BS

   5.3 **Apéndice C: Mensajes del protocolo de Optimización de Rutas en Redes Vehiculares para NEMO - VARON**. Se describe el formato de cada uno de los mensajes de VARON y sus distintas opciones.

   5.4 **Apéndice D: Instalación de OpenWrt en el router ASUS WL-500g Premium**. Estudio del proceso de cambio de *firmware* en el router ASUS WL-500g Premium.

   5.5 **Apéndice E: Instalación y uso de la plataforma de desarrollo de OpenWrt en PC**. Estudio de las herramientas de desarrollo que proporciona OpenWrt.

   5.6 **Apéndice F: Detalles de instalación y configuración necesarios en los routers**. Descripción del proceso de montaje y configuración del escenario en el que se realiza la evaluación experimental del prototipo.

# Parte II

# Estado del Arte

# Capítulo 2

# Movilidad de redes IPv6: NEMO BS

## 2.1. Introducción

Los avances de las tecnologías basadas en IP en los últimos tiempos y el desarrollo de Internet han facilitado la extensión del protocolo IP, sobre todo en su versión más reciente, IPv6, a otros campos para los que originalmente no fue diseñado. Una de estas extensiones desarrollada para incluir soporte de la movilidad, ha sido el protocolo MIPv6 (***M**obile IPv6*)[JPA04]. Siguiendo esa línea, han aparecido otras soluciones para variar distintos aspectos susceptibles de mejora o no tratados en la definición original. Este es el caso de la movilidad de redes completas. Este capítulo presenta una solución, llamada NEMO BS (***NE**twork **MO**bility **B**asic **S**upport*) [DWPT05] con mecanismos similares a los utilizados para el soporte de movilidad en IPv6 pero con la diferencia de que el elemento "en movimiento" es una red completa en lugar de un nodo. Se analizará el funcionamiento de este protocolo, así como sus ventajas e ineficiencias.

## 2.2. Movilidad

En los últimos años, gracias al crecimiento de la telefonía móvil y al desarrollo de dispositivos móviles cada vez con mayor capacidad y menor tamaño, incluir soporte para la movilidad de los usuarios se ha convertido en una necesidad, o casi una obligación, para un gran número de aplicaciones. No resulta extraño encontrar usuarios conectados a Internet o accediendo a otro equipo de forma remota en lugares como aeropuertos o estaciones de tren. Además, supone una gran ventaja por ejemplo, para un hombre de negocios, poder realizar gestiones o asistir a una reunión mientras viaja o se desplaza en un medio de transporte. Para poder ofrecer estos servicios, es necesario gestionar una serie de parámetros que permitan cambiar de punto de acceso a la red al producirse el movimiento y que este cambio se realice de forma transparente al usuario.

Aunque la gestión de la movilidad puede realizarse a distintos niveles, hacerlo a nivel de red (o a nivel IP) proporciona algunas ventajas:

- No es necesario modificar las aplicaciones ya existentes, así como el desarrollo de nuevas aplicaciones no se ve afectado por la evolución que los distintos protocolos de movilidad puedan sufrir, ocasionando posibles problemas de compatibilidad.

- Los cambios ocasionados por el movimiento del usuario requieren principalmente cambios en la localización en la red y el enrutamiento, cambios relacionados directamente con el nivel de red y el protocolo IP.
- Permite soportar redes de acceso heterogéneas (UMTS, WLAN, etc.).

Para gestionar el movimiento de los distintos dispositivos, se establecen unos principios básicos para dotar de cierta estructura a todo el proceso. Por ejemplo, se asume que el dispositivo móvil "pertenece" a una red, llamada *red hogar*, considerada su ubicación habitual, su punto de partida, por así decirlo, para el acceso a Internet. Cuando el dispositivo móvil se desplaza su punto de acceso cambia, llamando *red visitada* a la red a la que pertenece el router que proporciona conexión al nodo móvil en su nueva ubicación. Este router, coopera con un router perteneciente a la red hogar del nodo móvil para hacer posible que el nodo móvil reciba el tráfico dirigido a él, a pesar de haber cambiado de posición en la topología de la red (Figura 2.1).

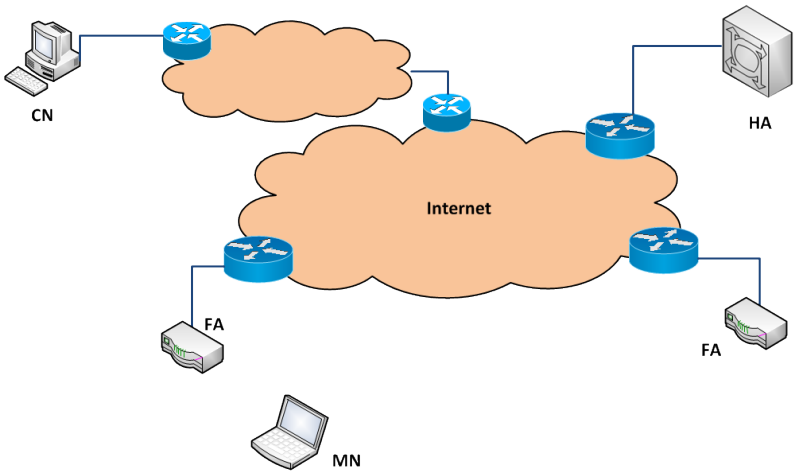


Figura 2.1: Escenario general de movilidad

En el protocolo de movilidad a nivel IP se definen cuatro entidades principales:

- Nodo móvil (***M**obile **N**ode*, MN): el dispositivo móvil.
- Agente local (***H**ome **A**gent*, HA): router en la red hogar que gestiona las conexiones del nodo móvil.
- Agente foráneo (***F**oreign **A**gent*, FA): router de acceso en la red visitada que facilita la movilidad del terminal cooperando con su agente local.
- Nodo corresponsal, también llamado nodo interlocutor (***C**orrespondent **N**ode*, CN): nodo, fijo o móvil, con el que se comunica el nodo móvil.

Entre estos cuatro elementos destacan el agente local y el agente foráneo, ya que son los principales responsables del encaminamiento del tráfico dirigido al nodo móvil, y por tanto, de que siga siendo alcanzable a través de la dirección IP que toma en su red hogar, a pesar de cualquier cambio en su localización. Para ello, el nodo móvil conserva esa dirección, o HoA (*Home Address*), pero además se le asigna una nueva dirección, o CoA (*Care-of Address*) cuando se conecta a una red visitada. El nodo móvil al detectar el movimiento

debe informar a su agente local, para que éste pueda almacenar la información relevante para la movilidad referente a ese nodo móvil (CoA, HoA) y así establecer un túnel entre el agente local y el nodo móvil, a través del cual el nodo móvil recibirá el tráfico dirigido a su HoA. De la misma manera, cuando debido al movimiento se produzca un cambio en el punto de acceso, es decir, ocurra un traspaso o *handover*, el nodo móvil deberá informar de nuevo a su agente local ya que su CoA habrá cambiado, para modificar el túnel entre ellos.

El soporte de movilidad se tuvo en cuenta cuando IPv6 fue diseñado, por lo que la integración en este protocolo es más sencilla que en su predecesor IPv4, resultando en un protocolo más eficiente. Por ejemplo, se eliminó la figura del agente foráneo. Las redes visitadas contarán con uno o varios routers de acceso, que a través de anuncios (*Router Advertisement* de ICMPv6) o por otros métodos, permitirán la autoconfiguración del nodo móvil, adquiriendo él mismo la CoA e informando a su agente local. Además también existe la posibilidad de realizar una optimización de rutas, ya que el nodo móvil puede registrarse con un nodo corresponsal además de con su agente local para que el tráfico dirigido a él no tenga que pasar forzosamente por su red hogar para ser enviado a su ubicación actual, dando solución al problema conocido como *triangular routing*, mostrado en la Figura 2.2.

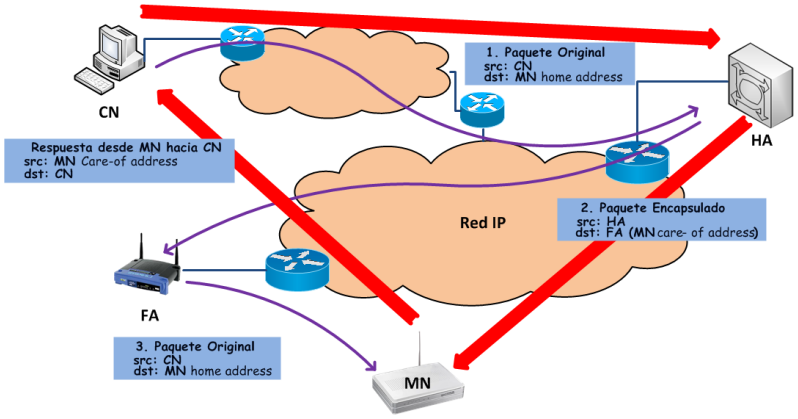


Figura 2.2: Ilustración del problema conocido como *triangular routing*

A pesar de ser más eficiente y de facilitar la gestión de la movilidad de los dispositivos, es posible que a veces no sea un sólo dispositivo el que se mueva, sino que sea la red completa. También es posible que considerar varios elementos que se mueven "juntos", como un todo, resulte más eficiente que gestionar la movilidad de cada elemento por separado. En la siguiente sección se explica con más detalle el funcionamiento del protocolo de movilidad de redes, que como se verá, difiere ligeramente del protocolo diseñado para gestionar la movilidad de los terminales.

## 2.3. Protocolo de soporte básico de movilidad de redes

Como ya se ha comentado en la sección anterior, los protocolos de movilidad diseñados para IPv4 [?] e IPv6 [JPA04], no contemplan el caso en el que el movimiento lo realice una red completa. Con el objetivo de ampliar el soporte de la movilidad a redes completas, en este caso en IPv6, el IETF (*Internet Engineering Task Force*) creó un grupo de trabajo, llamado NEMO, por las siglas en inglés de movilidad de redes (*NEtwork MObility*). Como resultado ha surgido el protocolo de soporte básico de movilidad de redes, NEMO BS

(***NE**twork **MO**bility **B**asic **S**upport*). En principio, el funcionamiento de este protocolo es similar al de MIPv6, pero se han introducido algunas diferencias como se verá a continuación.

En primer lugar, NEMO BS define la figura del router móvil (***M**obile **R**outer*, MR) como el elemento que proporciona conexión a la red móvil. Un escenario básico con una red móvil (gestionada por su MR), el router de acceso de la red visitada y el agente local en la red hogar, se muestra en la Figura 2.3.

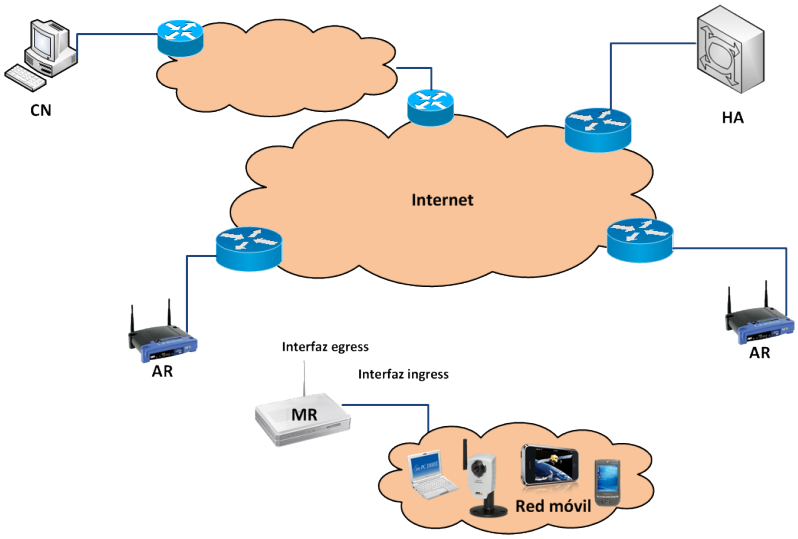


Figura 2.3: Escenario sencillo de movilidad de redes

El MR es el encargado de gestionar la movilidad en la red móvil, de forma que los nodos móviles no necesitan hacerse cargo del cambio de punto de acceso, que se realiza de forma transparente para ellos, liberándolos de carga computacional y reduciendo la señalización necesaria. Además, tampoco el router de acceso de la red visitada es consciente en ningún momento de la presencia de una red móvil al otro lado del router móvil, por lo que tampoco es necesario realizar en él ninguna modificación para dar soporte a la movilidad de redes.

Este router móvil proporciona a la red móvil conexión con el exterior, tanto si la red se encuentra en la red hogar (*Home Network*), como en una red visitada. El router móvil gestiona el registro con su agente local, la asociación con los routers de acceso de las redes que visita, y además el tráfico que cada uno de los nodos móviles intercambia con otros nodos.

El proceso mediante el cual el router móvil registra en su agente local la dirección a través de la cual la red móvil está localizable cuando está "fuera de casa", se basa en el intercambio de dos mensajes entre router móvil y agente local. El router móvil envía un mensaje *Binding Update* (BU), que contiene la dirección IPv6 que utilizará en la red visitada (la *Care-of Address*, o CoA). El agente local, una vez que ha almacenado la información en su lista de asociaciones (*Binding Cache*) envía un asentimiento, *Binding ACK*, para hacer saber al router móvil que todo ha ido bien. En ese momento, ambos establecen un túnel bidireccional IPv6 en IPv6, que encapsulará el tráfico entre la red móvil y la red hogar. De esta forma, la dirección destino de los paquetes dirigidos a un nodo de la red móvil será su dirección permanente, la que tiene sentido topológico en la red hogar. El agente local añadirá otra cabecera IPv6 con dirección destino la CoA del router móvil antes de enviar el datagrama por el túnel. El router móvil al recibirlo, eliminará la

cabecera exterior y entregará el datagrama tal y como era originalmente a su destinatario. En la Figura 2.4 se describe esta encapsulación de forma gráfica.

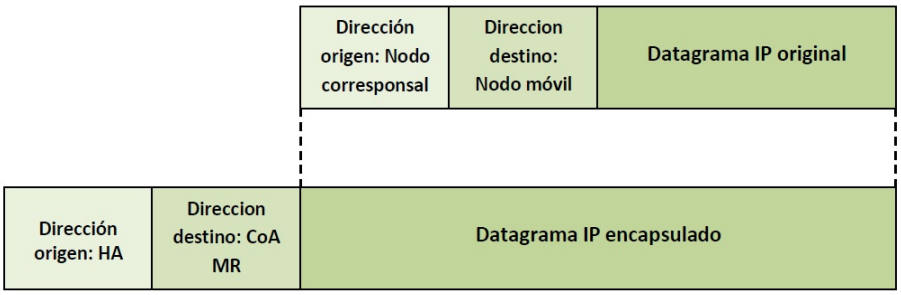


Figura 2.4: Encapsulación del tráfico dirigido a un nodo en la red móvil a través del túnel MR-HA

Esta forma de encaminar el tráfico de la red móvil, permite que los nodos conectados a ella puedan disfrutar de movilidad de forma totalmente transparente, sin ser conscientes de ello y sin estar obligados a tener soporte para ello.

Por otro lado, la gestión recae en el router móvil, lo que puede presentar un potencial punto de debilidad, ya que un fallo en este elemento puede dejar a la red móvil aislada, aunque esto depende mucho de la configuración de la red móvil y además podría también ocurrir estando en la red hogar, por lo que no constituye un riesgo añadido por la gestión de la movilidad.

En cambio, el mayor inconveniente del protocolo de movilidad de redes se encuentra en que todo el tráfico debe pasar por el agente local, lo que genera una ineficiencia ya que la ruta utilizada no es la óptima. En MIPv6 existe un procedimiento de optimización de rutas [JPA04], pero en NEMO BS esta optimización no está contemplada.

## 2.4. Aplicaciones de la movilidad de redes

En una sociedad como la actual, en la que los usuarios quieren, incluso necesitan, disfrutar de conexión a Internet en cualquier momento del día y en cualquier lugar, de forma continuada, surgen varias aplicaciones en las que un protocolo de movilidad de redes puede ser útil:

- Redes de área personal (PAN): distintos dispositivos electrónicos que una persona puede llevar encima (PDA, cámara de fotos, reproductor de música, etc.) que obtienen acceso a Internet a través de, por ejemplo, un teléfono móvil que actúa como router móvil.
- Conexión en un medio de transporte: la plataforma móvil permite mediante WiFi el acceso a Internet de los usuarios del medio de transporte.
- Redes vehiculares: al igual que en el transporte público, puede ser interesante dotar de acceso a Internet a los pasajeros que van en otra clase de vehículo, incluso dando origen a otro tipo de servicios, como permitir que hoteles, estaciones de servicio u otros elementos de interés elegidos por el usuario se "anuncien" al viajero en un radio de x kilómetros. En la Figura 2.5 se muestra un posible escenario de red vehicular.

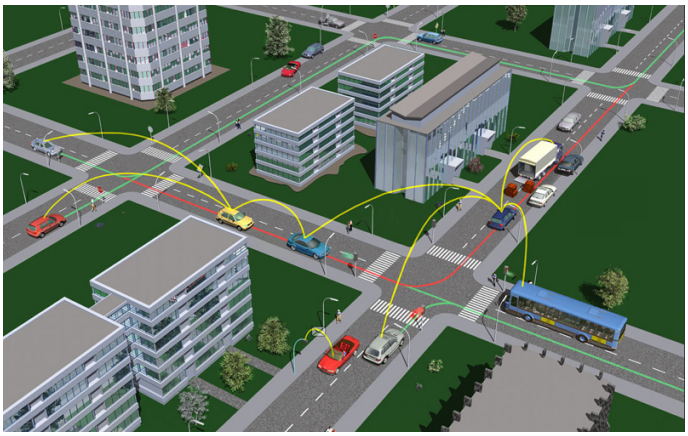

Figura 2.5: Escenario de una red vehicular.[car]

## 2.5. Conclusiones

NEMO BS presenta una solución válida para proveer movilidad a redes completas, facilitando así que nodos sin soporte explícito para ello puedan tener acceso a Internet en cualquier lugar y además en movimiento. Además, cada día surgen más aplicaciones para las redes móviles, como las redes de área personal o las redes vehiculares.

Aunque NEMO se considera un protocolo eficiente para gestionar la comunicación de una red móvil, existe un punto débil, que consiste en el encaminamiento de forma única a través de un túnel bidireccional entre la red hogar y la red móvil, sin ofrecer la posibilidad de adaptar esta ruta en casos en que se pudiera encontrar una más eficiente. A este respecto, en el presente proyecto fin de carrera se ha implementado una optimización de rutas para NEMO en redes ad-hoc vehiculares, que pretenden mejorar el encaminamiento cuando se detecte que un nodo con el que la red móvil quiere establecer una comunicación está presente en la misma red vehicular, mediante una ruta multisalto en la misma red ad-hoc, sin necesidad de enviarlo a través del túnel hasta la red hogar.

# Capítulo 3

# Redes vehiculares

## 3.1. Introducción

En la última década se ha producido un despegue de la investigación en el campo de las redes vehiculares, existiendo una gran cantidad de asociaciones entre diversas entidades (agencias gubernamentales, universidades, empresas) para producir numerosos avances. Originalmente, el objetivo era aumentar la seguridad vial, permitiendo a los vehículos comunicarse entre sí y con otras entidades de la infraestructura vial, para reducir el número de accidentes y sus consecuencias, así como ofrecer al conductor la posibilidad de anticiparse ante una incidencia ocurrida a cierta distancia. Pero la investigación ha abierto nuevas líneas de desarrollo, surgiendo la posibilidad de ofrecer servicios adicionales, como acceso a Internet, juegos en línea (para los pasajeros) u otras aplicaciones basadas en el entretenimiento, muy presentes en la industria de Internet. Sin embargo, las posibilidades no son sólo para los vehículos y el entretenimiento de sus ocupantes mientras viajan, también negocios en las inmediaciones pueden darse a conocer, o se puede recabar información sobre la ciudad de destino, qué hacer o qué ver, y descargar esta información directamente al sistema de navegación GPS en el vehículo, o al teléfono móvil del conductor, por ejemplo.

El interés en este tipo de redes se refleja, como ya se ha comentado, en los numerosos proyectos de investigación existentes tales como PRE-DRIVE C2X (2008-2010), Geonet (2008 - 2010), INTERSAFE-2 (2008-2011), SIM-TD (2009 - 2011), NoW (2006-2010) etc. Estos proyectos (Figura 3.1) tratan de dotar de capacidad de comunicación a los automóviles para la transmisión de mensajes, los cuales pretenden mejorar la seguridad en las carreteras.

Todo esto da una idea de la importancia de la investigación y desarrollo en el campo de las redes vehiculares.

Tecnológicamente, varias empresas de automoción están presentes en proyectos de investigación, colaborando con agencias e institutos tecnológicos para llegar a un consenso que aúne de forma eficiente todas las posibilidades de la forma más conveniente para todos. Las redes ad-hoc vehiculares (***Vehicular Ad-hoc NETworks***, VANETs) son la extensión de las redes ad-hoc móviles (MANET) al entorno vehicular. En este capítulo se va a presentar la situación actual de las redes vehiculares, así como sus principales características, ya que son un tipo especial de redes con algunos requerimientos más exigentes que una red convencional.


Figura 3.1: Diferentes proyectos y consorcios internacionales relacionados con las redes vehiculares

## 3.2. Características de las redes vehiculares (VANETs)

Debido al entorno en el que se desenvuelven este tipo de redes, tienen características especiales que las hacen diferentes de las redes inalámbricas ad-hoc convencionales, como son por ejemplo las redes de sensores. Las principales diferencias se detallan a continuación.

- Los vehículos cuentan con una reserva de energía mucho mayor que las que puede tener cualquier dispositivo móvil. La energía necesaria para los dispositivos de comunicaciones de la red puede obtenerse de las baterías instaladas en los vehículos y puede ser recargada.
- Los vehículos tienen unas características físicas (dimensiones, peso, capacidad) muy diferentes al tipo de dispositivos utilizados normalmente en redes inalámbricas, por lo que son capaces de soportar (y transportar) equipos más pesados y con capacidades computacionales mayores. Esto unido a las escasas restricciones de potencia, hacen posible instalar equipos más potentes, con mayor capacidad así como transmisores y receptores capaces de trabajar a tasas propias de redes cableadas.
- Los vehículos se mueven a distintas velocidades, lo que para una velocidad elevada, resulta en una comunicación inter-vehicular difícil de mantener por los frecuentes cambios de topología en la red. Sin embargo, existen ciertos patrones en el movimiento del tráfico que pueden ayudar a mantener la conectividad en un grupo de vehículos. Otros aspectos, como la provisión de calidad de servicio en distintos escenarios, requieren una tecnología de acceso al medio diseñada adecuadamente.
- Normalmente, hay vehículos en un área muy próxima y a unos pocos saltos de la infraestructura (WiFi, satélite, celular...), por lo que el acceso a Internet puede considerarse como una posibilidad factible a la hora de diseñar protocolos de red y aplicaciones.

## 3.3. Aplicaciones en Redes vehiculares

Las redes vehiculares permiten la aparición de aplicaciones muy innovadoras, algunas de ellas enfocadas a la compartición de archivos o P2P (*Peer-to-Peer*). Las previsiones de tener equipos sin restricciones de capacidad de procesado y almacenamiento en los

vehículos, hace posible el uso de este tipo de aplicaciones, al contrario que en otras redes ad-hoc tradicionales. A continuación se enumeran una serie de servicios de gran interés en el entorno vehicular, a modo de ejemplo.

1. Servicios vehiculares específicos: en esta categoría se encuentran las aplicaciones directamente relacionadas con el tráfico o con el estado del vehículo, como por ejemplo, descarga de la situación actual del tráfico, diagnóstico del vehículo, actualización del software instalado en el vehículo, información de aparcamiento, etc.

2. Servicios de acceso a Internet: cada vehículo debe facilitar la conexión a Internet, especialmente importante en medios de transporte público, por ejemplo, que ofrezcan conexión a sus pasajeros para poder consultar el correo o conectarse a una VPN (*Virtual Private Network*).

3. Servicios de comunicaciones personales: esta categoría acoge servicios tradicionales como llamadas de voz y vídeo, o sesiones de chat y mensajería instantánea. Algunos vehículos ya disponen de dispositivos manos libres integrados para utilizar junto al teléfono móvil. En un futuro se espera que los dispositivos instalados en el vehículo sean más sofisticados, permitiendo el uso de aplicaciones más complejas.

4. Servicios de entretenimiento: juegos en red, streaming multimedia, son algunos ejemplos de aplicaciones muy actuales que tienen también cabida en las redes vehiculares. Especialmente, para los pasajeros del vehículo, por ejemplo, los niños sentados en el asiento trasero viendo una película durante un viaje o una persona viendo un capítulo de su serie favorita mientras se desplaza a su lugar de trabajo.

5. Servicios *multicast*: en esta categoría se encuentra el acceso a información destinada a un grupo de usuarios. Este grupo de usuarios puede estar definido según la localización (información meteorológica, estado del tráfico, accidentes), bajo demanda (radio, televisión) o con otros fines (anuncios publicitarios de hoteles o restaurantes en las inmediaciones, etc.)

Al contrario de lo que se puede pensar en un primer momento, las aplicaciones P2P a las que se hacía mención antes no se refieren sólo a descarga de archivos o aplicaciones típicas. A continuación se incluye una lista de ejemplos reales, con el fin de ofrecer una visión de este nuevo sector de aplicaciones surgidas en las redes vehiculares.

1. Despliegue de redes vehiculares de sensores (VSN, *Vehicular Sensor Network*) para monitorizar condiciones ambientales y actividades sociales en zonas urbanas. La diferencia principal con las redes de sensores tradicionales consiste que los vehículos no tienen las mismas restricciones en cuanto a tamaño y energía. Así se pueden instalar potentes unidades de procesamiento y una gran variedad de sensores (químicos, acústicos, de vibración, luminosos), detectores, sistemas de posicionamiento (GPS) que permitan obtener información del entorno y procesarla en el mismo vehículo. Un ejemplo de este tipo de aplicación es *MobEyes* [LZG+06], que pretende ofrecer servicios de monitorización realizados por los propios vehículos, para detectar la ocurrencia de determinados eventos en las calles, mantener los datos almacenados de forma local, procesarlos (por ejemplo, reconocer números de matrícula) y enviarlos a vehículos próximos con un objetivo común (por ejemplo facilitar la posición de un determinado vehículo a las autoridades). Existen

otros ejemplos relacionados con las plataformas móviles de sensores, como CarTel [HBZ+06], Pothole Patrol propuesto por Eriksson (et al.) [EGH+08] o ZebraNet [JOW+02], aplicado en el Centro de Investigación Mpala en Kenya para rastrear la posición de los animales.

2. Streaming de vídeo de emergencia: V3 (*Vehicle-to-Vehicle Live Video*, V3) [GAZ05] ofrece soporte para streaming de vídeo basado en localización, para que los usuarios puedan ver vídeo de una región de interés remota. Para ello, se presupone que el vehículo cuenta con ordenador de a bordo, sistemas de comunicación inalámbrica, GPS y algunos de ellos, con una cámara de video para la transmisión. Esta transmisión podría resultar de utilidad en situaciones de emergencia, como desastres naturales, accidentes de tráfico, ataques terroristas, etc. para ayudar a los conductores a evitar el peligro y facilitar operaciones de rescate.

3. Mercadillo virtual en VANETs (*FleaNet*) [LPAG06]: esta propuesta considera la creación de un mercadillo virtual en redes vehiculares urbanas, de forma que los usuarios puedan compartir no sólo información relacionada con la seguridad vial, sino compartir intereses y encontrar un punto de encuentro, y por qué no, realizar transacciones de compra-venta. De la misma forma, no son sólo los vehículos los que pueden "anunciarse", sino también negocios en las proximidades.

4. Transferencia de información bajo demanda: *Vehicular Information Transfer Protocol* (VITP) [DNII05] establece el formato de los mensajes para transmitir solicitudes y respuestas sensibles a la localización, entre los nodos de una VANET. Con ellos, se puede solicitar y agrupar información referente al estado del tráfico, alertas o servicios en ruta. Es esencial en este protocolo de comunicaciones tener la información de localización actualizada en cada momento, sin embargo la información es ofrecida por cada nodo de acuerdo a una política de *best-effort*: cualquier nodo puede participar en la comunicación o salir del área de influencia antes de que el intercambio de información se complete.

5. Aplicaciones interactivas: *RoadSpeak* [SHSI08] propone un chat de voz en redes sociales vehiculares. Esta aplicación permitiría a los conductores formar grupos y comunicarse entre sí mediante mensajes de voz, utilizando una estructura cliente-servidor. El servidor estaría centralizado y los clientes tendrían que conectarse a través de la infraestructura de acceso a Internet (utilizando servicios 3G, WiFi, WiMax, etc).

Aunque, como se ha visto, las redes vehiculares ofrecen una gran variedad de aplicaciones basadas en la cooperación de los distintos vehículos, en algunos casos se hace imprescindible la existencia de una infraestructura fija de comunicaciones, normalmente situada en las inmediaciones de las vías de circulación. Esta infraestructura permitiría ofrecer a los vehículos una pasarela de conexión permanente a corta distancia y una arquitectura disponible en la que confiar distintas aplicaciones.

La infraestructura de red podría estar formada por diferentes dispositivos y tecnologías, como puntos de acceso 802.11 [iee07], receptores/transmisores vía satélite, servidores, unidades de carretera (RSU, *RoadSide Unit*) y estaciones base de redes celulares, desplegados a lo largo de la red vial.

En la siguiente sección se presentan brevemente las distintas tecnologías que pueden estar presentes en las redes vehiculares, desde la tecnología empleada en redes móviles celulares hasta la nueva variante de 802.11, en proceso de estandarización: 802.11p [iee10].

## 3.4. Tecnologías de comunicación inalámbricas en redes vehiculares

Existen varias tecnologías de acceso inalámbricas que pueden utilizarse en redes vehiculares. A continuación se presentan las principales, algunas de ellas existentes previamente, otras diseñadas específicamente para el entorno vehicular.

### 3.4.1. Redes celulares

Los sistemas celulares han sufrido una rápida evolución en los últimos años para adaptarse a la demanda de los usuarios. La gran inversión realizada por las operadoras ha hecho posible el despliegue de la infraestructura con gran número de estaciones base, repetidores y con cobertura en prácticamente todo el territorio. Sin embargo, como inconvenientes destacan que esa infraestructura es propiedad de los operadores de red y en ella basan su negocio, por lo que no está claro como podría utilizarse esto en una red vehicular y cómo encarecería el servicio, lo que podría incluso frenar su implantación. Además, el acceso mediante tecnología 3G presenta un retardo variable y no despreciable, así como anchos de banda bajos si se compara con otras tecnologías.

### 3.4.2. WiMAX (802.16-2004)

El objetivo de WiMAX (*Worldwide Interoperability for Microwave Access*) [iee04] es facilitar el acceso de banda ancha inalámbrico en la última milla, como alternativa a las redes cableadas (xDSL), proporcionando una tasa de datos considerable (alrededor de unos 30 Mbps), soporte para la movilidad (en la modificación 802.16e, conocida como *Mobile-WiMAX*) [iee05] y cobertura en una distancia de varios kilómetros. Además especifica distintos niveles de calidad de servicio y varios modos de acceso a nivel físico. Para algunas aplicaciones WiMAX constituye una alternativa viable, en competencia directa con las redes celulares, pero con una gran inversión. Sin embargo, también podría considerarse una alternativa a tener en cuenta para el despliegue de la infraestructura fija de red.

### 3.4.3. WLAN: 802.11a/b/g

Debido al bajo coste y a su gran aceptación, es muy probable que se convierta en la principal tecnología en las redes vehiculares. Entre sus ventajas cuenta con una banda de frecuencias de libre acceso, ancho de banda aceptable, distintos modos de comunicación (ad-hoc e infraestructura), resultando especialmente importante el modo ad-hoc para las redes vehiculares, que por el momento la sitúan como una firme candidata, siempre que se solventen ciertos aspectos relacionados con el retardo en el establecimiento de la comunicación y la gestión de los cambios de punto de acceso.

### 3.4.4. DSRC/WAVE/802.11p

DSRC (*Dedicated Short Range Communications*) es, a grandes rasgos, una adaptación de 802.11a. Además, la pila de protocolos para redes vehiculares está comenzando a ser estandarizada, por ejemplo, los niveles físico y de enlace se encuentran en proceso bajo

el estándar 802.11p [iee10], que es la modificación de 802.11 [iee07] para adaptarlo al contexto de las redes vehiculares. Los niveles superiores se encuentran definidos en la familia 1609 (WAVE, *Wireless Access in Vehicular Environments*), organizados en cuatro grupos principales:

- 1609.1- *WAVE Resource Manager*: define el nivel de aplicación.
- 1609.2- *WAVE Security Services*: establece los mecanismos de seguridad, autenticación, confidencialidad en la banda de frecuencias de DSRC[1].
- 1609.3- *WAVE Networking Services*: define las capas de red y transporte, incluyendo enrutamiento, gestión y configuración de direcciones.
- 1609.4- *WAVE Multichannel Operations*: se ocupa de la coordinación y gestión de la banda de frecuencias de DSRC.

En la Figura 3.2 se muestra la arquitectura con los distintos elementos.

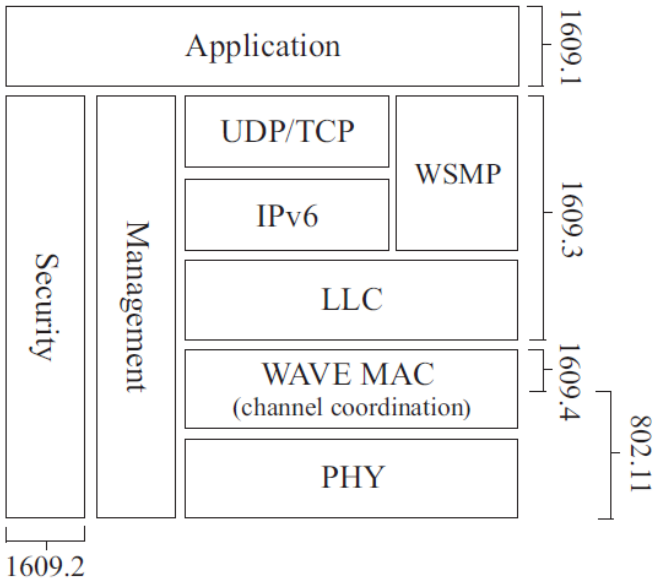


Figura 3.2: Pila de protocolos propuesta por IEEE involucrados en las comunicaciones vehiculares [HKHL10]

## 3.5. *Routing* en redes vehiculares

El enrutamiento es uno de los puntos clave en la investigación en redes vehiculares, ya que muchas aplicaciones requieren un camino multisalto entre vehículos. Las características propias de las VANETs (escenarios muy dinámicos, con grandes diferencias en velocidad y densidad de vehículos, además teniendo en cuenta el trazado de las carreteras o la presencia de intersecciones, y la participación de elementos fijos (*Roadside Units*, RSU) en las comunicaciones) hacen que el diseño de estos protocolos de cálculo de rutas sea un verdadero reto para los investigadores.

[1] Norteamérica: 5.850 - 5.925 GHz; Japón: 5.770 - 5.850 GHz; Europa: 5.875 - 5.905 GHz

A pesar de esto, las redes vehiculares no ofrecen sólo dificultades para el diseño de un protocolo de enrutamiento, también se pueden utilizar ciertas características exclusivas de estas redes de las que sacar partido, por ejemplo, conociendo la velocidad y la trayectoria de un vehículo se puede prever donde estará en el instante siguiente, se puede obtener información de su localización y también se tiene acceso a mapas. De forma general, hay un conjunto de aspectos clave que hay que tomar en consideración para diseñar una solución de forma ineludible:

- Escalabilidad: es uno de los principales requisitos. Las VANETs no tienen un límite en el número de vehículos presentes en la red, además este número puede ser muy variable, pasando por ejemplo, de un entorno urbano, con una gran densidad de vehículos a una carretera secundaria en un entorno rural. Idealmente, sería preferible que cada vehículo pudiera tomar sus propias decisiones de forma local, sin tener que conocer toda la topología de la red, lo que además reduce la sobrecarga de información de control.

- Descubrimiento de vecinos: es una parte fundamental en un protocolo de enrutamiento. Es habitual que cada nodo envíe de forma periódica anuncios indicando su posición o cualquier otra información relevante para informar a sus vecinos. Sin embargo, es crucial elegir bien el intervalo de envío de estos anuncios para poder mantener la información actualizada y controlar la sobrecarga en la red. Por ejemplo, se podría adaptar el intervalo de envío según las condiciones de la red (teniendo en cuenta la densidad de vehículos o el grado de movilidad).

- Identificación del destino: es necesario definir cual será el identificador de cada nodo. Existen protocolos de *routing* geográfico en los que cada nodo es identificado por su posición, por lo que es necesario conocer la posición del destino.

- Cálculo previo de la trayectoria: puede resultar interesante calcular o prever la trayectoria que será seguida por el vehículo y utilizar esa información para intentar establecer la ruta que seguirán los paquetes de datos que se envíen.

- Reenvío de datos: dado el escenario tan cambiante en una red vehicular, puede resultar ventajoso que la decisión del siguiente salto en la que se basan las comunicaciones en las VANETs (otros vehículos actúan como intermediarios entre dos nodos para facilitar la comunicación a través de un camino multisalto), se tome de forma independiente para cada paquete, según los nodos que estén más próximos en ese momento, o considerando cualquier otra métrica.

- División de la red: la densidad en una red vehicular puede ser muy desigual dependiendo del lugar o incluso de la hora del día. Aun en casos con tráfico denso, una red vehicular dada puede verse dividida por una intersección o un semáforo en rojo, por lo que será necesario establecer mecanismos para detectar, y más importante aún, rehacerse ante un fallo en la ruta por un vehículo que ha dejado de estar en las proximidades.

- Uso de información adicional: se espera que los vehículos sean capaces de usar su sistema de navegación y fuentes de información externas, que podrán ser utilizadas para los protocolos de comunicaciones también, teniendo una gran información no sólo de la posición del vehículo sino también del contexto.

Finalmente, se presenta una clasificación de los protocolos de diseño de rutas para VANETs, según el tipo de información utilizada (existen varias posibles clasificaciones). De acuerdo a este criterio, encontramos cuatro posibles grupos:

- Esquemas básicos: Funcionan sólo con información de los nodos vecinos. En este grupo se encuentran CAR [NG07] (*Connectivity-Aware Routing*) y GPCR [LMFH05](*Greedy Perimeter Coordinator Routing*).
- Basados en mapas: Se ayudan de un mapa para establecer un camino que permita llegar al destino. Ejemplos de este tipo de protocolos son SAR [THR03](*Spatially Aware Routing*) y GSR [LHT+03] (*Geographic Source Routing*)
- Basados en trayectoria: Trabajan con información sobre la trayectoria que será seguida por el vehículo para calcular el próximo siguiente salto. En esta categoría se encuentran GeOpps [LM07] (*Geographical opportunistic routing for vehicular networks*)y MoVe [LCGZ05] (*Motion Vector*)
- Basados en información de tráfico: Asumen que los vehículos pueden obtener información de la situación del tráfico, como por ejemplo de la densidad de vehículos. Existen varios ejemplos de protocolos de este tipo, como: A-STAR [SLL+04] (*Anchor-based Street- and Traffic-Aware Routing*), VADD [ZC08] (*Vehicle-Assisted Data Delivery*), MDDV [WFGH04] (*Mobility-centric Data Dissemination algorithm for Vehicular networks*) o SADV [DWX07] (*Static Node-Assisted Adaptive Routing Protocol*).

En la Figura 3.3 se muestra esta clasificación, para situar cada protocolo en su lugar de una forma más visual.


Figura 3.3: Clasificación de protocolos de enrutamiento en redes vehiculares

## 3.6. Conclusiones

Las redes vehiculares, o VANETs, proponen un nuevo escenario, con gran proyección y un futuro lleno de gran variedad de posibilidades, muchas de ellas con aplicaciones y servicios únicos para el entorno vehicular. La gran mayoría, así como la mayor parte de la investigación que se está llevando a cabo no es posible aún con las capacidades de comunicación de los vehículos actuales, por lo que aún queda mucho trabajo por delante. El futuro despliegue de las redes vehiculares dependerá de muchos factores, y del esfuerzo

conjunto por parte de las autoridades, los fabricantes de automóviles, las instituciones de investigación y la cooperación de otras entidades para dar a conocer los proyectos que se emprendan y facilitar su implantación en la sociedad. Sin embargo, actualmente la gran mayoría de la población tiene un automóvil, ó más en cada núcleo familiar, y además es habitual que muchas personas pasen mucho tiempo al cabo del año en su automóvil. En algunos casos, es un artículo de lujo, pero en otros en un elemento de uso cotidiano y todo el mundo quiere viajar lo más seguro y más cómodo posible. Por esta razón, si se realiza una labor de difusión adecuada, el despliegue de las redes vehiculares puede llegar a ser una realidad, cuando el despliegue tecnológico así lo permita, ya que resulta muy llamativo para los usuarios poder tener integrado en su vehículo esa cantidad de facilidades y propone una cantidad de servicios novedosos que pueden resultar atractivos a una gran variedad de sectores de la sociedad y del mundo empresarial.

La investigación está todavía muy abierta, aunque también hay gran número de participantes y de proyectos enfocados a facilitar el desarrollo de las redes vehiculares. Es importante llegar a puntos de encuentro para tomar decisiones comunes en un campo que puede afectar a la vida cotidiana de muchas personas.

# Capítulo 4

# Optimización de rutas para NEMO en redes vehiculares: VARON

## 4.1. Introducción

Tal y como se ha explicado en el capítulo 2, una vez que ha finalizado el proceso de señalización de NEMO y se ha establecido el túnel entre el HA y el MR, todo el tráfico dirigido al router móvil será encaminado a través de ese túnel, de tal forma que, aunque existiera una ruta más conveniente entre un nodo corresponsal y el router móvil (o uno de los nodos móviles gestionados por él) la ruta seguida sería a través del túnel, pasando por la red hogar.

Además, también se ha visto la aplicación directa de NEMO en el campo de las redes vehiculares. Dadas las características especiales de este tipo de redes, es sensato pensar que la comunicación a través de la red hogar puede suponer un retardo inasumible, además de ser una solución más lógica que dos nodos que están en la misma red vehicular puedan comunicarse directamente a través de ella.

Por esta razón, como una posible solución a esta ineficiencia surge VARON (*Vehicular Ad-hoc Route Optimization for NEMO*), diseñado para proporcionar una optimización de rutas basada en la que se realiza en MIPv6 y adaptándola al escenario y a las exigencias de seguridad de la red vehicular.

En este capítulo se describe el funcionamiento de esta propuesta y se explica el mecanismo de seguridad que permite a los routers móviles autenticar sus identidades. Además se describe cómo se realiza el enrutamiento una vez que se ha establecido una ruta multisalto en la red vehicular.

## 4.2. VARON: *Vehicular Ad-hoc Route Optimization for NEMO*

VARON propone una optimización de rutas para NEMO en entornos vehiculares, creando una red vehicular ad-hoc (VANET) de forma segura y utilizando esta red para optimizar las comunicaciones entre dos redes móviles.

Para ello, se asume que los routers móviles tienen, al menos, tres interfaces:

1. Una (o más) interfaces de entrada (*ingress*) para comunicar con los nodos dentro del vehículo, que pertenecen a la red móvil.
2. Una (o más) interfaces de salida (*egress*) para conectar con Internet.
3. Una interfaz ad hoc para comunicar con otros vehículos en las proximidades, que formen parte de la VANET.

La idea es que los vehículos puedan comunicarse y estén siempre disponibles a través de NEMO, y en el momento en que se detecte que un nodo, con el que se pretende establecer una comunicación o con el que se está ya comunicando, está presente en la red vehicular, se pueda redirigir el tráfico hacia él a través de una ruta multisalto en la red ad-hoc. De esta manera, nunca se perderá la comunicación a pesar de este cambio de configuración, ya que aunque el proceso para el establecimiento de la ruta puede tener una duración variable, como se verá en la sección 8.5, la comunicación se realiza a través de Internet hasta el momento en que se dispone de una ruta optimizada en la red vehicular. Un ejemplo del escenario básico, se muestra en la Figura 4.1.

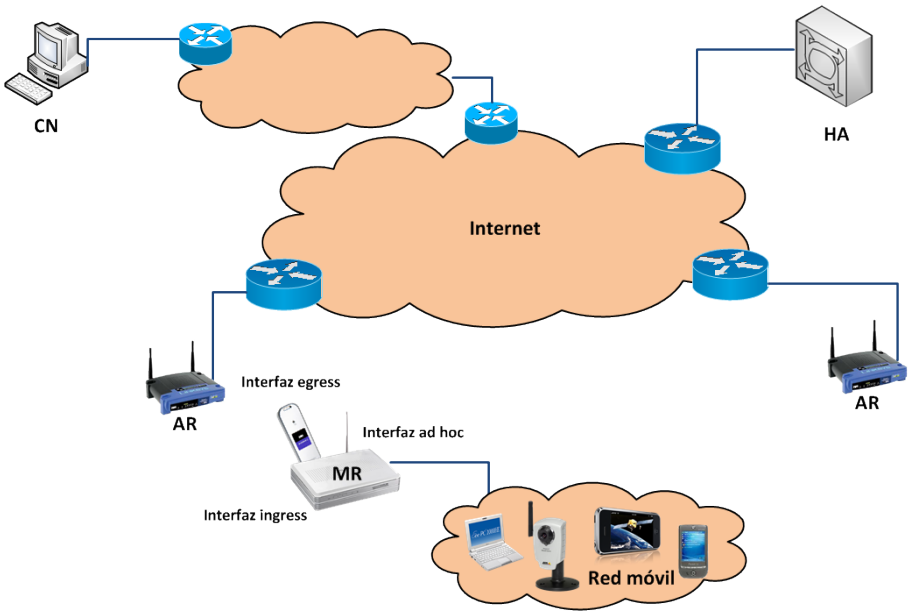


Figura 4.1: Escenario básico de movilidad de redes en la red vehicular

El router móvil es el elemento principal en el escenario propuesto por VARON. Como se explicará a continuación, es la entidad encargada de tomar las decisiones relacionadas con la optimización de la ruta, además de encargarse de todas las operaciones necesarias, de la generación y procesado de cada mensaje y del establecimiento de la ruta, ya sea como uno de los extremos de la comunicación, o como salto intermedio para facilitar la comunicación entre otro par de routers móviles.

El proceso de señalización se puede dividir en dos fases principales:

- Descubrimiento de prefijos de redes móviles: es necesario que cada router móvil anuncie en la red vehicular la presencia del prefijo de red móvil que gestiona.

- Creación de una ruta segura en la red ad-hoc: cuando se descubre que un prefijo de red con el que se quiere establecer una comunicación está disponible en la red ad-hoc, los dos routers móviles intercambian una secuencia de mensajes entre sí, que les permitirán por un lado, aprender una ruta hacia el otro y por otro lado, autenticar sus identidades mutuamente, para evitar suplantación de identidad u otros ataques relacionados con la seguridad.

A continuación se describen detalladamente las acciones llevadas a cabo por cada router en estas dos fases.

### 4.2.1. Descubrimiento de prefijos móviles

Cada MR debe anunciar por la interfaz ad-hoc (la que se conecta a la red vehicular) su prefijo de red móvil (MNP) para que todos los otros routers móviles puedan ser conscientes de la presencia de ese prefijo, y por tanto de la posibilidad de optimizar sus rutas hacia él. Este anuncio se realiza mediante el envío periódico de un mensaje, llamado *Home Address Advertisement*, (HoAA)[1]. Estos mensajes, dirigidos a la dirección *multicast* del grupo de todos los routers, contienen la dirección en la red hogar (HoA) del router móvil y un tiempo de vida asociado, para dotar a esa información de cierta temporalidad.

Los routers móviles al recibir estos mensajes deben reenviarlos mediante inundación, limitada por el valor del campo *Hop Limit* de la cabecera IPv6 [DH98], para propagar esa información al mayor número posible de nodos.

### 4.2.2. Establecimiento de una ruta segura en la red vehicular

Una vez que un router móvil ha descubierto la presencia en la red vehicular de un prefijo de red móvil, con el cual desea establecer una ruta a través de la red ad-hoc (por el motivo que sea, por ejemplo que ya estuviese comunicándose con él hasta ese momento a través de la infraestructura), debe enviar un mensaje CoRTI (*Care-of Route Test Init*) con el que informará de esa intención al otro router móvil. Este mensaje se envía a la dirección multicast de todos los routers. En la Figura 4.2 se muestra el intercambio de mensajes completo para la creación de la nueva ruta.

En el CoRTI, el router *origen* u *Originator MR*[2] incluye su HoA, la HoA del MR al que va dirigido el mensaje y las opciones relacionadas con la seguridad de la ruta, tales como las opciones de la firma y la dirección generada criptográficamente (ver apéndice B.3). Este mensaje, al igual que el HoAA, se reenvía mediante inundación, limitada por el campo *Hop Limit*, con la diferencia de que los routers intermedios aprenden una ruta hacia el router *origen*, a través del router del que reciben el mensaje.

El router *objetivo* (o *Target MR*), al recibir el CoRTI dirigido a él (lo sabrá tras comprobar que el campo *Target MR* coincide con su HoA) aprenderá una ruta hacia el MR *origen* y le enviará un mensaje CoRT (*Care-of route Test*), que viajará a través de la ruta aprendida por cada MR en el camino entre ambos. Este mensaje tiene prácticamente el mismo formato que el CoRTI, y también incluye información sobre las opciones de la

[1] El formato de todos los mensajes de VARON se describe en detalle en el Apéndice B.

[2] El router que comienza el establecimiento de la ruta es llamado *Originator MR* y el router con el que quiere establecer esa ruta es denominado *Target MR*. En esta memoria, por claridad, se les denominará MR *origen* y MR *objetivo* respectivamente.

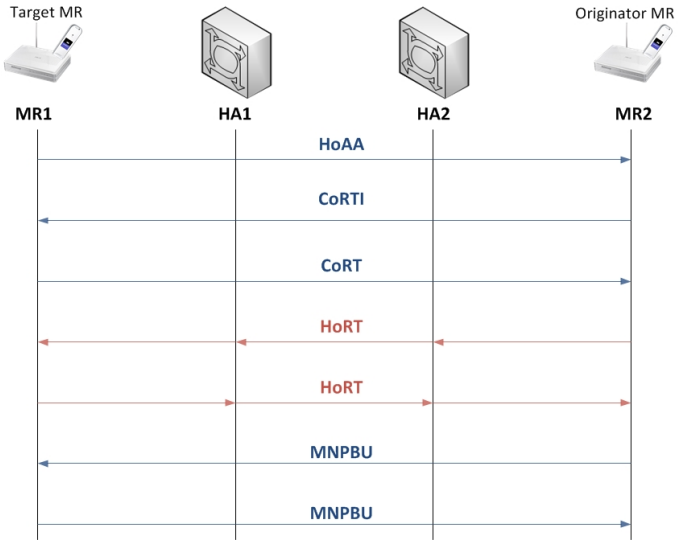


Figura 4.2: Intercambio de mensajes definido por VARON para crear la ruta en la red vehicular

firma y la dirección generada criptográficamente, para permitir a los demás nodos verificar la identidad del autor y la integridad del mensaje.

Cuando el router que inició el proceso recibe el mensaje CoRT envía un mensaje, denominado *Home Route Test* (HoRT) dirigido al router *objetivo* a través del túnel establecido con su red hogar. El objetivo de este mensaje, similar al enviado en el proceso de optimización de rutas de MIPv6, es comprobar que el router móvil está en realidad autorizado a gestionar el prefijo de red móvil que anuncia. Esta comprobación se basa en el hecho de que el router móvil debe recibir no sólo los mensajes dirigidos a su HoA en la red vehicular, sino también los mensajes dirigidos a una dirección perteneciente al prefijo de red móvil a través de la infraestructura, para demostrar que realmente el tráfico dirigido a esa red móvil pasa por él, redirigido por su agente local a través del túnel correspondiente. Cuando el router móvil recibe el mensaje HoRT, debe responder con otro mensaje del mismo tipo, que también viaja por el túnel hasta su red hogar para desde allí ser encaminado hacia la red hogar del otro MR, que, si es quién dice ser, lo recibirá a través de su túnel MR-HA. El último paso, es confirmar la recepción de todos los mensajes con el envío de un mensaje Mobile Node Prefix Binding Update (MNPBU), que contiene información encriptada con una clave que los dos routers móviles pueden formar a partir de información intercambiada durante el proceso (mecanismo de seguridad explicado en mayor detalle en la sección 4.2.2.1) de establecimiento de la ruta. El primero de los MR envía un MNPBU después de recibir el HoRT, y en ese momento, levanta un túnel cuyos extremos serán las HoA de cada MR y a través del cual irá encapsulado el tráfico entre la red móvil y el MNP con el que ha establecido la asociación. El MR del otro extremo responderá con otro MNPBU y levantará otro túnel para encapsular el tráfico dirigido a la otra red móvil.

#### 4.2.2.1. Autenticación de la ruta

El proceso de establecimiento de ruta a través de la red vehicular, se basa en el intercambio de varios mensajes que contienen además información criptográfica, la cual permite verificar la identidad de los routers móviles presentes en el proceso, así como

verificar la integridad de los mensajes que envían. Uno de los objetivos de este intercambio es evitar, por ejemplo, que un nodo malintencionado pueda entrometerse en la ruta haciéndose pasar por un MR encargado de gestionar un MNP, entre otros ataques. A continuación se describe en detalle como se realiza la codificación incluida en los mensajes, así como los mecanismos de firma digital que los routers móviles utilizan para confirmar sus identidades.

El mecanismo de seguridad se basa en el proceso llamado *Return Routability* de MIPv6, mediante el cual un nodo corresponsal puede asegurar, con un nivel de seguridad razonable, que un nodo móvil posee la HoA y la CoA que dice poseer, ya que sólo si recibe un mensaje enviado a cada una de las direcciones (que, como se ha explicado siguen caminos distintos a través de la red) será capaz de construir una clave que será utilizada por ambos (nodo móvil y nodo corresponsal) para autenticar el mensaje tras el cual ambos podrán establecer un túnel para comunicarse sin necesidad de ir a través de la red hogar.

Antes de describir el proceso y detallar la información incluida en cada mensaje de VARON, se va a introducir la terminología básica necesaria para entender mejor el procedimiento:

- Reto (*Nonce*): Es un número aleatorio generado y usado internamente por cada router móvil para la creación de un testigo de generación de claves (*Keygen token*). Debe permanecer secreto en el router móvil. No debe confundirse con el campo *Nonce* de los mensajes de VARON, que es un número aleatorio para identificar a un mensaje, distinguiéndolo de otras versiones del mismo tipo de mensaje, que se pueden recibir varias veces debido al mecanismo de inundación mediante el que se reenvían algunos mensajes.

- Índice de reto (*Nonce index*): El índice es utilizado para indicar qué reto ha sido utilizado para generar un testigo de generación de claves. Este índice permite por un lado, facilitar la identificación del testigo utilizado sin revelar el reto y, por otro lado, el router móvil puede distinguir entre los distintos retos que ha generado cuál es el utilizado en un determinado mensaje (como se explicará más adelante, el router móvil debe incluir estos índices y testigos de generación de claves en algunos de los mensajes del proceso de establecimiento de ruta).

- *Cookie*: Número aleatorio incluido en los mensajes para evitar que un nodo malicioso suplante la identidad de un router móvil.

- Testigo de generación de claves (*Keygen token*): Número generado a partir de un reto que será utilizado para calcular una clave, $K_{bm}$, generada para la autenticación del mensaje MNPBU. En el proceso de establecimiento de rutas cada MR utilizará dos testigos, uno incluido en el formato de los mensajes enrutados a través de la red hogar (*Home Keygen token*) y otro en los mensajes enviados por la VANET (*Care-of Keygen token*).

Además, en la formación de los mensajes y en el procesado de cada uno de ellos, tanto por los routers móviles de los extremos como de los nodos intermedios, se utiliza un mecanismo de firma digital (RSA - *Rivest, Shamir, Adleman*[3]) que se incluye en los mensajes CoRTI y CoRT, y una dirección IP generada criptográficamente (CGA -

[3]RSA: sistema criptográfico de clave pública. Es el más utilizado, y es válido tanto para cifrar como para firmar digitalmente.

*Cryptographically Generated Addresses*[4]) para asociar la dirección IP y la clave pública del router móvil, con el objetivo de verificar su identidad.

Con toda esta información, el siguiente paso es analizar el mecanismo de autenticación y verificación llevado a cabo por cada MR. Para ello, se va a describir el intercambio de testigos que tiene lugar durante el proceso de establecimiento de la ruta, mostrado en la Figura 4.3.

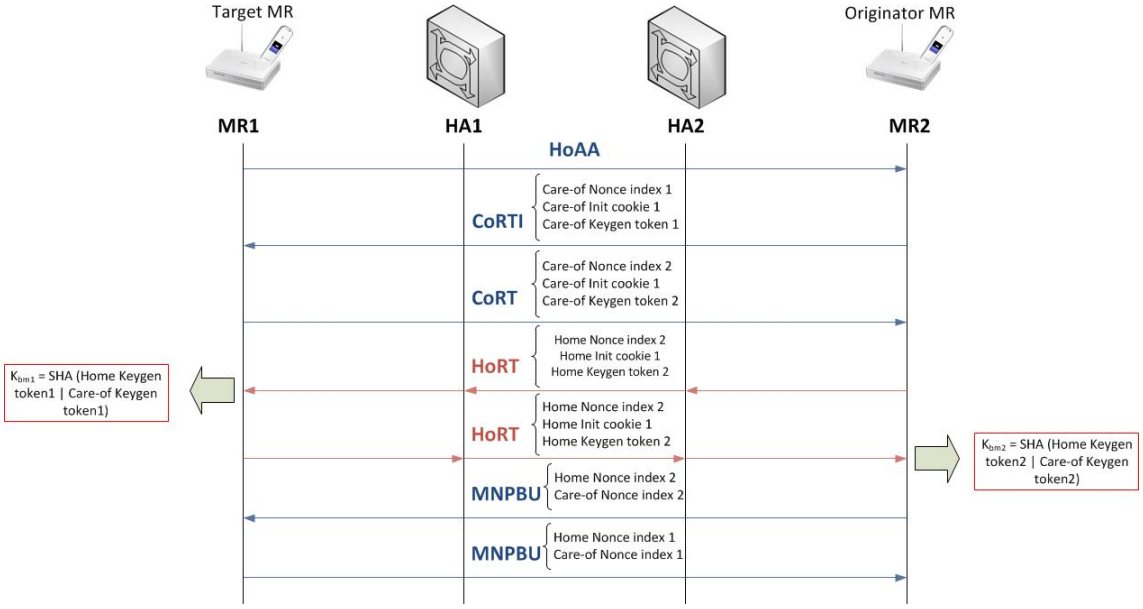


Figura 4.3: Intercambio de información en los mensajes de VARON para mecanismos de seguridad

El primero de los mensajes que contiene información criptográfica relevante es el CoRTI. En él, el MR *origen* incluye varios parámetros (los subíndices 1 y 2 son utilizados para diferenciar cuáles son generados por el MR *origen* y cuáles por el MR *objetivo*):

1. Índice del reto *Care-of*$_1$. El índice del reto utilizado en la generación del testigo *Care-of*$_1$, incluido en este mismo mensaje.
2. *Care-of init cookie*$_1$.
3. Testigo de generación de clave *Care-of*$_1$.

En el mensaje CoRT que el router móvil *objetivo* genera al recibir un CoRTI se debe copiar el campo *Nonce* (el utilizado para identificar el mensaje y distinguirlo frente a otros) del CoRTI recibido, además de incluir los siguientes parámetros:

1. Índice del reto *Care-of*$_2$. El índice del reto utilizado en la generación del testigo *Care-of*$_2$, incluido en este mismo mensaje.
2. *Care-of init cookie*$_1$. Debe copiarse del correspondiente CoRTI.

[4]CGA: dirección IPv6 que identifica a un equipo de la red, calculada a partir de una función *hash*. Es un procedimiento para asociar una clave pública a una dirección IPv6 según el protocolo SEND - Secure Neighbor Discovery Protocol

3. Testigo de generación de clave $Care\text{-}of_2$.

De forma análoga, en los mensajes HoRT, se incluyen los parámetros equivalentes a los anteriores pero generados para ser enviados a través de la infraestructura, por la llamada "ruta hogar" (*Home route*). En el primer HoRT, enviado por el router *origen*, se tiene:

1. Índice del reto $Home_1$. El índice del reto utilizado en la generación del testigo $Home_1$, incluido en este mismo mensaje.
2. *Home init cookie*$_1$.
3. Testigo de generación de clave $Home_1$.

En el segundo HoRT, generado por el router *objetivo* al recibir el mensaje anterior:

1. Índice del reto $Home_2$. El índice del reto utilizado en la generación del testigo $Home_2$, incluido en este mismo mensaje.
2. *Home init cookie*$_1$.
3. Testigo de generación de clave $Home_2$.

Tras la recepción de los respectivos HoRT, cada MR es capaz de generar la clave utilizada por el otro, $K_{bm1}$ y $K_{bm2}$, a partir de los testigos de generación de claves recibidos. El MR *objetivo* genera la clave 1 y el MR *origen* genera la clave 2. Estas claves son el resultado de aplicar el algoritmo SHA1 a la concatenación de *home keygen token* y *care-of keygen token* recibidos.

A continuación, en el MNPBU enviado por cada router móvil existe un campo llamado *Authenticator* que contiene las HoAs de ambos MRs codificadas con su clave. Al recibirlo, cada MR puede utilizar la clave del otro, que ha generado con la información recibida, para validar la identidad del otro MR.

Este mecanismo se basa en la presunción de que la ruta a través de la red hogar es segura, al igual que el proceso de optimización de rutas (*Return Routability*) de MIPv6. De esta manera, el mecanismo de seguridad seguido para autenticar el mensaje MNPBU se puede considerar seguro, ya que no introduce ninguna vulnerabilidad adicional, más que las ya existentes.

### 4.2.3. Encaminamiento a través de la nueva ruta

Una vez que el proceso de señalización para autenticar ambos MR y establecer la ruta ha finalizado, se crea un túnel bidireccional entre ambos routers móviles. Ahora el tráfico destinado a cada una de las redes móviles será encapsulado a través de ese túnel creado por VARON en la red ad-hoc, en lugar de ser enviado hacia la red hogar por el túnel creado por NEMO BS. Para ello, en el momento en que finaliza la señalización se crea una ruta específica para cada MNP, indicando que cualquier tráfico que se reciba con ese destino deberá ser enviado por el túnel creado por VARON. Es necesario mencionar que sólo los routers de los extremos han verificado sus identidades mutuamente. El resto de nodos intermedios, utilizados como saltos en el encaminamiento sólo han aprendido rutas

hacia distintos vehículos (identificados por su HoA) basándose en los mensajes CoRTI y CoRT recibidos durante el establecimiento de la ruta. Por ésta razón, el tráfico dirigido a los nodos pertenecientes al MNP gestionado por el router móvil se encapsula a través del túnel hasta el otro router móvil, porque sólo ellos se han autenticado como los nodos autorizados a gestionar el tráfico de ese MNP.

Por otro lado, si la dirección origen es la HoA del MR, ese tráfico no será encapsulado, sino que se enviará siguiendo el camino multisalto aprendido por todos los routers intermedios involucrados en el establecimiento de ruta.

El resto de tráfico intercambiado desde la red móvil con el exterior sigue utilizando la ruta existente anteriormente con NEMO, que sigue siendo válida y establece una ruta por defecto para que todo el tráfico para el que no se indique una ruta más específica, sea enviado a través del túnel hacia el HA, desde donde será reencaminado hacia su destino.

En la Figura 4.4 se detallan las rutas establecidas en cada router móvil para una comunicación entre un router móvil MR A y MR B, dependiendo de si el origen del tráfico es una dirección perteneciente a la red móvil o la HoA del router móvil.

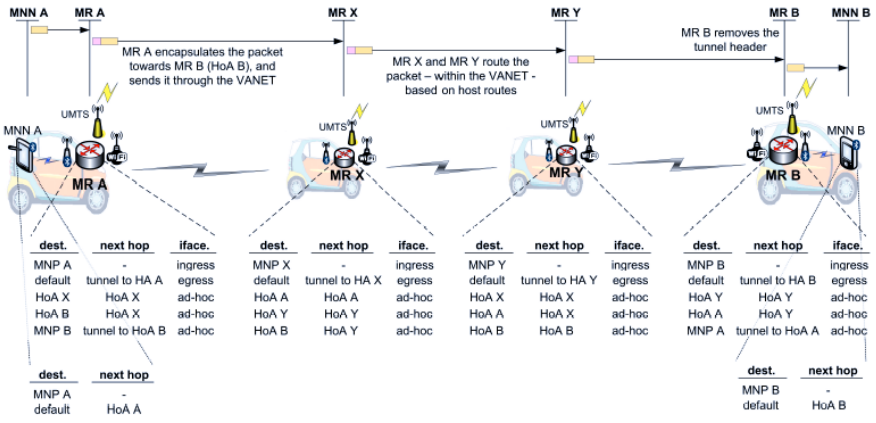


Figura 4.4: Tablas de rutas en cada MR de la red vehicular para una ruta entre A y B [Ber06]

## 4.3. Conclusiones

La movilidad de redes completas utilizando el protocolo NEMO BS carece de una optimización de rutas que permita encaminar el tráfico de manera eficiente, ya que todo el tráfico cuyo origen y/o destino sea la red móvil será encapsulado en un túnel bidireccional entre la red hogar y la red móvil (concretamente entre el agente local y el router móvil).

La solución presentada en este proyecto es una optimización de rutas adaptada al entorno de las redes vehiculares, basada en la formación de una red ad-hoc inalámbrica entre varias redes móviles, representadas por su router móvil. El router móvil se encarga de anunciar su propia HoA para darse a conocer al resto de vehículos. Cuando uno de ellos está interesado en optimizar la ruta hacia otro de los routers presentes en la red vehicular, se comienza un proceso de establecimiento de rutas al que se añade la implementación de ciertos mecanismos de seguridad para evitar posibles ataques. Una vez que ese proceso ha finalizado, se configura un túnel bidireccional entre los dos routers móviles, que pueden

comunicarse directamente o a través de un número indeterminado de nodos intermedios. Ese túnel estará destinado a encapsular el tráfico intercambiado entre ambas redes móviles, sin afectar al criterio establecido para encaminar el resto del tráfico.

En los próximos capítulos se describirán las distintas fases de desarrollo y el trabajo realizado para llevar a cabo una implementación del protocolo de optimización de rutas descrito, VARON.

# Parte III

# Trabajo realizado

# Capítulo 5

# Extensión de NEMO BS para varias interfaces inalámbricas

## 5.1. Introducción

Uno de los objetivos de este proyecto consiste en la implementación de un prototipo en un dispositivo real de bajo coste, para poder hacer una evaluación más realista. Dada la experiencia previa del Departamento en la utilización de OpenWrt, y la existencia de una implementación del protocolo de movilidad de redes diseñado para esta plataforma, se decidió utilizar un *hardware* igualmente soportado por OpenWrt. El principal obstáculo para encontrar el dispositivo adecuado fue que el router debía contar con más de una interfaz inalámbrica. El router Asus WL-500g Premium ofrecía la posibilidad de disponer de interfaces adicionales, utilizando un adaptador 802.11b/g conectado por USB.

En este capítulo se presenta el trabajo realizado para extender la funcionalidad de la implementación de NEMO BS para el caso de tener varias interfaces inalámbricas, conectando al router móvil un adaptador inalámbrico USB. Para ello, fue necesario elegir uno de los muchos existentes en el mercado, para que se ajustase mejor a las necesidades del proyecto. Una vez elegida la interfaz inalámbrica adicional se realizaron una serie de pruebas para evaluar su rendimiento.

Por otro lado, se comprobó de forma satisfactoria que la implementación de NEMO BS existente era compatible con el router ASUS WL-500g Premium, ya que inicialmente no fue diseñada para este dispositivo. Por último, se realizó la adaptación de esta implementación para ser utilizada en un dispositivo con varias interfaces inalámbricas.

## 5.2. Estudio de las interfaces inalámbricas USB

Con el objetivo de ampliar el número de interfaces inalámbricas del router ASUS WL-500g Premium y poder utilizar esas interfaces en los escenarios que se pretenden modelar, es necesario encontrar un adaptador USB que cumpla unos determinados requisitos:

- Compatibilidad con Linux.
- Compatibilidad con estándares 802.11b/g.

- Controlador del dispositivo incluido en OpenWrt.

Finalmente, se encontraron dos dispositivos que reunían todas estas características:

- WUSB54GC de Linksys.
- WLU-803G de Pheenet.

Ambos adaptadores pueden ser utilizados en equipos con distribuciones Linux, son compatibles con 802.11b/g y OpenWrt incluye soporte para ambos controladores. Las características técnicas, así como los pasos a seguir para instalarlos en OpenWrt y en las distribuciones de Linux utilizadas se describen en el apéndice D.2.

### 5.2.1. Evaluación del rendimiento de las interfaces inalámbricas USB

Dado que este tipo de interfaces no ha sido utilizado previamente y no se tiene ningún antecedente de su rendimiento en la red, se han realizado pruebas para conocer qué resultados se pueden esperar cuando se introduzcan en el escenario de pruebas. También es interesante saber hasta que punto la conexión de otra interfaz inalámbrica interfiere ó no en el funcionamiento normal del router. El conjunto de pruebas realizadas se puede dividir de la siguiente forma:

- Funcionamiento en un PC: Inicialmente, para evaluar de forma aislada el funcionamiento del adaptador USB y evaluar posibles restricciones debidas a limitaciones del router móvil, se realizaron pruebas conectándolos a un PC en lugar de al router ASUS. El escenario de la prueba se observa en la Figura 5.1. Se contemplaron diversas variaciones dentro del mismo escenario:
  - Las dos interfaces conectadas en los canales 1 y 11.
  - Configuración en modo ad-hoc y configuración en modo Punto de acceso/Estación. (Figuras 5.1 y 5.2)

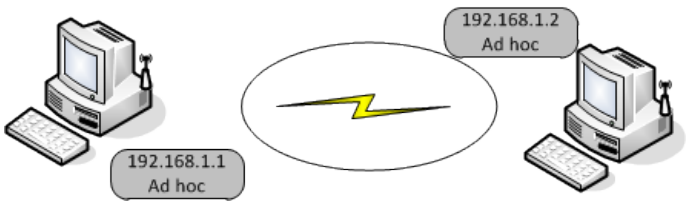


Figura 5.1: Escenario de prueba con un adaptador conectado en un PC (mod ad-hoc)

- Rendimiento en una red ad-hoc multisalto: El escenario de la prueba realizada se muestra en la Figura 5.3. El mismo escenario fue utilizado con diversas variaciones:
  - Todos los elementos conectados en el mismo canal (en los canales 1, 5 y 11).
  - Elementos conectados en distintos canales que se solapan: canales 1, 3 y 5.
  - Elementos conectados en distintos canales que no se solapan: canales 1, 6 y 11.

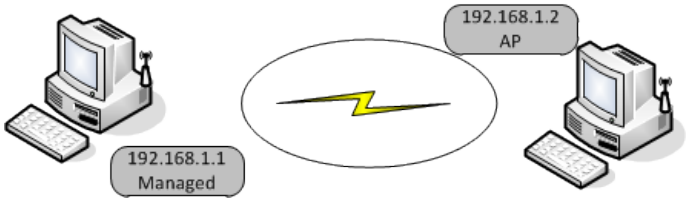


Figura 5.2: Escenario de prueba con un adaptador conectado en un PC (modo infraestructura)

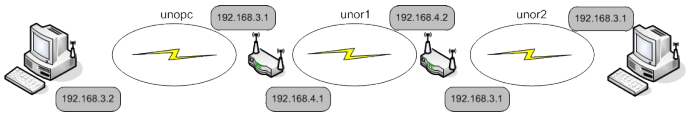


Figura 5.3: Escenario de prueba red ad-hoc con routers ASUS y adaptadores USB

- Rendimiento en una red en modo infraestructura multisalto: El escenario de la prueba realizada se muestra en la Figura 5.4. El mismo escenario fue utilizado con diversas variaciones:
  - Todos los elementos conectados en el mismo canal (en los canales 1, 5 y 11).
  - Elementos conectados en distintos canales que se solapan: canales 1, 3 y 5.
  - Elementos conectados en distintos canales que no se solapan: canales 1, 6 y 11.

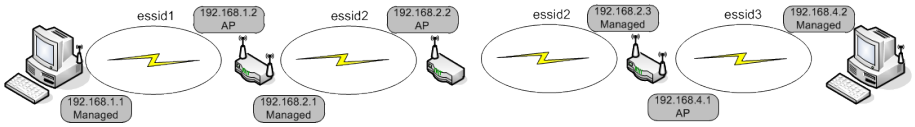


Figura 5.4: Escenario de prueba red en infraestructura con routers ASUS y adaptadores USB

Cada una de las pruebas anteriores se ha realizado para los dos adaptadores USB, Linksys y Pheenet.

Los resultados obtenidos se describen a continuación. En la Tabla 5.1 se muestra la tasa obtenida entre dos PC's conectados con la interfaz inalámbrica USB en modo ad-hoc.

| | |
|---|---|
| **Canal 1** | 12.18 Mbps ±83.6 Kbps |
| **Canal 11** | 14.433 Mbps ±776.75 Kbps |

Tabla 5.1: Medidas de la tasa de envío con los adaptadores USB conectados en modo ad-hoc

En la Tabla 5.2 se muestra la tasa obtenida entre dos PC's conectados con la interfaz inalámbrica USB conectada en modo estación a la interfaz inalámbrica del PC, en modo AP.

| | |
|---|---|
| **Canal 1** | 13.3 Mbps ±1.82 Mbps |
| **Canal 11** | 20.9 Mbps ±800 Kbps |

Tabla 5.2: Medidas de la tasa de envío con el adaptador USB conectado en modo estación

Como se puede apreciar, los resultados con los adaptadores inalámbricos USB conectados a un PC muestran el rendimiento esperado para una transmisión UDP a 35 Mbps. En la Tabla 5.3 se recogen los resultados obtenidos para las redes multisalto en modo ad-hoc y en modo infraestructura, con la interfaz inalámbrica USB ***Linksys*** conectada en los routers móviles. Los mismos resultados, pero utilizando el adaptador USB ***Pheenet*** se muestran en la Tabla 5.4. Como se puede apreciar, los resultados obtenidos son mucho más pobres que para el caso de utilizar un PC. En esta caída del rendimiento influyen varios factores, por un lado, la limitación del router móvil, cuya capacidad es menor que la de un ordenador. Por otro lado, las versiones de los controladores instaladas en los ordenadores, permiten configurar parámetros que no ha sido posible modificar en el router Asus. Por último, la conexión USB, o mejor dicho, la velocidad de procesamiento del router móvil para gestionar la conexión del puerto USB también ha podido influir.

| **Escenario** | **Modo ad-hoc** | **Modo Infraestructura** |
|---|---|---|
| **Todos canal 1** | 2.34 Mbps ±1.43 Mbps | 4.28 Mbps ±1.54 Mbps |
| **Todos canal 5** | 2.48 Mbps ±0.917 Mbps | 2.588 Mbps ±0.21 Mbps |
| **Todos canal 11** | 1.64 Mbps ±0.946 Mbps | 1.76 Mbps ±0.5 Mbps |
| **Canales 1/3/5** | 9.652 Mbps ±73.11 Kbps | 1.45 Mbps ±1.33 Mbps |
| **Canales 1/6/11** | 9.81 Mbps ±57.16 Kbps | 3.91 Mbps ±1.7 Mbps |

Tabla 5.3: Medida de la tasa de envío con el adaptador USB Linksys conectado en los routers móviles

| **Escenario** | **Modo ad-hoc** | **Modo Infraestructura** |
|---|---|---|
| **Todos canal 1** | 3.06 Mbps ±2.56 Mbps | 8.3 Mbps ±0.54 Mbps |
| **Todos canal 5** | 646.4 Kbps ±331 Kbps | 9.35 Mbps ±1.42 Mbps |
| **Todos canal 11** | 1.647 Mbps ±670.3 Kbps | 9.47 Mbps ±0.3 Mbps |
| **Canales 1/3/5** | 3.57 Mbps ±935 Kbps | 9.07 Mbps ±0.35 Mbps |
| **Canales 1/6/11** | 4.03 Mbps ±257 Kbps | 8.6 Mbps ±0.47 Mbps |

Tabla 5.4: Medida de la tasa de envío con el adaptador USB Pheenet conectado en los routers móviles

## 5.3. Implementación de NEMO BS y escenario de pruebas

Una vez que se ha comprobado el funcionamiento de las interfaces inalámbricas USB, el siguiente paso consiste en asegurar que el soporte de movilidad de redes puede ser utilizado en los routers ASUS. Los adaptadores USB serán utilizados como interfaces adicionales para conseguir extender la funcionalidad del soporte de movilidad de redes, que ya estaba implementado como parte del trabajo de otro proyecto fin de carrera anterior [SG10]. En ese proyecto se desarrolló una implementación del router móvil (MR) y del agente local (HA) de acuerdo a lo definido en [DWPT05]. Debido a que el *software* desarrollado para el MR fue inicialmente diseñado para otro dispositivo, era necesario comprobar primero su funcionamiento en el router ASUS. A su vez, para realizar esta prueba, es necesario desplegar un escenario similar al escenario de movilidad de redes para el que fue ideado. En este escenario, se necesita contar con los siguientes equipos:

- Ordenador portátil: Este equipo se utiliza como agente local.

- Router Linksys WRT54GL: Este equipo será el router de acceso al que se deberá conectar el router móvil en la red visitada. Para extender la funcionalidad de la aplicación al caso con varias interfaces inalámbricas se utilizarán dos routers de acceso, para poder realizar el cambio de punto de acceso según la calidad del enlace.

- Router ASUS WL-500g Premium: Este equipo será el router móvil.

A grandes rasgos, la implementación para el soporte básico de movilidad de redes se encarga de gestionar todo el proceso de señalización necesario para que el router móvil, al entrar en la red visitada, envíe una asociación a su agente local y éste encamine todo el tráfico dirigido a la red móvil hacia su nueva dirección en la red visitada (*Care-of Address*), creando un túnel entre agente local y router móvil. El escenario desplegado se muestra en la Figura 5.5.

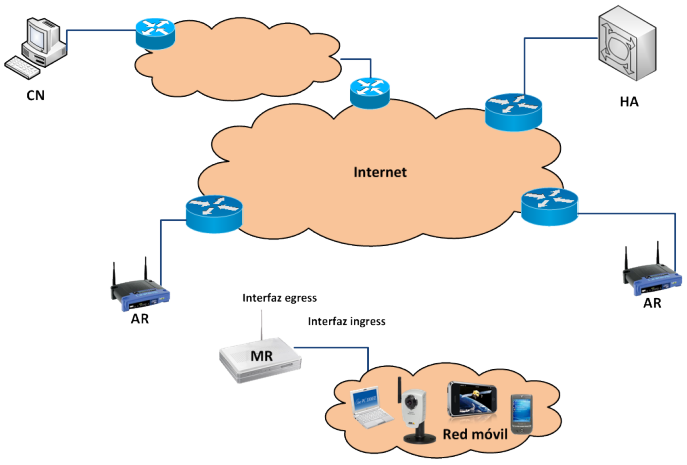


Figura 5.5: Escenario básico de movilidad de redes

Para que este *software* funcione correctamente, además de los archivos ejecutables del agente local y el router móvil, se necesitan unos ficheros de configuración en ambas entidades. Por su parte, los routers de acceso están pre-configurados para facilitar el funcionamiento de la aplicación. Una vez que se ha comenzado con la ejecución de ambos programas, lo único que necesita el router móvil para comenzar el proceso de señalización de NEMO BS es recibir un *Router Advertisement* del router de acceso de la red visitada, que anuncia el prefijo de red a la que puede conectarse, configurando así su *Care-of Address* o CoA.

Aunque la implementación del *software* del router móvil fue diseñada y optimizada para otro tipo de router, si los ficheros de configuración cumplen con el formato establecido, la aplicación funciona sin problemas en el router ASUS sin necesidad de hacer ninguna modificación, a pesar de tener otra arquitectura y distinto número de interfaces que el router móvil original.

## 5.4. Adaptación de NEMO BS para varias interfaces inalámbricas

Una vez que se ha conseguido instalar, por una parte el *software* para el soporte básico de movilidad de redes y por otra, la interfaz inalámbrica adicional mediante un adaptador USB, el siguiente paso consiste en unir estas dos funcionalidades. El objetivo final es tener la capacidad de gestionar la movilidad de la red, de forma que si estando conectado a una red visitada a través de uno de las interfaces inalámbricas, la calidad del enlace se sitúa por debajo de un determinado umbral, se pueda buscar un nuevo punto de acceso que proporcione un enlace de mayor calidad conectándose a través de la otra interfaz. Inicialmente, la conexión es la que se muestra en la Figura 5.6.

Para conseguirlo, se ha desarrollado un *script* para detectar cuando la calidad del enlace es menor que el umbral establecido y en ese momento, se configura la otra interfaz inalámbrica y se impide la recepción de *Router Advertisements* por la interfaz conectada al enlace de mala calidad. Se adoptó esta solución ya que utilizar un *script* en lugar de modificar el código de la implementación de NEMO, proporcionaba mayor flexibilidad para realizar las modificaciones, ya que no es necesario realizar compilación cruzada y es posible acceder directamente a la configuración de las interfaces inalámbricas. Para poder obtener el comportamiento deseado, era imprescindible realizar alguna modificación sobre el *software* existente, ya que éste *software* escucha en todas las interfaces *egress* del router móvil para recibir *Router Advertisements* y detectar el movimiento de la red. Por tanto, al tener dos interfaces es necesario gestionar de alguna forma cuál es la interfaz a través de la cual se quiere conectar al punto de acceso de una red visitada, así como compaginar la conexión existente con la utilización de la otra interfaz cuando baja la calidad del enlace.

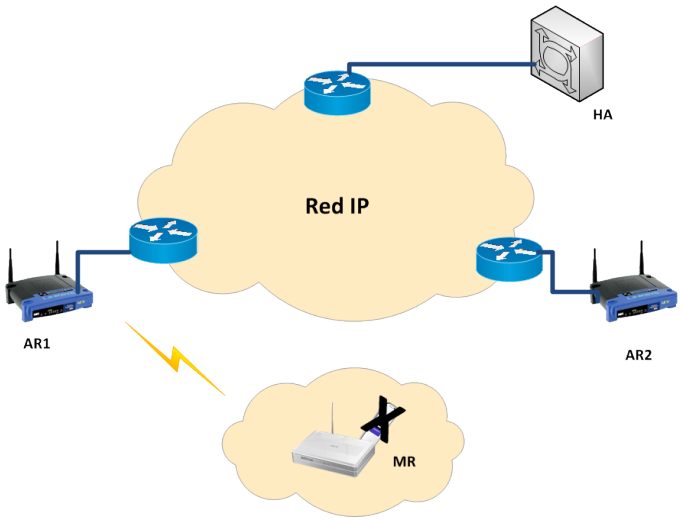


Figura 5.6: Escenario de prueba con dos interfaces inalámbricos

Entonces, el programa de NEMO BS, cuando se reciba un anuncio a través de la "nueva" interfaz, comenzará el proceso de señalización necesario al detectar que se ha cambiado de punto de acceso, modificando también su *Care-of Address*, por lo que será necesario informar a su agente local para que actualice en su tabla de asociaciones la entrada relacionada con ese MR y modifique el túnel. La Figura 5.7 muestra la conexión tras el cambio de punto de acceso e interfaz.

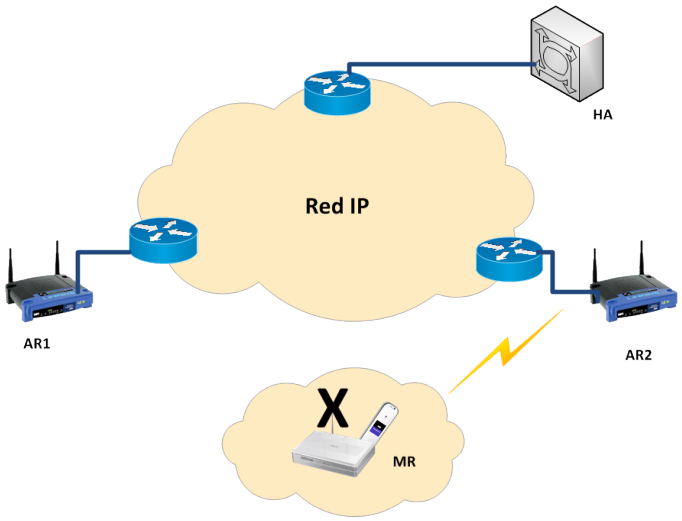


Figura 5.7: Escenario de prueba con dos interfaces inalámbricos tras detectar enlace de mala calidad y conectarse a un nuevo punto de acceso mediante el interfaz adicional

También se comprueba que la calidad del enlace a través de la segunda interfaz está por encima de un determinado umbral. Estos umbrales dependen de la interfaz inalámbrica en cuestión, ya que por características y especificaciones técnicas, ambos dispositivos tienen distinta sensibilidad de recepción, por lo que los umbrales no pueden ser iguales para ambos.

## 5.5. Conclusiones

La posibilidad de contar con varias interfaces inalámbricas en el router móvil es un aspecto fundamental para la realización de este proyecto. Por un lado para poder adaptar la implementación de NEMO existente, lo que proporciona una mayor libertad para futuras aplicaciones en las que varíe el escenario de evaluación. Por otro lado, para poder implantar la red móvil en un entorno vehicular es imprescindible, con la solución propuesta, contar con más de una interfaz inalámbrica.

Antes de poder utilizar directamente los adaptadores USB en el prototipo de la optimización de rutas desarrollado, es necesario conocer su comportamiento, para poder aislar posibles errores o fallos en el rendimiento de la aplicación.

A este respecto, se ha visto que los adaptadores USB imponen algunas restricciones en la configuración, pero su funcionamiento es adecuado para esta aplicación. Quizá en un futuro próximo, existan nuevos dispositivos en el mercado, que permitan extender esta funcionalidad.

# Capítulo 6

# Desarrollo y estructura del *software*

## 6.1. Introducción

En este capítulo se va a explicar la estructura y el funcionamiento del código en C desarrollado para implementar la optimización de rutas propuesta por VARON [Ber06]. Además también se incluyen otras líneas de desarrollo necesarias para poder probar la funcionalidad de VARON, como son el soporte de movilidad de redes (implementado en otro proyecto fin de carrera [SG10]) y la realización de operaciones criptográficas que se llevan a cabo para generar y/o procesar algunos de los mensajes.

El protocolo de movilidad de redes constituye una parte fundamental para la implementación realizada, ya que una parte de la señalización en el proceso de establecimiento de la nueva ruta, se basa en el envío de un mensaje a través del túnel entre cada MR y su HA. Para establecer este túnel deben haber intercambiado previamente un mensaje BU y un mensaje BA.

Por otro lado, el mecanismo de optimización de rutas diseñado en VARON no sólo pretende establecer una ruta más eficiente para las comunicaciones de la red móvil sino también hacerlo de una forma segura, o al menos, sin añadir ninguna posible vulnerabilidad a las ya existentes. Para ello, se incluyen opciones como la firma digital y el uso de direcciones generadas criptográficamente (CGA). Estas opciones de seguridad consumen recursos y tiempo en el router móvil, más aún teniendo en cuenta las limitaciones de los dispositivos utilizados. Por esta razón, se han decidido incluir estos retardos en la implementación realizada.

## 6.2. *Software* desarrollado

La aplicación desarrollada debe ocuparse de realizar toda la señalización descrita en el Capítulo 4 (VARON). Por simplicidad y por comodidad a la hora de depurar y aislar errores se ha dividido en distintos bloques, cada uno de ellos encargado, por ejemplo, de recibir uno de los mensajes y generar la respuesta apropiada, así como de aprender la ruta correspondiente o levantar el túnel entre los dos MR. En las siguientes secciones se describe cada uno de estos bloques.

Además, aunque no se han implementado en el código desarrollado las distintas

operaciones criptográficas utilizadas para firmar, validar, autenticar mensajes, etc. en el procesado y generación de los mensajes, todas las operaciones se han simulado utilizando *OpenSSL*[1]. Esta simulación ha permitido calcular el tiempo empleado por el router móvil para llevar a cabo cada operación (firma digital, *hash*[2], validación de firma, etc.). Debido a su complejidad, y dado que el objetivo principal es analizar la viabilidad y rendimiento de la solución desarrollada, no se realizó la implementación de estas operaciones. Sin embargo, conocer el tiempo empleado por el router móvil en realizar estas operaciones es importante para determinar los retardos que se introducen a la hora de procesar y generar los distintos mensajes.

### 6.2.1. Envío de HoAA

El primer módulo es el encargado del envío periódico de los mensajes HoAA (*HoA Advertisement*). Este mensaje será enviado por cada router móvil en la red ad-hoc, a la dirección *multicast* que agrupa a todos los routers. El objetivo de este mensaje es anunciar la HoA del router móvil que lo envía, para informar al resto de dispositivos presentes en la red de que ese prefijo de red móvil (***M**obile **N**etwork **P**refix*, MNP) está disponible a través de la red ad-hoc. El mensaje lleva asociado un número de secuencia y un tiempo de vida, que limita el tiempo de validez del mensaje. El formato se muestra en detalle en el Apéndice B. La Figura 6.1 muestra los diagramas de flujo de este módulo.

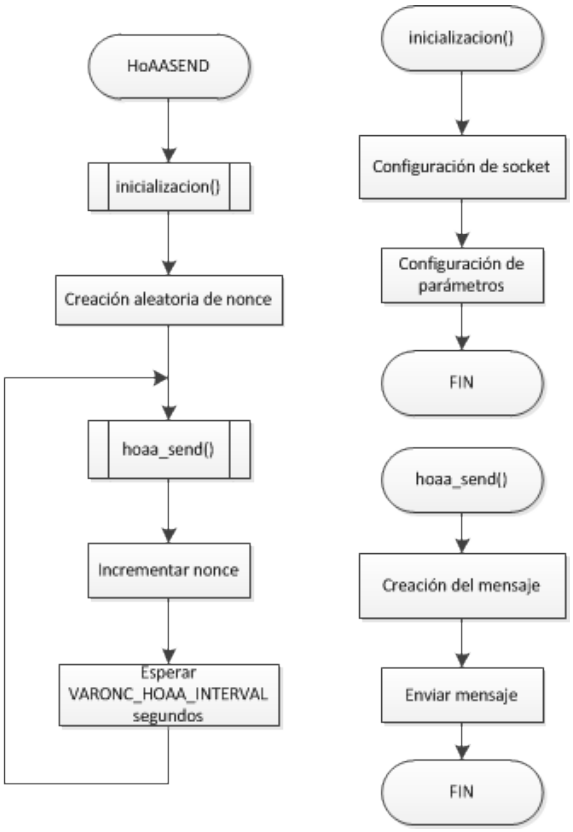


Figura 6.1: Diagramas de flujo de las funciones para el envío periódico del HoAA

[1] http://www.openssl.org

[2] *Hash*: resumen para generar claves o llaves que representen de manera casi unívoca a un documento o registro.

El comando para lanzar la ejecución de este bloque admite como parámetro el número de segundos entre el envío de dos mensajes consecutivos. Si no se indica ningún intervalo se utiliza el valor por defecto, `VARONC_HOAA_INTERVAL`, igual a 20 segundos. Un ejemplo de la ejecución de este módulo se puede observar en la Figura 6.2.

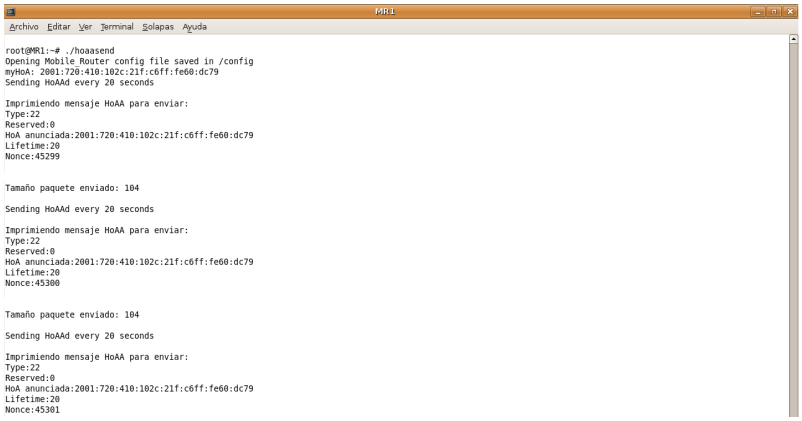


Figura 6.2: Ejemplo de ejecución del módulo de envío de HoAA

### 6.2.2. Recepción de HoAA

Este módulo se encarga de recibir los anuncios periódicos o HoAA y comenzar, si se está interesado, el proceso para establecer una ruta a través de la red vehicular, o *Care-of route*, con ese MR. Para tomar esa decisión, se debe incluir en el fichero de configuración inicial una lista de prefijos de red con los que establecer una comunicación si fuera posible a través de la red ad-hoc. En el momento de recibir un anuncio de HoA, el router móvil debe comprobar si el prefijo anunciado se encuentra en su lista de posibles objetivos. Si es así, el router enviará un mensaje CoRTI para comenzar el proceso de establecimiento de una ruta segura con el MR del que ha recibido el anuncio, tras haber realizado una serie de comprobaciones sobre el mensaje. En un escenario real, existirá algún mecanismo para decidir hacia qué prefijos se debe optimizar la ruta en la red vehicular, en lugar de utilizar una lista pre-configurada.

Para realizar estas comprobaciones, el router almacena en una tabla la dirección anunciada, el tiempo de vida y el número de secuencia enviados en cada mensaje recibido. El número de secuencia sirve para distinguir los mensajes y asociarlos unívocamente a la HoA anunciada, ya que el mismo mensaje es reenviado por distintos MRs, para facilitar que todos los routers conozcan la existencia de ese prefijo en la red. Si se recibe un HoAA con el mismo número de secuencia que otro ya recibido anteriormente, el mensaje se descarta. Si por el contrario, se recibe un HoAA anunciando la misma HoA pero con distinto número de secuencia, se considera un mensaje distinto, que sirve para refrescar y actualizar esa entrada en la tabla. De esa forma se refresca su tiempo de vida en la tabla y se reenvía ese mensaje, si el campo *Hop Limit* lo permite. En la Figura 6.3 se muestran los diagramas de flujo de las funciones implementadas en este bloque.

Si se ha decidido comenzar el proceso para establecer la ruta por la red ad-hoc, se envía un mensaje CoRTI. Este mensaje se envía también a la dirección multicast del grupo de todos los routers. El formato se describe con más detalle en el Apéndice B. En la Figura 6.4 se muestra la salida por pantalla de este módulo.

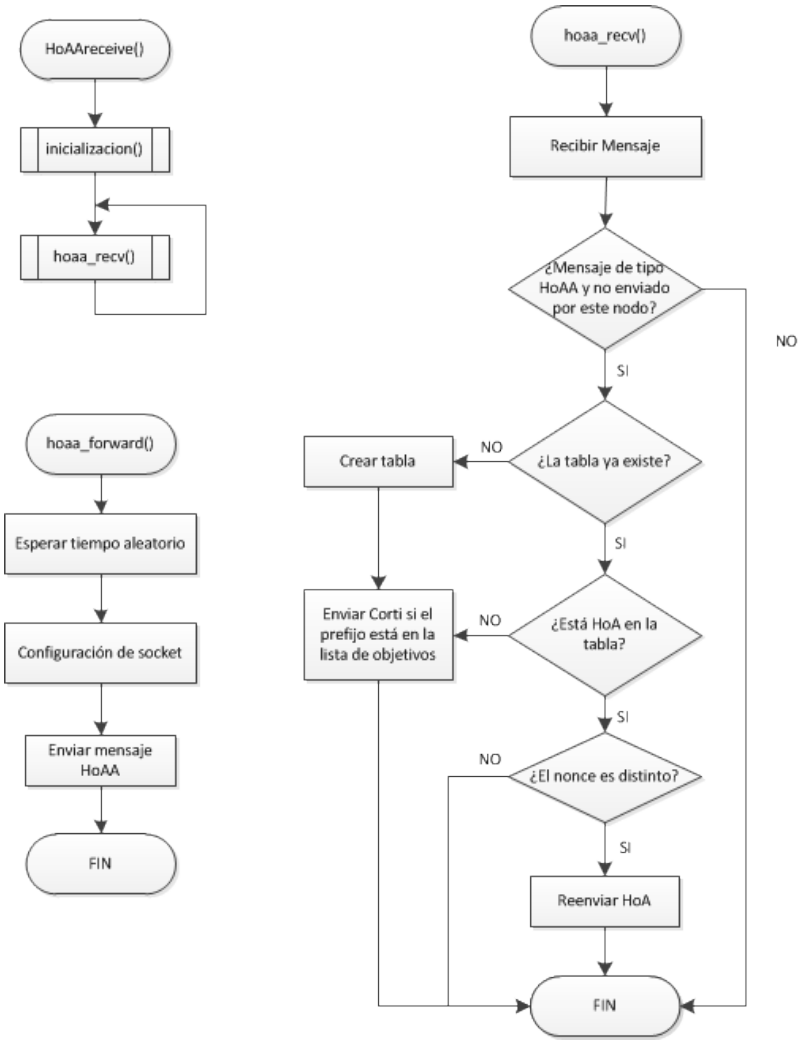


Figura 6.3: Diagramas de flujo de las funciones para la recepción y procesado de HoAA

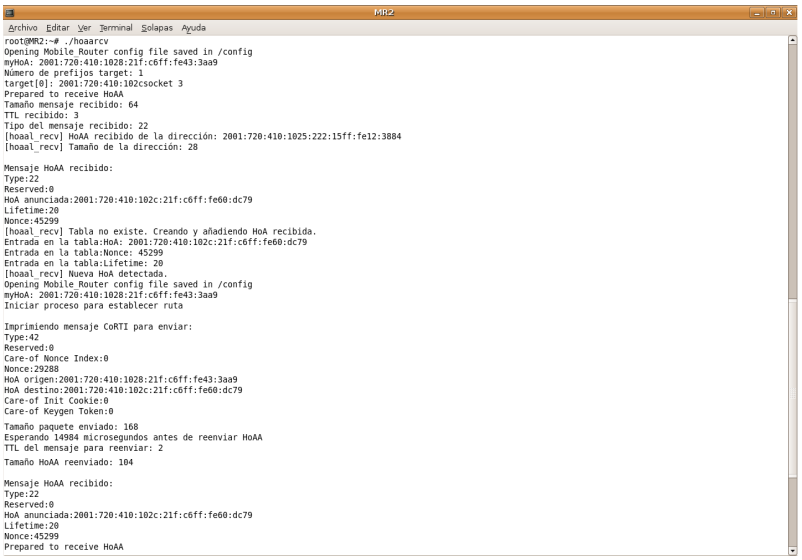


Figura 6.4: Ejemplo de ejecución del módulo de recepción de HoAA

### 6.2.3. Recepción de CoRTI

Una vez que se ha enviado el CoRTI, es necesario recibirlo. Para ello se ha realizado el siguiente bloque, cuyo diagrama de flujo de las distintas funciones implementadas se puede observar en la Figura 6.5.

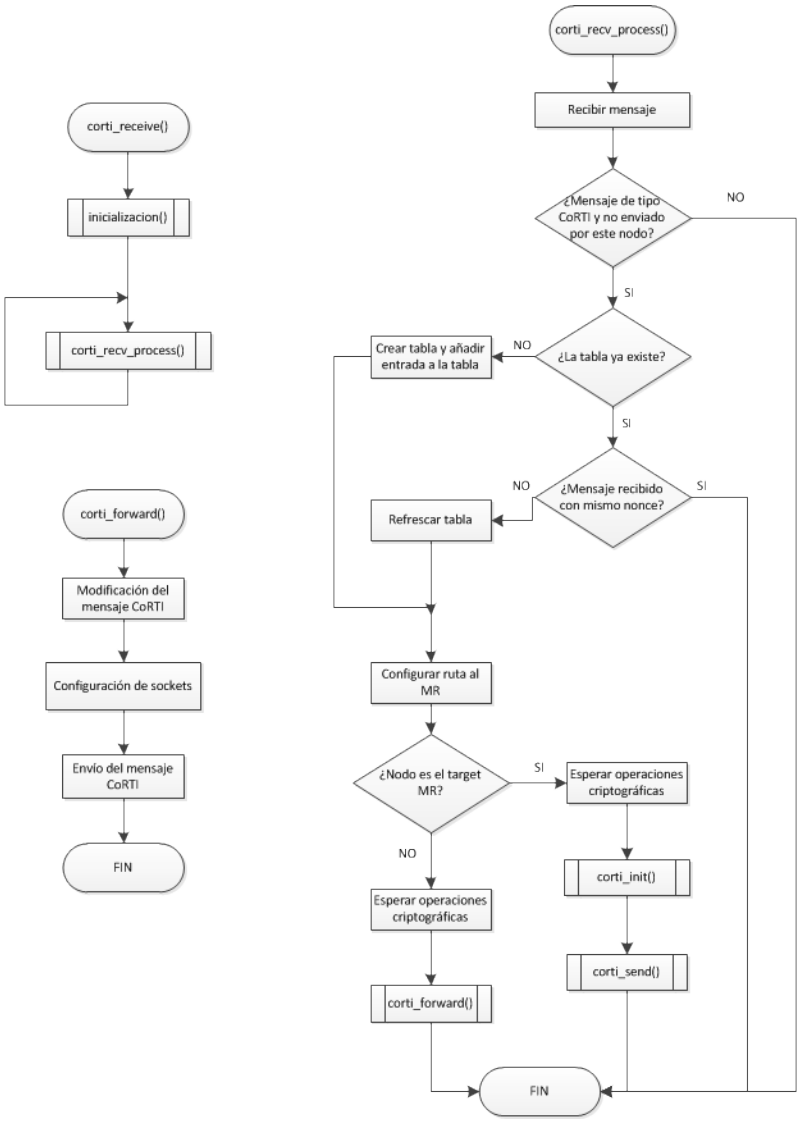


Figura 6.5: Diagramas de flujo de las funciones para la recepción y procesado de CoRTI

Además de recibir el mensaje, el *software* comprueba si el router móvil es el objetivo de ese CoRTI, es decir, si el router que lo envió quería establecer una ruta con él o con otro router móvil, en cuyo caso reenviará el mensaje decrementando el *Hop Limit*. En cualquier caso, el mensaje CoRTI será utilizado para aprender una ruta hacia el router móvil que lo envió originalmente, gracias al campo *Originator HoA*[3], que almacena su dirección HoA. El router que lo recibe, creará una ruta temporal hacia ese destino a través del router del que ha recibido el mensaje. Una vez que el mensaje llegue al router que envió el anuncio y con el que el router inicial (MR *origen*) quería establecer la ruta por la red ad-hoc, éste responderá con un mensaje CoRT. La Figura 6.6 muestra la ejecución de este módulo en

[3]El formato de todos los mensajes se explica con más detalle en el Apéndice B

el router destinatario final de este mensaje, que detecta que es el router *objetivo* y genera el mensaje CoRT.

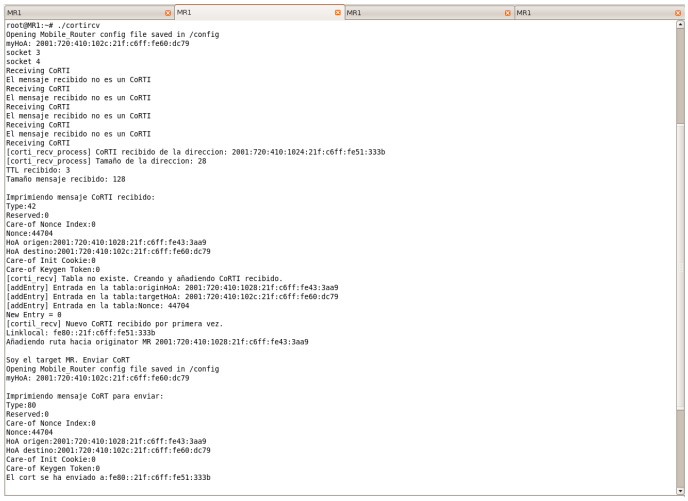


Figura 6.6: Ejemplo de ejecución del módulo de recepción de CoRTI

El mensaje CoRT, al contrario que los mensajes enviados hasta el momento, se envían con dirección destino la dirección local IPv6 [4] router del que se ha aprendido la ruta con la recepción del CoRTI.

### 6.2.4. Recepción de CoRT

El diagrama de flujo de las funciones que forman este módulo se muestra en la Figura 6.7. Al recibir un CoRT, el router debe comprobar si es el destinatario final, o si por el contrario debe reenviar el mensaje. Para ello, debe comparar la dirección en el campo *Originator HoA*, con su propia HoA. Si el mensaje no va dirigido a él lo reenviará al siguiente salto en la ruta que aprendió anteriormente. Si por el contrario, es el destinatario final, deberá enviar un mensaje HoRT dirigido al MR *objetivo* a través de la infraestructura, pasando por sus respectivas redes hogar. La Figura 6.8 muestra un ejemplo de la ejecución de este bloque.

[4] Dirección local IPv6: válida y única tan sólo en un enlace. El formato de la dirección es: fe80::/64. Los últimos 64 bits suelen corresponder a la dirección hardware de la interfaz siguiendo el formato EUI-64.

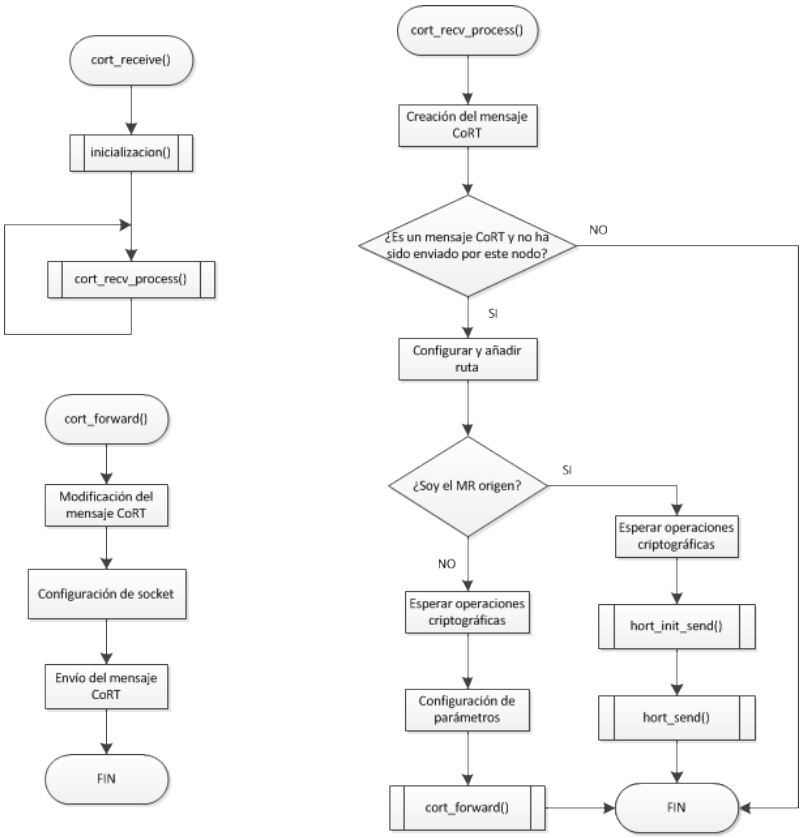


Figura 6.7: Diagramas de flujo de las funciones para la recepción y procesado de los mensajes CoRT

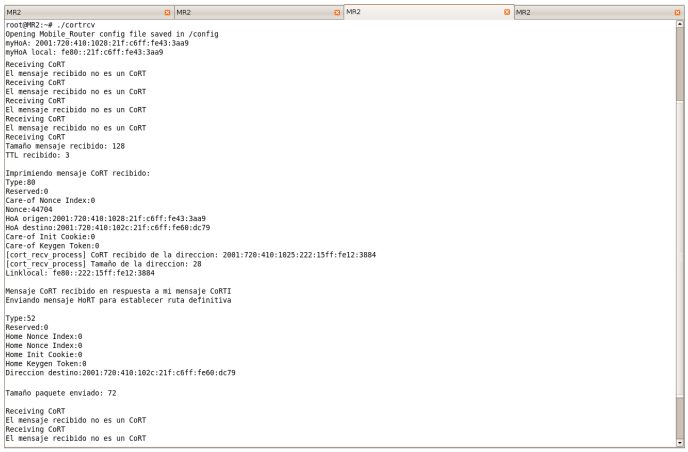


Figura 6.8: Ejemplo de ejecución del módulo de recepción de CoRT

### 6.2.5. Recepción de HoRT. Envío y recepción de MNPBU

El último bloque se encarga de la recepción de los mensajes que son enviados a través de las redes hogar de los routers móviles (HoRT) y también de los mensajes que sirven para constatar que los routers móviles están autorizados a gestionar los prefijos que anuncian, y por tanto finalizan el proceso de establecimiento de la ruta en la red ad-hoc (MNPBU).

Cuando un router móvil recibe un HoRT, primero comprueba si es un mensaje enviado

como respuesta a otro HoRT, en cuyo caso debe enviar un MNPBU o no, debiendo responder entonces con otro HoRT. Por este motivo el módulo que se encarga de la recepción y envío de los HoRT se encarga también de la recepción y envío de los MNPBU, porque no puede ocurrir uno sin el otro. La diferencia radica en que el envío de los HoRT se realiza por la interfaz que conecta al router móvil con su red hogar, mientras que los MNPBU circulan por la red ad-hoc, por la nueva ruta que será establecida a través de los distintos saltos intermedios entre los MR *origen* y *objetivo*.

El diagrama de flujo de las funciones implementadas en este módulo se muestra en la Figura 6.9, para la gestión de los mensajes de tipo HoRT, y en la Figura 6.10, para la gestión de los mensajes de tipo MNPBU. Por otro lado, la Figura 6.11 recoge un ejemplo de la ejecución de este bloque.

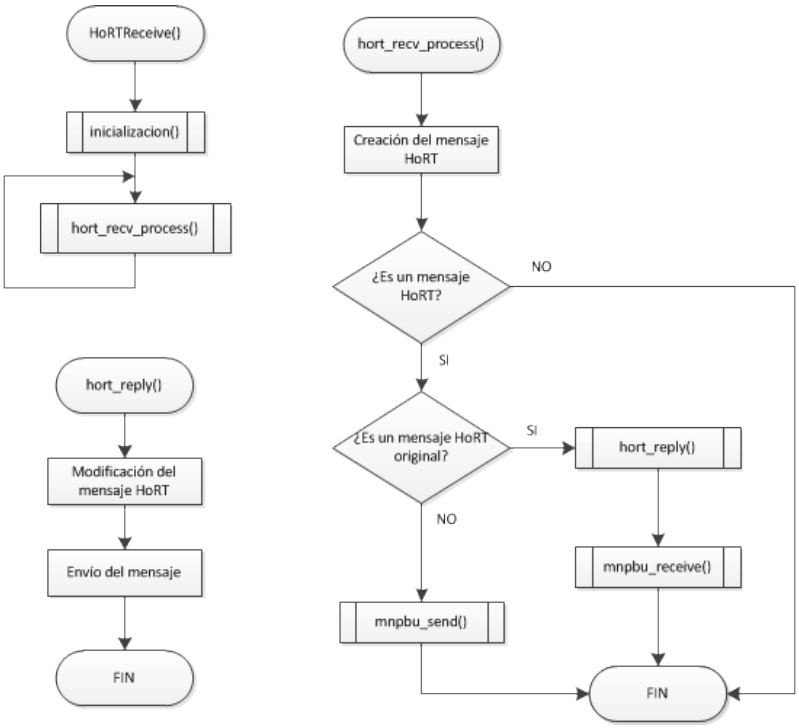


Figura 6.9: Diagramas de flujo de las funciones para la recepción y procesado relacionados con los mensajes HoRT

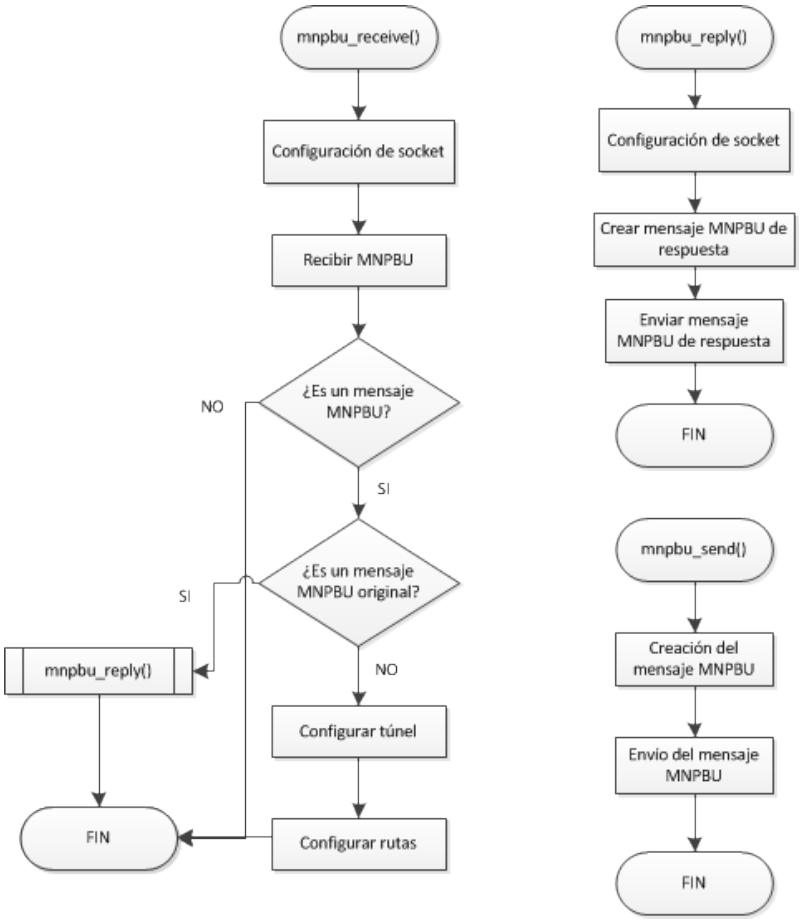


Figura 6.10: Diagramas de flujo de las funciones para la recepción y procesado relacionados con los mensajes MNPBU

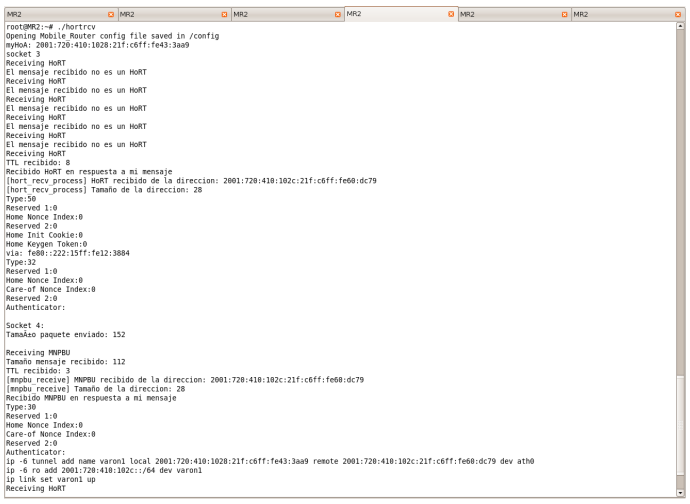


Figura 6.11: Ejemplo de ejecución del módulo de recepción y envío de mensajes HoRT y MNPBU

## 6.3. Integración de los distintos bloques

Todos los bloques que componen la implementación han sido probados por separado para facilitar la eliminación de errores y para comprobar el funcionamiento de cada uno de ellos. A pesar de ello, todos son plenamente compatibles y pueden ejecutarse a la vez, pues aunque todos están diseñados para recibir mensajes, y la configuración en algunos de ellos es la misma, se realizan las comprobaciones oportunas para separar los distintos tipos de mensaje y continuar con el procesamiento sólo en el caso de que el mensaje recibido sea el adecuado. Por ejemplo:

- Se comprueba que la dirección origen del paquete IP no sea la propia.
- Se comprueba el tipo de mensaje (campo *Type* en cada uno de los mensajes de VARON).
- En el caso del CoRTI, sólo se procesa y en su caso, se reenvía el mensaje cuando se recibe por primera vez un CoRTI, identificándolo por su número de secuencia y sus direcciones en los campos *Originator HoA* y *Target HoA*, para limitar la inundación de mensajes.

Por las características del proceso de optimización de rutas, cada router móvil puede jugar un papel distinto en cada proceso para el establecimiento de la *Care-of route*, debiendo realizar distintas tareas en cada caso. Por ejemplo:

- Los routers que actúan como extremos (MR *origen* y MR *objetivo*) deben generar los mensajes (CoRTI, CoRT, HoRT y MNPBU) además de aprender las rutas y al final del proceso establecer un túnel entre ellos.
- Por su parte, los routers intermedios deben actuar como intermediarios entre los que serán los dos extremos de ese túnel, reenviando los mensajes que se envían entre ellos, facilitándoles la comunicación.

  Por este motivo, si se quieren simplificar las operaciones que deben realizar los routers intermedios, puede prescindirse en ellos del bloque encargado de los mensajes HoRT y MNPBU, ya que un router intermedio nunca recibirá ni tendrá que enviar un HoRT, y los MNPBU sólo tiene que reencaminarlos hacia su destino, para lo cual no necesita ningún *software*.

Aún así, cualquier router puede actuar en cualquier momento como router intermedio, router *origen* o router *objetivo*, dependiendo de la topología de la red y de los intereses del usuario de la aplicación en cada momento. De la misma manera, puede disminuirse la carga de mensajes de señalización en la red si los HoAA sólo son enviados por los routers que, por cualquier razón, interese al usuario que puedan establecer una comunicación con otro de los nodos en la red. Así, por ejemplo, los nodos que vayan a actuar como routers intermedios en todo momento pueden no utilizar la aplicación de envío de HoAA.

## 6.4. Integración con el soporte de movilidad de redes, NEMO BS

Para poder realizar la optimización de rutas propuesta por VARON, es necesario contar con varias (al menos dos) redes móviles, o al menos con dos routers móviles para poder comprobar la creación de la nueva ruta entre ellos. Además, es imprescindible contar con una implementación del protocolo NEMO BS que realice la comunicación entre cada router móvil y su agente local estableciendo un túnel bidireccional IPv6 en IPv6.

En este proyecto fin de carrera se ha utilizado una implementación ya existente de NEMO BS, realizada en otro proyecto de esta universidad [SG10]. Así, la realización de este proyecto se ha podido centrar en desarrollar y evaluar la optimización de rutas para NEMO.

A su vez, los routers utilizados constan de varias interfaces, y el uso de una interfaz inalámbrica adicional (por USB) hace posible simular el comportamiento de la red vehicular por una de las interfaces, que será la interfaz ad-hoc, configurada en uno de los canales de 802.11a, mientras que la otra interfaz inalámbrica se utiliza para conectarse a los routers de acceso de las redes visitadas y establecer el túnel que comunica a la red móvil con el exterior mediante NEMO.

El prototipo del protocolo de movilidad de redes utilizado consta de dos módulos *software*: uno para realizar las funciones del router móvil y otro para las del agente local. El router móvil detecta el movimiento de la red gracias a la recepción de mensajes ICMPv6 de tipo *Router Advertisement* (RA) y configura su nueva dirección IPv6 en la red visitada (CoA). Para informar a su agente local de su nueva localización, el router móvil envía un mensaje BU, al que el agente local responde con un BA tras actualizar la información correspondiente a esa red móvil en su lista de asociaciones o *Binding Cache*. En ese momento, ambos establecen un túnel bidireccional IPv6 en IPv6 para encapsular el tráfico generado por y/o con destino la red móvil. A partir de ahí, la red móvil tiene plena conectividad a través del túnel que lo comunica con su agente local y ningún nodo en la red móvil es consciente del cambio. Sólo el router móvil ha modificado la dirección de una de sus interfaces para poder mantener una configuración topológicamente correcta en la nueva red.

Así, la tabla de rutas más básica de un router móvil ejecutando NEMO constaría de:

- Una entrada por cada MNP gestionado por él a través de su interfaz *ingress* o interna.
- Una entrada para llegar al router de acceso en la red visitada, configurada tras la recepción del RA.
- Una ruta por defecto a través del túnel con su agente local, que se encargará de redirigir el tráfico de forma adecuada.

Además de estas rutas, también puede haber otras específicas, dependiendo de la situación. En la Figura 6.12 se puede observar la tabla de rutas del router móvil ejecutando NEMO.

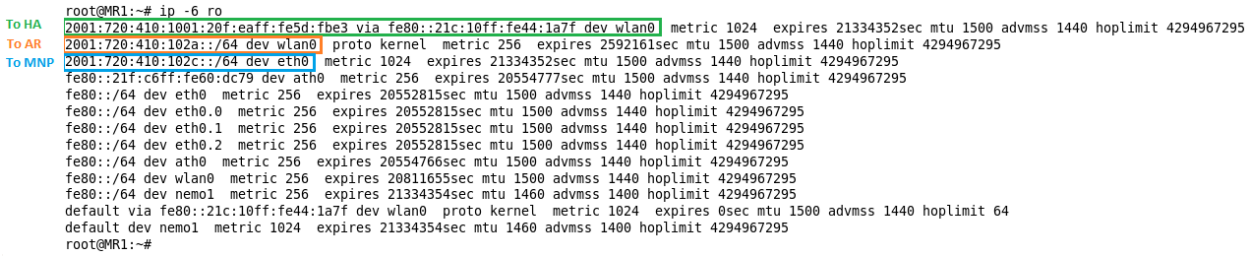


Figura 6.12: Tabla de rutas de un router móvil ejecutando NEMO

Por su parte, el agente local también debe configurar un camino para llegar a la red móvil, por lo que, en el caso del escenario desarrollado [5], su tabla de rutas debería tener las siguientes entradas:

- Una entrada por cada router de acceso con destino la dirección de la interfaz inalámbrica de dichos routers.
- Una entrada por cada MNP de la red móvil a través del túnel.
- Una ruta por defecto para conectarse a la red.

En la Figura 6.13 se observa la tabla de rutas de un agente local en el que se está ejecutando NEMO tras configurar el túnel hasta la red móvil. Cabe destacar que la

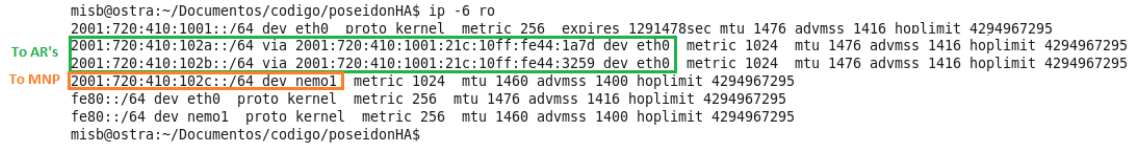


Figura 6.13: Tabla de rutas de un agente local ejecutando NEMO

utilización del prototipo de NEMO no ha necesitado ninguna adaptación para ser utilizado, más allá de la configuración de las direcciones correspondientes para que el router móvil conociera sus prefijos de red móvil y su HoA.

## 6.5. Simulación de operaciones criptográficas

Como se ha comentado en el capítulo 4, los mensajes para la creación de la ruta optimizada a través de la red vehicular van acompañados de una serie de medidas de seguridad con el fin de proteger las comunicaciones entre redes móviles, así como verificar las identidades de los routers móviles, a los que se confiará el tráfico dirigido hacia los nodos móviles de su red.

Para ello, se utilizan varios mecanismos:

- Direcciones IPv6 generadas criptográficamente: Son direcciones IPv6 que permiten asociar la dirección con la clave pública de su dueño. Serán las que los router móviles

[5] En un caso real, la tabla de rutas del agente local podría ser distinta. Por ejemplo, no sería habitual que existiera una ruta específica hacia los routers de acceso, desconocidos a priori, o podría no tener ruta por defecto.

utilizarán como HoA, que a su vez es la dirección que les identificará como destino dentro de la red ad-hoc vehicular. La generación y validación de estas direcciones supone la realización de dos operaciones *hash*. El algoritmo utilizado para ello es SHA1.

- Firma digital: Cada router móvil tiene sus propios pares de clave pública y privada. Así, el MR firma varios campos de los mensajes CoRTI y CoRT, incluyendo las direcciones IPv6 origen y destino, las opciones relacionadas con la generación de su dirección CGA, y la del router intermedio que le reenvió el mensaje, ya que estos routers también incluyen las opciones relacionadas con la firma digital y su CGA al procesar los mensajes para reenviarlos. El mecanismo utilizado es RSA, con distintos tamaños de clave: 1024, 768 y 512 bits.

- Autenticación de MNPBU: Algunos campos de los mensajes del proceso de creación de la nueva ruta incluyen los testigos de generación de claves y los índices de los retos que permiten a los routers móviles intercambiar información para generar dos claves, una por cada uno de ellos, con la que autenticarán los mensajes MNPBU. Concretamente, se utiliza un código de autenticación de mensajes basado en *hash* (SHA1) utilizando una de estas claves para encriptar las HoAs de los MRs que quieren optimizar la ruta entre ellos.

## 6.6. Conclusiones

El desarrollo del software para la creación de la ruta optimizada para NEMO en la red vehicular constituye una fase importante del trabajo realizado en este proyecto. Cada uno de los bloques ha sido desarrollado de forma independiente, lo cual facilita la depuración de errores, porque el código tiene una extensión menor, y la realización de pruebas para comprobar el funcionamiento son más sencillas, ya que se puede depurar paso a paso. Además, la dependencia entre los distintos bloques imponía la necesidad de que el bloque anterior funcionase correctamente para poder comprobar el siguiente.

Por otro lado, la ejecución en el dispositivo utilizado como router móvil requiere un proceso de compilación cruzada, por la diferencia de arquitectura entre el ordenador, donde se desarrolla el software, y el router, donde se va a ejecutar. Además, el proceso de prueba del funcionamiento en el router permite ajustar distintos parámetros, diseñados de forma teórica, adaptándolos a la implementación física, en un dispositivo real.

La introducción de los retardos debidos a la realización de operaciones criptográficas, requirió repetir varias simulaciones para obtener un valor medio del tiempo empleado por el router móvil para llevarlas a cabo. Estas repeticiones hicieron posible comprobar que el nivel de carga computacional era muy exigente dadas las capacidades del router móvil. A pesar de esto, el funcionamiento del router fue satisfactorio, aunque un equipo más potente permitiría reducir el tiempo necesario para la optimización de rutas.

# Capítulo 7

# Escenario desplegado

## 7.1. Introducción

Tras haber realizado una descripción teórica del proceso de optimización de rutas realizado, así como del desarrollo *software*, en este capítulo se va a explicar cómo se ha llevado a cabo la implementación física de VARON. Para ello, se van a comentar las características de los dispositivos que se han utilizado, el escenario de comunicaciones en el que se ha validado su funcionamiento y las principales características de la plataforma desplegada.

## 7.2. Equipos y dispositivos de comunicaciones

En el escenario desplegado para validar el *software* desarrollado se encuentran varios elementos. Por un lado, se encuentran los dispositivos que forman parte de la red vehicular, en la que sucede la optimización de rutas, y por otro, los dispositivos que gestionan la movilidad de cada una de las redes móviles utilizando el protocolo NEMO BS.

En las siguientes secciones se detallan cada uno de estos elementos y las estructuras de red que conforman el escenario completo.

### 7.2.1. El router móvil

El router móvil es el elemento principal de este proyecto. Es un dispositivo fundamental, tanto para la gestión de la movilidad como para la optimización de rutas llevada a cabo en la red vehicular. El equipo utilizado para realizar estas tareas es el router ASUS WL-500g Premium, mostrado en la Figura C.1. Entre otras características (descritas con más detalle en el apéndice C) destaca la presencia de varias interfaces LAN y sobre todo de dos puertos USB. Estos puertos USB dotan de gran flexibilidad al router porque posibilitan la conexión de adaptadores de distintas tecnologías. En este proyecto se han utilizado adaptadores 802.11b/g, pero también existen adaptadores para conectar a redes 3G o Bluetooth, permitiendo realizar diferentes configuraciones y simular entornos de comunicaciones heterogéneos.

La interfaz interna original del router ASUS fue reemplazada por una tarjeta de

la familia Atheros, debido a las limitaciones en la funcionalidad que presentaba la tarjeta original, Broadcom. La tarjeta Atheros permitió la utilización de 802.11a para la comunicación en la red vehicular y un mejor rendimiento, ya que la flexibilidad en los parámetros de configuración de la interfaz original era muy limitada.

La capacidad de este router es limitada, pero aún así es capaz de realizar varias tareas de forma simultánea. Por una parte, ejecuta el *software* de NEMO BS (el módulo correspondiente al router móvil) y por otro, el *software* de VARON, realizando al mismo tiempo todas las tareas de configuración de las interfaces de red y rutas a petición de ambos. Cabe destacar que, aunque el rendimiento del router es bueno, sí se aprecian diferencias según el número de tareas que esté realizando, ya que se ha podido comparar el comportamiento de un router móvil que realiza la optimización de la ruta y el de un router que actúa de salto intermedio en el proceso. Este router intermedio sólo ejecuta el *software* correspondiente a VARON y además no tiene que gestionar ni configurar la interfaz USB, que es la interfaz a través de la cual NEMO establece el túnel con la red hogar. De la misma manera, la configuración de las interfaces inalámbricas conectadas al puerto USB es algo más lenta que la configuración de la interfaz interna del router.

A pesar de este consumo de recursos, no se han detectado fallos o limitaciones en el rendimiento más allá de las esperadas por utilizar un dispositivo con capacidades reducidas.

### 7.2.2. Interfaces inalámbricas adicionales

Las interfaces inalámbricas adicionales, conectadas al puerto USB del router móvil (descripción más detallada en la sección D.2 del Apéndice D), han sido utilizadas para la conexión a Internet de forma permanente de las redes móviles utilizando el protocolo de soporte básico de movilidad de redes, NEMO BS.

## 7.3. Infraestructura de red

El escenario desplegado se presenta en la Figura 7.1. En éste se observan varias redes móviles representadas por su router móvil, que forman la red vehicular. Además, cada red móvil puede conectar a través de Internet con su red hogar, donde se encuentra su agente local, con el que gestiona su propia movilidad. La estructura de cada una de estas redes se detalla en las siguientes secciones.

### 7.3.1. Redes móviles y redes hogar

Las redes móviles constan de un router móvil ASUS. En principio, la red móvil debería tener al menos un nodo móvil conectado al router, que se encargaría de gestionar la movilidad de la red completa. Por simplicidad y por infraestructura, la conexión de nodos móviles se ha simulado añadiendo varias direcciones pertenecientes al MNP gestionado por el router móvil a su interfaz *ingress*.

Los routers móviles que optimizan la ruta entre ellos a través de la red vehicular son los que constan de dos interfaces inalámbricas, ya que necesitan comunicación con sus redes hogar. Los routers móviles que actúan como saltos intermedios sólo tienen su interfaz inalámbrica interna.

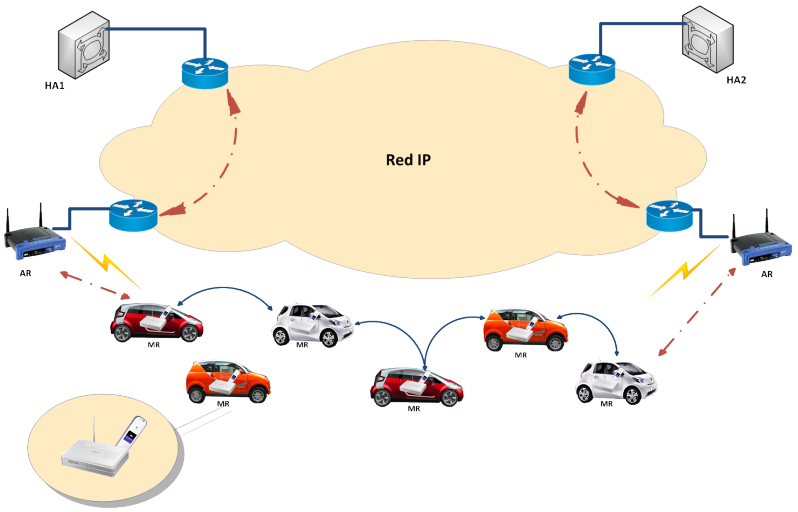


Figura 7.1: Escenario desplegado con redes móviles, redes hogar, puntos de acceso y red vehicular

Los agentes locales que se encuentran en la red hogar son ordenadores personales ubicados en el laboratorio 4.1.F.04 de la universidad, con sistema operativo Linux (Ubuntu) y kernel 2.6. Concretamente, estas son las características de las máquinas utilizadas:

- Coleóptero: Ubuntu 8.04 (*hardy*). *Kernel* 2.6.24-23 *generic*. Este equipo fue utilizado como agente local.
- Ostra: Ubuntu 8.10 (*intrepid*). *Kernel* 2.6.27-7 *generic*. Este equipo fue utilizado como agente local.
- Mosquito: Ubuntu 8.04 (*hardy*). *Kernel* 2.6.24-23 *generic*. Este equipo fue utilizado para configuración de los routers móviles, desarrollo *software* y evaluación del rendimiento.

### 7.3.2. La red vehicular

La red vehicular está formado por un conjunto de redes móviles, en este caso por un conjunto de routers móviles conectados entre sí formando una red ad-hoc inalámbrica. La tecnología utilizada es 802.11a, por varios motivos:

1. Similitud con el estándar propuesto para las redes vehiculares [iee10]. En él se define la banda de frecuencias de 5.9 GHz. De las posibilidades existentes en 802.11, la variante más cercana que podemos utilizar es 802.11a.
2. Eficiencia, velocidad y rendimiento. Aunque la tasa máxima efectiva es similar a la alcanzada con 802.11g se evita la posibilidad de trabajar en modo de compatibilidad con 802.11b si en algún momento entrase en la red alguna estación o punto de acceso con esta tecnología, lo que disminuiría considerablemente el rendimiento. Por esta razón además, los adaptadores USB se utilizan para la movilidad de la red, ya que no pueden trabajar en 802.11a, así como los puntos de acceso disponibles en el escenario.
3. Se evitan interferencias y colisiones en el actualmente sobre-utilizado rango de frecuencias utilizado por 802.11b/g.

Para emular realmente el comportamiento de una red vehicular y poder obtener resultados en la evaluación del trabajo realizado más fieles a la realidad, se debería haber realizado un escenario en el que se pudiera introducir la movilidad de las redes, para evaluar el comportamiento de la optimización de rutas cuando los vehículos se mueven juntos (ó no), a distintas velocidades, etc. Debido a la limitación de recursos y a las dimensiones del laboratorio en el que se han desarrollado las pruebas no se ha podido realizar.

A pesar de que en el escenario de pruebas todos los routers móviles empleados estaban dentro del rango de visibilidad del resto, se ha simulado un escenario con distintas topologías utilizando la herramienta *ip6tables*. Este comando integrado en el *kernel* permite filtrar el tráfico en la red, limitando así la visibilidad de los routers móviles y poder establecer rutas con distinto número de saltos intermedios. En concreto, utilizando una herramienta desarrollada en un proyecto fin de carrera previo [GE10], se ha podido simular un escenario en el que cada vehículo sólo podía recibir mensajes de uno (en el caso de los extremos) o dos vehículos (en el caso de los intermedios), como se muestra en la Figura 7.2. Para ello, se utilizan las coordenadas de cada vehículo y el radio de cobertura de cada uno, por lo que eligiendo adecuadamente estos parámetros, se puede simular una topología de red multisalto. Para cada router, se calcula si la distancia que les separa de cada uno de los otros vehículos es menor que la zona de visibilidad entre ellos:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} < r_1 + r_2 \tag{7.1}$$

De esta forma, se genera una regla para descartar los paquetes recibidos procedentes de la dirección MAC de los vehículos que no estén dentro de la zona de visibilidad de cada red móvil.

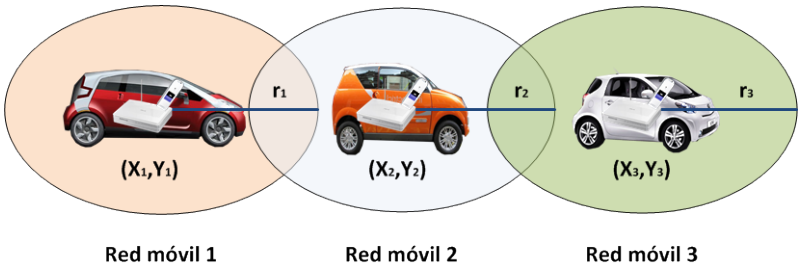


Figura 7.2: Topología de red simulada mediante el uso de *ip6tables*

## 7.4. Conclusiones

El escenario desplegado consta de varios elementos que le confieren cierta complejidad. Por eso es importante que la configuración de cada elemento esté lo más automatizada posible. Además, así se evitan errores difíciles de detectar cada vez que se pone en marcha un experimento.

Por otro lado, la utilización de varios routers móviles permite evaluar el comportamiento de la solución desarrollada para distinto número de saltos intermedios en la ruta ad-hoc. Esta es una prueba especialmente importante, ya que el éxito de la optimización de rutas depende totalmente de que se pueda utilizar en un escenario con un número variable (e impredecible) de redes móviles. Aunque el número es reducido, debido a la complejidad y limitación de espacio se consideró que era un número suficiente para realizar las pruebas más relevantes.

La utilización de 802.11a y de las interfaces inalámbricas USB para comunicar con la infraestructura de red fue una decisión posterior, que facilitó la homogeneización de los dispositivos en la red vehicular, ya que no todos los routers móviles cuentan con adaptadores USB.

# Capítulo 8

# Evaluación

## 8.1. Introducción

En este capítulo se va a presentar la validación de la implementación desarrollada, así como las pruebas llevadas a cabo para medir su eficiencia. Entre otras cosas, es necesario comprobar el comportamiento del protocolo de optimización y del rendimiento del router utilizado, ya que algunas de las tareas pueden ser muy exigentes y añadir retardos al tiempo total de señalización que no son debidos directamente al proceso de establecimiento de la ruta en la red vehicular.

## 8.2. Validación del funcionamiento

En primer lugar, se debe comprobar que el código en C desarrollado funciona como se espera. Ya que el proceso de optimización de rutas está dividido en distintos bloques, esta comprobación se puede hacer fácilmente, así como la depuración de errores de cada uno de los bloques.

Para comprobar que los mensajes se envían adecuadamente, siguiendo la secuencia apropiada y verificar su formato, se utilizan los analizadores de redes *Wireshark* (en los ordenadores) y *tcpdump* (en los routers móviles).

En el escenario de prueba se utilizan dos routers para actuar como MR *origen* y MR *objetivo* en el proceso de optimización. Inicialmente, se comprobó el funcionamiento sólo con estos dos routers, para más tarde añadir un tercero, que actúe como salto intermedio, probando así todos los posibles casos de funcionamiento que se pueden encontrar para un router móvil en la red vehicular.

El MR *objetivo* comienza enviando de forma periódica los anuncios de su HoA. El MR *origen* recibe uno de estos anuncios y genera un mensaje de tipo CoRTI que será enviado a la dirección multicast del grupo de todos los routers. El MR *objetivo* lo recibe y genera el mensaje de tipo CoRT correspondiente, dirigido a la dirección *link-local* del router móvil del que recibió el mensaje CoRTI, en este caso, directamente del MR *origen*. Este intercambio de mensajes se puede observar en la Figura 8.1.

A continuación los mensajes HoRT intercambiados entre ambos MR deben ser encapsulados en el túnel de cada uno de ellos con su agente local, creado por el protocolo

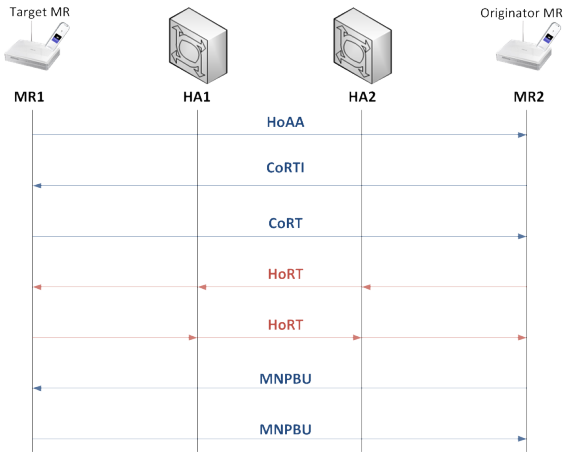


Figura 8.1: Intercambio de mensajes para crear ruta en la red vehicular

de soporte de movilidad de redes.

La Figura 8.2 muestra como tras recibir los mensajes HoRT los routers móviles envían y reciben los correspondientes mensajes de tipo MNPBU que finalizan el proceso de establecimiento de la ruta. En ese momento, los dos crean un túnel IPv6 en IPv6 y las rutas adecuadas para dirigir el tráfico hacia el prefijo de red móvil del otro a través del túnel y la nueva ruta, como se puede comprobar en la Figura 8.3 que muestra la tabla de rutas tras concluir el proceso de señalización.

```
root@MR2:~# tcpdump -i ath0 -p udp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ath0, link-type EN10MB (Ethernet), capture size 96 bytes
02:38:05.358372 IP6 2001:720:410:102c:21f:c6ff:fe60:dc79.65432 > ff02::2.65432: UDP, length 56
02:38:05.401897 IP6 2001:720:410:1024:21f:c6ff:fe51:333b.65432 > ff02::2.65432: UDP, length 56
02:38:05.446543 IP6 2001:720:410:102d:21f:c6ff:fe51:3333.65432 > ff02::2.65432: UDP, length 56
02:38:05.491313 IP6 2001:720:410:1029:21f:c6ff:fe23:3fc2.65432 > ff02::2.65432: UDP, length 56
02:38:05.547065 IP6 2001:720:410:1023:222:15ff:fe12:3970.65432 > ff02::2.65432: UDP, length 56
02:38:05.587162 IP6 2001:720:410:1027:222:15ff:fe12:395a.65432 > ff02::2.65432: UDP, length 56
02:38:05.638076 IP6 2001:720:410:1025:222:15ff:fe12:3884.65432 > ff02::2.65432: UDP, length 56
02:38:05.777454 IP6 2001:720:410:1028:21f:c6ff:fe43:3aa9.23456 > ff02::2.23456: UDP, length 120
02:38:05.849868 IP6 2001:720:410:1028:21f:c6ff:fe43:3aa9.65432 > ff02::2.65432: UDP, length 56
02:38:05.985001 IP6 2001:720:410:1025:222:15ff:fe12:3884.23456 > ff02::2.23456: UDP, length 120
02:38:06.211405 IP6 2001:720:410:1027:222:15ff:fe12:395a.23456 > ff02::2.23456: UDP, length 120
02:38:06.435329 IP6 2001:720:410:1023:222:15ff:fe12:3970.23456 > ff02::2.23456: UDP, length 120
02:38:06.892574 IP6 2001:720:410:1029:21f:c6ff:fe23:3fc2.23456 > ff02::2.23456: UDP, length 120
02:38:07.122727 IP6 2001:720:410:102d:21f:c6ff:fe51:3333.23456 > ff02::2.23456: UDP, length 120
02:38:07.350179 IP6 2001:720:410:1024:21f:c6ff:fe51:333b.23456 > ff02::2.23456: UDP, length 120
02:38:09.370745 IP6 2001:720:410:1025:222:15ff:fe12:3884.23456 > fe80::21f:c6ff:fe43:3aa9.23456: UDP, length 120
02:38:09.672415 IP6 2001:720:410:1028:21f:c6ff:fe43:3aa9.34567 > 2001:720:410:102c:21f:c6ff:fe60:dc79.34567: UDP, length 104
02:38:09.770440 IP6 2001:720:410:102c:21f:c6ff:fe60:dc79.34567 > 2001:720:410:1028:21f:c6ff:fe43:3aa9.34567: UDP, length 104
02:38:25.368632 IP6 2001:720:410:102c:21f:c6ff:fe60:dc79.65432 > ff02::2.65432: UDP, length 56
02:38:25.400985 IP6 2001:720:410:1024:21f:c6ff:fe51:333b.65432 > ff02::2.65432: UDP, length 56
02:38:25.448553 IP6 2001:720:410:102d:21f:c6ff:fe51:3333.65432 > ff02::2.65432: UDP, length 56
02:38:25.474339 IP6 2001:720:410:1029:21f:c6ff:fe23:3fc2.65432 > ff02::2.65432: UDP, length 56
02:38:25.535199 IP6 2001:720:410:1023:222:15ff:fe12:3970.65432 > ff02::2.65432: UDP, length 56
02:38:25.577913 IP6 2001:720:410:1027:222:15ff:fe12:395a.65432 > ff02::2.65432: UDP, length 56
02:38:25.602485 IP6 2001:720:410:1025:222:15ff:fe12:3884.65432 > ff02::2.65432: UDP, length 56
02:38:25.631955 IP6 2001:720:410:1028:21f:c6ff:fe43:3aa9.65432 > ff02::2.65432: UDP, length 56
^C
26 packets captured
26 packets received by filter
0 packets dropped by kernel
root@MR2:~#
```

Figura 8.2: Intercambio de mensajes para crear ruta en la red vehicular

Al introducir un router como salto intermedio se debe comprobar que la comunicación entre los MR de los extremos pasa realmente por él. Para ello, se ha configurado con el comando *ip6tables* una regla que evita que ambos routers móviles reciban los mensajes procedentes del otro, filtrándolos por su dirección MAC origen. El router intermedio debe reenviar todos los mensajes que reciba, ya sea los dirigidos a una dirección multicast o a una dirección unicast perteneciente a uno de los routers de los extremos.

Los mensajes *multicast* son reenviados por cualquier router, ya que se propagan por la red mediante inundación. Para reenviar los mensajes dirigidos a una dirección destino

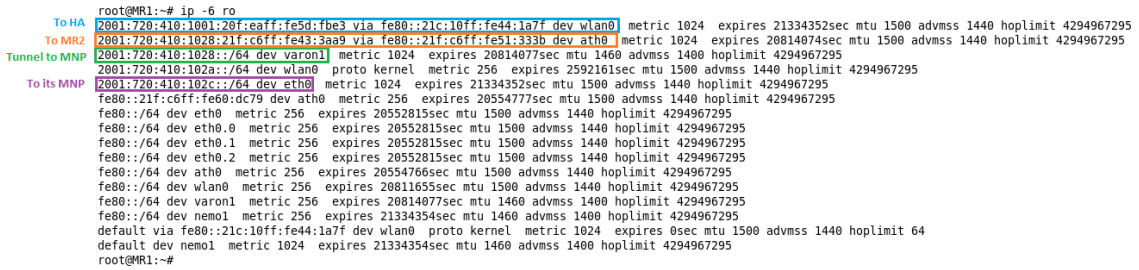


Figura 8.3: Intercambio de mensajes para crear ruta en la red vehicular

*unicast*, debe haber aprendido la ruta en un paso previo del proceso de señalización. Es importante señalar que el reenvío de los mensajes MNPBU por parte de los routers intermedios, se realiza sin necesidad de ejecutar ninguno de los bloques de *software* implementados. Esto es debido a que los routers intermedios no necesitan recibir el mensaje, sólo reenviarlo, a diferencia de los mensajes de tipo CoRT, que necesita recibir para comprobar las opciones CGA y RSA del router anterior a él en la cadena y añadir las suyas, además de aprender la ruta inversa. Gracias al intercambio de mensajes de tipo CoRTI y CoRT, los routers intermedios aprenden como encaminar los mensajes dirigidos al MR origen y al MR objetivo, por lo que, al no tener que hacer ninguna operación con el mensaje MNPBU, el router intermedio puede simplemente reenviarlo hacia el siguiente salto, para lo que no necesita ningún *software* adicional, sólo la configuración adecuada de las tablas de rutas.

Por tanto, el módulo dedicado al envío/recepción de los mensajes HoRT y MNPBU sólo es ejecutado por los routers que tienen soporte de movilidad de redes (ejecutando NEMO BS) y dos interfaces inalámbricas. El resto de routers en la red se comportan como saltos intermedios en la optimización de rutas entre dos routers móviles del otro tipo. En la Figura 8.2 se comprueba como los mensajes pasan a través del router intermedio, observándose además como cada router decrementa el campo *Hop Limit* de cada mensaje que reenvía.

Una vez que se ha comprobado que el código desarrollado funciona de la forma esperada para los tres papeles que puede desempeñar un router móvil, ya se pueden introducir más saltos intermedios en la red, como se hará en la sección 8.4 para medir el tiempo invertido en el proceso de establecimiento de la ruta optimizada para distinto número de routers móviles.

## 8.3. Tiempo necesario para realizar operaciones criptográficas

En la optimización de rutas descrita por VARON la seguridad juega un papel importante. Para alcanzar dicho compromiso de seguridad, es necesario que cada router:

1. Utilice direcciones generadas criptográficamente (que los demás routers móviles deben verificar).

2. Incluya su firma en algunos de los mensajes (CoRTI y CoRT).

3. Verifique la firma digital enviada por otros routers.

Todas estas operaciones requieren un tiempo, dependiendo del tamaño de la clave utilizada y del conjunto de datos que es necesario cifrar, codificar, firmar o validar. Estas operaciones, además, son más exigentes para el router móvil, debido a su capacidad reducida. Por este motivo es necesario calcular y medir de algún modo el tiempo invertido por el router móvil en realizar cada una de las operaciones involucradas en la seguridad del proceso de establecimiento de rutas. Para realizar este cálculo se ha utilizado *OpenSSL*, que permite realizar, entre otras operaciones, firmas digitales, codificar mensajes mediante funciones *hash* y validar ambos procesos. Para llevar a cabo las medidas se realizaron treinta repeticiones de cada operación realizada en cada mensaje, distinguiéndose varios casos para cada tipo de mensaje, como se detalla a continuación:

- **CoRTI 1**: Se ha denominado así al tiempo necesario para que el MR *origen* genere el mensaje CoRTI. Esto requiere firmar 198, 166 o 134 bytes dependiendo del tamaño de clave utilizado (1024, 768 o 512 bits).

- **CoRTI 2**: Este es el tiempo empleado por un router intermedio para procesar y reenviar un CoRTI recibido directamente del router móvil que lo originó. Requiere verificar la firma del anterior MR (tamaño 198, 166 o 134 bytes), dos operaciones de *hash* para comprobar la CGA (153 bytes) del router origen y firmar el nuevo mensaje (de tamaño 510, 414 o 318 bytes).

- **CoRTI 3**: Se ha llamado así al tiempo necesario para que un router intermedio procese y reenvíe un CoRTI que ha recibido de otro router intermedio. Es necesario hacer esta distinción porque el tamaño de los datos que se deben verificar o firmar en el caso de un mensaje previamente reenviado es mayor, aumentando también la carga computacional de cada operación. Concretamente, se necesita verificar la firma del router origen (verificar 198, 166 o 134 bytes), la firma del router intermedio anterior (verificar 510, 414 o 318 bytes), comprobar la opción CGA de ambos (4 operaciones de *hash* de 153 bytes) y firmar el nuevo mensaje, eliminando la firma y opción CGA del router intermedio anterior.

- **CoRTI 4**: Este es el tiempo necesario para que el router objetivo procese el mensaje CoRTI recibido a través de un router intermedio y generar el mensaje CoRT correspondiente. Este proceso implica verificar la firma del MR origen (198, 166 o 134 bytes), verificar la firma del MR intermedio (510, 414 o 318 bytes), comprobar las opciones CGA de ambos routers (4 operaciones *hash* de 153 bytes) y firmar el CoRT generado (198, 166 o 134 bytes).

- **CoRTI 5**: Se ha denominado *CoRTI 5* al tiempo necesario para que el MR *objetivo* procese un CoRTI recibido directamente del MR origen y genere el mensaje CoRT. Esto requiere verificar la firma del MR origen (198, 166 o 134 bytes), comprobar la opción CGA del router origen (operación *hash* de 153 bytes) y firmar el nuevo mensaje (198, 166 o 134 bytes).

- **CoRT 2**: Este es el tiempo necesario para que un router intermedio procese y reenvíe un mensaje CoRT recibido directamente del MR *objetivo*. Este proceso implica verificar la firma del MR (198, 166 o 134 bytes), comprobar la opción CGA del MR *objetivo* (operación *hash* de 153 bytes) y firmar el nuevo mensaje añadiendo sus opciones de firma y CGA (510, 414 o 318 bytes). Estas operaciones coinciden con las realizadas en el caso del tiempo *CoRTI 2*, por lo que sólo se ha realizado la simulación de ese tiempo, utilizando el mismo resultado para ambos casos.

- **CoRT 3**: Este es el tiempo empleado por un router intermedio para procesar y reenviar un CoRT recibido de otro router intermedio. Requiere verificar las firmas del router origen y del router intermedio, comprobar las opciones CGA de ambos y firmar el nuevo mensaje. Al igual que con el tiempo *CoRT 2*, esta simulación no se ha realizado porque coincide con la correspondiente al caso *CoRTI 3*.
- **CoRT 4**: Este es el tiempo necesario para que el router destinatario del mensaje lo procese tras recibirlo de un router intermedio. Requiere verificar la firma del MR que envió el mensaje y del MR intermedio que lo ha reenviado, además de verificar sus opciones CGA.
- **CoRT 5**: Por último, este es el tiempo necesario para que el router procese un mensaje CoRT recibido directamente del router móvil que lo originó. Requiere verificar la firma del MR origen y su CGA.

En la tabla 8.1 se recogen los resultados obtenidos para la realización de todas estas operaciones por parte de un router móvil.

| Tamaño de clave | CoRTI1 | CoRTI2 | CoRTI3 | CoRTI4 | CoRTI5 | CoRT4 | CoRT5 |
|---|---|---|---|---|---|---|---|
| **512 bits** | 42.5 ±4.4 | 70.1 ±4.5 | 90.7 ±8.6 | 86.5 ±7.5 | 66.1 ±6.7 | 44.4 ±5.6 | 23.6 ±7.05 |
| **768 bits** | 69.5 ±7.8 | 97.5 ±5.1 | 121.01 ±7.6 | 116.4 ±10.002 | 91 ±7.4 | 45.9 ±4.7 | 22.97 ±3.09 |
| **1024 bits** | 130.16 ±8.9 | 164.7 ±5.23 | 183.5 ±5.3 | 177.5 ±14.95 | 151.63 ±9.1 | 51.2 ±5.6 | 26.234 ±8.25 |

Tabla 8.1: Medida del tiempo (en ms) empleado por el router móvil en operaciones criptográficas

A la vista de los resultados obtenidos se puede constatar que la operación más exigente es la validación de la firma. Por ejemplo, observando el tiempo *CoRT5*, que consiste en la validación de la firma de un MR y de su CGA, y comparándolo con el tiempo *CoRTI5* en el que se realizan las mismas operaciones más una firma, el aumento es muy considerable. Además, por esta razón la gestión de los mensajes que requieren la validación de dos firmas (CoRTI3) y realizar la firma, son aquellas en las que el router móvil invierte más tiempo.

Para poder confirmar que el router tiene una capacidad limitada y determinar en qué medida ésta influye en el rendimiento, se han realizado las mismas medidas en un ordenador del laboratorio (Mosquito). La Tabla 8.2 recoge los resultados obtenidos. La Figura 8.4 muestra los resultados obtenidos, comparándolos con los del router móvil.

| Tamaño de clave | CoRTI1 | CoRTI2 | CoRTI3 | CoRTI4 | CoRTI5 | CoRT4 | CoRT5 |
|---|---|---|---|---|---|---|---|
| **512 bits** | 7.1 ±1.1 | 15.5 ±1.2 | 22 ±1.9 | 18.5 ±0.7 | 12.2 ±0.6 | 18.5 ±0.7 | 6.5 ±1.2 |
| **768 bits** | 9.9 ±1.5 | 17.7 ±5.1 | 26.9 ±5.5 | 25.1 ±1.8 | 19.2 ±1.8 | 25.6 ±2.7 | 7 ±1.5 |
| **1024 bits** | 22.03 ±5.9 | 28.1 ±1.4 | 35.1 ±3.2 | 37.9 ±7.8 | 27.4 ±1.5 | 35.4 ±7.6 | 7.7 ±2.2 |

Tabla 8.2: Medida del tiempo (en ms) empleado por un PC en operaciones criptográficas

Una vez que se han realizado estas medidas, se introducen en las partes adecuadas del código programado para poder evaluar de la forma más fiel a la realidad las siguientes medidas. De esta forma, se inserta una espera del tiempo correspondiente al procesado de cada mensaje en las partes del código en C desarrollado donde corresponda, por ejemplo, al recibir un CoRTI, los campos *Hop Limit* de la cabecera IPv6 o *Originator HoA* del mensaje de VARON, indican si se trata de un mensaje reenviado, de la misma manera que el router móvil puede averiguar si el mensaje va dirigido a él o debe reenviarlo. Dependiendo

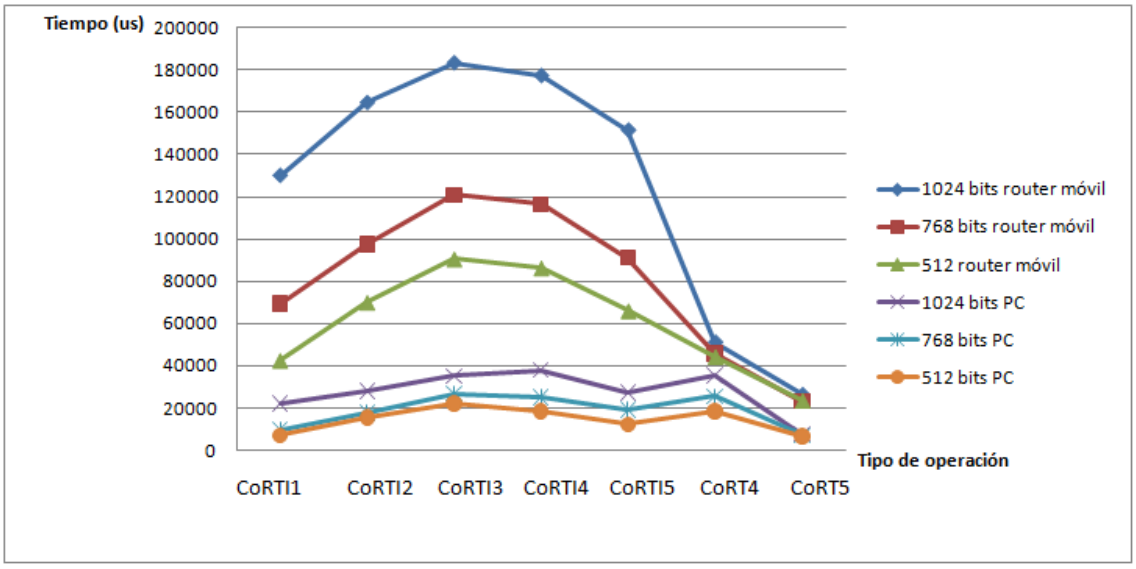


Figura 8.4: Comparativa del tiempo empleado por un PC y por el router móvil (en $\mu$s) para los tres tamaños de clave

del caso, el retardo introducido será mayor o menor, coincidiendo con uno de los tiempos medidos en esta prueba.

## 8.4. Medida del tiempo de señalización

La siguiente prueba realizada consiste en medir el tiempo necesario para establecer una ruta optimizada en la red vehicular para distinto número de saltos intermedios, concretamente desde ningún salto intermedio hasta 7 (de 2 a 9 routers móviles en total).

El escenario realizado para las pruebas se muestra en la Figura 8.5. En él, los routers denominados MR1 y MR2 son los extremos de la ruta que se quiere establecer. El MR1 envía periódicamente los HoAA y el MR2 se encarga de iniciar el proceso de creación de la ruta en la VANET. Para simular una red multisalto, se ha configurado una regla mediante *ip6tables*, para forzar que cada router móvil sólo esté en la zona de visibilidad de uno (en el caso de los routers de los extremos, MR1 y MR2), o dos (en el caso de los routers intermedios) routers móviles, descartando todos los mensajes que reciba del resto. Así se ha formado una cadena de redes móviles en la que MR1 y MR2 sólo se pueden comunicar a través de un número determinado de saltos intermedios.

En cada prueba se mide el tiempo comprendido entre el envío del CoRTI por parte del MR2 hasta que éste recibe el MNPBU, porque en ese momento es cuando configura el túnel hacia el MR1 (que ya lo ha configurado al recibir el primer MNPBU). Para tomar estas medidas se han realizado 35 repeticiones, para cada número de saltos y cada tamaño de clave utilizado en las operaciones criptográficas.

Los resultados obtenidos se recogen en la tabla 8.3.

En la Figura 8.6 se observa el tiempo obtenido para los tres tamaños de clave. Como era de esperar, el tiempo transcurrido en el proceso de creación de la ruta es mayor cuanto mayor número de saltos intermedios hay en la ruta y cuanto mayor es el tamaño de la clave

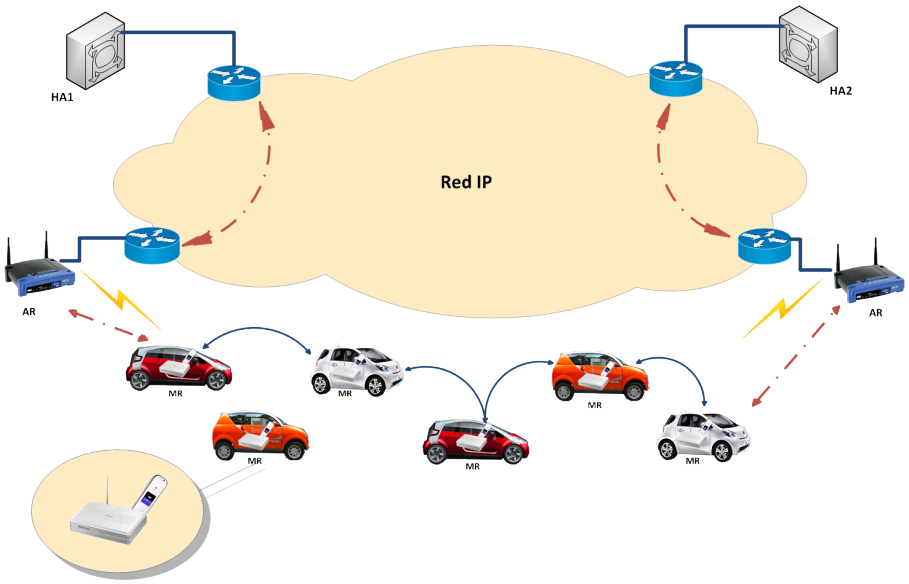


Figura 8.5: Escenario para la medida del tiempo empleado en la creación de la ruta (de 2 a 7 saltos intermedios)

| Tamaño de clave | Número de saltos intermedios | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1024 bits | 662 ±173 | 1147.5 ±92.82 | 1599.7 ±49.5 | 2092.8 ±21.2 | 2612.2 ±116.7 | 3103.2 ±166.4 | 3591.3 ±58.3 | 4024.3 ±53.5 |
| 768 bits | 568.7 ±21.1 | 939.4 ±74.2 | 1291.1 ±28.2 | 1637.7 ±17.4 | 2009.5 ±33.7 | 2382.5 ±92.9 | 2764.2 ±82.6 | 3150.97 ±172.2 |
| 512 bits | 549.6 ±26.5 | 853.5 ±61.7 | 1166.5 ±76.4 | 1434.96 ±47.7 | 1759.8 ±63.5 | 2055.2 ±85.1 | 2377.5 ±97.6 | 2639.4 ±74.8 |

Tabla 8.3: Medida del tiempo (en ms) empleado para optimizar la ruta entre dos routers móviles

utilizada, ya que las operaciones criptográficas añaden un mayor retardo. Esta prueba es de gran valor porque permite conocer el tiempo empleado por los routers móviles para crear una ruta segura a través de la red vehicular. Es importante señalar también que el funcionamiento del prototipo es el esperado, incluso en un escenario con un gran número de saltos intermedios, ya que en un escenario vehicular real, se espera tener un máximo de 3 o 4 saltos.

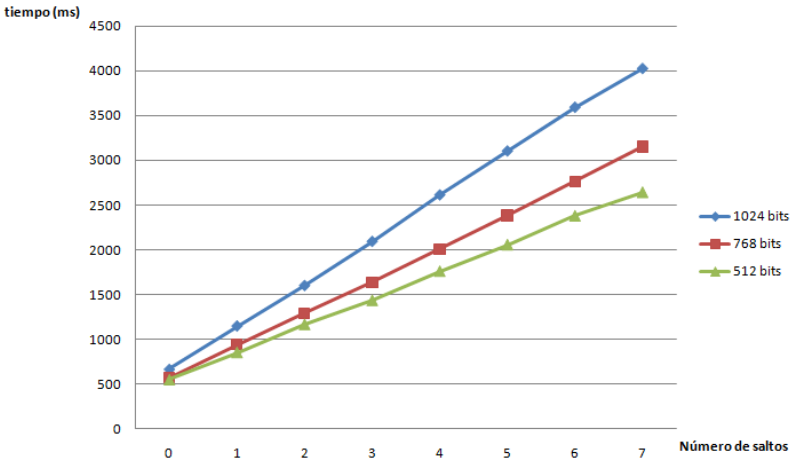


Figura 8.6: Tiempo necesario para crear la nueva ruta para tres tamaños de clave

Cabe destacar que en una situación real no será habitual que los routers de acceso y

los agentes locales estén conectados a la misma infraestructura de red, como ocurre en el escenario de pruebas del laboratorio. Por esta razón el tiempo medido en esta prueba se considera como un tiempo base, al que habría que añadir retardos debidos al RTT entre redes visitadas y redes hogar, así como entre las dos redes hogar de los dos routers móviles que optimizan una ruta entre ellos.

## 8.5. Comunicación a través de la nueva ruta

Para comprobar la validez de la implementación realizada, además de comprobar la realización de las operaciones de la optimización de rutas, es necesario verificar que el tráfico entre dos nodos móviles pertenecientes a cada una de las redes móviles se transmite a través del túnel creado por VARON. Para ello, se configura en la interfaz *ingress* o interna de cada router móvil extremo otra dirección, distinta de su HoA, para simular la presencia de un nodo en la red móvil.

A continuación, se utiliza el comando *ping6* para asegurar que se mantiene la conexión entre ambos MRs en todo momento. Antes de ejecutar VARON, el tráfico entre los dos nodos móviles debe viajar a través del túnel con sus respectivos agentes locales y después, cuando el proceso de optimización de rutas haya finalizado, la comunicación debe pasar a realizarse a través del nuevo túnel en la red vehicular. Para realizar estas comprobaciones se utiliza el analizador de redes *tcpdump*.

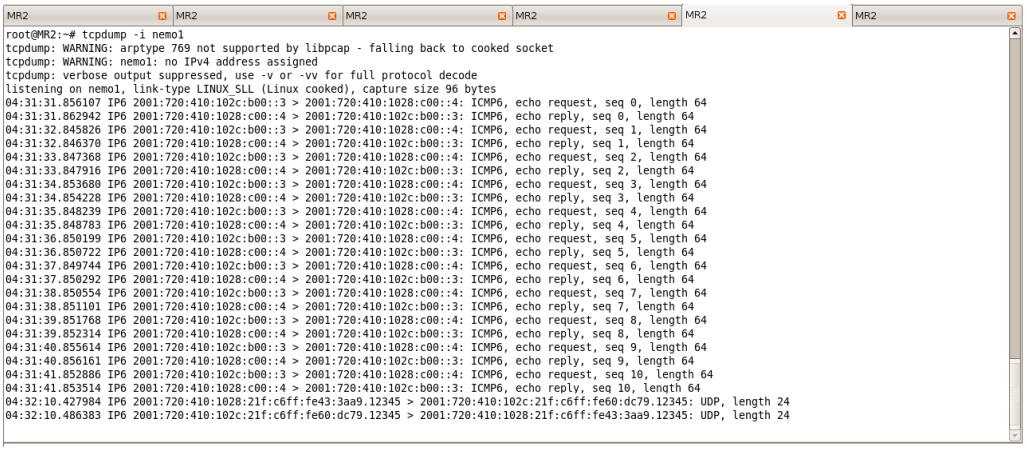


Figura 8.7: Captura de tráfico entre dos redes móviles en la interfaz *nemo1*

En la Figura 8.7 se observa el tráfico en la interfaz *nemo1*, que es el túnel creado por el *software* del protocolo de movilidad de redes. En esa misma figura, se observan dos mensajes UDP al final de la captura, correspondientes a los dos mensajes HoRT intercambiados por los routers móviles para la optimización de la ruta entre ellos. Tras esos mensajes deja de haber mensajes ICMPv6 *echo request* y *echo reply*, ya que se ha creado el túnel *varon1* y esos mensajes se transmiten ahora a través de ese nuevo túnel, como se muestra en la Figura 8.8.

Por último, la Figura 8.9 sirve para verificar en las estadísticas del ping realizado que no se pierde ningún paquete durante el cambio de una ruta a otra, ya que en ningún momento se deja de tener conectividad.

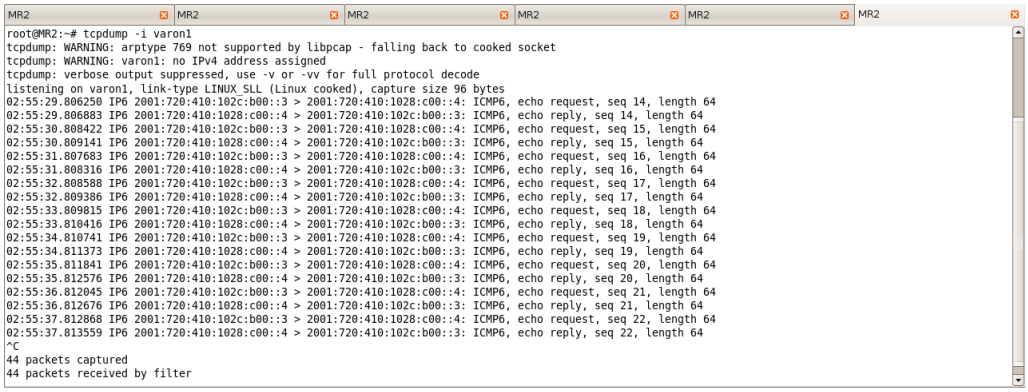


Figura 8.8: Captura de tráfico entre dos redes móviles en la interfaz *varon1*

```
64 bytes from 2001:720:410:102c:b00::3: seq=307 ttl=64 time=9.655 ms
64 bytes from 2001:720:410:102c:b00::3: seq=308 ttl=64 time=9.755 ms
64 bytes from 2001:720:410:102c:b00::3: seq=309 ttl=64 time=9.928 ms
64 bytes from 2001:720:410:102c:b00::3: seq=310 ttl=64 time=9.814 ms
64 bytes from 2001:720:410:102c:b00::3: seq=311 ttl=64 time=9.995 ms
64 bytes from 2001:720:410:102c:b00::3: seq=312 ttl=64 time=9.717 ms
64 bytes from 2001:720:410:102c:b00::3: seq=313 ttl=64 time=10.356 ms
64 bytes from 2001:720:410:102c:b00::3: seq=314 ttl=64 time=9.590 ms
64 bytes from 2001:720:410:102c:b00::3: seq=315 ttl=64 time=9.717 ms
64 bytes from 2001:720:410:102c:b00::3: seq=316 ttl=64 time=12.933 ms
^C
--- 2001:720:410:102c:b00::3 ping statistics ---
317 packets transmitted, 317 packets received, 0% packet loss
round-trip min/avg/max = 8.198/10.712/23.217 ms
root@MR2:~#
```

Figura 8.9: Estadísticas del comando *ping6* entre las dos redes móviles. No hay pérdida de paquetes

## 8.6. Medida del RTT: *Round Trip Time*

El RTT, *Round Trip Time* es el tiempo necesario para que un paquete viaje desde el emisor hasta el receptor y vuelva nuevamente hacia el emisor, es decir, el retardo de ida y vuelta entre emisor y receptor. Esta medida es interesante para poder determinar el impacto de la optimización de rutas, y la mejora en la eficiencia respecto a la ruta establecida por el protocolo de movilidad de redes, NEMO BS.

Para llevar a cabo esta medida, se han realizado 25 repeticiones de un *ping* entre dos direcciones pertenecientes a los MNP de las redes móviles que han creado una ruta entre ellos a través de la red vehicular. Se ha calculado la media de los retardos máximo, mínimo y promedio de las estadísticas facilitadas por el comando *ping6*, fijando una transmisión de 40 paquetes de tipo *Echo Request* en cada repetición. Así mismo, se ha realizado esta medida para distintos casos:

- Medida del RTT en las comunicaciones a través del túnel *nemo1*, creado por el protocolo de movilidad de redes, antes de optimizar la ruta.
- Medida del RTT en las comunicaciones a través del túnel *varon1*, creado en el protocolo de optimización de rutas, para el caso de tener comunicación directa entre los dos routers móviles, sin ningún salto intermedio.

- Medida del RTT en las comunicaciones a través del túnel *varon1*, creado en el protocolo de optimización de rutas, para el caso de tener siete saltos intermedios.

En la Tabla 8.4 se muestra la media de los retardos máximo, mínimo y promedio obtenidos utilizando las distintas rutas, expresando los resultados en media ±desviación estándar. Como se puede apreciar en la tabla, la optimización de la ruta supone una disminución del retardo entre ambos MNP, incluso para el caso extremo de siete saltos intermedios en la red vehicular. El aumento de la eficiencia en el caso de tener comunicación directa, con ningún salto intermedio, entre los dos routers móviles, es significativo si se compara con el retardo obtenido utilizando la ruta establecida por NEMO. Además, hay que recordar que en un escenario real, los agentes locales no estarán conectados a la misma red, ni estarán conectados a la misma red que los routers de acceso, como es el caso del escenario desplegado en el laboratorio, por lo que el retardo entre las dos redes móviles a través del túnel creado por NEMO con sus respectivas redes hogar sería mayor que el medido en esta prueba, obteniendo una mayor ventaja en el caso de utilizar la optimización de rutas en la red vehicular.

| **Ruta** | | **RTT (ms)** | | |
|---|---|---|---|---|
| | | **Mínimo** | **Promedio** | **Máximo** |
| **NEMO** | | 8.27 ±0.17 | 12.1 ±0.47 | 25.39 ±9.03 |
| **VARON** | **0 saltos** | 3.15 ±0.29 | 4.05 ±0.06 | 7.21 ±0.96 |
| | **7 saltos** | 9.18 ±0.12 | 10.07 ±0.19 | 12.83 ±0.57 |

Tabla 8.4: Medida del RTT (en ms) a través de la ruta utilizada por NEMO y la ruta optimizada por VARON, para número de saltos mínimo y máximo

## 8.7. Conclusiones

La batería de pruebas realizada es de vital importancia para formar una opinión sobre el funcionamiento de la optimización de rutas propuesta. Por un lado, es necesario cuantificar el retardo y la sobrecarga introducida en la red para llevarla a cabo. Por otro lado, es necesario comprobar que la comunicación en la red vehicular es factible a través de la nueva ruta.

El tiempo invertido en la creación de la nueva ruta es un parámetro a tener en cuenta a la hora de tomar la decisión de optimizar la ruta o no, dependiendo por ejemplo, del tipo de comunicación que se quiera mantener o establecer con el otro nodo. De esta forma, sabiendo que se tiene un retardo máximo de 4 segundos, la aplicación puede decidir si ese tiempo es asumible o no. También dependiendo del entorno, en una red vehicular 4 segundos pueden suponer un cambio total en la topología de la red. De todas formas, éste es un caso extremo, ya que en una red vehicular, en general, se espera tener un máximo deseable de saltos en torno a 3 o 4 saltos. Existe un compromiso que será necesario estudiar según el escenario, ya que el número de saltos en la red influye en varios parámetros, como el alcance de la comunicación, el retardo o la estabilidad del enlace.

Desde el punto de vista de la comunicación, ese retardo no es decisivo, ya que en ningún momento se pierde la conexión con la otra red móvil, pudiendo seguir en contacto a través de la infraestructura hasta el momento en que el nuevo túnel esté disponible en la red vehicular.

Desde el punto de vista del router móvil, las simulaciones de las operaciones criptográficas han permitido comprobar las limitaciones del dispositivo utilizado. Aunque en los vehículos puedan instalarse equipos con menores restricciones, que el router móvil realice todas las operaciones (ejecutar los *softwares* de NEMO BS y VARON, realizar operaciones criptográficas, enviar y recibir mensajes, configurar rutas e interfaces, etc) ha sido un trabajo bastante exigente para el modelo de router utilizado en este proyecto. Sobre todo, el cálculo de la firma digital y su validación, ya que estas operaciones son en las que más tiempo invierte durante el proceso de creación de la nueva ruta. Además, el uso de otro equipo como router móvil permitiría aumentar el tamaño de clave, por ejemplo, en un intento de aumentar la seguridad en la red, lo que significaría también un aumento en la carga computacional.

# Parte IV

# Conclusiones

# Capítulo 9

# Conclusiones y líneas de trabajo futuro

## 9.1. Introducción

En este capítulo se presentan las principales conclusiones extraídas del trabajo realizado en el presente proyecto fin de carrera, describiendo los principales puntos a destacar y las dificultades encontradas en cada fase del desarrollo. Además se proponen unas líneas de trabajo futuro para añadir funcionalidad a la implementación desarrollada y plantear otras mejoras pendientes de realización.

## 9.2. Conclusiones

La realización de este proyecto fin de carrera conduce a varias conclusiones sobre la solución propuesta, la implementación desarrollada y los resultados obtenidos tras la batería de pruebas:

- En primer lugar, tras el estudio de NEMO BS, cabe destacar que se trata de un protocolo interesante y con aplicación práctica. El campo de aplicación podría ampliarse si se tratase de mejorar su rendimiento en todos los aspectos posibles. El funcionamiento descrito en este proyecto es muy básico, el protocolo contempla diferentes topologías de red móvil, mecanismos de seguridad (IPsec), etc. que lo convierten en una solución de espectro más amplio. Aún así, en lo que a definición de rutas se refiere, necesita alguna optimización o alternativa para mejorar el rendimiento en caso de aplicaciones más exigentes.

- La redes vehiculares están en plena actualidad en el mundo de la investigación. Es una línea abierta en muchos campos, en pleno proceso de estandarización. La información existente sobre la situación actual es dispersa y muy variada. Existen multitud de aplicaciones, servicios y procedimientos, al menos siendo desarrollados, diseñados o en proceso en este momento. Es necesario alcanzar un acuerdo común para facilitar la difusión de estas redes, y posibilitar un despliegue fácil, haciendo que sea accesible para obtener un mayor alcance e impacto en la sociedad.

- La utilización del firmware *OpenWrt* ha sido muy satisfactoria. Esta tecnología, que puede no resultar muy conocida, pero para la que existen multitud de proyectos de desarrollo a nivel usuario, para distintos dispositivos de red y arquitecturas y se siguen diseñando nuevas distribuciones con considerables mejoras sobre versiones anteriores (mejoras en el soporte inalámbrico, nuevos módulos incluidos en el *kernel*, interfaces gráficas, etc). Esta plataforma ha proporcionado una distribución ligera de Linux fundamental en el desarrollo de este proyecto, ya que ha sido la base sobre la que se ha desarrollado la configuración del router móvil y la ejecución de los módulos *software*. En definitiva, confiere gran flexibilidad para la configuración y es de gran utilidad para el testeo de nuevos protocolos o implementaciones desarrolladas a nivel usuario en dispositivos reales (en los que normalmente el usuario tiene poca libertad para hacer modificaciones).

- En este proyecto se ha probado el funcionamiento de la implementación desarrollada, mostrando el comportamiento práctico en dispositivos reales de la optimización de rutas para NEMO. Es importante conocer los resultados obtenidos para poder obtener una visión de esta solución, de la que hasta ahora sólo existían datos teóricos, basados en simulaciones.

- Un punto importante de esta optimización de rutas, consiste en el hecho de que no se pierde tráfico mientras se está realizando el proceso, porque la conexión sigue disponible a través del interfaz NEMO en todo momento. La única modificación introducida es una ruta específica para un prefijo de red móvil, que aparece en la red vehicular. De la misma forma, si se perdiese este prefijo en la red vehicular (o algún nodo a lo largo del camino multisalto establecido), la nueva ruta expiraría pasado un tiempo. En ese caso, se podría perder la comunicación, por lo que es necesario diseñar cuidadosamente el mecanismo para detectar el fallo en la ruta. Rehacerse ante la desaparición de esta ruta es tan sencillo como volver a encaminar el tráfico a través de la infraestructura utilizando NEMO para llegar a la red hogar, hasta que se decidiera realizar cualquier otra optimización.

- Las mediciones realizadas muestran que el proceso de creación de la ruta no introduce demasiado retardo, que además puede disminuir con el incremento en la capacidad de procesamiento de los equipos que se prevé estarán instalados en los vehículos. Es importante limitar el número de saltos en la red ad-hoc, por un lado, para no incrementar demasiado el RTT, por otro lado, por las características tan cambiantes de la red y por último para no sobrecargar la red en exceso con los mecanismos de inundación.

- Respecto al mecanismo de inundación, aunque es utilizado en una gran cantidad de protocolos, es un posible punto susceptible de mejora en el protocolo propuesto. Una posible alternativa podría ser que el mensaje HoAA pudiera hacer también el papel de CoRTI, evitando así la inundación de los dos mensajes, aunque incrementando el tamaño del HoAA. Otra solución podría ser enviar el CoRTI a la dirección anunciada en el HoAA recibido en lugar de a la dirección *multicast*, aunque esto también cambiaría otros aspectos del protocolo.

- Por otra parte, la optimización propuesta necesita madurez, que se vería apoyada por el diseño de una batería de pruebas en un entorno vehicular más fiel a la realidad y más exigente. La propuesta contempla la seguridad, aspecto imprescindible en una red tan potencialmente atractiva (y relativamente fácil de atacar) para nodos maliciosos, por la facilidad de entrada en la red.

## 9.3. Líneas de trabajo futuro

Por último, tras finalizar el trabajo de este proyecto fin de carrera se proponen algunas ampliaciones y posibles mejoras, para futuras aplicaciones.

- Implementación de los mensajes de error tras detectar un fallo en una ruta de la red vehicular (CoRE).
- Implementación de un mecanismo que permita configurar el tiempo de vida de la ruta creada.
- Interacción con otras tecnologías, como por ejemplo 3G.
- Implementación de los mecanismos de seguridad empleados en la creación de los mensajes (RSA, SHA1).
- Expandir el escenario de evaluación para simular un escenario más aproximado a la realidad, al menos, para incluir movimiento, handover, etc.
- Expandir el escenario de evaluación, incluyendo otros elementos presentes en una red vehicular, como sistemas de navegación autónoma, routing geográfico, varios dispositivos en la red móvil, etc.
- Aprovechar el hecho de que el router móvil tiene varios interfaces para probar a introducir otras tecnologías, construyendo una red heterogénea, por ejemplo con un adaptador 3G. Este tipo de redes es el escenario más factible en el entorno vehicular, con tecnologías de la familia 802.11(a/b/g/p), 3G, quizá Bluetooth u otra tecnología interconectando los distintos dispositivos en la red móvil, etc.
- Permitir la configuración, en tiempo de ejecución, de ciertos parámetros como el *Hop Limit* de los mensajes, u otros de mayor interés según la aplicación. En la versión actual, es configurable así el intervalo de envío de los mensajes HoAA.

# Parte V

# Apéndices

# Apéndice A

# Planificación de tareas y presupuesto

## A.1. Introducción

En este apéndice se presenta una relación de tareas en las que se ha dividido la realización de este proyecto fin de carrera y un desglose justificado de los costes para llevarlas a cabo.

Finalmente, se presenta un presupuesto total de ejecución del proyecto, incluyendo el coste de cada tarea así como de los distintos materiales utilizados.

## A.2. Descomposición en tareas

La realización de este proyecto fin de carrera engloba la ejecución de diversas tareas que han sido clasificadas según sus objetivos. Esta división en tareas se presenta en la Tabla A.1.

A continuación, se describen estas tareas, detallando los objetivos específicos, la relación con otras tareas, la duración y el esfuerzo dedicados a cada una de ellas:

- TAREA A: DOCUMENTACIÓN Y ANÁLISIS DEL ESTADO DEL ARTE.
  - Subtarea A.1: OpenWrt y proceso cambio *firmware*
    - Descripción: en esta primera tarea se realiza un estudio del *firmware* OpenWrt, que se instala en el router Asus, así como del proceso de cambio del mismo.
    - Objetivos: Instalar en los routers Asus el *firmware* necesario para llevar a cabo la aplicación.
    - Dependencia (o relación) con otras tareas: esta tarea dará comienzo al inicio del proyecto.
    - Duración: 6 semanas.
    - Recursos: Ingeniero 0.5 hombres/mes.
  - Subtarea A.2: entorno de compilación cruzada y desarrollo
    - Descripción: familiarización con el entorno de compilación requerido para el *firmware* de los routers móviles. Se compila el *firmware*, con lo que se

obtiene una imagen del mismo. Con ello se permite compilar aplicaciones para este tipo de routers y además se obtienen los módulos de movilidad necesarios que se han de instalar en dichos routers.

- Objetivos: Compilar el *firmware* de OpenWrt para obtener una imagen del mismo, y poder así desarrollar y compilar aplicaciones para los routers Asus.
- Dependencia (o relación) con otras tareas: esta tarea dará comienzo tras la tarea A.1.
- Duración: 3 semanas.
- Recursos: Ingeniero 0.5 hombres/mes.

- Subtarea A.3: estudio de los protocolos de movilidad y movilidad de redes
  - Descripción: en esta tarea se ha realizado un estudio previo de las soluciones para gestionar la movilidad en IPv4 e IPv6, así como del funcionamiento del protocolo NEMO BS.
  - Objetivos: Con este estudio se pretende tener una visión global de las soluciones de movilidad y la extensión para la gestión de la movilidad de redes IP.
  - Dependencia (o relación) con otras tareas: esta tarea dará comienzo tras la tarea A.2.
  - Duración: 2 semanas.
  - Recursos: Ingeniero 0.5 hombres/mes.
- Subtarea A.4: estudio de la situación actual de la investigación en redes vehiculares
  - Descripción: en esta tarea se realiza una aproximación al mundo de las redes vehiculares.
  - Objetivos: Con este estudio se pretende tener una visión global de la situación general de la investigación en redes vehiculares, así como de alguna soluciones que se han propuesto.
  - Dependencia (o relación) con otras tareas: .
  - Duración: 2 semanas.
  - Recursos: Ingeniero 0.5 hombres/mes.
- Subtarea A.5: estudio de la optimización de rutas para redes móviles en redes ad-hoc vehiculares
  - Descripción: en esta tarea se ha realizado un estudio previo de la solución propuesta en VARON.
  - Objetivos: Con este estudio se pretende tener una visión global del funcionamiento de la solución propuesta, del formato de los mensajes, de la optimización de rutas y de la interacción de los distintos elementos.
  - Dependencia (o relación) con otras tareas: .
  - Duración: 2 semanas.
  - Recursos: Ingeniero 0.5 hombres/mes.

- TAREA B: INSTALACIÓN Y CONFIGURACIÓN DE DISPOSITIVOS.

  - Descripción: A lo largo de esta tarea se van a realizar las tareas de cambio del *firmware* a los routers Asus. También se van a crear los ficheros o *scripts* para la configuración de las interfaces que se conectan a las distintas redes.

- Objetivos: en esta fase se pretende instalar los módulos necesarios y realizar la configuración para preparar a los routers móviles para la posterior ejecución del *software* de VARON.
- Dependencia (o relación) con otras tareas: esta tarea comenzará una vez que haya terminado la tarea de documentación A.
- Duración: 5 semanas.
- Recursos: Ingeniero 0.5 hombres/mes.

▪ TAREA C: DESARROLLO DE *SOFTWARE*

- Subtarea C.1: *HoAA_send*
  - Descripción: El *software* desarrollado está dividido en varios módulos, y además cada módulo depende directamente de la implementación del anterior.
  - Objetivos: Desarrollo del módulo *software* para el envío periódico de mensajes HoAA.
  - Dependencia (o relación) con otras tareas: esta tarea se comenzará una vez que se haya terminado la tarea B.
  - Duración: 2 semanas.
  - Recursos: Ingeniero 1 hombre/mes.
- Subtarea C.2: *HoAA_rcv*
  - Descripción: en esta tarea se programa el *software* para la recepción de los mensajes de tipo HoAA. En él se implementa la recepción del mensaje, la gestión de la tabla en la que se almacena la información de cada mensaje recibido, el reenvío de los HoAA mediante inundación y el envío del mensaje CoRTI, si procede.
  - Objetivos: Implementación del módulo para la recepción de mensajes HoAA.
  - Dependencia (o relación) con otras tareas: esta tarea se comenzará una vez que se haya terminado la tarea C.1.
  - Duración: 2 semanas.
  - Recursos: Ingeniero 1 hombre/mes.
- Subtarea C.3: *CoRTI_rcv*
  - Descripción: en esta tarea se programa el *software* para la recepción de los mensajes de tipo CoRTI. En él se implementa la recepción del mensaje, la gestión de la tabla en la que se almacena la información de cada mensaje recibido, el reenvío de los CoRTI mediante inundación, el aprendizaje y configuración de la ruta en la red vehicular y el envío del mensaje CoRT, si procede.
  - Objetivos: Implementación del módulo para la recepción de mensajes CoRTI.
  - Dependencia (o relación) con otras tareas: esta tarea se comenzará una vez que se haya terminado la tarea C.2.
  - Duración: 2 semanas.
  - Recursos: Ingeniero 1 hombre/mes.

- Subtarea C.4: *CoRT_rcv*
  - Descripción: en esta tarea se programa el *software* para la recepción de los mensajes de tipo CoRT. En él se implementa la recepción del mensaje, el reenvío según la ruta aprendida anteriormente, el aprendizaje y configuración de la ruta hacia el otro extremo de la comunicación y el envío del mensaje HoRT, si procede.
  - Objetivos: Implementación del módulo para la recepción de mensajes CoRT.
  - Dependencia (o relación) con otras tareas: esta tarea se comenzará una vez que se haya terminado la tarea C.3.
  - Duración: 2 semanas.
  - Recursos: Ingeniero 1 hombre/mes.
- Subtarea C.5: *HoRT_rcv*
  - Descripción: en esta tarea se programa el *software* para la recepción de los mensajes de tipo HoRT y MNPBU. En él se implementa la recepción de los mensajes, la gestión y el procesamiento según cada caso, la generación del mensaje de respuesta correspondiente y la configuración de la nueva ruta y el túnel en la red vehicular.
  - Objetivos: Implementación del módulo para la recepción de mensajes HoRT y MNPBU.
  - Dependencia (o relación) con otras tareas: esta tarea se comenzará una vez que se haya terminado la tarea C.4.
  - Duración: 2 semanas.
  - Recursos: Ingeniero 1 hombre/mes.

- TAREA D: ESTUDIO DE LAS INTERFACES INALÁMBRICAS USB
  - Descripción: A lo largo de esta tarea se realizará la instalación necesaria para utilizar las interfaces inalámbricas USB y el estudio de su rendimiento.
  - Objetivos: Instalar, configurar y evaluar el comportamiento de los adaptadores inalámbricos por USB.
  - Dependencia (o relación) con otras tareas: esta tarea se lleva a cabo tras la tarea B.
  - Duración: 5 semanas.
  - Recursos: Ingeniero 0.25 hombres/mes.

- TAREA E: FUNCIONAMIENTO Y DEPURACIÓN DEL SOFTWARE
  - Subtarea E.1: Funcionamiento de cada módulo
    - Descripción: En esta tarea se evaluará el funcionamiento de los módulos *software* implementados.
    - Objetivos: Comprobar el funcionamiento de la aplicación desarrollada.
    - Dependencia (o relación) con otras tareas: esta tarea se lleva a cabo tras las tareas C y D.
    - Duración: 2 semanas.
    - Recursos: Ingeniero 0.25 hombres/mes.
  - Subtarea E.2: Medida del tiempo empleado en operaciones criptográficas

- Descripción: En esta tarea se medirá el tiempo necesario en un router móvil y en un PC para realizar las operaciones criptográficas asociadas con el procesado de los distintos mensajes.
- Objetivos: Medir el retardo que será introducido en el proceso de creación de la ruta en la red vehicular, debido a la carga computacional de las operaciones criptográficas asociadas a VARON.
- Dependencia (o relación) con otras tareas: esta tarea se lleva a cabo tras la tarea E.1.
- Duración: 2 semanas.
- Recursos: Ingeniero 0.25 hombres/mes.

- TAREA F: INTEGRACIÓN CON EL PROTOCOLO DE MOVILIDAD DE REDES, NEMO BS
  - Descripción: En esta tarea se comprobará que ambas implementaciones, la de NEMO BS y la de VARON, son compatibles y pueden ejecutarse a la vez en el router móvil.
  - Objetivos: Configurar el router móvil para la utilización de la implementación de NEMO BS y comprobar el funcionamiento simultáneo de ambas aplicaciones.
  - Dependencia (o relación) con otras tareas: esta tarea se lleva a cabo tras la tarea E.2.
  - Duración: 1 semanas.
  - Recursos: Ingeniero 0.25 hombres/mes.
- TAREA G: DESPLIEGUE Y CONFIGURACIÓN ESCENARIO
  - Descripción: A lo largo de esta tarea se realizará el despliegue del escenario, que permitirá posteriormente evaluar el prototipo.
  - Objetivos: Configurar un escenario que permita realizar pruebas para la validación del prototipo.
  - Dependencia (o relación) con otras tareas: esta tarea se lleva a cabo tras la tarea F.
  - Duración: 1 semana semanas.
  - Recursos: Ingeniero 0.25 hombres/mes.
- TAREA H: EVALUACIÓN DEL PROTOTIPO
  - Subtarea H.1: Creación del camino multisalto en la red vehicular
    - Descripción: esta tarea se centra en verificar que los routers móviles configuran adecuadamente la ruta en la red vehicular, así como el túnel entre dos redes móviles . Además, es necesario impedir que todos los routers móviles puedan recibir mensajes del resto, sino limitar su visibilidad a uno o dos routers.
    - Objetivos: Comprobar la configuración de la ruta optimizada en la red vehicular.
    - Dependencia (o relación) con otras tareas: esta tarea se lleva a cabo tras la tarea G.
    - Duración: 2 semanas.
    - Recursos: Ingeniero 0.5 hombres/mes.

- Subtarea H.2: Medida del tiempo necesario para crear la ruta optimizada
  - Descripción: esta tarea se centra en medir el tiempo empleado por el prototipo para crear una ruta optimizada en la red vehicular, con distinto número de saltos intermedios.
  - Objetivos: Analizar la eficiencia de la implementación realizada.
  - Dependencia (o relación) con otras tareas: esta tarea se lleva a cabo tras la tarea H.1.
  - Duración: 3 semanas.
  - Recursos: Ingeniero 0.5 hombres/mes.
- Subtarea H.3: Estabilidad de la conexión de la red móvil
  - Descripción: en esta tarea se comprueba que el cambio al crearse la nueva ruta no afecta a la conectividad de las redes móviles, sino que sólo se cambia la forma de encaminar el tráfico entre ellas.
  - Objetivos: Comprobar mediante el comando *ping6* que el encaminamiento pasa de realizarse a través del túnel creado por NEMO a realizarse a través del túnel en la red vehicular, sin pérdida de información.
  - Dependencia (o relación) con otras tareas: esta tarea se lleva a cabo tras la tarea H.1.
  - Duración: 1 semana.
  - Recursos: Ingeniero 0.5 hombre/mes.
- Subtarea H.4: Medida del RTT
  - Descripción: en esta tarea se mide el retardo que sufren los paquetes enviados de una red móvil a otra, comparando las distintas rutas: la creada por NEMO y la optimizada por VARON, para distinto número de saltos intermedios.
  - Objetivos: Determinar el impacto de la optimización de rutas, comparando su eficiencia con la ruta utilizada en el protocolo de movilidad de redes.
  - Dependencia (o relación) con otras tareas: esta tarea se lleva a cabo tras la tarea H.2.
  - Duración: 1 semana.
  - Recursos: Ingeniero 0.5 hombre/mes.

- TAREA I: MEMORIA

  - I.1 Organización y estructura del documento
    - Descripción: A lo largo de esta tarea se organiza el documento, estructurándolo adecuadamente.
    - Objetivos: Organización y estructura de la memoria del documento.
    - Dependencia (o relación) con otras tareas: esta tarea comienza tras la tarea H.
    - Duración: 1 semana.
    - Recursos: Ingeniero 0.5 hombres/mes.
  - I.2 Realización del documento final
    - Descripción: A lo largo de esta tarea se redacta el documento final del proyecto.
    - Objetivos: Redacción de cada uno de los capítulos (y apéndices) de este proyecto.

    - Dependencia (o relación) con otras tareas: esta tarea se lleva a cabo tras la tarea I.1.
    - Duración: 4 semanas.
    - Recursos: Ingeniero 0.5 hombres/mes.

- TAREA J: PRESENTACIÓN

    - Descripción: en esta tarea se va a preparar la presentación de este proyecto.
    - Objetivos: elaborar un conjunto de transparencias adecuado que permitan tener una visión general, clara y completa del trabajado realizado.
    - Dependencia (o relación) con otras tareas: esta tarea se ejecuta tras la tarea I.2.
    - Duración: 1 semanas.
    - Recursos: Ingeniero 0.5 hombres/mes.

| **Tarea** | Duración(s) | Recursos (Ing/m) | Total(h) |
|---|---|---|---|
| *Documentación y análisis del estado del arte* | | | |
| A.1 OpenWrt y proceso cambio *firmware* | 6 | 0.5 | 120 |
| A.2 Entorno de compilación cruzada y desarrollo *software* | 3 | 0.5 | 60 |
| A.3 Estudio protocolos de movilidad y movilidad de redes | 2 | 0.5 | 40 |
| A.4 Estudio de la situación actual de la investigación en redes vehiculares | 2 | 0.5 | 40 |
| Total | | | **260** |
| *Instalación y configuración de dispositivos* | | | |
| B.1 Configuración de dispositivos | 5 | 0.5 | 100 |
| Total | | | **100** |
| *Desarrollo de software* | | | |
| C.1 HoAA_send | 2 | 1 | 80 |
| C.2 HoAA_rcv | 2 | 1 | 80 |
| C.3 CoRTI_rcv | 2 | 1 | 80 |
| C.4 CoRT_rcv | 2 | 1 | 80 |
| C.5 HoRT_rcv | 2 | 1 | 80 |
| Total | | | **400** |
| *Estudio de las interfaces inalámbricas USB* | | | |
| D.1 Estudio y configuración | 5 | 0.25 | 50 |
| Total | | | **50** |
| *Funcionamiento y depuración del software* | | | |
| E.1 Funcionamiento de cada módulo | 6 | 0.5 | 120 |
| E.2 Medida del tiempo empleado en operaciones criptográficas | 2 | 0.25 | 20 |
| Total | | | **140** |
| *Integración con el protocolo de movilidad de redes, NEMO BS* | | | |
| F.1 Integración | 2 | 0.25 | 20 |
| Total | | | **20** |
| *Despliegue y configuración escenario* | | | |
| G.1 Despliegue y configuración | 3 | 0.5 | 60 |
| Total | | | **60** |
| *Evaluación del prototipo* | | | |
| H.1 Creación del camino multisalto en la red vehicular | 2 | 1 | 80 |
| H.2 Medida del tiempo necesario para crear la ruta optimizada | 3 | 0.5 | 60 |
| H.3 Estabilidad de la conexión de la red móvil | 2 | 0.25 | 20 |
| Total | | | **160** |
| *Memoria* | | | |
| I.1 Organización y estructura del documento | 1 | 0.5 | 20 |
| I.2 Realización del documento final | 8 | 0.5 | 160 |
| Total | | | **180** |
| *Presentación* | | | |
| J.1 Realización de la presentación final | 2 | 0.5 | 40 |
| Total | | | **40** |
| **Total** | | | **1410** |

Tabla A.1: Resumen descomposición en tareas

## A.3. Planificación con el diagrama de fases de ejecución detallado

En la Figura A.1, se presenta el diagrama de *Gantt* reducido de las principales tareas. A continuación en la Figura A.2 se muestra la planificación detallada, según las fases, del proyecto.

| Nombre | Duracion | Inicio | Fin | Predecesoras |
|---|---|---|---|---|
| Tarea A: DOCUMENTACION | 65d | 17/08/2009 | 13/11/2009 | |
| Tarea B: INSTALACION Y CONFIGURACION DISPOSITIVOS | 30d | 16/11/2009 | 25/12/2009 | 1,5 |
| Tarea C: DESARROLLO DE SOFTWARE | 50d | 28/12/2009 | 05/03/2010 | 6,7 |
| Tarea D: ESTUDIO DE INTERFACES INALAMBRICAS USB | 25d | 08/03/2010 | 09/04/2010 | 8FI+20d |
| Tarea E: FUNCIONAMIENTO Y DEPURACION DEL SW | 40d | 12/04/2010 | 04/06/2010 | 14 |
| Tarea F: INTEGRACION CON PROTOCOLO NEMO BS | 10d | 07/06/2010 | 18/06/2010 | 16,18 |
| Tarea G: DESPLIEGUE Y CONFIGURACION ESCENARIO | 15d | 21/06/2010 | 09/07/2010 | 16 |
| Tarea H: EVALUACION DEL PROTOTIPO | 35d | 12/07/2010 | 27/08/2010 | 21,22 |
| Tarea I: MEMORIA | 45d | 30/08/2010 | 29/10/2010 | 23,26 |
| Tarea J: PRESENTACION | 10d | 01/11/2010 | 12/11/2010 | 27,29 |

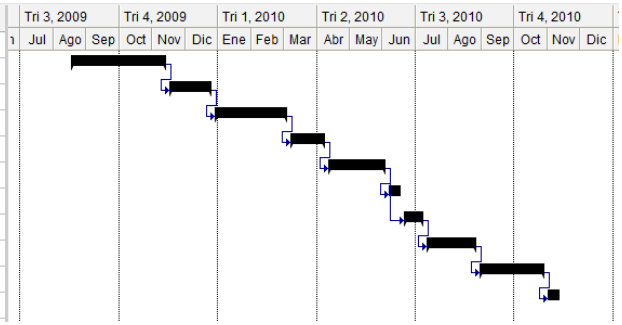


Figura A.1: Diagrama de Gantt reducido.

| Nombre | Duracion | Inicio | Fin | Predecesoras |
|---|---|---|---|---|
| **Tarea A: DOCUMENTACION** | 65d | **17/08/2009** | **13/11/2009** | |
| A.1 OpenWrt y procesos de cambio de firmware | 30d | 17/08/2009 | 25/09/2009 | |
| A.2 Entorno de compilacion cruzada y desarrollo software | 15d | 28/09/2009 | 16/10/2009 | 2 |
| A.3 Estudio protocolos movilidad y movilidad de redes | 10d | 19/10/2009 | 30/10/2009 | 3 |
| A.4 Investigación en redes vehiculares | 10d | 02/11/2009 | 13/11/2009 | 4 |
| **Tarea B: INSTALACION Y CONFIGURACION DISPOSITIVOS** | 30d | **16/11/2009** | **25/12/2009** | **1,5** |
| B.1 Instalación y configuración de dispositivos | 30d | 16/11/2009 | 25/12/2009 | 5 |
| **Tarea C: DESARROLLO DE SOFTWARE** | 50d | **28/12/2009** | **05/03/2010** | **6,7** |
| C.1 HoAA_send | 10d | 28/12/2009 | 08/01/2010 | 7 |
| C.2 HoAA_rcv | 10d | 11/01/2010 | 22/01/2010 | 9 |
| C.3 CoRTI_rcv | 10d | 25/01/2010 | 05/02/2010 | 10 |
| C.4 CORT_rcv | 10d | 08/02/2010 | 19/02/2010 | 11 |
| C.5 HoRT_rcv | 10d | 22/02/2010 | 05/03/2010 | 12 |
| **Tarea D: ESTUDIO DE INTERFACES INALAMBRICAS USB** | 25d | **08/03/2010** | **09/04/2010** | **8FI+20d** |
| D.1 Estudio y configuracion | 25d | 08/03/2010 | 09/04/2010 | 13 |
| **Tarea E: FUNCIONAMIENTO Y DEPURACION DEL SW** | 40d | **12/04/2010** | **04/06/2010** | **14** |
| E.1 Funcionamiento de cada modulo | 30d | 12/04/2010 | 21/05/2010 | 15 |
| E.2 Medida del tiempo empleado en operaciones criptograficas | 10d | 24/05/2010 | 04/06/2010 | 17 |
| **Tarea F: INTEGRACION CON PROTOCOLO NEMO BS** | 10d | **07/06/2010** | **18/06/2010** | **16,18** |
| F.1 Integración con NEMO BS | 10d | 07/06/2010 | 18/06/2010 | 18 |
| **Tarea G: DESPLIEGUE Y CONFIGURACION ESCENARIO** | 15d | **21/06/2010** | **09/07/2010** | **16** |
| G.1 Despliegue y configuracion | 15d | 21/06/2010 | 09/07/2010 | 20 |
| **Tarea H: EVALUACION DEL PROTOTIPO** | 35d | **12/07/2010** | **27/08/2010** | **21,22** |
| H.1 Creacion del camino multisalto en la red vehicular | 10d | 12/07/2010 | 23/07/2010 | 22 |
| H.2 Medida del tiempo de necesario para crear la ruta optimizada | 15d | 26/07/2010 | 13/08/2010 | 24 |
| H.3 Estabilidad de la conezión de la red móvil | 10d | 16/08/2010 | 27/08/2010 | 25 |
| **Tarea I: MEMORIA** | 45d | **30/08/2010** | **29/10/2010** | **23,26** |
| I.1 Organizacion y estructura del documento | 5d | 30/08/2010 | 03/09/2010 | 26 |
| I.2 Realizacion del documento final | 40d | 06/09/2010 | 29/10/2010 | 28 |
| **Tarea J: PRESENTACION** | 10d | **01/11/2010** | **12/11/2010** | **27,29** |
| J.1 Realizacion de la presentacion final | 10d | 01/11/2010 | 12/11/2010 | 29 |

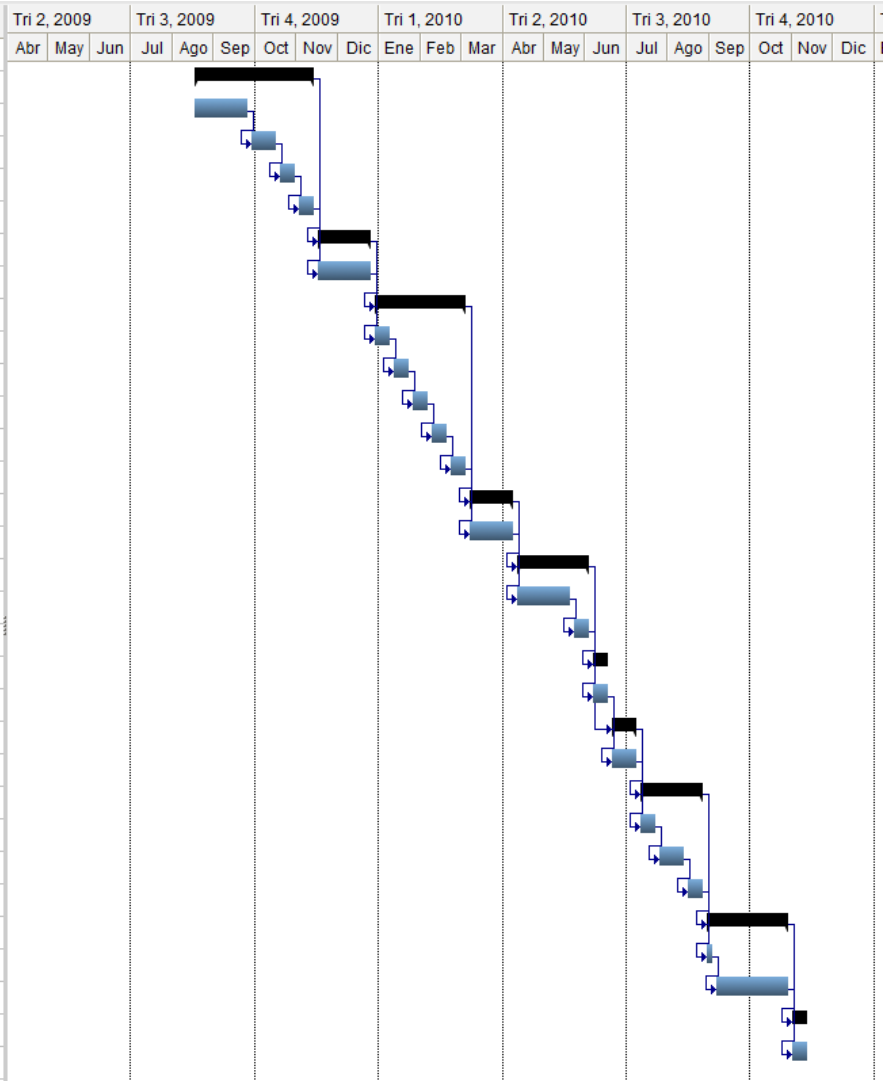


Figura A.2: Diagrama de Gantt.

## A.4. Recursos

En esta sección se describen los distintos recursos que son necesarios para la realización del proyecto:

- Recursos materiales:
  - Ordenadores de sobremesa.
    - 3 PCs con procesador Intel Pentium 4 de doble núcleo, 3.4 GHz. 1 GB RAM. Sistema operativo: Linux, distribución Ubuntu 8.04 y Ubuntu 8.10.
  - Routers Linksys: 2 routers modelo WRT54G.
  - Routers Asus: 9 dispositivos modelo WL-500g Premium.
  - Adaptadores inalambricos 802.11b/g USB.
    - Dos adaptadores Linksys, modelo WUSB54GC.
    - Dos adaptadores Pheenet, modelo WLU-803G.
  - Tarjetas inalámbricas con tecnología *Atheros*: Nueve tarjetas *Alfa Networks* AWPCI085S para reemplazar la tarjeta original de los routers Asus.
  - Equipamiento de red: 1 hub, varios cables Ethernet.
- Recursos de trabajo: 1 Ingeniero de telecomunicaciones

## A.5. Presupuesto de Proyecto

En esta sección se muestra el presupuesto final del Proyecto (Tabla A.2).

1. - Autor: María Isabel Sánchez Bueno
2. - Departamento: Ingeniería Telemática
3. - Descripción del Proyecto:
   - Título: Optimización de rutas para redes móviles en redes ad-hoc vehiculares
   - Duración: 15 meses
   - Tasa de costes indirectos: no se especifican.
4. - Presupuesto total del Proyecto (valorado en Euros): 59121.3 euros
5. Subcontratación de tareas: no se especifican.
6. Otros costes directos del proyecto: no se especifican.

| **Concepto** | **Cantidad** | **Coste (€)** | **Total (€)** |
|---|---|---|---|
| **Recursos materiales** | | | |
| Ordenadores de sobremesa | 3 | 550 | 1650 |
| Routers Linksys | 2 | 60 | 120 |
| Routers Asus | 9 | 65 | 585 |
| Adaptadores 802.11b/g USB Linksys | 2 | 22.5 | 45 |
| Adaptadores 802.11b/g USB Pheenet | 2 | 27 | 54 |
| Tarjetas inalámbricas Alfa Networks AWPCI085S | 9 | 29.70 | 267.3 |
| Equipamiento de red (hub, cable Ethernet) | - | - | - |
| Total | | | **2721.3** |
| **Recursos de trabajo** | | | |
| Ingeniero de telecomunicaciones | 1 (1410 horas) | 40€/hora | 56400 |
| Total | | | **56400** |
| **Otros** | | | |
| Documentación | - | - | - |
| Reuniones | - | - | - |
| Total | | | - |
| **Total** | | | **59121.3 €** |

Tabla A.2: Tabla presupuesto

# Apéndice B

# Mensajes del protocolo de Optimización de Rutas en Redes Vehiculares para NEMO - VARON

## B.1. Introducción

En este apéndice se va a realizar una descripción detallada de los mensajes del protocolo VARON. Los mensajes se encapsulan utilizando UDP.

## B.2. Home Address Advertisement (HoAA)

Este mensaje es enviado periódicamente por todos los routers móviles para anunciar su propia HoA al resto de routers móviles presentes en la red. La dirección de destino es la dirección local IPv6 multicast que identifica al grupo de todos los routers: FF02::02. El campo *Hop Limit (TTL)* del paquete IPv6 es VARONC_HOAA_TTL_MAX, cuyo valor por defecto es 10 segundos. En la Figura *B*.1 se muestra el formato de este mensaje. A continuación, sigue una breve descripción de los campos del mensaje:

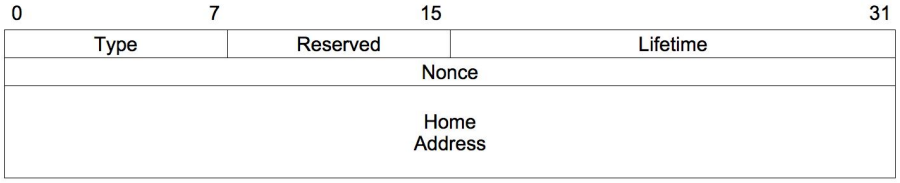


Figura B.1: Formato del mensaje HoAA

**Type** : campo de 8 bits que identifica al mensaje como un HoAA.

**Reserved** : campo de 8 bits reservado para un uso futuro. Debe ser inicializado a cero por el emisor e ignorado por el receptor.

**Lifetime** : entero sin signo de 16 bits. Representa el tiempo de vida del anuncio, en

segundos. Es utilizado para controlar la caducidad de las direcciones anunciadas. El valor 0 no debe ser utilizado y el valor máximo es de 18.2 horas.

**Nonce** : campo de 32 bits que contiene un número de secuencia que identifica de manera unívoca al mensaje HoAA enviado por un router móvil, para evitar el procesamiento de mensajes previamente recibidos, ya que estos mensajes son reenviados por los routers móviles mientras lo permita el campo *Hop Limit* del paquete IPv6. Al recibir un HoAA, éste será reenviado si no ha sido recibido previamente, es decir, si el campo *Nonce* no coincide con el de un mensaje que anunciara la misma HoA recibido anteriormente.

**Home Address** : dirección en la red hogar (128 bits) del router móvil emisor. Debe ser una dirección *unicast* válida y además debe coincidir con la dirección origen del paquete IPv6 que contiene el mensaje HoAA.

## B.3. Care-of Route Test Init (CoRTI)

Este mensaje se envía por un router móvil que ha detectado (mediante la recepción de un HoAA) que un nodo con el que tiene una comunicación establecida a través de Internet, está presente en la VANET. Este router móvil será denominado **MR origen**.

El mensaje es enviado a la dirección *multicast* de todos los routers: FF02::02 y la dirección origen será la HoA del router móvil que lo envía. El campo *Hop Limit* del paquete IPv6 es VARONC_CORTI_TTL_MAX, cuyo valor por defecto es de 10 saltos. En la Figura B.2 se muestra el formato de este mensaje. A continuación, sigue una breve descripción de los campos del mensaje:

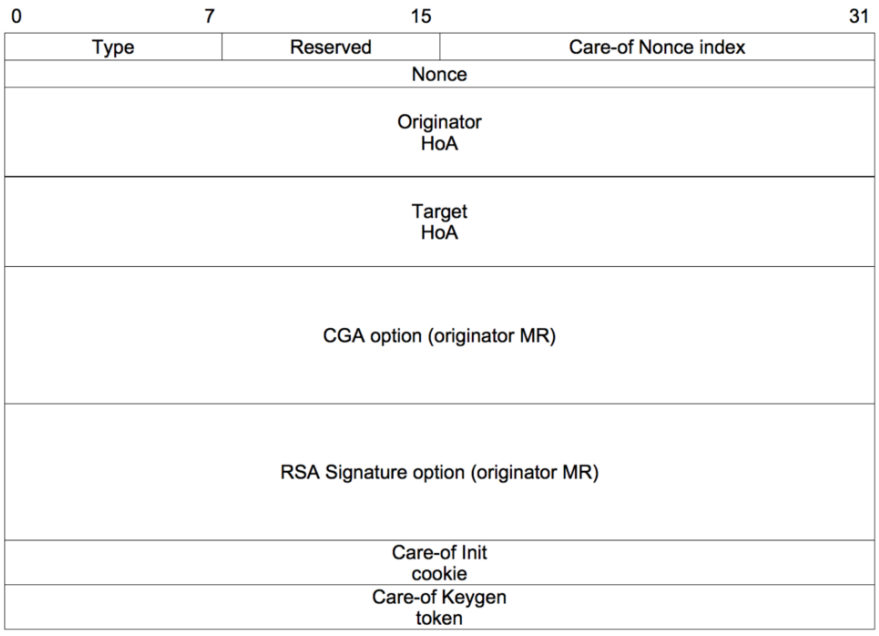


Figura B.2: Formato del mensaje CoRTI

**Type** : campo de 8 bits que identifica al mensaje como un CoRTI.

**Reserved** : campo de 8 bits reservado para un uso futuro. Debe ser inicializado a cero por el emisor e ignorado por el receptor.

**Care-of Nonce Index** : campo de 16 bits que contiene el índice del reto (*nonce index*) utilizado por el MR origen para generar el testigo *Care-of Keygen token*. Este campo debe ser duplicado en el mensaje MNPBU (***M**obile **N**etwork **P**refix **B**inding **U**pdate*) enviado de vuelta por el router móvil *objetivo* al final del proceso de establecimiento de la ruta.

**Nonce** : campo de 32 bits que contiene un número de secuencia, $N_A$, generado por el MR origen, que identifica al mensaje CoRTI enviado. Este valor debe ser copiado en el campo *Nonce* del mensaje CoRT generado como respuesta por el router *objetivo*.

**Originator HoA** : dirección en la red hogar (128 bits) del MR origen (el emisor del mensaje, que quiere establecer una ruta hacia el MR *objetivo*, *Care-of Route*). Debe ser una dirección *unicast* válida.

**Target HoA** : dirección en la red hogar (128 bits) del router móvil *objetivo*. Al igual que la del MR origen, debe ser una dirección *unicast* válida.

**CGA option (MR origen)** : campo de longitud variable que contiene la estructura de datos de la opción CGA (definida en la sección 5.1 de [AKZN05]).

**RSA Signature option (MR origen)** : campo de longitud variable que contiene la opción de la firma RSA (definida en la sección 5.2 de [AKZN05])

**Care-of Init cookie** : campo de 64 bits que contiene un valor aleatorio, *Care-of init cookie*, utilizado para proteger el proceso es establecimiento de rutas. Debe ser copiado en el campo *Care-of Init Cookie* del mensaje CoRT generado como respuesta por el router *objetivo*.

**Care-of Keygen token** : campo de 64 bits que contiene el testigo *Care-of Keygen Token* utilizado para generar la clave $K_{bm}$ que será utilizada en el proceso de autenticación de la *Care-of Route*.

Un mensaje CoRTI reenviado por un router móvil intermedio tiene un formato ligeramente distinto al descrito, ya que cada router añade la información referente a su CGA y firma RSA. Este formato se muestra en la Figura B.3.

### B.3.1. Opciones RSA

Esta opción incluye información sobre la firma digital realizada por cada router móvil, que se incluye en los mensajes de tipo CoRTI y CoRT. El formato de esta opción, definida en [AKZN05], se muestra en la Figura B.4. A continuación, se describe cada uno de los campos de esta opción.

**Type** : Campo de 1 byte de longitud con valor 11.

**Length** : Longitud de la opción, incluyendo los campos *Type, Length, Reserved, Key Hash, Digital Signature, Padding* en unidades de 8 octetos.

**Reserved** : Campo de 16 bits reservado para un uso futuro. Debe ser inicializado a cero por el emisor e ignorado por el receptor.

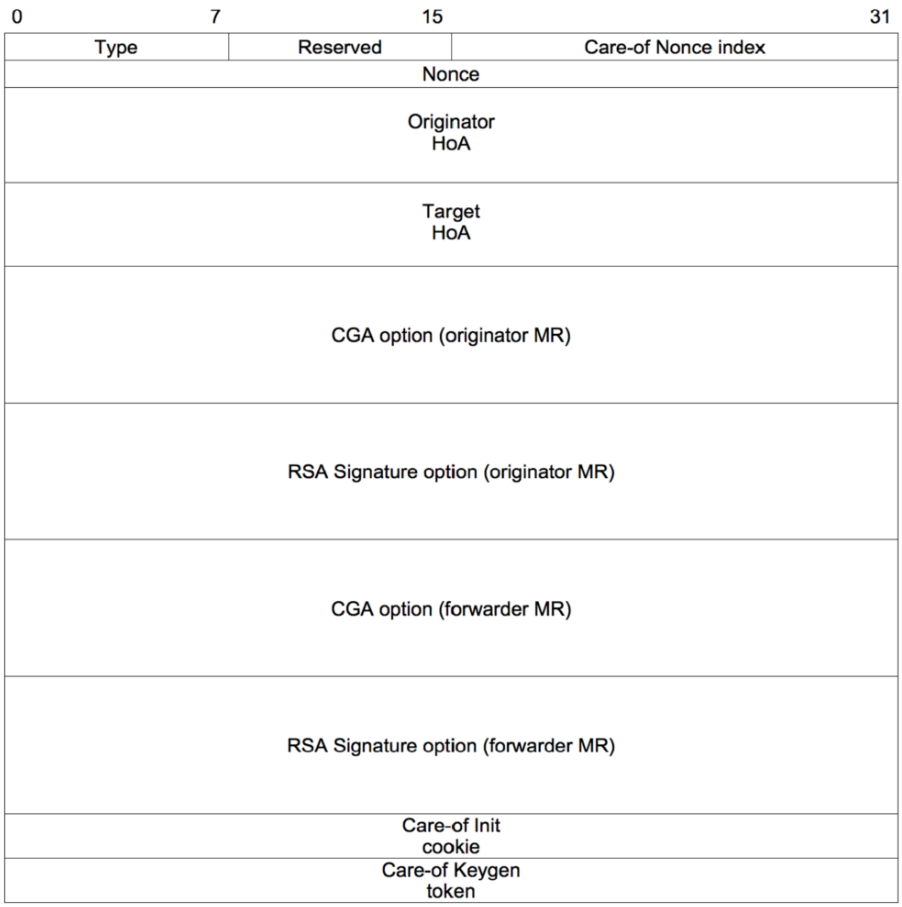


Figura B.3: Formato del mensaje CoRTI reenviado por un router intermedio

**Key Hash** : Campo de 128 bits que contiene los 128 bits más significativos (por la izquierda) del resumen hash (SHA1) de la clave pública utilizada para firmar el mensaje correspondiente.

**Digital Signature** : Campo de longitud variable que contiene la firma, realizada con la clave privada del emisor del mensaje sobre los siguientes datos:

- Campo *Type* del mensaje de VARON correspondiente.
- Campo *Reserved* del mensaje de VARON correspondiente.
- Campo *Nonce* del mensaje de VARON correspondiente.
- Dirección IPv6 origen del mensaje, es decir, la HoA del emisor del mensaje.
- Dirección IPv6 destino del mensaje, es decir, la HoA del destinatario final del mensaje.
- Opción CGA, de longitud variable.
- Si se trata de un mensaje reenviado, opciones CGA y RSA del MR que originó el mensaje por primera vez.

**Padding** : Campo de longitud variable para hacer que la longitud total de la opción sea un múltiplo de 8 octetos.

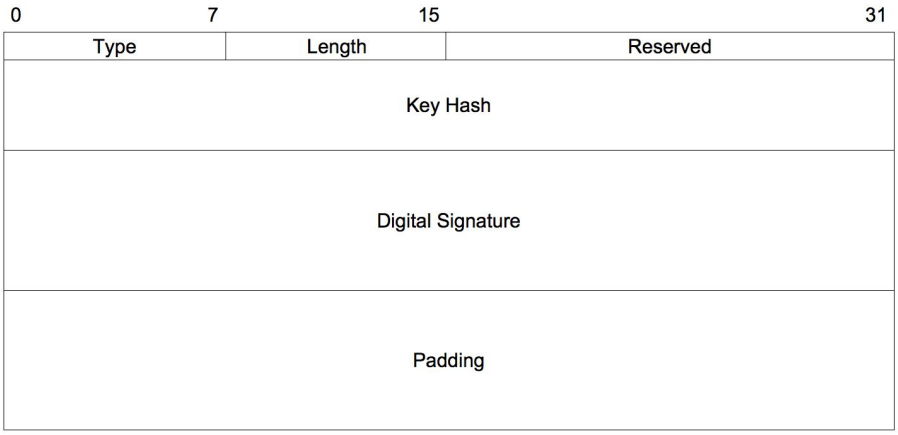


Figura B.4: Formato de la opción con la información necesaria sobre la firma digital (RSA) del router emisor del mensaje

### B.3.2. Opciones CGA

Esta opción se incluye en los mensajes CoRTI y CoRT para permitir verificar la identidad de los routers móviles *origen* y *objetivo*. El formato de esta opción, definida en [AKZN05], se muestra en la Figura B.5. A continuación, se describe cada uno de los campos de esta opción.

**Type** : Campo de 1 byte de longitud con valor 11.

**Length** : Longitud de la opción, incluyendo los campos *Type, Length, Pad Length, Reserved, CGA Parameters, Padding* en unidades de 8 octetos.

**Pad Length** : Número de octetos de padding al final de la opción CGA (contenidos en el campo *Padding*). Debe ser inicializado a cero por el emisor e ignorado por el receptor.

**Reserved** : Campo de 8 bits reservado para un uso futuro. Debe ser inicializado a cero por el emisor e ignorado por el receptor.

**CGA Parameters** : Campo de longitud variable, conteniendo los siguientes campos:

- **Modifier** : Campo de 128 bits, que contiene un entero sin signo, que puede tener cualquier valor. Es utilizado en la generación de la CGA, añadiendo mayor aleatoriedad a la dirección generado.
- **Subnet Prefix** : Campo de 64 bits que contiene el prefijo de red de la CGA, es decir el MNP en el caso de VARON.
- **Collision count** : Entero sin signo de 8 bits, de valor 0, 1 o 2. Es incrementado en la generación de la CGA al detectar que la dirección generada está duplicada (Por el mecanismo DAD, *Duplicate Address Detection*).
- **Public Key** : Campo de longitud variable que contiene la clave pública del emisor del mensaje.
- **Extension Fields** : Campo de longitud variable que no se utiliza en la especificación estudiada. Puede que se incluya en futuras versiones.

**Padding** : Campo de longitud variable para hacer que la longitud total de la opción sea múltiplo de 8 octetos.

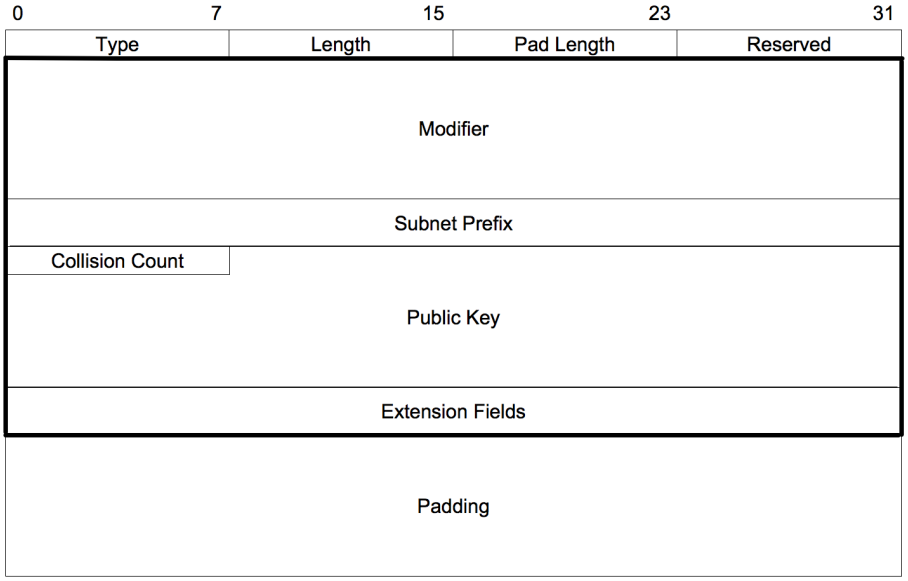


Figura B.5: Formato de la opción con la información necesaria sobre la CGA del router emisor del mensaje

## B.4. Care-of Route Test (CoRT)

Este mensaje es enviado por el MR objetivo en respuesta a un mensaje CoRTI. Este router móvil establece su dirección hogar como la dirección origen del mensaje, y la del router móvil que le envió el CoRTI como dirección destino del paquete enviado. El formato del mensaje, cuyos campos son análogos a los del mensaje CoRTI, se muestra en la Figura B.6.

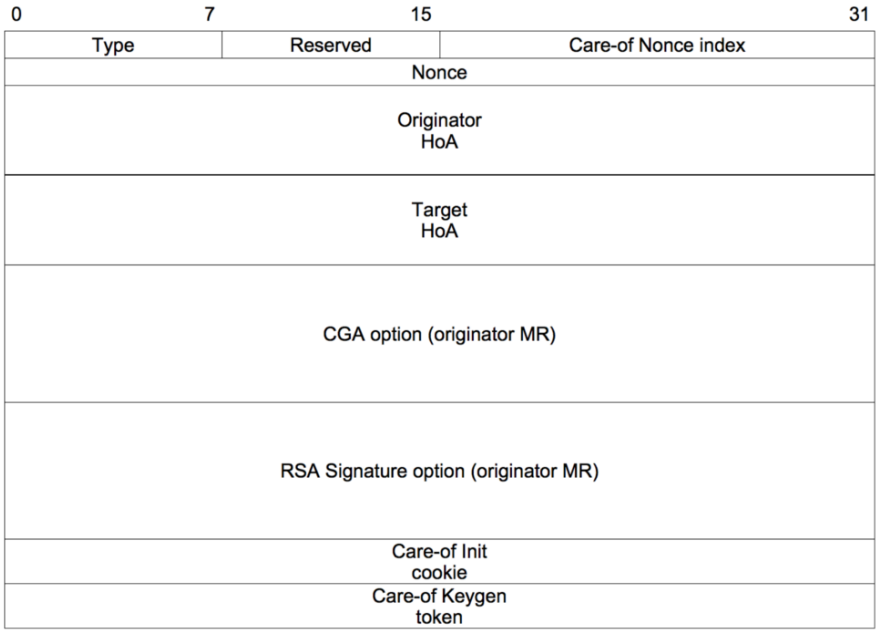


Figura B.6: Formato del mensaje CoRT

El MR emisor del mensaje CoRT debe incluir el número de secuencia (*nonce*) y la *Care-*

*of Init cookie* recibidos en el mensaje CoRTI correspondiente. El formato de un mensaje CoRTI reenviado por un router móvil intermedio, al igual que en el caso del mensaje CoRTI, es ligeramente diferente, ya que cada router a lo largo del camino entre origen y destino añade sus propias opciones CGA y RSA eliminando las del salto anterior, tras haberlas comprobado. El formato de éste se puede ver en la Figura B.7.

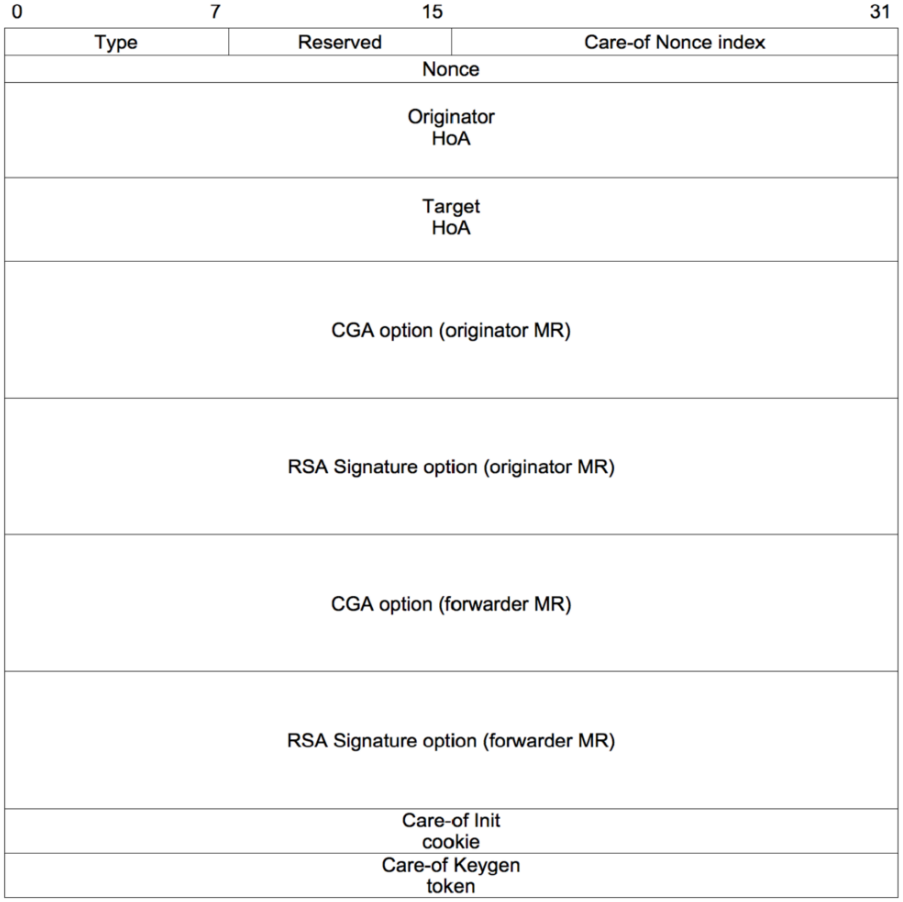


Figura B.7: Formato del mensaje CoRT reenviado por un router intermedio

## B.5. Home Route Test (HoRT)

Este mensaje es enviado por un router móvil tras la recepción de un CoRT o en respuesta a otro HoRT. El objetivo del mensaje es verificar que el router móvil con el que se quiere establecer una ruta a través de la red ad-hoc, es verdaderamente el encargado de gestionar el prefijo de la red móvil que anuncia, para evitar posibles ataques. Por esta razón, el mensaje es enrutado a través de la infraestructura hacia la red hogar, intentando probar que el router móvil recibe el tráfico dirigido a su HoA (o a una dirección perteneciente a su MNP) por ambas redes.

Las direcciones origen y destino del paquete UDP son las HoAs de cada uno de los MR involucrados. En la Figura *B.8* se muestra el formato de este mensaje. A continuación, sigue una breve descripción de los campos del mensaje:

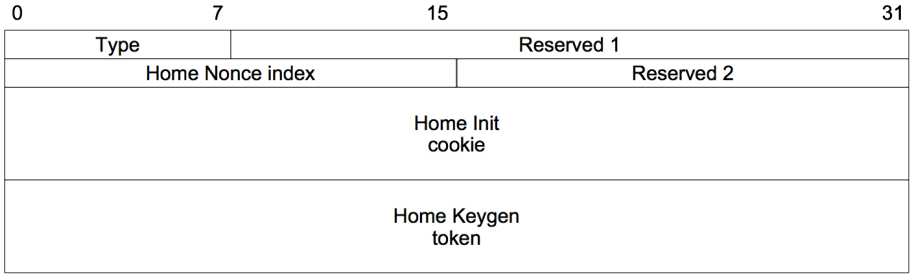


Figura B.8: Formato del mensaje HoRT

**Type** : campo de 8 bits que identifica al mensaje como un HoRT. Para identificar si el mensaje es una petición o una respuesta, ya que en el proceso de señalización se envían dos HoRT, uno por cada MR, debe tener el valor 52 si es el primero de los dos mensajes, o el valor 50 si se envía en respuesta a otro HoRT recibido previamente.

**Reserved 1** : campo de 24 bits reservado para un uso futuro. Debe ser inicializado a cero por el emisor e ignorado por el receptor.

**Home Nonce Index** : campo de 16 bits que contiene el índice del reto (*nonce index*) utilizado por el MR emisor para generar el testigo *Home Keygen token*. Este campo debe ser duplicado en el mensaje MNPBU (***M**obile **N**etwork **P**refix **B**inding **U**pdate*) enviado de vuelta por el router móvil *objetivo* al final del proceso de establecimiento de la ruta.

**Reserved 2** : campo de 16 bits reservado para un uso futuro. Debe ser inicializado a cero por el emisor e ignorado por el receptor.

**Home Init cookie** : campo de 64 bits que contiene un valor aleatorio, *Home init cookie*.

**Home Keygen token** : campo de 64 bits que contiene el testigo *Home Keygen token* utilizado en el proceso de señalización para generar la clave $K_{bm}$ que será utilizada en el proceso de autenticación de la *Care-of Route*.

## B.6. Mobile Network Prefix Binding Update (MNPBU)

Este mensaje es enviado por un router móvil tras la recepción de un HoRT o en respuesta a otro MNPBU. El objetivo del mensaje es verificar que el router móvil con el que se quiere establecer una ruta a través de la red ad-hoc, es verdaderamente el encargado de gestionar el prefijo de la red móvil que anuncia, para evitar posibles ataques. El mensaje es enrutado a través de la red ad-hoc incluyendo un código de autenticación de mensaje (MAC, por sus siglas en inglés) calculado a partir de la clave $K_{bm}$, así como la información necesaria para que el router móvil receptor del mensaje pueda obtener esa clave y comprobar la validez del mensaje autenticado.

Las direcciones origen y destino del paquete UDP son las HoAs de cada uno de los MR involucrados. En la Figura *B.9* se muestra el formato de este mensaje. A continuación, sigue una breve descripción de los campos del mensaje:

**Type** : campo de 8 bits que identifica al mensaje como un MNPBU. Para identificar si el mensaje es una petición o una respuesta, ya que en el proceso de señalización

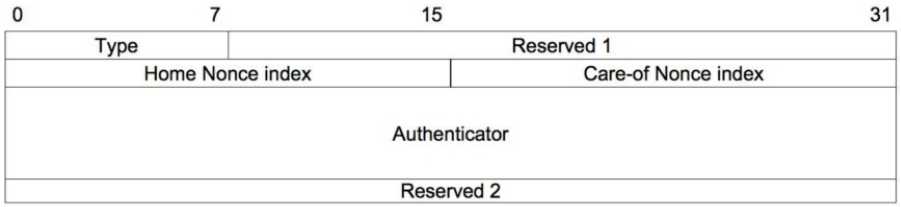


Figura B.9: Formato del mensaje MNPBU

se envían dos MNPBU, uno por cada MR, debe tener el valor 32 si es el primero de los dos mensajes, o el valor 30 si se envía en respuesta a otro MNPBU recibido previamente.

**Reserved 1** : campo de 24 bits reservado para un uso futuro. Debe ser inicializado a cero por el emisor e ignorado por el receptor.

**Home Nonce Index** : campo de 16 bits que contiene el índice del reto (*nonce index*) utilizado por el MR emisor para generar el testigo *Home Keygen token*. Este campo es una copia del enviado en el HoRT por el otro router móvil involucrado en el proceso de señalización.

**Care-of Nonce Index** : campo de 16 bits que contiene el índice del reto (*nonce index*) utilizado por el MR emisor para generar el testigo *Care-of Token*. Este campo es una copia del enviado en el CoRTI o en el CoRT (según el MR que lo envíe).

**Authenticator** : campo de 96 bits que contiene la información necesaria para autenticar al MR: consiste en una función *hash* (SHA-1) de la clave $K_{bm}$ y las HoAs de los dos MR involucrados en el establecimiento de la ruta.

**Reserved 2** : campo de 32 bits reservado para un uso futuro. Debe ser inicializado a cero por el emisor e ignorado por el receptor.

## B.7. Care-of Route Error (CoRE)

Este mensaje no ha sido implementado en este proyecto fin de carrera, pero se incluye su definición por completitud. Este mensaje es enviado por un router intermedio cuando detecta que la ruta entre dos MRs ha dejado de funcionar por cualquier motivo. En la Figura *B*.10 se muestra el formato de este mensaje. A continuación, sigue una breve descripción de los campos del mensaje:

**Type** : campo de 8 bits que identifica al mensaje como un CoRE.

**Reserved** : campo de 24 bits reservado para un uso futuro. Debe ser inicializado a cero por el emisor e ignorado por el receptor.

**Nonce** : campo de 32 bits que contiene el número de secuencia (*nonce*) utilizado por el MR emisor del mensaje. El objetivo de este campo es asegurar la originalidad del mensaje.

**Source HoA** : dirección de 128 bits, *Home Address* del origen de la ruta que ha dejado de funcionar.

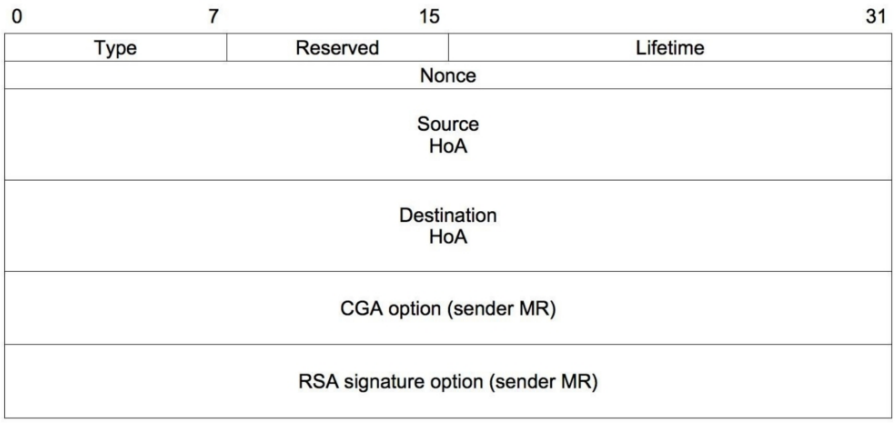


Figura B.10: Formato del mensaje CoRE

**Destination HoA** : dirección de 128 bits, *Home Address* del otro extremo de la ruta que ha dejado de funcionar.

**CGA option (sender MR)** : campo de longitud variable que contiene la estructura de datos de la opción CGA.

**RSA signature option (sender MR)** : campo de longitud variable que contiene los datos de la opción RSA.

Al igual que en los mensajes CoRTI y CoRT, los routers intermedios que reenvían el mensaje añaden sus propias opciones CGA y RSA, eliminando las de otro MR intermedio si las hubiera, tras haberlas comprobado. El formato de un mensaje reenviado se puede ver en la Figura *B*.11.

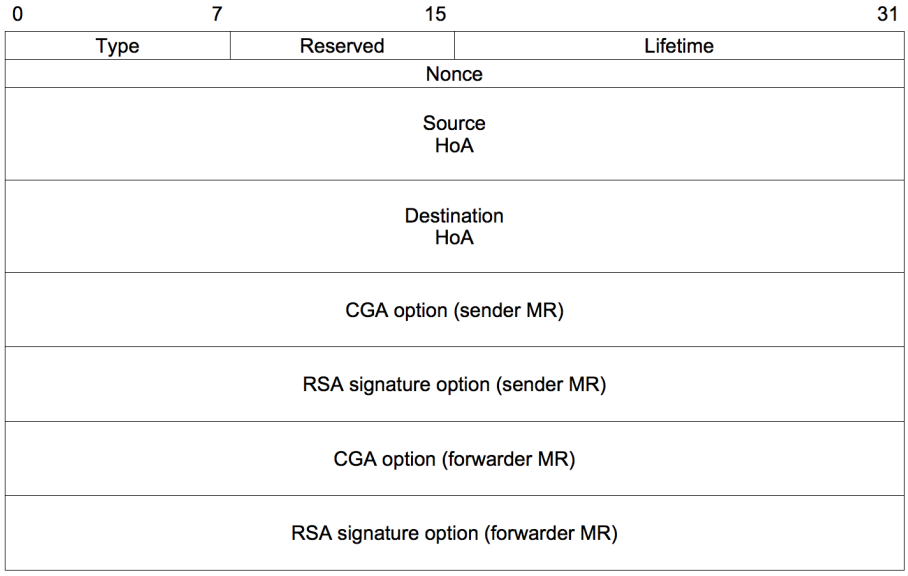


Figura B.11: Formato del mensaje CoRE reenviado por un router intermedio

# Apéndice C

# Instalación de OpenWrt en el router ASUS WL-500g Premium

## C.1. Introducción

En este apéndice se presenta el router inalámbrico Asus WL500g Premium (Figura C.1). Al hablar de la marca ASUS generalmente se asocia con componentes de PC tales como tarjetas gráficas y placas base, a pesar de que la compañía tiene presencia en otros sectores de mercado también. Muestra un desarrollo en el campo de los *netbooks* (Asus EeePc) y en los dispositivos en red.

## C.2. El router ASUS WL-500g Premium

Figura C.1: Router Asus WL500g Premium

Este router puede ser considerado como un pequeño ordenador debido a sus características. En la Tabla C.1 se muestran las especificaciones del router ASUS WL-500G PREMIUM.

Además, una de las características que hacen este router más adecuado para este proyecto es la presencia de varias interfaces, así como dos puertos USB, a los que se puede conectar, por ejemplo, un adaptador inalámbrico para tener también varias interfaces

| | |
|---|---|
| **Arquitectura** | MIPS |
| **Vendor** | Broadcom |
| **Bootloader** | CFE |
| **CPU Speed** | 266 MHz |
| **System-On-Chip** | Broadcom BCM94704 |
| **Flash size** | 8MiB (Spansion S29GL064M90) |
| **RAM** | 32MiB (2 HY50U281622ETP-5, algunas unidades antiguas tienen sólo 16MiB) |
| **Wireless** | MiniPCI Broadcom 802.11b/g BCM4318 802.11 Wireless LAN Controller[1] |
| **Ethernet** | Robo switch BCM5325 |
| **USB** | Sí |
| **Serial** | Sí |
| **JTAG** | No |

Tabla C.1: Especificaciones técnicas del router ASUS WL-500G PREMIUM

inalámbricas. En la Figura C.2 se observa la parte trasera del router con las cuatro interfaces LAN, una interfaz WAN y dos puertos USB, así como los botones RESTORE (esencial para cambiar el firmware) y EZSETUP (de color rojo).

Figura C.2: Vista trasera del router Asus WL500g Premium

Aunque el router viene de fábrica con muchas funcionalidades que lo hacen un dispositivo muy atractivo y de gran utilidad, para realizar las pruebas de este proyecto sería más útil tener un firmware como OpenWrt, que además de estar basado en Linux, permitiera configurar todos los parámetros que es necesario cambiar.

## C.3. El *firmware* OpenWrt

El *firmware* es un bloque de instrucciones de programa para propósitos específicos, grabado en una memoria de tipo no volátil (ROM, EEPROM, *flash*,...), que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo. Funcionalmente, el *firmware* es el intermediario (interfaz) entre las órdenes externas que recibe el dispositivo y su electrónica.

El *firmware* se encuentra en memorias ROM de los sistemas de diversos dispositivos periféricos, como en monitores de vídeo, unidades de disco, impresoras, etc., pero también en los propios microprocesadores, chips de memoria principal y en general en cualquier circuito integrado.

En un microprocesador el *firmware* es el que recibe las instrucciones de los programas y las ejecuta en la circuitería del mismo, emitiendo órdenes a otros dispositivos del sistema. Muchos de los *firmwares* almacenados en ROM están protegidos por Derechos de Autor.

OpenWrt (Figura *C.3*) es un *firmware* libre, basado en GNU/Linux optimizado para routers con un *hardware* determinado y reducidas capacidades. El proyecto comenzó en 2004, basándose en un primer momento en el código fuente del router Linksys WRT54G, como mera referencia. Pero pronto se comenzó a dar soporte a modelos parecidos, incluso de otros fabricantes y de ahí a diferentes arquitecturas y *chipsets*.

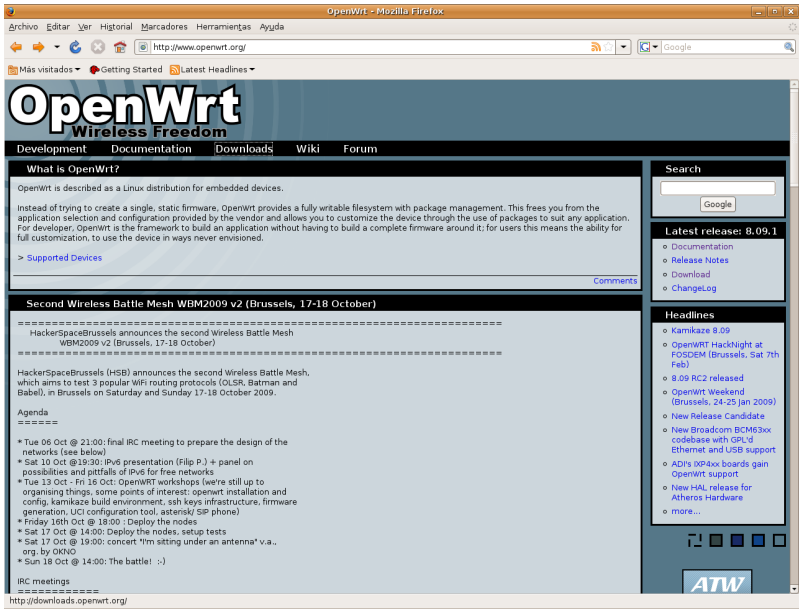


Figura C.3: Página web de OpenWrt.

Básicamente, OpenWrt permite elegir dos tipos de sistemas de archivos:

- Squashfs: Un sistema de ficheros que proporciona una partición de sólo lectura. Esta partición contiene un entorno Linux mínimo, preparado para arrancar el router y proveer lo básico. Esto hace necesaria una partición JFFS2 para poder configurar el router, es decir, para posibilitar volver a la configuración por defecto si por un fallo se deja éste inaccesible.

- JFFS2: Un sistema de ficheros con una partición de lectura y escritura. Especialmente diseñado para transacciones en memorias *flash*. Todo lo que no provee la partición *squashfs* puede ser incorporado en esta partición.

OpenWrt tiene tres líneas de desarrollo:

- *WhiteRussian*. Distribución ya estable, que utiliza un kernel Linux 2.4.x y sólo soporta unas pocas arquitecturas *hardware*.

- *Kamikaze*. Es la que más se está desarrollando actualmente. Utiliza kernel Linux 2.4 y 2.6. Soporta más arquitecturas, por lo tanto, mayor número de dispositivos. Kamikaze v8.09.1 es la última versión estable con fecha de julio de 2009.

- *Backfire*. Es la versión más actual. Incluye mejoras, como mayor soporte de interfaces inalámbricas, interfaz gráfica, etc. En el momento en el que se comenzó el trabajo de este proyecto no estaba disponible aún.

Debido a que la distribución *WhiteRussian* no soporta un kernel Linux 2.6 y tampoco tiene desarrollo en la actualidad (última actualización con fecha de febrero de 2007 versión rc6) se descarta su uso. Por tanto, la distribución de OpenWrt que se decidió instalar es Kamikaze, en su versión 8.09, que en su momento era la versión estable más actual. Es posible elegir entre un kernel 2.4 y 2.6. El kernel 2.4 funciona completamente bien, pero no incluye algunos módulos imprescindibles para esta implementación, como el soporte para IPv6, por lo que se descartó. Por otro lado, debido a las características del router, para el kernel 2.6 el funcionamiento de la interfaz inalámbrica está limitado, no siendo posible que funcione en modo AP. Por esta razón, y para evitar otros posibles problemas de compatibilidad, se cambió la tarjeta inalámbrica original del router por una tarjeta Atheros, que puede funcionar a pleno rendimiento.

## C.4. Cambio del *firmware*

El proceso de cambio de firmware puede realizarse de varias maneras. La utilizada en este caso, y quizá la más sencilla, es transferir el archivo con la imagen que se quiere instalar mediante TFTP, mientras el router está en un modo "a prueba de fallos"(*diag mode*). Estos son los pasos a seguir:

- Primero, el router debe estar en este modo especial, que ha sido llamado "a prueba de fallos". Para ello:

  1. Desconectar el router de la alimentación.
  2. Conectar el puerto 1 del router directamente al ordenador utilizado para realizar la instalación.
  3. Apretar el botón RESTORE, de color negro utilizando un lápiz por ejemplo, y mantenerlo pulsado unos segundos.
  4. Mientras se mantiene pulsado, conectar el cable de alimentación.
  5. Cuando la luz de *power* esté parpadeando, eso indicará que el router ya se encuentra en el modo "a prueba de fallos". Ahora el router será accesible en la dirección 192.168.1.1, lo que se puede confirmar mediante un *ping*.

- En esta situación, el router aceptará una imagen vía TFTP. Para transferir el archivo con la imagen del firmware e instalarlo, estos son los pasos a seguir:

  1. Situándose en el directorio en el que previamente habremos descargado la versión del firmware que se desea instalar [2], comprobando que la imagen sea la adecuada para la arquitectura del dispositivo que se va a utilizar, en este caso el fichero es `openwrt-brcm47xx-squashfs.trx` , habrá que ejecutar la siguiente secuencia de comandos:

     ```
     tftp 192.168.1.1
     tftp> binary
     tftp> trace
     tftp> put openwrt-brcm47xx-squashfs.trx
     ```

     **binary** : activación del modo de transferencia binaria

[2] http://downloads.openwrt.org/kamikaze/8.09/brcm47xx/openwrt-brcm47xx-squashfs.trx

**trace** : activación del modo de depuración para poder ver las trazas del programa

**put** : transferencia de la imagen del *firmware*

2. Una vez que se haya transferido completamente la imagen, habrá que esperar unos minutos, ya que primero se carga la imagen en la RAM y después se procede a la instalación del *firmware*. Cuando la instalación se haya completado, es necesario reiniciar el router, desconectando y volviendo a conectar el cable de la alimentación. El router seguirá siendo accesible en la dirección 192.168.1.1, así que haciendo *telnet* a esa dirección es posible comprobar que la instalación se ha realizado con éxito y comenzar a configurar el router.

Puede resultar conveniente habilitar el acceso por *ssh* en lugar de telnet para tener una interfaz de acceso más segura. Para ello, se utiliza el comando *passwd*, que solicitará una contraseña para el usuario (en este caso el usuario *root*). La próxima vez que se acceda al router tendrá que ser mediante *ssh* y utilizando la contraseña elegida.

# Apéndice D

# Detalles de instalación y configuración necesarios en los routers

Una vez que se ha cambiado el firmware original del router por la distribución de OpenWrt de acuerdo a lo explicado en el apéndice C, ya se puede empezar a configurar el router según las necesidades de la implementación realizada.

En primer lugar, se instalan los módulos necesarios para incluir funcionalidades que serán necesarias para el establecimiento de rutas, configuración de interfaces de red y túneles, y el uso de los adaptadores USB inalámbricos.

A continuación, se cargan los archivos ejecutables de la solución implementada y se añaden los ficheros de configuración para que todo funcione correctamente.

## D.1. Configuración inicial

Por defecto, en la imagen del firmware básica que se le ha instalado a los routers, hay algunos módulos propios del kernel que no vienen incluidos. Por tanto, lo primero será instalar una serie de paquetes que serán de mucha utilidad para configurar interfaces de red y rutas, o para que los adaptadores USB inalámbricos sean reconocidos por el sistema. Para ello, se conecta el router a Internet mediante un cable de red por el puerto WAN y se introducen los siguientes comandos:

```
opkg update
```

Es necesario actualizar la lista de paquetes disponibles cada vez que se quiera realizar alguna operación con opkg.

```
opkg install ip kmod-ip6tables kmod-ip6-tunnel
```

Estos paquetes son necesarios para incluir soporte de IPv6, toda la configuración de las direcciones y rutas, así como la formación de túneles.

```
opkg install kmod-usb2 kmod-net-zd1211rw kmod-rt73-usb
```

Estos paquetes son necesarios para utilizar los adaptadores inalámbricos USB. La instalación y configuración necesarias en el router para la utilización de estos adaptadores se explicará en la sección D.2. Si se quisieran utilizar los puertos USB para conectar dispositivos de almacenamiento o de otro tipo, se necesitaría instalar otros paquetes, pero ese no es el caso, por lo que sólo se instalan los relacionados con los adaptadores inalámbricos.

Otra herramienta que puede resultar interesante instalar es *tcpdump* para poder comprobar el intercambio de mensajes entre los distintos routers móviles. Para instalarlo simplemente hay que ejecutar:

```
opkg install tcpdump
```

Continuando con la configuración necesaria, se modifican los ficheros `/etc/config/network` y `/etc/config/wireless`, quedando como se muestra a continuación:

```
# Copyright (C) 2006 OpenWrt.org

config switch eth0
        option vlan0    "1 5*"
        option vlan1    "2 3 4 5"
        option vlan2    "0 5"

config interface loopback
        option ifname   "lo"
        option proto    static
        option ipaddr   127.0.0.1
        option netmask  255.0.0.0

config interface lan
        option ifname   eth0.0
        #option type    bridge
        option proto    static
        option ipaddr   192.168.1.8
        option netmask  255.255.255.0

config interface lan2
        option ifname   eth0.1
        option proto    static
        option ipaddr   192.168.2.8
        option netmask  255.255.255.0

config interface wan
        option ifname   eth0.2
        option proto    static

config interface wifi
```

```
        option ifname   ath0
        option proto    static
        option ipaddr   192.168.3.8
        option netmask  255.255.255.0


config wifi-device  wl0
        option type     atheros
        option country  ES
        option channel  40
        option agmode   a
        #REMOVE THIS LINE TO ENABLE WIFI:
        #option disabled 1

config wifi-iface
        option device   ath0
        #option network  lan
        option mode     ad-hoc
        option encryption none

onfig wifi-device  wifi1
        option type     mac80211
        option country  ES
        option channel  4
        option hwmode   11g

config wifi-iface
        option device   wlan0
        #option network  lan
        option mode     sta
        option ssid     nemo
        option encryption none
```

Se salvan los cambios en ambos archivos y se reinicia el router para que la configuración se actualice:

```
/etc/init.d/network restart
```

En este fichero se puede comprobar como la interfaz Atheros se utilizará en la red ad-hoc, que además utiliza 802.11a y la interfaz USB se utilizará para conectar con la infraestructura.

Para evitar posibles conflictos y que el router no se comporte de la manera esperada, se inhabilitan el *firewall* y los servidores de *DNS* y *HTTP* de la siguiente forma:

```
OpenWrt:/# /etc/init.d/firewall stop
OpenWrt:/# rm /etc/init.d/firewall
OpenWrt:/# /etc/init.d/dnsmasq stop
OpenWrt:/# rm /etc/init.d/dnsmasq
```

```
OpenWrt:/# /etc/init.d/httpd stop
OpenWrt:/# rm /etc/init.d/httpd
```

Por último, es necesario habilitar el reenvío de tráfico, para que el router realmente se comporte como tal, reencaminando los paquetes que no van dirigidos a él de acuerdo a su tabla de rutas. Para ello, la línea que aparece comentada en el archivo `/etc/sysctl.conf` se debe descomentar:

```
net.ipv6.conf.all.forwarding=1
```

### D.1.1. Cambio de la interfaz inalámbrica original del router Asus

Como se ha comentado en varias ocasiones, la tarjeta inalámbrica con la que viene de fábrica el router Asus WL-500g Premium fue reemplazada por otra tarjeta, *Atheros*, concretamente, el modelo utilizado fue el AWPCI085S de *Alfa Networks*. Este cambio, permite una mayor flexibilidad en la configuración, así como un mayor rango de funcionamiento, por ejemplo, en el rango de frecuencias de 802.11a.

Los pasos a seguir para cambiar una tarjeta inalámbrica por otra se detallan a continuación. En la Figura D.1 se puede ver una imagen del interior del router, antes y después del cambio.

- Abrir la carcasa del router con un destornillador. En la parte inferior del router se encuentran unos tornillos pequeños, tapados con una cubierta de goma.
- Apartar dos pestañas a los lados de la tarjeta inalámbrica para poder extraerla de su posición.
- Colocar la nueva tarjeta inalámbrica en la posición de la anterior, volviendo a ajustar las pestañas de los laterales.
- Atornillar de nuevo la carcasa.

Figura D.1: Vista del interior del router móvil antes y después de cambiar la tarjeta inalámbrica original

Tras realizar este cambio, es necesario instalar el paquete `kmod-madwifi`, antes de empezar a configurar la nueva tarjeta con tecnología *Atheros*:

```
opkg update
opkg install kmod-madwifi
```

## D.2. Configuración para las interfaces inalámbricas USB

Los adaptadores USB permiten tener varias interfaces inalámbricas, ya que el router Asus cuenta con una interfaz propia y dos puertos USB. Esto también podría utilizarse para conectar un adaptador con tecnología Bluetooth o 3G en lugar de 802.11b/g.

Inicialmente, se realizaron pruebas para evaluar el funcionamiento de los adaptadores inalámbricos conectados en un ordenador, para aislar las limitaciones debidas al dispositivo y las ocasionadas por las del propio router. Para que la interfaz inalámbrica funcione, es necesario instalar los controladores del dispositivo. La instalación difiere según se trate del adaptador Linksys o Pheenet, y según la distribución de Línux. A continuación se describen los pasos necesarios para instalar los controladores de ambas interfaces en un sistema *Ubuntu* y en uno *Debian*.

Por otro lado, para poder utilizar estas interfaces en el router móvil es necesario realizar en él una serie de cambios. Además, el uso de estas interfaces se verá limitado, ya que no es posible configurarlas en modo AP, sólo en modo ad-hoc o como estación, conectándose a un punto de acceso. De todas formas, estas características son suficientes para el papel que van a jugar en la implementación desarrollada. Se ha estudiado el funcionamiento de dos adaptadores: WUSB54GC de Linksys y WLU-803G de Pheenet, cuyas principales características y proceso de configuración se explica a continuación.

### D.2.1. Linksys WRT54GC

Figura D.2: Adaptador USB Linksys utilizado como interfaz inalámbrica adicional

En la Tabla D.1 se muestran las especificaciones del USB Linksys 54GC.

Las características clave son las siguientes:

- Compatible con los estándares 802.11g y 802.11b (2,4 GHz)
- Admite USB 2.0 a un máximo de 54 Mbps de transferencia de datos con reserva automática
- Admite seguridad de encriptación WEP y WPA de hasta 128 bits
- Compatible con la configuración Wi-Fi protegida (WPS), que permite una configuración segura y sencilla

| | |
|---|---|
| **Modelo** | WUSB54GC |
| **Dimensiones** | 21.05 mm x 73.63 mm x 9.71 mm |
| **Canales** | 13 canales (Europa) |
| **Modulación** | 802.11b: CCK (11 Mbps), DQPSK (2 Mbps), DBPSK (1 Mbps); 802.11g: OFDM |
| **Protocolo** | Enlace |
| **Potencia de transmisión** | 802.11g: 14 ±1,5 dBm (habitualmente) 802.11b: 17 ±1,5 dBm (habitualmente) |
| **Sensibilidad de recepción** | 11 Mbps: -87 dBm (habitualmente) 54 Mbps: -71 dBm (habitualmente) |
| **Funciones de seguridad** | Encriptación WEP y WPA |
| **Bits de clave de seguridad** | 64 bits y 128 bits |

Tabla D.1: Especificaciones técnicas de la antena USB Linksys 54GC

#### D.2.1.1. Instalación necesaria en PC

El primer paso, obviamente, consiste en descargar el controlador y descomprimirlo en el PC en cuestión. A partir de ahí, la configuración para cada distribución de Linux utilizada se describe en las siguientes secciones.

***Ubuntu*** En el caso de utilizar una distribución Ubuntu es necesario situar cada archivo en el lugar adecuado. La secuencia de comandos necesaria es la siguiente:

```
cd RT73_Linux_STA_Drv1.0.4.0/Module
cp Makefile.6 Makefile
```

Es conveniente comprobar que el dispositivo que se quiere instalar se encuentra entre los listados en la documentación del controlador. Se puede averiguar el identificador del adaptador inalámbrico con el comando `lsusb`. En este caso, el identificador del WUSB54GC es `0x13b1,0x0020`. En el archivo `rtmp_def.h` debe aparecer ese identificador, en el apartado VID/PID (Vendor ID/Product ID). Habiendo realizado esta comprobación, compilar el controlador, situar el módulo en el directorio adecuado y cargarlo en el *kernel*:

```
make
cp rt73.ko /lib/modules/$(uname -r)/kernel/drivers/net/wireless/
insmod /lib/modules/$(uname -r)/kernel/drivers/net/wireless/rt73.ko
```

Actualizar `modules.dep` y situar los ficheros binarios en sus directorios correspondientes:

```
depmod -a
cp rt73.bin /lib/firmware
cp rt73sta.dat /etc/Wireless/RT73STA/
```

Para comprobar que la instalación ha ido bien y que se ha cargado el módulo, se puede utilizar el comando *dmesg*. Si no hay ningún problema, la salida por pantalla debería ser similar a la siguiente:

```
[2823581.268032] usb 1-7: new high speed USB device using ehci_hcd and address 5
[2823581.586662] usb 1-7: configuration #1 chosen from 1 choice
[2823581.587167] idVendor = 0x13b1, idProduct = 0x20
[2823581.883308] usbcore: registered new interface driver rt73usb
```

O se puede utilizar también, para obtener la ruta donde se encuentra el archivo con extensión .ko: `sudo modprobe -l rt73`

Finalmente, al ejecutar el comando `iwconfig` aparecerá un interfaz llamado *rausb0*, que se puede configurar utilizando los comandos habituales. También es posible editar el fichero binario con extensión .dat, que anteriormente se situó en el directorio `/etc/Wireless/RT73STA/`: `vim -b rt73sta.dat`. Este archivo permitirá comprobar los parámetros configurables de la interfaz y modificar su valor.

Es posible que aún no funcione bien el dispositivo, o que la configuración no sea efectiva. Esto puede ser debido a otro módulo que puede encontrarse instalado, según la versión del *kernel*, por ejemplo (También se puede detectar porque al crear la interfaz se llame *wlan1* en lugar de *rausb0*). Para solucionarlo, editar el archivo `/etc/modprobe.d/blacklist` añadiendo al final `blacklist rt73usb`:

`sudo nano /etc/modprobe.d/blacklist`

Añadir: `blacklist rt73usb`

Y ya está todo listo para configurar el dispositivo.

***Debian*** En primer lugar crear la estructura de directorios y poner cada archivo en su lugar:

```
cd /RT73_Linux_STA_Drv1.0.4.0/Module
cp Makefile.6 ./Makefile
make all
cd /etc
mkdir Wireless
mkdir RT73STA
cd /RT73_Linux_STA_Drv1.0.4.0/Module
cp rt73.bin /etc/Wireless/RT73STA/
```

El archivo `rt73sta.dat` permite cambiar la configuración del dispositivo, incluso con algunas opciones que no están soportadas por línea de comandos. Primero, hay que convertir el archivo a formato UNIX (para usar el comando `dos2unix` es necesario haber instalado previamente `torfodos`):

```
dos2unix rt73sta.dat
cp rt73sta.dat /etc/Wireless/RT73STA/rt73sta.dat
```

A continuación se carga el módulo: `insmod rt73.ko` y ya se puede levantar la interfaz: `ifconfig rausb0 up`

#### D.2.1.2. Instalación necesaria en router con OpenWrt

Para la utilización de este adaptador como interfaz inalámbrica sólo es necesario instalar el paquete `kmod-rt73-usb` ya que el controlador de esta interfaz inalámbrica es el rt73 de *Realtek*. De hecho, si antes de instalarlo se conecta al puerto USB, el router lo reconoce, pero la configuración no se puede realizar. Tras la instalación de este módulo, se puede comprobar que se ha cargado utilizando el comando *lsmod*, por ejemplo.

### D.2.2. Pheenet WLU-803G

Como segundo adaptador USB se ha elegido un modelo de *PheeNet Technology* (Figura D.3), proveedor de soluciones de red.

Figura D.3: Logo de la empresa Pheenet Technology Corp.

PheeNet WLU-803G es un adaptador USB de 54Mbps de 802.11b/g que se muestra en la Figura D.4.

Figura D.4: Adaptador USB Linksys utilizado como interfaz inalámbrica adicional

En la Tabla D.2 se muestran las especificaciones de la antena Pheenet WLU-803G.

#### D.2.2.1. Instalación necesaria en PC

El controlador de este adaptador USB es de la familia Zydas. Este *driver* ya se encuentra cargado en las distribuciones de Linux utilizadas. Si se quiere comprobar la ubicación de este módulo en el equipo, se puede utilizar el comando *modprobe* para localizar el paquete `zd1211rw.ko`:

```
sudo modprobe -l zd1211rw
```

| | |
|---|---|
| **Rango de temperatura funcionamiento** | 0-55 C |
| **Dimensiones** | 127 x 140 x 35 mm |
| **Banda de frecuencia** | 2.400-2.497 GHz |
| **Modulación** | OFDM (IEEE 802.11g); DSSS (IEEE 802.11b) |
| **Antena** | Omnidireccional 5 dBi (*Reserve SMA Connector*) |
| **Rango de sensibilidad** | -88 dBm (1 Mbps); -62 dBm (54 Mbps) |
| **Rango de potencia de salida** | 802.11b: 18dBm; 802.11g: 16dBm |
| **Compatibilidad** | Windows 98SE/ME/200/XP, WinXP64 CPU 64(bit)Frame, Linux 2.4x, 2.6x Kernel (Optional), MAC (Optional) |
| **Estándar** | IEEE 802.11b/g |
| **USB** | USB2.0, USB1.1 compliant |
| **Seguridad** | 64 / 128 / 256-bit WEP; TKIP, WPA, 802.11i |

Tabla D.2: Especificaciones técnicas de la antena Pheenet WLU-803G

#### D.2.2.2. Instalación necesaria en router con OpenWrt

Para poder utilizar este adaptador como interfaz inalámbrica adicional no basta sólo con instalar el paquete `kmod-net-zd1211rw`, que es el nombre que recibe el módulo implementado para *OpenWrt*. Además hay que proporcionarle el firmware del dispositivo. Para ello, tras descargarse el archivo comprimido que contiene el firmware y transferirlo al router, se descomprime en el directorio `/lib/firmware` y se renombra la carpeta como zd1211.

Ahora sí, el adaptador inalámbrico está listo para ser utilizado correctamente. Se puede comprobar que los módulos necesarios se cargan en el kernel utilizando *lsmod*.

## D.3. Configuración para la utilización del *software* desarrollado

### D.3.1. Compilación de *OpenSSL* para OpenWrt para la realización de operaciones criptográficas

En primer lugar, para que la implementación realizada se aproxime más a la propuesta en [Ber06] es necesario calcular los tiempos que emplea el router móvil en realizar las operaciones criptográficas involucradas en cada mensaje, para introducir esos retardos en el *software*. Para ello, se ha modificado el código fuente de *OpenSSL* del paquete instalado en el router para que tome referencias temporales antes y después de realizar cada operación, con el fin de poder calcular la diferencia entre ellas, que será el tiempo necesario para que el router realice cada operación.

Para poder instalar en el router el paquete con el código fuente modificado, se utiliza una herramienta proporcionada por OpenWrt llamada SDK (*Software Development Kit*).

Este entorno de desarrollo es de gran utilidad ya que permite compilar (o mejor dicho, cross-compilar) paquetes para una determinada arquitectura sin tener que compilar toda la imagen del sistema. Además, es posible compilar nuevas versiones de un paquete o personalizar paquetes existentes realizando alguna modificación, como en este caso.

Después de descargarse la versión adecuada de *OpenSSL*, para OpenWrt la versión más reciente soportada es la 0.9.8o, se debe descomprimir el archivo en la carpeta `/build/kamikaze_8.09/build_dir/mipsel/OpenWrt-SDK/package` asegurándose de que existen una carpeta llamada *patches* y un *makefile*, dentro de la nueva carpeta openssl-0.9.8o.

Si el paquete que se pretende compilar tiene alguna dependencia adicional, será necesario también descargarla y crear una carpeta para ella en el directorio `package` del SDK. En este caso, para compilar *openssl*, es necesario compilar también *zlib*. Una vez que se tiene el código fuente modificado, se puede compilar. La secuencia de comandos para realizar todo el proceso sería la siguiente:

```
$ svn export svn://svn.openwrt.org/openwrt/trunk/package/openssl package/openssl
$ grep DEPENDS package/openssl/Makefile
DEPENDS:=+zlib
$ svn export svn://svn.openwrt.org/openwrt/trunk/package/zlib package/zlib
$ make package/openssl/prepare
$ make package/openssl/compile
```

Si la compilación se realiza con éxito, se creará un archivo .ipk en el directorio `/bin/packages/mipsel/` del SDK. Ese archivo ya se puede enviar al router e instalarlo haciendo uso del comando `opkg`.

### D.3.2. Utilización del *software* desarrollado

Para poder utilizar el *software* desarrollado en los routers móviles, es necesario transferir los archivos ejecutables generados en la compilación cruzada, desde el ordenador al router. Una vez que tenemos todos los archivos disponibles, para que la ejecución funcione deberá crearse un directorio `config` en el que se incluirán unos ficheros de texto a los que se accede desde el programa:

- *my_ home_ address.txt*, que contiene la información de la HoA del propio MR y la lista de prefijos de redes móviles, con los que debe iniciar un proceso para establecer la ruta ad-hoc en caso de recibir un anuncio de uno de esos prefijos.
- *ruta_ inversa.txt*, del que se leerá la información para enviar los mensajes del proceso de señalización a partir de la tabla de rutas.

# Glosario

La lista de los acrónimos utilizados en este proyecto es la siguiente:

**AR** *Access Router*

**AP** *Access Point*

**BA** *Binding ACK*

**BR** *Binding Refresh*

**BU** *Binding Update*

**CoA** *Care-of-Address*

**CoRE** *Care-of Route Error*

**CoRT** *Care-of Route Test*

**CoRTI** *Care-of Route Test Init*

**CN** *Correspondent Node*

**CGA** *Cryptographically Generated Addresses*

**DNS** *Domain Name Server*

**DHCP** *Dynamic Host Configuration Protocol*

**DSRC** Dedicated Short Range Communications

**FA** Foreign Agent

**GPS** Global Positioning System

**HA** *Home Agent*

**HL** *Home Link*

**HMIPv6** *Hierarchical MIPv6*

**HoA** *Home Address*

**HoAA** *Home Address Advertisement*

**HoRT** *Home Route Test*

**HTTP** *Hypertext Transfer Protocol*

**IETF** Internet Engineering Task Force

**IP** *Internet Protocol*

**IPsec** *Internet Protocol Secure*

**MANET** Mobile Ad-hoc Network

**MIP** *Mobile IP*

**MN** *Mobile Node*

**MNP** *Mobile Network Prefix*

**MNPBU** *Mobile Network Prefix Binding Update*

**MR** *Mobile Router*

**MRHA** *Mobile Router-Home Agent* bidirectional tunnel

**NEMO BS** *Network Mobility Basic Support*

**P2P** Peer to peer

**PAN** Personal Area Network

**RA** *Router Advertisement*

**RSA** *Rivest, Shamir, Adleman*

**RTSP** *Real Time Streaming Protocol*

**RTT** *Round Trip Time*

**SEND** *Secure Neighbor Discovery Protocol*

**SSH** *Secure SHell*

**SSL** *Secure Socket Layer*

**TCP** *Transmission Control Protocol*

**TELNET** *TELetype NETwork*

**TFTP** *Trivial File Transfer Protocol*

**UDP** *User Datagram Protocol*

**UMTS** Universal Mobile Transmission System

**USB** *Universal Serial Bus*

**VANET** Vehicular Ad-hoc Network

**VARON** Vehicular Ad-hoc Route Optimization for NEMO

**VLAN** *Virtual Local Area Network*

**VPN** Virtual Private Network

**VSN** Vehicular Sensor Network

**WAVE** *Wireless Access in Vehicular Environments*

**WLAN** *Wireless Local Area Network*

# Bibliografía


[AKZN05] J. Arkko, J. Kempf, B. Zill, and P. Nikander, *SEcure Neighbor Discovery (SEND)*, RFC 3971 (Proposed Standard), March 2005.

[Aur05] T. Aura, *Cryptographically Generated Addresses (CGA)*, RFC 3972 (Proposed Standard), March 2005, Updated by RFCs 4581, 4982.

[AVH07] J. Arkko, C. Vogt, and W. Haddad, *Enhanced Route Optimization for Mobile IPv6*, RFC 4866 (Proposed Standard), May 2007.

[Ber06] Carlos J. Bernardos, *Route optimisation for mobile networks in ipv6 heterogeneous environments (optimización de rutas para redes móviles en entornos ipv6 heterogéneos)*, Ph.D. thesis, Universidad Carlos III de Madrid, November 2006.

[BFA07] Roberto Baldessari, Andreas Festag, and Julien Abeille, *NEMO meets VANET: A deployability analysis of network mobility in vehicular communication*, 7th International Conference on ITS Telecommunications (ITST 2007) (ITST), ITST (Intelligent Tansports Systems Telecommunications), 2007, pp. 375–380.

[BFL07] R. Baldessari, A. Festag, and M. Lenardi, *C2C-C Consortium Requirements for NEMO Route Optimization*, (Informational), July 2007.

[BOC+06] Carlos J. Bernardos, Antonio De La Oliva, María Calderón, Dirk von Hugo, and Holger Kahle, *NEMO: Network Mobility. Bringing ubiquity to the Internet access*, IEEE INFOCOM 2006, April 2006, demonstration.

[BSC+05a] Carlos J. Bernardos, Ignacio Soto, María Calderón, Dirk von Hugo, and Emmanuel Riou, *NEMO: Movilidad de Redes en IPv6*, Novatica (2005), no. 174, 37–43.

[BSC+05b] Carlos J. Bernardos, Ignacio Soto, María Calderón, Dirk von Hugo, and Emmanuel Riou, *NEMO: Network Mobility in IPv6*, UPGRADE - The European Journal for the Informatics Professional **VI** (2005), no. 2, 36–42.

[BSC+07] Carlos J. Bernardos, Ignacio Soto, María Calderón, Fernando Boavida, and Arturo Azcorra, *VARON: Vehicular ad hoc route optimisation for NEMO*, Comput. Commun. **30** (2007), 1765–1784.

[BZFL08] Roberto Baldessari, Wenhui Zhang, Andreas Festag, and Long Le, *A MANET-centric solution for the application of nemo in vanet using geographic routing*, TridentCom '08: Proceedings of the 4th International Conference on Testbeds and research infrastructures for the development of networks & communities (ICST, Brussels, Belgium), ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 1–7.

[car] *Car 2 Car Communication Consortium.*

[CBB+06] María Calderón, Carlos J. Bernardos, Marcelo Bagnulo, Ignacio Soto, and Antonio de la Oliva, *Design and Experimental Evaluation of a Route Optimization Solution for NEMO*, IEEE Journal on Selected Areas in Communications **24** (2006), no. 9, 1702–1716.

[CDG06] A. Conta, S. Deering, and M. Gupta, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*, RFC 4443 (Draft Standard), March 2006, Updated by RFC 4884.

[CN06] S. Chakrabarti and E. Nordmark, *Extension to Sockets API for Mobile IPv6*, RFC 4584 (Informational), July 2006.

[DH98] S. Deering and R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, RFC 2460 (Draft Standard), December 1998, Updated by RFC 5095.

[dlOBC05] Antonio de la Oliva, Carlos J. Bernardos, and María Calderón, *Practical evaluation of a network mobility solution*, Proceedings of the 11th Open European Summer School (EUNICE 2005: Networked Applications) (Colmenarejo, Madrid (SPAIN)), July 2005, pp. 60 – 66.

[DNII05] Marios D. Dikaiakos, Tamer Nadeem, Saif Iqbal, and Liviu Iftode, *VITP: an information transfer protocol for vehicular computing*, In Workshop on Vehicular Ad Hoc Networks, 2005, pp. 30–39.

[DWPT05] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, *Network Mobility (NEMO) Basic Support Protocol*, RFC 3963 (Proposed Standard), January 2005.

[DWX07] Ding, Yong, Wang, Chen, and Xiao, Li, *A static-node assisted adaptive routing protocol in vehicular networks*, VANET '07: Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks (New York, NY, USA), ACM, 2007, pp. 59–68.

[EGH+08] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan, *The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring*, The Sixth Annual International conference on Mobile Systems, Applications and Services (MobiSys 2008) (Breckenridge, U.S.A.), June 2008.

[Ern07] T. Ernst, *Network Mobility Support Goals and Requirements*, RFC 4886 (Informational), July 2007.

[GAZ05] Meng Guo, M.H. Ammar, and E.W. Zegura, *V3: a vehicle-to-vehicle live video streaming architecture*, Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on, 2005, pp. 171 – 180.

[GE10] Antonio Guerrero Espartero, *Estudio Experimental del funcionamiento de OSPF-MANET y OLSR en redes malladas multisalto inalámbricas*, Proyecto fin de carrera, Universidad Carlos III de Madrid . Escuela Politécnica Superior, Leganés, 2010.

[GTB+03] R. Gilligan, S. Thomson, J. Bound, J. McCann, and W. Stevens, *Basic Socket Interface Extensions for IPv6*, RFC 3493 (Informational), February 2003.

[Han04] Jun-ichiro itojun Hangino, *IPv6 network programming*, Digital Press, Newton, MA, USA, 2004.

[HBZ+06] Bret Hull, Vladimir Bychkovsky, Yang Zhang, Kevin Chen, Michel Goraczko, Allen K. Miu, Eugene Shih, Hari Balakrishnan, and Samuel Madden, *CarTel: A Distributed Mobile Sensor Computing System*, 4th ACM SenSys (Boulder, CO), November 2006.

[HKHL10] Kai-Yun Ho, Po-Chun Kang, Chung-Hsien Hsu, and Ching-Hai Lin, *Implementation of WAVE/DSRC devices for vehicular communications*, Computer Communication Control and Automation (3CA), 2010 International Symposium on, vol. 2, May 2010, pp. 522 –525.

[iee04] *IEEE 802.16-2004 Standard for Local and metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, 2004.

[iee05] *IEEE 802.16e-2005 Standard for Local and metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems. Amendment 2: Physical Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands*, 2005.

[iee07] *IEEE 802.11 Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements*, 2007.

[iee09] *IEEE 1609 - Family of Standards for Wireless Access in Vehicular Environments (WAVE)*, 2009.

[iee10] *802.11p-2010 IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments*, 2010.

[JOW+02] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li shiuan Peh, and Daniel Rubenstein, *Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet*, 2002.

[JPA04] D. Johnson, C. Perkins, and J. Arkko, *Mobility Support in IPv6*, RFC 3775 (Proposed Standard), June 2004.

[LCGZ05] Lebrun, J., Chuah, Chen-Nee, Ghosal, D., and Zhang, Michael, *Knowledge-Based Opportunistic Forwarding in Vehicular Wireless Ad Hoc Networks*, vol. 4, 2005, pp. 2289–2293.

[LHT+03] C. Lochert, H. Hartenstein, J. Tian, H. Fussler, and M. Hermann, D.and Mauve, *A routing strategy for vehicular ad hoc networks in city environments*, Proc. IEEE Intelligent Vehicles Symposium, Lecture Notes in Computer Science, 2003, pp. 156–161.

[LM07] Leontiadis, Ilias and Mascolo, Cecilia, *GeOpps: Geographical Opportunistic Routing for Vehicular Networks*, World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a, 2007, pp. 1–6.

[LMFH05] Christian Lochert, Martin Mauve, Holger Füßler, and Hannes Hartenstein, *Geographic routing in city scenarios*, Mobile Computing and Communications Review **9** (2005), no. 1, 69–72.

[LPAG06] Uichin Lee, Joon-Sang Park, E. Amir, and M. Gerla, *Fleanet: A virtual market place on vehicular networks*, Mobile and Ubiquitous Systems, Annual International Conference on **0** (2006), 1–8.

[LZG+06] Uichin Lee, Biao Zhou, M. Gerla, E. Magistretti, P. Bellavista, and A. Corradi, *Mobeyes: smart mobs for urban monitoring with a vehicular sensor network*, Wireless Communications, IEEE **13** (2006), no. 5, 52 –57.

[mob04] *Mobile IPv6 Overview*, Tech. report, Cisco Systems, Diciembre 2004.

[NBG06] Naumov, Valery, Baumann, Rainer, and Gross, Thomas, *An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces*, MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing (New York, NY, USA), ACM, 2006, pp. 108–119.

[NCL07] E. Nordmark, S. Chakrabarti, and J. Laganier, *IPv6 Socket API for Source Address Selection*, RFC 5014 (Informational), September 2007.

[nem10] *Network Mobility (NEMO)*, Septiembre 2010.

[NG07] Valery Naumov and Thomas R. Gross, *Connectivity-aware routing (CAR) in vehicular ad-hoc networks*, INFOCOM, 2007, pp. 1919–1927.

[NTWZ07] C. Ng, P. Thubert, M. Watari, and F. Zhao, *Network Mobility Route Optimization Problem Statement*, RFC 4888 (Informational), July 2007.

[NZWT07] C. Ng, F. Zhao, M. Watari, and P. Thubert, *Network Mobility Route Optimization Solution Space Analysis*, RFC 4889 (Informational), July 2007.

[ope] *OpenWrt Wireless Freedom.*

[OW09] Stephan Olariu and Michele C. Weigle (eds.), *Vehicular networks: From theory to practice*, CRC Press Taylor & Francis, mar 2009.

[SBBC09] Ignacio Soto, Roberto Baldessari, Carlos J. Bernardos, and Maria Calderon, *Vehicular Networks: Techniques, Standards and Applications Book*, ch. Network Mobility (NEMO) in Vehicular Networks, CRC Press, 2009.

[SBC+09] Ignacio Soto, Carlos J. Bernardos, María Calderón, Albert Banchs, and Arturo Azcorra, *NEMO-Enabled Localized Mobility Support for Internet Access in Automotive Scenarios*, IEEE Communications Magazine, Special Issue on Automotive Networking - Technology, Design, and Applications series **47** (2009), no. 5, 152–159.

[SG10] José Pablo Salvador García, *Implementación de mecanismos avanzados de movilidad de redes sobre routers Fonera*, Proyecto fin de carrera, Universidad Carlos III de Madrid . Escuela Politécnica Superior, Leganés, 2010.

[SHSI08] Stephen Smaldone, Lu Han, Pravin Shankar, and Liviu Iftode, *Roadspeak: enabling voice chat on roadways using vehicular social networks*, Proceedings of the 1st Workshop on Social Network Systems (New York, NY, USA), SocialNets '08, ACM, 2008, pp. 43–48.

[SK08] Ignacio Soto and José Félix Kukielka, *Wireless Local Area Networks: WLANs*, Slides, Universidad Carlos III de Madrid, 2008.

[SLL+04] Seet, Boon-Chong, Liu, Genping, Lee, Bu-Sung, Foh, Chuan-Heng, Wong, Kai-Juan, and Lee, Keok-Kee, *A-STAR: A Mobile Ad Hoc Routing Strategy for Metropolis Vehicular Communications*, 2004.

[ST98] W. Stevens and M. Thomas, *Advanced Sockets API for IPv6*, RFC 2292 (Informational), February 1998, Obsoleted by RFC 3542.

[Ste90] W. Richard Stevens, *Unix network programming*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1990.

[STNJ03] W. Stevens, M. Thomas, E. Nordmark, and T. Jinmei, *Advanced Sockets Application Program Interface (API) for IPv6*, RFC 3542 (Informational), May 2003.

[Tan02] Andrew Tanenbaum, *Computer networks*, Prentice Hall Professional Technical Reference, 2002.

[THR03] Tian, Jing, Han, Lu, and Rothermel, K., *Spatially aware packet routing for mobile ad hoc inter-vehicle radio networks*, vol. 2, 2003, pp. 1546–1551 vol.2.

[veh] *National Highway Traffic Safety Administration.*

[WED07] W. Ivancic W. Eddy and T. Davis, *NEMO Route Optimization Requirements for Operational Use in Aeronautics and Space Exploration Mobile Networks*, (Informational), December 2007.

[WFGH04] Wu, Hao, Fujimoto, Richard, Guensler, Randall, and Hunter, Michael, *MDDV: A mobility-centric data dissemination algorithm for vehicular networks*, VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks (New York, NY, USA), ACM, 2004, pp. 47–56.

[Wor92] Working Group 16, *CALM - Communications Access for Land Mobiles*, Tech. report, ISO TC 204, 1992.

[ZC08] Zhao, Jing and Cao, Guohong, *VADD: Vehicle-assisted data delivery in vehicular ad hoc networks*, IEEE Transactions on Vehicular Technology **57** (2008), no. 3, 1910–1922.

[Zul] Holger Zuleger, *Mobile Internet Protocol v6. MIPv6. A short introduction*, Tech. report.