UNIVERSIDAD DE MURCIA

DEPARTAMENTO DE INGENIERÍA DE LA INFORMACIÓN Y LAS COMUNICACIONES

Propuesta de Diseño y Despliegue de Servicios de Seguridad y Movilidad en Redes Vehiculares

DIRECTORES

D. José Santa Lozano - **D. Antonio F. Skarmeta Gómez**

DOCTORANDO

D. Pedro Javier Fernández Ruiz

2015

Agradecimientos

Son muchos los años compartidos con personas estupendas con las que he pasado muy buenos momentos y de las que he podido aprender muchísimo. Mi primer agradecimiento va dirigido a Antonio Skarmeta, que desde un principio confió en mí, dándome la oportunidad de trabajar de investigador en la misma Universidad en la que realicé mis estudios, que para mí ha sido como una prolongación de los mismos. He tenido oportunidad de viajar, de conocer gente de otros países y trabajar en equipo con distintos compañeros que han ido acompañándome a lo largo de estos años. Nunca podré olvidar los momentos vividos con cada uno de vosotros, Juan Antonio, Alejandro, Cristian, Jordi y Fernando Bernal, gracias a todos vosotros por hacer de mi trabajo algo tan agradable. Muchísimas gracias a José Santa que, aparte de compañero, ha sido un gran apoyo en estos últimos años y sobre todo un guía que ha sabido dirigir bien mis pasos. Gracias también al resto de gente del departamento, pues siempre habéis estado ahí para echar una mano. También agradecer a otros profesores como Diego Sevilla, Rafael Marín López, Pedro M. Ruiz y Gregorio Martínez, pues también he compartido con ellos momentos de colaboración que me han enriquecido.

Agradecer también a la gente que ha estado cerca de mí en estos años y sobre todo en este último donde el esfuerzo ha sido intenso. A mi familia, siempre fiel y apoyándome en todo momento. A mis amigos por hacerme pasar muy buenos ratos, sobre todo a Antonio, Dorna y especialmente a Sonia por su gran ayuda en los últimos momentos de este trabajo.

También tengo que agradecerle mucho al baile, que ha hecho de mí una persona más completa y que me ha servido como fuente de energía a lo largo de todos estos años.

Me he reservado el último agradecimiento a mis padres y a mi hermano, que lo son todo en mi vida. Les agradezco toda su paciencia y sobre todo la confianza que han tenido en mí en todo momento. ¡Os quiero!

Índice general

Índice de Figuras IX

Índice de Tablas XII

Acrónimos XV

1. Introducción **5**
1.1. Sistemas Inteligentes de Transporte (ITS) 6
1.2. Sistemas Inteligentes de Transporte Cooperativos (C-ITS) 7
1.3. Redes Vehiculares 9
1.3.1. Elementos de una Red Vehicular 9
1.3.2. Estandarización de las Redes Vehiculares 10
1.3.3. Proyectos relacionados 12
1.4. Objetivos de la Tesis Doctoral y Aportaciones 14
1.4.1. Objetivos 14
1.4.1.1. Incorporar IPv6 a Entornos ITS 14
1.4.1.2. Explotar la Sinergia entre los Estándares ISO/ETSI para Redes Vehiculares y los Provenientes de IETF e Internet 15
1.4.1.3. Dotar de Esquemas de Seguridad a las Redes Vehiculares 15
1.4.1.4. Conseguir una Experiencia sin Cortes en Redes Vehiculares 16
1.4.1.5. Incorporar un Servicio de Movilidad Seguro en C-ITS . 16
1.4.1.6. Aportar una Metodología de Pruebas en Entornos Reales 16
1.4.2. Aportaciones 17
1.4.2.1. Propuesta de Arquitectura y Pila de Red IPv6 para Entornos C-ITS 17
1.4.2.2. Implementación de IKEv2: OpenIKEv2 17
1.4.2.3. Integración de los Servicios de Movilidad y Seguridad de IPv6 para Redes Vehiculares 18
1.4.2.4. Mejora en los Procesos de Traspaso mediante la Integración de Tecnologias IETF e IEEE 18
1.4.2.5. Desarrollo de un Entorno de Pruebas Reales para Redes Móviles Vehiculares 19

1.5. Trabajos relacionados 19
1.6. Organización del Documento 23
1.7. Publicaciones Derivadas de la Tesis Doctoral 24
1.7.1. Artículos en Revistas 24
1.7.2. Capítulos en Libros 27
1.7.3. Congresos 28

2. Estándares y Tecnologías basadas en IPv6 para entornos ITS 31
2.1. Arquitectura de Referencia ISO/ETSI 31
2.1.1. Tipos de estaciones ITS 32
2.1.2. Capas de la Pila de la Arquitectura de Referencia 33
2.1.2.1. Capa de Acceso 33
2.1.2.2. Capa de Red y Transporte 33
2.1.2.3. Capa de Facilidades 33
2.1.2.4. Capa de Aplicaciones 34
2.1.2.5. Planos de Gestión y Seguridad 34
2.2. Tecnologías Inalámbricas Apropiadas para Entornos Vehiculares 34
2.2.1. Tecnologías 3GPP 35
2.2.2. IEEE 802.11p y ETSI G5 36
2.2.3. Asistencia al Traspaso mediante IEEE 802.21 37
2.3. Tecnologías IPv6 de Interés para C-ITS 38
2.3.1. Asignación Automática de Direcciones 39
2.3.2. Servicio de Movilidad Basado en *Mobile IP* (MIPv6) 41
2.3.2.1. Principios de Funcionamiento de MIPv6 41
2.3.2.2. Movilidad de Redes con *Network Mobility* (NEMO) . . 42
2.3.2.3. Uso Simultáneo de Múltiples Interfaces de Red mediante *Multiple Care-of Address* (MCoA) 43
2.3.3. Servicio de Seguridad Basado en IPsec 44
2.3.3.1. Motivación del uso de IPsec 45
2.3.3.2. Protocolos 46
2.3.3.3. Modos de funcionamiento y escenarios típicos 46
2.3.3.4. Asociaciones y Políticas de Seguridad 47
2.3.3.5. *Internet Keying Exchange* versión 2 (IKEv2) 50
2.3.3.5.1. Detalles del Protocolo 51
2.3.3.5.2. Nuevas Características Incorporadas 51
2.3.3.6. Necesidad de IPsec e IKEv2 en Redes Vehiculares . . . 53
2.4. Conclusiones 53

3. Propuesta de Arquitectura de Red 55
3.1. Nuestra Propuesta de Arquitectura de Comunicaciones 56
3.1.1. Estación Vehicular ITS 58
3.1.2. Estación de Carretera ITS 59
3.1.3. Estación Central ITS 59
3.2. Distribución de Protocolos IPv6 sobre la Arquitectura 60

3.2.1. Servicio de Autoconfiguración 60
3.2.2. Servicio de Movilidad 61
3.2.3. Servicio de Seguridad 64
3.2.4. Asistencia al Traspaso 67
3.2.5. Elementos de Transición de IPv4 a IPv6 68
3.3. Conclusiones 69

4. IKEv2 y Nuestra Implementación OpenIKEv2 71
4.1. Implementaciones de Código Abierto: OpenIKEv2 71
4.1.1. Comparativa de Implementaciones de IKEv2 de Código Abierto 72
4.1.2. Escenario de Pruebas para las Diferentes Implementaciones de IKEv2 74
4.2. Incidencias Encontradas en la Implementación de OpenIKEv2 79
4.2.1. Cookie inválida 80
4.2.2. Refresco de Material Criptográfico en una IPsec SA Doble . . . 80
4.2.3. SPI Inválido en el Campo DELETE 81
4.2.4. Colisión de Intercambios 82
4.3. Conclusiones 83

5. Integración de los servicios de Movilidad y Seguridad 85
5.1. Aproximaciones al Problema 85
5.1.1. Caso A: Movilidad Encapsulada en la Seguridad 86
5.1.2. Caso B: Seguridad Encapsulada en la Movilidad 89
5.1.3. Estudio y Comparativa de los Acercamientos Planteados 95
5.1.3.1. Grado de Integración de las Implementaciones 96
5.1.3.2. Grado de Protección del Tráfico 96
5.1.3.3. Comportamiento en los Traspasos 96
5.1.3.4. Control de Acceso: Autenticación y Autorización . . . 97
5.1.3.5. Grado de Estandarización 97
5.1.4. Conclusiones y Selección de una de las Soluciones 98
5.2. Incorporación de la Solución Seleccionada a Nuestra Arquitectura . . . 98
5.2.1. Selección de Flujos de Tráfico de Datos 99
5.2.2. Ajustes de Diseño en nuestra Propuesta de Solución 100
5.2.2.1. CoA vs HoA 101
5.2.2.2. Imposible Identificar al MR a través de la CoA 101
5.2.2.3. Identificación de las IKE SAs mediante la HoA 102
5.3. Conclusiones 103

6. Mejoras en el Traspaso 105
6.1. El Traspaso (*Handover*) 105
6.1.1. Tipos de Traspasos 105
6.1.2. Fases en el Traspaso 106
6.1.3. Traspasos sin Cortes 107
6.2. Optimización de la Fase de Autenticación y Autorización 108

6.2.1. Método de Autenticación mediante Firma Digital 109
6.2.2. Métodos de Autenticación mediante EAP 109
6.3. Uso Simultáneo de más de una Interfaz: MCoA 112
6.4. Asistencia al Traspaso mediante IEEE 802.21 114
6.4.1. Implementación de IEEE 802.21 114
6.4.2. Resultados de la Incorporación de IEEE 802.21 a Nuestra Propuesta 116
6.5. Conclusiones 117

7. Pruebas y Análisis de Resultados 119
7.1. Configuración del Escenario de Pruebas 119
7.1.1. Despliegue de la Arquitectura de Red 120
7.1.2. Equipamiento de la Red Vehicular 120
7.2. Plan de Pruebas 121
7.2.1. Metodología 122
7.2.2. Métricas 124
7.3. Resultados 126
7.3.1. Estudio del Retardo en el Envío de Paquetes 126
7.3.2. Estudio del Ancho de Banda Máximo 127
7.3.3. Estudio de la Fiabilidad del Envío de Paquetes 129
7.3.4. Estudio del Impacto de la Aplicación de Seguridad 133
7.3.5. Estudio del Comportamiento de la Red en los Traspasos 133
7.3.5.1. Traspaso de 3G a 802.11p 134
7.3.5.2. Traspaso de 802.11p a 3G 135
7.3.6. Estudio de la Aplicación de 802.21 al Proceso de Traspaso . . . 136
7.3.6.1. Traspaso de 3G a 802.11p 137
7.3.6.2. Traspaso de 802.11p a 3G 139
7.3.7. Análisis Final de los Resultados 141
7.4. Conclusiones 142

8. Conclusiones y Trabajos Futuros 143
8.1. Conclusiones Generales 143
8.2. Trabajos Futuros 146
8.2.1. Evolución de los Traspasos 146
8.2.2. Mejoras en la Integración de Servicios de Movilidad y Seguridad 147
8.2.3. Explorar las Comunicaciones V2V 147
8.2.4. Interfaces Físicas Vs. Interfaz Virtual 147

Bibliografía 149

A. Detalles del protocolo IKEv2 159
A.1. Transporte de IKEv2 159
A.2. Retransmisión de Mensajes 159
A.3. Identificadores de Mensaje 160

A.4. Ventana Deslizante de Peticiones . . . 160
A.5. Sincronización de Estado entre los Extremos . . . 160
A.6. Ataque de DoS en el Iniciador de una IKE_SA . . . 160
A.7. Número de Versión y Compatibilidad con Futuras Versiones . . . 161
A.8. Cookies . . . 161
A.9. Negociación de los Algoritmos Criptográficos . . . 162
A.10. Generación de Material Criptográfico . . . 162
A.10.1. IKE_SA . . . 163
A.10.2. CHILD_SA . . . 164
A.11. Regeneración de Claves . . . 164
A.12. Selectores de Tráfico . . . 165
A.13. Mecanismo de Asignación de Direcciones . . . 165
A.14. Autenticación de la IKE_SA . . . 165
A.15. Manejo de Errores . . . 165
A.16. NAT Trasversal . . . 166

B. Diseño e Implementación de OpenIKEv2 167
B.1. Diseño de OpenIKEv2 . . . 167
B.1.1. Componentes de la Librería *libopenikev2* . . . 168
B.1.1.1. Subsistema de Hilos . . . 168
B.1.1.2. Subsistema de Red . . . 169
B.1.1.3. Subsistema Criptográfico . . . 170
B.1.1.4. Subsistema de Interfaz con IPsec . . . 171
B.1.1.5. Subsistema de Registro de Eventos . . . 171
B.1.1.6. Sistema de Configuración . . . 172
B.1.1.7. Sistema de Bus de Eventos . . . 173
B.2. Implementación de OpenIKEv2 . . . 173
B.2.1. Herramientas de Trabajo . . . 174

C. Herramientas para la Toma y Procesamiento de Datos 175
C.1. Entorno para las Herramientas de Pruebas . . . 175
C.1.1. Generación y Captura de Datos . . . 175
C.1.1.1. Subdirectorio MR . . . 176
C.1.1.2. Subdirectorio AR . . . 179
C.1.1.3. Subdirectorio HOST . . . 180
C.1.2. Procesamiento de los Datos Obtenidos . . . 181
C.1.2.1. Subdirectorio DATA . . . 181
C.1.2.2. Subdirectorio SCRIPTS . . . 183

Índice de figuras

1.1. Comparativa de arquitecturas más influyentes en redes vehiculares [1] . 12

2.1. Visión simplificada de la pila de comunicaciones de referencia del ISO/ETSI 32
2.2. Dispositivo Laguna de Commsignia compatibles con ITS-G5 37
2.3. Esquema de entidades IEEE 802.21 y sus mensajes 38
2.4. Paso de una dirección MAC EUI-48 a EUI-64 40
2.5. Escenario típico de uso de NEMO 42
2.6. Aplicación de las cabeceras AH y ESP 48
2.7. Escenarios típicos de uso de IPsec 49
2.8. Funcionamiento de la arquitectura de seguridad IPsec 50
2.9. Intercambios IKEv2 y sus respectivas asociaciones 52

3.1. Arquitectura de comunicaciones basada en tecnologías IPv6 57
3.2. Presencia de ICMPv6 en nuestra propuesta de arquitectura 62
3.3. Presencia de NEMO y MCoA en nuestra propuesta de arquitectura . . 63
3.4. Presencia de NEMO y MCoA en nuestra propuesta de arquitectura . . 66
3.5. Incorporación de los servicios MIH al plano de gestión de la arquitectura de referencia OSI/ETSI 69
3.6. Elementos que permiten el uso de IPv4 en nuestra arquitectura basada en IPv6 70

4.1. Diferentes pasos de la negociación IKEv2 76
4.2. Tiempos de negociaión IKEv2 en milisegundos agrupados por tipo de interfaz IPsec. 79
4.3. Ejemplo de una colisión de intercambios 83

5.1. Diferentes enfoques en aplicación simultánea de servicios de movilidad y seguridad 87
5.2. Diagrama de secuencia del establecimiento de la seguridad y movilidad en el caso A 88
5.3. Túneles IPsec establecidos entre MR y HA para proporcionar seguridad al tráfico de movilidad 90
5.4. Diagrama de secuencia del establecimiento de la movilidad y seguridad en el caso B 91

5.5. Distintas estrategias de cooperación entre los servicios de movilidad y seguridad ante el cambio de direccionamiento IP 94
5.6. Diagrama de secuencia de las negociaciones necesarias para crear las IPsec SAs que protegen el tráfico de datos 95
5.7. Selección de interfaz basado en flujos de datos 99
5.8. Diagrama de secuencia de la creación de una CoA adicional 102

6.1. Intercambios en la autenticación basada en firma digital en IKEv2 . . . 110
6.2. Intercambios en la autenticación IKEv2 con EAP 111
6.3. Comparativa de tiempos de autenticación con diferentes métodos 112
6.4. PDR en una vuelta con tecnologías 3G y 802.11p usando MCoA 113
6.5. Implementación mínima de IEEE 802.21 115
6.6. Valor de RTT durande la realización de traspasos entre 3G y 802.11p . 116
6.7. PDR en una vuelta con tecnologías 3G y 802.11p usando MCoA asistido con IEEE 802.21 117

7.1. Escenario ITS desplegado en la Universidad de Murcia 121
7.2. Equipamiento usado en el escenario de pruebas 122
7.3. Recorrido de cada prueba y el nivel de señal 802.11p a lo largo de ella . 125
7.4. Evaluación del tráfico ICMPv6 en términos de RTT 127
7.5. Ancho de banda máximo TCP con y sin seguridad a 20 Km/h 128
7.6. PDR (%) a diferentes tasas de transferencia de tráfico UDP a 20 Km/h 131
7.7. Comparación de anchos de banda de tráfico UDP a diferentes tasas de transferencia a 20 Km/h 132
7.8. Valores medios de los traspasos de 3G a 802.11p, con y sin seguridad . 135
7.9. Representación gráfica de los valores medios del tiempo empleado en los traspasos de 3G a 802.11p a 30 Km/h con y sin 802.21 138
7.10. PDR (%) a diferentes tasas de transferencia de tráfico UDP en laboratorio y en circuito con asistencia en el traspaso mediante 802.21 . 140

A.1. Estructura lógica de una propuesta IPsec 162

B.1. Interacción entre *libopenikev2* y *libopenikev2_impl*. 168
B.2. Patrón de diseño “estrategia”. 169
B.3. Subsistema de configuración. 172
B.4. Subsistema de eventos. 173

C.1. Estructura de directorios del entorno de pruebas 176
C.2. Origen de los datos obtenidos en las pruebas 181
C.3. Flujo de trabajo en el procesado de los datos 184
C.4. Representación gráfica del PDR en el plano terrestre 185

Índice de tablas

2.1. Distintas tasas de transferencia teóricas de las tecnologías inalámbricas dependientes del 3GPP 35

4.1. Tabla comparativa de características IKEv2 74
4.2. Características Hardware y Software de los equipos usados en las pruebas 75
4.3. Comparativa de tiempos de negociación medios de diferentes implementaciones IKEv2 con intervalos de confianza al 95 % 77

7.1. Componentes usados en el escenario de pruebas 123
7.2. Parámetros de configuración usados 124
7.3. Valores medios obtenidos en las pruebas con ICMPv6 con intervalo de confianza al 95 % 126
7.4. Valores medios de ancho de banda máximo TCP y sus intervalos de confianza al 95 % 128
7.5. Valores medios del PDR obtenidos en las pruebas con UDP y sus intervalos de confianza al 95 % 130
7.6. Valores medios del tiempo empleado en los traspasos de 3G a 802.11p con un intervalo de confianza al 95 % 134
7.7. Valores medios del tiempo empleado en los traspasos de 3G a 802.11p a 30 Km/h con y sin 802.21, con un valor de confianza al 95 % 137
7.8. Pros y contras de las tecnologías 3G y 802.11p 141

Acrónimos

3G 3rd Generation (Cellular Networks)

3GPP 3rd Generation Partnership Project

AAA Authentication, Authorization and Accounting

AH Authentication Header

API Application Programming Interface

AR Access Router

BA Binding Acknowledgement

BID Binding Identity

BR Border Router

BU Binding Update

C2C Car to Car

CALM Communications Access for Land Mobiles

CAM Cooperative Awareness Message

CGA Cryptographically Generated Address

C-ITS Cooperative Intelligent Transport Systems

CN Correspondent Node

CoA Care of Address

DAD Duplicated Address Detection

DENM Decentralized Environmental Notification Message

DH Diffie-Hellman

DHCP Dynamic Host Configuration Protocol

DoS Denial of Service

DNS Domain Name Server

DNS64 DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers

EAP Extensible Authentication Protocol

ESP Encapsulating Security Payload

ETSI European Telecommunications Standards Institute

EUI Extended Unique Identifier

FNTP Fast Networking and Transport layer Protocol

FRM Fast Re-authentication Method

GPS Global Positioning System

HA Home Agent

HoA Home Address

HMAC Keyed-Hash Message Authentication Code

HTTP Hypertext Transfer Protocol

I2V Infrastructure to Vehicle

ICMP Internet Control Message Protocol

ID Identifier

IEEE Institute of Electrical and Electronics Engineers

IETF Internet Engineering Task Force

IKEv2 Internet Keying Exchange version 2

IP Internet Protocol

IPsec Internet Protocol security

IPv6 Internet Protocol version 6

ISO International Organization for Standardization

IST Information Society Technologies

ITS Intelligent Transportation System

IVC Inter-Vehicle Communications

LAN Local Area Network

MAC Media Access Control

MANET Mobile Ad-hoc Networks

MCoA Multiple Care-of Addresses

MH Mobility Header

MICS Media-Independent Command Service

MIES Media-Independent Event Service

MIH Media-Independent Handover

MIHF Media-Independent Handover Function

MIHU Media-Independent Handover User

MIIS Media-Independent Information Server

MN Mobile Node

MNN Mobile Network Node

MNP Mobile Network Prefix

MPS Mobile Prefix Solicitation

MPA Mobile Prefix Advertisement

MR Mobile Router

MTU Maximum Transmission Unit

NAT Network Address Translation

NAT64 Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers

NEMO Network Mobility

OBU On-Board Unit

OSI Open Systems Interconnection

P2P Peer to Peer

PC Personal Computer

PDR Packet Delivery Ratio

PoA Point of Attachment

PSK Pre-shared Key

RA Router Advertisement

RFC Request For Comments

RTT Round-Trip Delay Time

RS Router Solicitation

RSU Road-Side Unit

SA Security Association

SAD Security Association Database

SAP Service Access Point

SG Security Gateway

SP Security Policy

SPD Security Policy Database

SPI Security Policy Identifier

SSH Secure Shell

SSL Secure Sockets Layer

TC Technical Committee

TCP Transmission Control Protocol

TLS Transport Layer Security

TS TRaffic Selector

UDP User Datagram Protocol

URL Uniform/Universal Resource Locator

USB Universal Serial Bus

V2I Vehicle to Infrastructure

V2R Vehicle to Roadside

V2V Vehicle to Vehicle

VANET Vehicular Ad-hoc Networks

VPN Virtual Private Network

WEP Wired Equivalent Privacy

WiFi Wireless Fidelity

WiMAX Worldwide Interoperability for Microwave Access

WLAN Wireless LAN

WPA WiFi Protected Access

WPA2 WiFi Protected Access version 2

Resumen

En la revolución tecnológica que estamos viviendo en estos primeros años del siglo XXI, la aparición y el auge de las tecnologías inalámbricas junto a dispositivos cada vez más pequeños y portables está suponiendo un gran impulso en la creación de nuevas aplicaciones y áreas de negocio antes nunca vistas. Las redes, cuya naturaleza hasta nuestros días había tenido un carácter estático, dan un salto cualitativo hacia la movilidad, hacia los vehículos, apareciendo así las *redes vehiculares*. Dichas redes están en el punto de mira del desarrollo tecnológico, dado el previsible número de aplicaciones y beneficios que se pueden aportar al mundo de la conducción.

Este gran avance no ha dejado indiferente a las instituciones europeas e institutos de investigación, que se preguntan si realmente se están aprovechando los beneficios que aportan dichas tecnologías emergentes en el campo de la conducción, carreteras y vehículos. Para ello se trabaja por parte de los institutos de estandarización ISO y ETSI en la definición de estándares que unifique el uso de estas tecnologías mediante una arquitectura de comunicaciones común que permita compartir la misma infraestructura entre los diferentes servicios ofrecidos. Este trabajo ya está dando sus primeros frutos con la aparición de nuevos estándares que definen un conjunto de protocolos y una arquitectura común para implementar redes vehiculares dentro del campo de los Sistemas de Transporte Inteligentes Cooperativos (C-ITS).

IPv6 tiene cada vez más presencia en estos tipos de redes y en los organismos de regulación asociados, ya que es un medio de transporte de datos independiente de la tecnología de transmisión utilizada, permitiendo la integración del mundo vehicular con el Internet del Futuro. Además, IPv6 incluye tecnologías útiles y estandarizadas en las áreas de movilidad de red y la seguridad. Como podrá comprobar el lector a lo largo de las líneas que componen este trabajo, el principal avance aportado por esta Tesis es la definición de una pila de comunicaciones híbrida que incorpore protocolos basados en Internet (IETF) para proporcionar servicios de movilidad y seguridad sobre IPv6 en redes vehiculares, siguiendo en todo momento las directrices marcadas por los órganos de estandarización ISO y ETSI.

Otro de los ámbitos de relevancia en el uso de IPv6 y, en general, en las redes que requieren de una estructura topológica, es la movilidad de los nodos. Sin duda, en el mundo vehicular éste es un aspecto crítico debido a la alta movilidad de los mismos. Los traspasos de red, generalmente conocidos como *handovers*, ocurren entorno a los C-ITS cuando un vehículo cambia su punto de acceso a la red. Esto está generalmente asociado a las comunicaciones vehículo-a-infraestructura (V2I), y puede realizarse de

dos modos posibles: traspaso de un punto de acceso a otro de la misma tecnología (traspaso horizontal), y traspaso entre puntos de distinta tecnología (traspaso vertical). Cuando en dichos traspasos se produce un cambio en el direccionamiento de red, esto puede conllevar asociados numerosos inconvenientes en aplicaciones y servicios que usen protocolos orientados a sesión, como por ejemplo TCP, que obliga a renegociar de nuevo todas las sesiones. Para dar solución a este problema en la presente Tesis, se apuesta por el servicio de *movilidad* del que dispone IPv6, gracias al protocolo *Mobile IPv6* (MIPv6), que permite mantener la continuidad de red usando una única dirección IP, independientemente de los diferentes esquemas de direccionamiento de las posibles redes a las que podamos conectarnos. Esto puede lograrse gracias a la encapsulación de datagramas IP dentro de otros, denominado de forma más coloquial como "Túnel IP". Una variante de MIPv6 es *Network Mobility* (NEMO), soportando el movimiento entre redes no de un solo nodo, sino de toda una red. Puesto que un vehículo es generalmente una "pequeña" red en movimiento, la aplicabilidad de NEMO en las redes vehiculares que hacen uso de IPv6 es directa.

Un error muy extendido en el pasado a la hora de definir y desarrollar arquitecturas de comunicaciones ha sido la de incluir las funcionalidades de seguridad a posteriori, una vez que la red ya estaba definida, implementada e incluso operativa. Puesto que la seguridad se ha convertido en una necesidad real demostrada en los últimos años de historia de Internet, los nuevos campos de investigación que han aparecido alrededor de las redes inalámbricas tales como las redes móviles ad-hoc (MANET), el Internet de las cosas (IoT) o los sistemas inteligentes de transporte cooperativos (C-ITS), engloban actualmente tecnologías para aportar confidencialidad, integridad y autenticidad a las comunicaciones. Precisamente en este último ámbito, referente a las redes vehiculares, la arquitectura telemática de referencia definida por el ISO/ETSI provee una capa trasversal con diferentes servicios de seguridad disponibles para los diferentes niveles dentro de la pila de comunicaciones. Esto establece las bases para el diseño y la integración de protocolos de seguridad, manejo de material criptográfico, sistemas de cifrado y capacidades de filtrado, que harán a las redes vehiculares seguras desde la etapa de diseño.

Las líneas seguidas actualmente para implementar esta seguridad en los C-ITS, tanto en el ámbito de la estandarización como en la literatura científica, en general, se han orientado a la protección de mensajes de red individuales usando criptografía asimétrica. Esto no supone ningún problema si la tasa de envío de mensajes es baja y el destino de los mismos es diverso. Sin embargo, el gran consumo de recursos que supone el uso de la criptografía asimétrica hace inadecuada esta forma de protección para elevadas tasas de envío de mensajes, sobre todo cuando éstos se destinan a determinados nodos de la red tanto en el lado de la carretera como en la infraestructura. Por ello surge la necesidad de abrir otras líneas orientadas a la protección de flujos de datos. En la presente Tesis, el uso de IPv6, y en concreto el protocolo *IPsec*, se presenta como una estrategia para transmitir datos de forma más eficiente usando asociaciones de seguridad extremo a extremo, normalmente implementadas con criptografía simétrica. Esto es considerablemente más rápido y presenta un menor consumo que la alternativa de seguridad asimétrica. Para el establecimiento automático de estas asociaciones de

seguridad tendrá un papel fundamental el protocolo *IKEv2*, que, también operando sobre IPv6, permite un uso fácil y escalable de este tipo de seguridad ofrecida por *IPsec*, además de aportar mecanismos para posibilitar la autenticación y autorización de los vehículos para el acceso a la red. Cabe destacar en este punto que mediante *IKEv2* es posible el uso del protocolo de autenticación extensible (EAP), que nos permite el uso de diferentes métodos de autenticación, pudiendo elegir aquellos más rápidos que nos permitan reducir más aún el tiempo total del traspaso.

Ofrecer las funciones de seguridad y movilidad en IPv6 de manera simultánea requiere de una adaptación especial de los protocolos involucrados para que ambos servicios sean interoperables. Estas adaptaciones están descritas por el IETF a nivel conceptual, sin embargo, en la práctica, se han descubierto lagunas de funcionamiento que requieren de una reconsideración a nivel de diseño. Estas situaciones son difícilmente detectables sobre el papel y comúnmente suelen aparecer en la fase de implementación. En esta Tesis se proponen soluciones a estas situaciones, al tiempo que se implementan, verifican y se evalúa su rendimiento en un escenario real.

La Tesis también presta un especial interés en la evaluación y detección de problemas en los traspasos. Como parte final de esta Tesis, se propone el uso de tecnologías complementarias que ayuden a reducir el tiempo invertido en estos traspasos y ganar en calidad de servicio. En concreto, el uso de *Multiple Care-of Address Registration* (MCoA) permite, además de mantener varios canales de comunicación simultáneos, una reducción en el tiempo de configuración de los enlaces hacia una red destino del traspaso. *IEEE 802.21* se propone igualmente para asistir los traspasos con información del entorno y del propio equipo móvil que ayuda a decidir el mejor momento para realizarlos, maximizando la calidad de servicio ofrecida. De hecho, estas mejoras han sido implementadas y validadas en un entorno de pruebas real, siendo la definición y realización de este proceso una parte fundamental de la Tesis, que además nos ha permitido obtener resultados relevantes sobre el rendimiento de las comunicaciones.

Capítulo 1

Introducción

Las comunicaciones inalámbricas están en un momento de rápida evolución que influye positivamente en la creación de diferentes tipos de redes. Las redes vehiculares, o también llamadas *VANETs*, no son ajenas a este gran empuje tecnológico, ya que están beneficiando en mucho su implantación en el parque móvil actual. En primer lugar y debido a este auge tecnológico, numerosos proyectos de investigación e iniciativas particulares empezaron a explorar el mundo de las comunicaciones vehiculares, descubriendo sus evidentes ventajas pero también sus no tan evidentes inconvenientes. Los investigadores no tardaron en darse cuenta de lo importantes que son los conceptos de seguridad y movilidad para este tipo de redes, creando así dos grandes líneas de investigación. Una se dedica principalmente a detectar los riesgos, vulnerabilidades y posibles ataques que pueden sufrir dichas redes, proponiendo posibles soluciones normalmente apoyadas en métodos criptográficos. Este interés está más que justificado, pues en este tipo de redes donde hay implicados vehículos en movimiento, un ataque puede poner en riesgo vidas humanas. La otra gran línea de investigación se dedica a identificar los problemas inherentes al movimiento de los nodos de este tipo de redes, como son los cortes que se producen en las comunicaciones en el momento que ocurren los traspasos (o también conocidos como *handovers* en la literatura científica), o cómo lidiar con la complejidad de movernos en redes heterogéneas donde intervienen diferentes tipos de tecnologías. También en esta línea se proponen soluciones que mejoren las comunicaciones, haciéndola lo más fluida y sin cortes posible.

Según avanzaban las investigaciones y las iniciativas comerciales, se hizo evidente la necesidad de estándares para la armonización de estas redes. Por este motivo, numerosos institutos de estandarización no tardaron en ponerse a trabajar para buscar una estructuración unificada de los componentes que las implementaban, definiendo una arquitectura de red de referencia para todos.

Una de las aplicaciones más importantes de este tipo de redes vehiculares es la incorporación de las mismas a los llamados Sistemas Inteligentes de Transporte (ITS, según sus siglas en inglés), permitiendo que los distintos elementos de la carretera (vehículos, señales de tráfico, semáforos, etc.) se comuniquen para colaborar entre sí y así poder implementar servicios y aplicaciones que aumenten la seguridad en

las carreteras y la eficiencia del transporte. De estos sistemas hablaremos en este capítulo, además de exponer de forma resumida la situación actual de dichas redes, proyectos de investigación en progreso y estándares definidos hasta la fecha. También estableceremos los objetivos y aportaciones de la presente Tesis Doctoral. De forma resumida exploraremos los trabajos que otros investigadores han aportado en líneas similares a la nuestra. Y, por último, veremos las publicaciones fruto de nuestras investigaciones que han dado como resultado este trabajo.

1.1. Sistemas Inteligentes de Transporte (ITS)

Los Sistemas Inteligentes de Transporte, o *Intelligent Transportation Systems* (ITS), conforman un esfuerzo común entre gobiernos, industria privada y centros de investigación, para aplicar las nuevas tecnologías de la información y las comunicaciones en los problemas actuales del transporte mundial [2]. Los objetivos actuales de tales desarrollos son muy diversos, aunque los incluidos de raíz en la concepción inicial de ITS son los siguientes [3]:

- Como máxima prioridad, mejorar la **seguridad** de todos los medios de transporte actuales. El vehículo es la principal preocupación, debido a que supone la mayoría de las muertes y lesiones que se producen en el transporte mundial.
- Mejorar la **eficiencia** de los sistemas de transporte, reduciendo los tiempos de viaje y las congestiones.
- Incentivar la **intermodalidad**, mediante la combinación de varios medios de transporte para completar un viaje.
- Integrar el transporte dentro de las políticas de **desarrollo sostenible**; en particular, reduciendo las emisiones de gases de los turismos y vehículos pesados, y optimizando el uso de las infraestructuras.
- Mejorar el **confort** de los pasajeros, con un gran número de servicios de información, servicios de ayuda a la decisión, sistemas de guiado y navegación, etc.

La preocupación actual de ITS trata problemas relativos al transporte marítimo, aéreo y por carretera. El despliegue tecnológico en el transporte marítimo y aéreo ha ido avanzado en gran medida, liderando la carrera tecnológica en la que están envueltos todos los medios de transporte. En este sentido, diversas tecnologías dentro de las telecomunicaciones y la electrónica ya se encuentran incorporadas en completos sistemas de información para incrementar su rendimiento. De esta manera, si bien es necesario realizar un amplio estudio sobre cómo ofrecer servicios añadidos para la seguridad y la intermodalidad, es sin embargo el transporte por carretera el que aglutina la mayoría de las preocupaciones de la comunidad ITS. Aquí es precisamente donde se centra la presente Tesis Doctoral.

La actual demanda de movilidad terrestre ha excedido la capacidad del sistema de carreteras y, debido a que éste no puede ser expandido al mismo ritmo, la infraestructura actual debe ser usada de una manera mucho más eficiente para manejar el aumento de la demanda. La congestión en las zonas urbanas y en las vías de comunicación principales continúa creciendo rápidamente, a la misma velocidad que las pérdidas económicas por el descenso de la productividad en la actividad laboral. Los enormes costes en combustible y el daño medioambiental no se han considerado hasta hace muy poco tiempo. Además, y lo que es más importante, los accidentes de tráfico producen más de 1.2 millones de muertes cada año, y alrededor de cuatro veces esta cantidad de heridos [4]. Aunque esa cifra no ha subido en los últimos años, es inaceptablemente elevada, lo que nos empuja a seguir investigando en tecnologías que ayuden de manera decisiva a disminuir esa cifra año a año.

Con la intención de sobrellevar los crecientes problemas en el transporte por carretera, los ITS pretenden mejorar la eficiencia de la red viaria, usando información tanto histórica como en tiempo real sobre el estado del tráfico y de las infraestructuras, para detectar necesidades en los distintos componentes del sistema [5]. Integrando nuevas funcionalidades en el ámbito de ITS, tanto en el lado de la infraestructura como en el vehículo, es posible reducir los tiempos de viaje, disminuir la frecuencia y severidad de los accidentes, reducir costes colaterales, y mejorar la satisfacción de los usuarios.

Los ITS aplican sistemas de procesamiento de información avanzada, comunicaciones, sensorización, y tecnologías de control por ordenador para intentar solventar los problemas del transporte terrestre. Sin embargo, es necesario mantener, e incluso aumentar, los recursos invertidos en la investigación y desarrollo para estas tecnologías, pues el potencial que tienen para poder desplegar nuevos servicios y aplicaciones es considerable. Por tanto es necesario incentivar programas de aplicación real de estos nuevos avances.

1.2. Sistemas Inteligentes de Transporte Cooperativos (C-ITS)

Un paso más para alcanzar los objetivos marcados en los ITS han sido la aparición de los Sistemas Inteligentes de Transporte Cooperativos, del inglés *Cooperative Intelligent Transport System (C-ITS)*. Estos son ITS donde entidades como los vehículos, las señales de tráfico, los semáforos y, en definitiva, los elementos que componen la vía, pueden comunicarse entre sí intercambiando información con el objetivo de aportar seguridad, sostenibilidad, eficiencia y confort de una forma más efectiva que los ITS autónomos. Esto es así ya que la información que pueden obtener en su conjunto todas las entidades dentro de las C-ITS es mucho mayor que en los ITS autónomos donde la información obtenida por los sensores es mucho más limitada. Además, se abre una interesante vía para el desarrollo de nuevos servicios que pueden implementarse mediante esta colaboración de las diferentes entidades que conforman la carretera y los

propios vehículos. Según [6] y [7], estos nuevos servicios pueden categorizarse en estos tres grupos:

- Seguridad: Estos servicios persiguen reducir la tasa de accidentes y aumentar la seguridad de tanto los ocupantes de los vehículos como de los peatones. Algunos ejemplos destacables son el sistema anticolisión y notificaciones como el que nos informa la proximidad de un accidente o que un vehículo de emergencia se aproxima.

- Eficiencia del tráfico y asistencia al conductor: Servicios orientados a incrementar la capacidad de la red de carreteras y reducir el tiempo de los trayectos. Como ejemplos podemos mencionar el servicio de límite de velocidad variable, planificación dinámica de rutas, control de intersecciones y servicios de detección y mitigación de congestiones.

- Entretenimiento e información, o *infotaiment*: La principal motivación de estos servicios es la de incrementar el confort de los ocupantes de los vehículos, proporcionando servicios de valor añadido, acceso a Internet y contenidos multimedia. Algunos ejemplos son la guía turística contextual, vídeo bajo demanda, pronóstico del tiempo y videoconferencias.

Para dar soporte a esta gran diversidad de servicios de manera escalable se necesita de una arquitectura de red genérica que asegure la compatibilidad de todos ellos y de los futuros que están por llegar de la mano de diferentes proveedores. Debido a esto, durante los últimos años, diversos grupos de estandarización han estado trabajando intensamente para conseguir ese objetivo en ITS cooperativos. Primero, el Comité Técnico 204 del ISO publicó el concepto *Communication Access for Land Mobiles* (CALM) [8], que fue mejorado después por otro Comité Técnico del ETSI dedicado al mundo ITS basándose en los resultados del proyecto Europeo de investigación COMeSafety a través de una nueva arquitectura de comunicaciones ITS [9]. Ambos comités unieron sus esfuerzos para definir una nueva arquitectura de comunicaciones estándar para ITS [8], la cual podía ser instanciada en todos los elementos que componen la red (vehículos, estaciones de carretera, centros de control, etc.).

A pesar de que se ha establecido una arquitectura general, existen trabajos en marcha para definir cómo interactúan los diferentes módulos que integran la pila de comunicaciones. Uno de estos puntos que se encuentra en discusión tiene que ver con los protocolos a utilizar en la capa de red. Esto es debido a que existen dos familias de protocolos estándar con gran presencia en escenarios ITS:

- Por un lado, protocolos específicos del mundo ITS, como *Wave Short Message Protocol (WSMP)* [10] y *GeoNetworking* [11], definidos por el *Institute of Electrical and Electronics Engineers* (IEEE) y el *European Telecommunications Standards Institute* (ETSI) respectivamente.

- Por otro lado, se encuentra el protocolo IP versión 6 (IPv6) [12], basado en la evolución de los protocolos usados en Internet por décadas, definido por el *Internet Engineering Task Force* (IETF).

Aunque *WSMP* y *GeoNetworking* ofrecen funcionalidades adaptadas a comunicaciones vehiculares, estos sólo contemplan las comunicaciones Vehículo-a-Vehículo (V2V). IPv6 representa una solución más dirigida a comunicaciones Vehículo-a-Infraestructura (V2I), capaz de interoperar con el que será el Internet del mañana, dando soporte a conceptos como el Internet de las Cosas (IoT) o *Smart Cities*, entre otros. Pero uno de los mayores beneficios de usar IPv6 es la posibilidad de aplicar soluciones ya existentes de seguridad, movilidad, *multi-homing*, etc. En la presente Tesis Doctoral se apuesta por el uso de IPv6 como pilar para establecer una arquitectura de comunicaciones para entornos vehiculares como este, con la posibilidad de aprovechar las bondades de IPv6 vistas antes, así como sus servicios asociados, sin olvidar la gran ventaja de poder integrar fácilmente dichas redes en el Internet del Futuro.

1.3. Redes Vehiculares

Como hemos visto antes, las C-ITS aspiran a proporcionar soluciones a los problemas de seguridad de los pasajeros y congestión del tráfico, además de mejorar el confort y la información que reciben tanto los conductores (asistencia al viaje) como los pasajeros (*infotaiment*). El medio principal por el que se quiere llegar a este objetivo es el uso de las comunicaciones inalámbricas en los sistemas de transporte, desplegando sobre ellas redes vehiculares.

1.3.1. Elementos de una Red Vehicular

Un nodo de una red C-ITS puede ser un vehículo equipado con una unidad de abordo (*On-board Unit*, OBU) con un sistema inalámbrico de comunicaciones de corto alcance con las que poder formar redes ad-hoc, pero también pueden ser nodos el equipamiento situado al lado de la carretera (*Road-Side Units*, RSUs) utilizado para conectar los vehículos a la infraestructura de red cableada. Podemos distinguir aquí entre dos tipos de comunicaciones:

- Vehículo-a-Vehículo (V2V), comunicaciones realizadas entre los vehiculos de un modo ad-hoc, espontáneo. De esta forma, un vehículo puede intercambiar información valiosa con otros vehículos como las condiciones del tráfico o aviso de accidentes, entre otras.
- Vehículo-a-Infraestructura (V2I), comunicaciones realizadas entre los vehículos y estaciones de carretera (RSUs), para intercambiar información sobre las condiciones de la carretera y avisos para que los conductores tomen medidas de seguridad. También se utiliza como enlace para comunicarse con redes externas como Internet.

A parte de la OBU, principalmente orientada a interactuar con los ocupantes, los vehículos también incorporan una serie de sensores (radar frontal y trasero, etc.)

que reciben información útil del entorno que generalmente el conductor es incapaz de percibir por sí mismo. No puede faltar un sistema de posicionamiento global, como es el *Global Positioning System* (GPS), necesario para localizar el vehículo y proporcionar asistencia a la conducción. Y, cómo no, para poder comunicarse con otros vehículos o con las estaciones de carretera, se necesita un sistema de comunicaciones, un pequeño sistema de computación embebido para ponerlo en funcionamiento y un sistema de registro de eventos a prueba de manipulación, similar a como actúa una caja negra de un avión. Por último, para identificar a los vehículos, algunos autores proponen el uso de matrículas electrónicas y no usar las convencionales por motivos de seguridad.

1.3.2. Estandarización de las Redes Vehiculares

Como en toda nueva tecnología, para facilitar el proceso de producción y reducir los costes y el tiempo de llevar los productos y servicios al mercado, la estandarización y normalización de dichas tecnologías de información y comunicaciones son un paso fundamental para asegurar su inter-operatibilidad y rápida implementación. En el caso de las redes vehiculares, la estandarización afecta a todas las diferentes capas que conforman la pila de comunicaciones, desde la capa física hasta la capa de aplicación.

Para la capa física, el Gobierno de los Estados Unidos, representado por el *Federal Communicacion Commission* (FCC), estableció en 1999 una banda del espectro radio-eléctrico de 75 MHz de ancho que comprende desde los 5850 hasta los 5925 GHz para realizar las comunicaciones necesarias dentro de las redes C-ITS. Esta banda se incluyó dentro de la tecnología denominada en Norte América como *Dedicated Short Range Communications* (DSRC). En Europa, el *European Telecommunications Standards Institute* (ETSI) también hizo lo propio. Sin embargo, la configuración de los canales dentro de esa banda se realizó de formas diferentes. Mientras que FCC definió siete canales de 10 Mhz (172, 174, 176, 178, 180, 182 y 184), ETSI definió sólo cinco (172, 174, 176, 178 y 180). Cada uno de estos canales tienen un propósito distinto, dividiéndose en canales de servicio (SCH) y canales de control (CCH). En el caso de Estados Unidos, el canal designado para control es el 178. Sin embargo, en Europa es el 180. El resto de canales en ambos casos son designados como de servicio. Una vez definidos los canales por donde transmitir, el *Institute of Electrical and Electronics Engineers* (IEEE) expandió la familia de estándares 802.11, añadiendo el estándar 802.11p para dar mejor soporte a las redes vehiculares a nivel físico y de acceso al medio, usando las bandas contempladas por DSRC. Este estándar, definido en IEEE 802.11p-2010 [13], adapta las capas físicas (PHY) y de acceso al medio (MAC) ya definidas en el documento IEEE 802.11-2007 [14] para cumplir con los requisitos de las redes vehiculares. Se puede resumir que está basado en el ya existente 802.11a al que se le ha suprimido la fase de asociación. Además, su esquema de QoS está basado en el 802.11e. En Europa, 802.11p es también considerado dentro del estándar ITS-G5 [15] definido por el ETSI para establecer igualmente las capas físicas y de acceso al medio en entornos ITS.

El IEEE también definió posteriormente una familia de estándares en el documento IEEE 1609 y bajo las siglas WAVE (*Standard for Wireless Access in Vehicular*

Environments), que definen una arquitectura, protocolos, servicios e interfaces necesarios para que una estación *WAVE* pueda comunicarse en un entorno vehicular y establecer comunicaciones V2V y V2I. Esta arquitectura también define aspectos de seguridad como la protección de los mensajes intercambiados por las estaciones *WAVE*. Esta familia de estándares forma la base para implementar aplicaciones y desplegar servicios dentro del mundo del transporte, sobre todo de las ITS.

Uno de los estándares de la familia 1609 [16], concretamente el IEEE 1609.3 [10], define los servicios de red y transporte que debe ofrecer una estación *WAVE*. Dentro de estos servicios se incluyen el servicio de enrutamiento de paquetes y direccionamiento. WAVE ha enfocado sus esfuerzos en la descripción de un protocolo de red y transporte específico para redes vehiculares que llamaron *WAVE Short Message Protocol* (WSMP). Aunque también se define una alternativa a WSMP para poder dar soporte a aplicaciones IP, los esfuerzos aplicados en esta línea de investigación son claramente insuficientes, dejando de lado la incorporación de IPv6 a dichas redes y dificultando a su vez la integración de servicios basados en IP. De hecho, como hemos adelantado antes, este es uno de los motivos fundamentales que han inspirado la presente Tesis, proponiendo el uso de IPv6 como alternativa a protocolos específicos como WSMP.

En la capa de aplicación podemos destacar la presencia del estándar J2735 [17] definido por el *Society of Automotive Engineers* (SAE). En él se establece el formato de mensajes, campos y demás elementos que se usan para intercambiar información para mejorar la seguridad vial tanto entre vehículos (V2V) como con la infraestructura (V2I).

Esta arquitectura definida por WAVE ha resultado ser demasiado rígida en el plano de las tecnologías de acceso a utilizar, pues sólo contempla 802.11p para comunicar los nodos de la red vehicular, ya fueran comunicaciones V2V o V2I. Esta fue una de las razones por las que la organización de estandarización *International Standard Organization* (ISO) propuso una arquitectura de red más flexible llamada *Communication Access for Land Mobiles* (CALM), fruto de los trabajos llevados a cabo por uno de sus grupos de trabajo (WG16) dentro del comité técnico 204, estando abierta a cualquier tecnología de nivel físico y de enlace. La organización europea ETSI, más afín a lo establecido por ISO en CALM, surgió posteriormente para aportar nuevos trabajos que, a la postre, serían integrados en una arquitectura conjunta de referencia ISO/ETSI. En este caso, las denominaciones de los elementos que componen las redes vehiculares cambiaron con respecto a las establecidas por WAVE, denominando a los nodos que conforman una red C-ITS como *estaciones ITS*, que es un concepto más genérico e independiente de las tecnologías utilizadas.

La industria automovilística dentro de la Unión Europea, representada por el *Car to Car Communication Consortium (C2C-CC)*, también ha desarrollado su propia arquitectura que primero se denominó *Car to Car Networking (C2CNet)*. Esta fue posteriormente incorporada en una evolución de la misma gracias al trabajo realizado en el proyecto *COMeSafety* [18]. Sería a partir de este proyecto de donde surgirían los primeros trabajos del ETSI TC ITS, comité técnico encargado de este área. A esta nueva evolución se la llamó *Geographical Networking (GeoNetworking)*, que contempla

tanto las comunicaciones V2V (C2C) y V2I (IPv6). Dentro de las posibles tecnologías, además de todas las posibles variantes de la familia 802.11 definidas por el IEEE, se contempla el uso de tecnologías establecidas por el *The 3rd Generation Partnership Project* (3GPP), como son GRPS, 3G/LTS, etc. En la Figura 1.1 podemos ver, de forma resumida, una comparativa de las tres arquitecturas que hemos destacado.

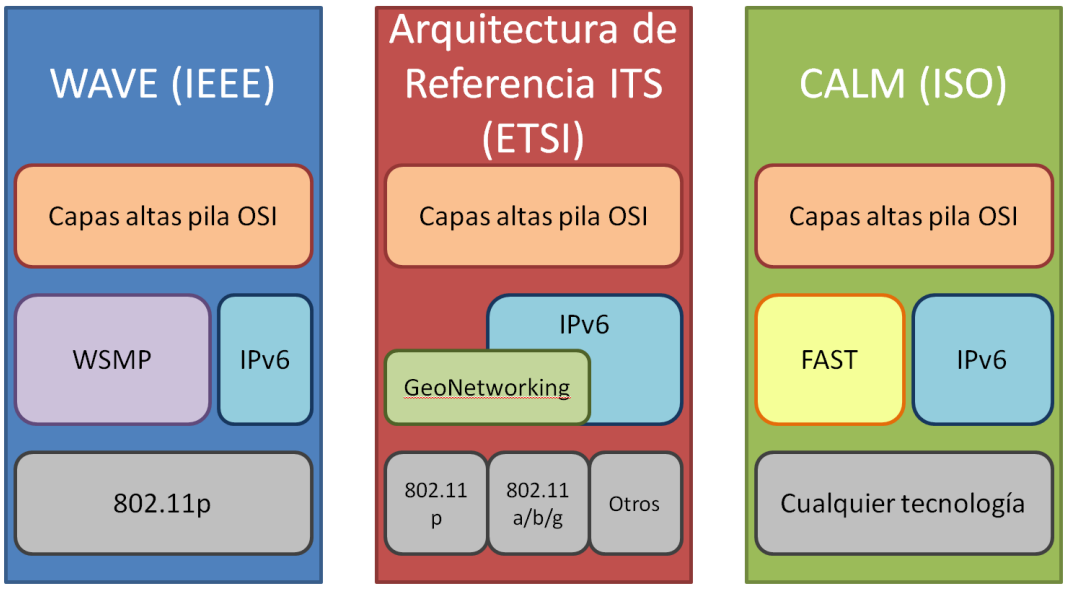


Figura 1.1: Comparativa de arquitecturas más influyentes en redes vehiculares [1]

1.3.3. Proyectos relacionados

La principal motivación que lleva a las autoridades de los distintos países a crear proyectos relacionados con ITS y las redes vehiculares es alcanzar un nivel razonable de seguridad y control en las carreteras para reducir el número de accidentes y los problemas generados por el rápido aumento del tráfico. En esta línea, los Programas Marco de la Comisión Europea han incluido en los últimos años una gran cantidad de proyectos relacionados con los vehículos. Podemos destacar aquí el trabajo llevado a cabo por algunos de estos proyectos:

- **ITSSv6** [19], proyecto europeo donde se ha diseñado con éxito una pila de comunicaciones basada en la arquitectura de referencia ISO/ETSI, buscando unificar criterios para el desarrollo de futuras iniciativas. Además, se han incorporado servicios de movilidad y seguridad importados del mundo IETF e Internet. En este proyecto la Universidad de Murcia participó activamente aportando una implementación para el protocolo IKEv2 y la integración de la misma con otras implementaciones de servicios de movilidad. También se hicieron trabajos de validación de la pila finalmente definida, elaborando para ello un

entorno de pruebas real de donde obtener resultados y conclusiones. Gran parte de los resultados obtenidos aquí forman parte de la presente Tesis.

- **FOTsis** [20], proyecto europeo de gran envergadura cuya principal misión ha sido la evaluación experimental en entornos reales de servicios que usan los avances generados a nivel de red en otros proyectos, como por ejemplo la pila del proyecto ITSSv6. La Universidad de Murcia también tuvo la gran oportunidad de participar en este proyecto, probando dicha pila en diferentes puntos de la red vial de países europeos, entre los que podemos destacar Grecia, Portugal, Alemania y España. Parte de lo aprendido en dichas pruebas influyó positivamente en la elaboración exhaustiva de un entorno de validación que nos proporcionó resultados interesantes que abordamos en la presente Tesis.
- **DRIVE-C2X** [21], proyecto ambicioso que recoge los resultados de otros proyectos anteriores centrados en cumplir los requisitos de las redes C-ITS, como PreVENT, CVIS, SAFESPOT, COOPERS y Pre-DRIVE C2X. Sin embargo, la cobertura que ofrece a protocolos y servicios basados en IPv6 es insuficiente.
- **OVERSEE** [22], proyecto que provee una plataforma genérica y estandarizada para entornos vehiculares, además de un entorno para desarrollo de aplicaciones.
- **PRESERVE** [23], proyecto europeo centrado en la seguridad de las redes C-ITS que recoge el resultado de proyectos anteriores para desarrollar un subsistema de seguridad completo, escalable y eficiente para comunicaciones V2X.
- **COMeSafety** [18], proyecto que tiene como principales vías de trabajo la coordinación e integración de los resultados obtenidos en otros proyectos europeos sobre comunicaciones vehiculares, y la colaboración en labores de estandarización en los protocolos usados.
- **SeVeCOM** [24], proyecto donde se tratan los problemas de seguridad de las comunicaciones vehiculares, considerando la creación de un entorno de comunicación seguro.
- **EVITA** [25], proyecto que trata problemas de seguridad en el tratamiento y transporte de la información relativa a soluciones cooperativas ITS, proponiendo una plataforma segura de comunicaciones vehiculares.

La Unión Europea ya tiene puesta la vista en el futuro inmediato, creando el nuevo *Horizon 2020*, como continuación de los Programas Marco, donde hay definidas distintas áreas de interés dentro del sector de la automoción. Una de ellas es la llamada *Smart, Green and Integrated Transport*, donde ya hay definido un programa de trabajo [26] en el que podemos ver que se define una gran convocatoria para proyectos de investigación llamada *Mobility for Growth*, y dentro de ella una específica que incluye a las C-ITS llamada *Cooperative ITS for safe, congestion-free and sustainable mobiliy* (MG.3.5-2014). Es por tanto evidente el interés de la Unión Europea por seguir apoyando este área de interés.

Aparte de los Programas Marco, la Comisión Europea mantiene una iniciativa conjunta entre la industria y el sector público, para acelerar el despliegue de las tecnologías de la información y las comunicaciones en los vehículos. A esta iniciativa se la llamó *Intelligent Car Initiative*, donde destaca la propuesta *eSafety* [27], que se encarga de la parte que concierne al despliegue del ITS relacionado con la seguridad vial.

La presente Tesis ha estado enmarcada en los proyectos ITSSv6 y FOTsis, proyectos que tienen como objetivo común incorporar diferentes tecnologías y protocolos a las redes C-ITS, que permitan mejorar aspectos como la seguridad y la movilidad de las comunicaciones.

1.4. Objetivos de la Tesis Doctoral y Aportaciones

En esta sección establecemos cuáles son los objetivos generales de la presente Tesis, cuáles son sus aportaciones y cómo se desarrollan en el propio documento a lo largo de sus capítulos.

1.4.1. Objetivos

El objetivo final para decidir potenciar las tecnologías de la información y las comunicaciones en el mundo vehicular es la de reducir el número de muertes por accidente de tráfico que se producen cada año, que a nuestro parecer es un número inaceptable y que debemos trabajar duro para reducir en medida de lo posible, y creemos que las tecnologías de la información y las comunicaciones pueden contribuir en mucho para conseguirlo. También se persigue reducir el impacto medio-ambiental que producen los vehículos, trabajando en la optimización de las rutas y patrones de conducción sostenibles gracias a una correcta asistencia a la conducción que sin las tecnologías de la información y las comunicaciones sería muy difícil proveerlo. Sin embargo, cabe destacar otros importantes objetivos que ayudarán positivamente a alcanzar este gran objetivo general:

1.4.1.1. Incorporar IPv6 a Entornos ITS

El mundo de las comunicaciones móviles en entornos vehiculares está evolucionando con paso firme en los últimos años. Sin embargo, se está avanzando más hacia protocolos y formas de comunicación específicas para dichos entornos vehiculares, donde la localización geográfica de los vehículos juega un papel importante, como por ejemplo tiene en cuenta el protocolo Geonetworking. Los métodos para dotar de seguridad a estas propuestas también han derivado en esquemas específicos orientados a la protección individual de mensajes gracias a la criptografía asimétrica. En definitiva, soluciones muy alejadas de las ya empleadas en redes de amplia adopción como las que forman parte de Internet, ya que, hasta el momento, se han considerado redes de naturaleza distinta. Estas soluciones ya existentes en Internet están basadas

principalmente en IP, protocolo ampliamente extendido en su versión 4 sobre el que se han desarrollado numerosos protocolos y servicios, que el paso de los años ha hecho que lleguen a un alto estado de madurez. La aparición de IP en su versión 6 ha supuesto un hito importante en Internet, mejorando los servicios ya existentes e incrementando algunas características de su antecesor, como la cantidad de direcciones IP disponibles. Nuestro objetivo aquí es contar con IPv6 como protocolo potencial para realizar comunicaciones en entornos vehiculares, tanto Vehículo a Infraestructura (V2I) como Vehículo a Vehículo (V2V). Una ventaja implícita de contar con IPv6 en entornos vehiculares es la de facilitar al máximo la conectividad de este tipo de redes al Internet del Futuro.

1.4.1.2. Explotar la Sinergia entre los Estándares ISO/ETSI para Redes Vehiculares y los Provenientes de IETF e Internet

El mundo vehicular y de Internet, junto con sus Estándares asociados, deben de tenerse en cuenta mutuamente ya que son muchos los beneficios que pueden aportarse entre sí, sobre todo en el caso del mundo vehicular, que puede sacar partido de muchos de los avances conseguidos en el seno de Internet. Esto debe hacerse de una forma normalizada y, para ello, los organismos de estandarización ISO y ETSI han definido una arquitectura de referencia para las redes cooperativas ITS, de tal forma que, independientemente de los protocolos utilizados, quedan organizados gracias a unas determinadas capas y planos dentro de una pila bien estructurada. Esta arquitectura de referencia es lo suficientemente abierta como para aceptar a IPv6 como protocolo de red y también a TCP y UDP como protocolos de transporte. Además serán importantes los sistemas de auto-configuración de IPV6 que jugarán un papel relevante en este tipo de redes.

1.4.1.3. Dotar de Esquemas de Seguridad a las Redes Vehiculares

Hasta la fecha es muy frecuente encontrar en redes vehiculares el uso de esquemas criptográficos para la protección de la información intercambiada, en su mayoría apoyados en la criptografía asimétrica y el uso de certificados y firma digital, métodos muy convenientes para pequeñas cantidades de datos. Para casos como las transmisiones multimedia que demandan gran cantidad de datos transmitidos, los esquemas asimétricos dejan de ser tan atractivos. Esto nos motiva a buscar una alternativa, en este caso usando esquemas simétricos, mucho menos demandantes de recursos, para la protección de este tipo de tráfico. Esta necesidad puede ser satisfecha gracias a la incorporación de IPv6 en redes vehiculares, ya que hemos podido contar en consecuencia con algunos de los servicios que han crecido a su lado y así aprovechar muchas de sus ventajas sin necesidad de desarrollar protocolos nuevos pero inmaduros. Entre estos servicios destacamos el de seguridad (orientada a la protección de flujos de paquetes mediante criptografía simétrica, más rápida que su alternativa asimétrica) gracias a los protocolos *IP Security*(IPsec) e *Internet Keying Exchange version 2* (IKEv2), que parecen ser una alternativa madura y razonable a los esquemas

asimétricos.

1.4.1.4. Conseguir una Experiencia sin Cortes en Redes Vehiculares

Los servicios de movilidad en redes vehiculares tan necesarios para el mantenimiento de una misma dirección para usar a lo largo de todo el recorrido del vehículo independientemente de las distintas redes visitadas, todavía tienen puntos débiles que pueden ser mejorados. Por este motivo, otro de nuestros objetivos es aportar una solución a estos problemas de continuidad que se presentan en situaciones de cambio de tecnología o direccionamiento, en los denominados traspasos o *handovers*, tan frecuentes en las comunicaciones inalámbricas en entornos vehiculares. Por ello, el objetivo es establecer y desarrollar las posibles líneas de investigación para la mejora de los procesos acontecidos en los traspasos, de tal forma que podamos llegar a una solución combinada que permita que dichos traspasos sean lo más imperceptibles dentro de lo posible. Esto implica realizar una ambiciosa batería de pruebas e identificar los puntos débiles de los traspasos y proponer tecnologías que puedan poner solución a dichas debilidades.

1.4.1.5. Incorporar un Servicio de Movilidad Seguro en C-ITS

Otro de los beneficios de incorporar IPv6 a las redes vehiculares ha sido el poder contar con el servicio de movilidad, gracias a los protocolos *Mobile IP version 6* (MIPv6) y su versión para movilidad de redes *Network Mobility* (NEMO). Esto permitirá a los nodos usar una misma IP en todo momento, independientemente de la red que nos esté danto conectividad en cada momento, lo que supone múltiples ventajas, como el poder conservar las sesiones establecidas a nivel de transporte y aplicación después de un cambio de red. Sin embargo, este objetivo entra en conflicto con el de aportar un servicio de seguridad simétrico, ya que ambos servicios, tan necesarios para las comunicaciones inalámbricas, deben ponerse de acuerdo para funcionar conjuntamente. Este proceso de integración entre ambos servicios no es directo tanto a nivel de diseño como de implementación. Estos deben salvar la dificultad de establecerse el uno sobre el otro, es decir, tener a la vez un servicio de seguridad móvil y un servicio de movilidad seguro, algo que no es baladí. Los propios grupos de trabajo del IETF ya han trabajado en ello y han establecido una forma estándar de interactuar entre estos dos servicios para poderlos ofrecer simultáneamente. Sin embargo, estos estándares todavía adolecen de algunas lagunas que se ponen de manifiesto cuando se llega a la fase de implementación, como ha sido en nuestro caso. Identificar dichas lagunas y establecer propuestas de solución desde la etapa de diseño es uno de los objetivos principales de esta Tesis.

1.4.1.6. Aportar una Metodología de Pruebas en Entornos Reales

La existencia prácticamente nula de iniciativas de redes vehiculares reales sobre las que realizar pruebas y obtener resultados en el mundo real nos ha motivado para realizar un enfoque investigador dirigido al diseño y a la implementación real de una

arquitectura de red vehicular. Además, la reciente aparición de nuevas tecnologías especialmente pensadas para las redes vehiculares, como es el caso de 802.11p, es otra motivación más para probar dicha tecnología en un entorno vehicular real. Para conseguir este objetivo, hay que alcanzar otros objetivos dependientes de este, como el de conseguir una implementación de los servicios involucrados, construir un prototipo funcional con el que hacer pruebas, y que dichas pruebas sigan una metodología diseñada de antemano con la rigurosidad necesaria para poder confiar en los valores y resultados obtenidos por dichas pruebas.

1.4.2. Aportaciones

Como producto resultante de perseguir los objetivos marcados en el apartado anterior, destacamos aquí las aportaciones surgidas de dicho proceso investigador.

1.4.2.1. Propuesta de Arquitectura y Pila de Red IPv6 para Entornos C-ITS

Para incorporar el protocolo IPv6 y sus servicios asociados de movilidad y seguridad, hace falta establecer primero cómo van a relacionarse los distintos elementos de la red. Por ello, se propone una arquitectura de red basada en IPv6 y sus servicios asociados siguiendo la arquitectura de referencia establecida por el ISO/ETSI. Esta es, sin duda, una de las grandes aportaciones de la presente Tesis, que será tratada en profundidad en el **Capítulo 3**. Adaptar nuestra solución a dicha arquitectura de referencia es un paso necesario para dirigirnos hacia un sistema único y compatible donde poder desarrollar todos los servicios para ITS compartiendo la misma pila y, por tanto, el mismo hardware. De no hacerlo así supondría que cada servicio fuera acompañado con su hardware específico, sin poderlo compartir con otros servicios, hecho que todas las organizaciones de estandarización quieren evitar a toda costa. La arquitectura que proponemos se ha implementado como prototipo pero es lo suficientemente completa como para poder realizar evaluaciones de rendimiento de las comunicaciones y establecer conclusiones.

1.4.2.2. Implementación de IKEv2: OpenIKEv2

Una de las tecnologías asociadas a IPv6 es la seguridad aportada por IPsec e IKEv2. Nuestra intención de llevar a cabo una red vehicular real ha promovido la creación de una nueva implementación del protocolo IKEv2. Esta implementación, llamada OpenIKEv2 por ser de código abierto, fue realizada en el marco de la presente Tesis debido a la escasez de implementaciones libres que había en el momento de iniciar este proyecto investigador. Este esfuerzo en desarrollar nuestra propia solución IKEv2 ha sido recompensado con creces en diversas ocasiones, ya que nos ha abierto las puertas a proyectos de investigación europeos como Enable [28] o ITSSv6 [19], donde se hacía necesario la adaptación de parte del protocolo o su integración en nuevas propuestas. Además, durante la implementación del mismo, se descubrieron lagunas

en la documentación del estándar, que propiciaban situaciones inconsistentes. Se tomó nota de dichas situaciones y se propusieron soluciones, siempre consensuadas con los propios redactores del estándar. Concretamente, las aportaciones se incluyeron en el RFC 4718 [29], que describía guías y clarificaciones útiles para implementar IKEv2, que posteriormente fueron incorporadas al RFC de definición del protocolo, el RFC 5996 [30] en 2010, y actualizado por el vigente RFC 7296 [31] en 2014. En el **Capítulo 4** y los **Apéndices A y B** se entra en mayor detalle, tanto sobre el propio estandar IKEv2 como en nuestra implementación OpenIKEv2.

1.4.2.3. Integración de los Servicios de Movilidad y Seguridad de IPv6 para Redes Vehiculares

Esta aportación, tratada con todo detalle en el **Capítulo 5**, tiene que ver con la problemática de incorporar los servicios de movilidad y seguridad en redes C-ITS. Esto es, adaptar las implementaciones de estos servicios ya existentes (OpenIKEv2 y Mip6d del proyecto UMIP [32]) para que puedan funcionar juntos. Ambos servicios usan tecnologías de enrutamiento de paquetes basadas en túneles, es decir, encapsulamiento de paquetes. Esto provoca que entren en conflicto, pues hay que decidir qué encapsulamiento hay que realizar primero. Además, durante el proceso de integración se originaron situaciones inconsistentes que el estándar no especificaba ni aclaraba, derivadas de la problemática de cómo compartir la información interna que manejan ambos servicios. Se propone pues una solución para este problema, que además ha sido presentada en publicaciones de prestigio como veremos más adelante. Como resultado, gracias a nuestra implementación OpenIKEv2 y conjuntamente con la implementación Mip6d, se ha llegado a una implementación funcional especialmente pensada para entornos de movilidad donde se le da una especial importancia a la seguridad. Tanto es así que el resultado de dicha integración forma parte de la pila de comunicaciones diseñada por el proyecto europeo ITSSv6, proyecto que busca unificar criterios a nivel europeo con vistas a diseñar las que serán las C-ITS del futuro.

1.4.2.4. Mejora en los Procesos de Traspaso mediante la Integración de Tecnologias IETF e IEEE

Por último, al ser el traspaso un elemento tan importante en las comunicaciones inalámbricas en movilidad, esta Tesis también aporta en su **Capítulo 6**, desde varios frentes diferentes, una mejora sustancial en dichos traspasos. Como producto de una gran campaña de pruebas, se pudo identificar dónde se podían mejorar dichos traspasos y qué tecnologías podíamos usar para conseguirlo. Estas líneas de investigación no se han limitado al nivel conceptual, sino que también se han realizado implementaciones y comprobado la efectividad de cada una de las diferentes técnicas para reducir el tiempo de traspaso. Estas no son excluyentes, sino que pueden aplicarse conjuntamente, sumando cada uno de sus beneficios. Las técnicas que hemos identificado para mejorar los traspasos son:

- Reducción de los tiempos en los procesos de autenticación. Para ello el

protocolo EAP ha sido determinante, ya que permite el uso de diferentes métodos de autenticación y así poder seleccionar el que mejor relación *nivel de protección-consumo de tiempo* tenga.

- Uso simultáneo de varias interfaces de red. Esto nos permite seguir teniendo conectividad mientras se negocia el establecimiento de otra interfaz de red. El principal responsable de esta capacidad es la extensión de movilidad de NEMO llamada *Multiple Care-of Addresses Regristration* (MCoA).

- Asistencia al traspaso mediante un protocolo específico para dicha función. IEEE 802.21 es aquí el principal protagonista. Se ha implementado parte de las entidades de 802.21 para asistir el proceso de traspaso mediante una plataforma en desarrollo llamada ODTONE [33], con la suficiente funcionalidad para llevar a cabo una asistencia basada en el nivel de fuerza de la señal recibida.

La combinación de las diferentes líneas permiten reducir o incluso eliminar los tiempos invertidos en los traspasos, consiguiendo una experiencia fluida para los usuarios.

1.4.2.5. Desarrollo de un Entorno de Pruebas Reales para Redes Móviles Vehiculares

Otra de las aportaciones es el trabajo realizado para definir y elaborar un flujo de trabajo bien definido a la hora de realizar las numerosas pruebas que se han efectuado a lo largo de este proceso investigador. Desde la planificación de las pruebas, establecimiento de las métricas, obtención de los datos, procesamiento de los mismos, hasta la generación de resultados acompañados de apoyo gráfico para su fácil comprensión y detección de anomalías. El fruto de estas buenas prácticas se verá reflejado a lo largo de la Tesis, pero con más incidencia en el **Capítulo 7**, donde se muestran los resultados de las pruebas realizadas. También cabe destacar la experiencia ganada en cuanto a desarrollo de prototipos, así como la capacidad para introducir equipamiento adecuado a los vehículos con sistemas de comunicaciones híbridos.

1.5. Trabajos relacionados

Como hemos visto en la sección de contribuciones de esta Tesis, una de ellas ha sido el desarrollo de un entorno de pruebas real en el área de las comunicaciones vehiculares V2I, siguiendo las recomendaciones de los estándares definidos para las C-ITS. Este acercamiento donde se le da preferencia a la experimentación real usando vehículos y carreteras reales es difícil de encontrar en la literatura. Esto es debido a lo duro y costoso que resulta diseñar y desplegar campos de pruebas, también llamados FOTs (*Field Operational Test*), derivando finalmente en simuladores. En este contexto podemos encontrar algunos trabajos, como el de Stahlmann et al. [34], donde presentan una plataforma de pruebas diseñada para el proyecto *DRIVE C2X*, incluyendo metodologías

de las pruebas, métricas, especificaciones del sistema e implementación, herramientas, etc. Sin embargo, este trabajo no está relacionado con el uso de IP, sino con protocolos específicos de entornos C-ITS. Un trabajo similar sobre el proyecto *simTD* fue presentado por Weib [35], pero esta vez haciendo una mención especial a la seguridad y privacidad basada en infraestructura de clave pública, de nuevo dejando el protocolo IP de lado. Sin embargo, el trabajo llevado a cabo en el proyecto ITSSv6 fue enfocado desde un principio hacia una red vehicular basada en IPv6, en el cual también se han realizado pruebas reales con la ayuda del proyecto FOTsis. Gracias en gran parte a esto, también hemos podido experimentar con la tecnología 802.11p, tecnología de acceso inalámbrico de la familia 802.11 especialmente diseñada para redes vehiculares. Ya se han dado algunos pasos probando esta tecnología, como por ejemplo en Shagdar et al. [36] (2012), donde los autores reportaron resultados similares a los presentados en esta Tesis en cuanto al comportamiento de la red en función del rendimiento, rango de cobertura, etc. También en Lin et al. [37] (2012) se describe una evaluación exhaustiva de comunicaciones V2I usando 802.11p. Al igual que la presente Tesis, estos trabajos concluyen que la calidad de la señal se ve muy influenciada por la distancia entre emisor y receptor, y también por los obstáculos del entorno. Sin embargo, la velocidad del vehículo no parece afectar tanto como cabría esperar. En Teixeira et al. [38] (2014) también se han realizado trabajos en un escenario de pruebas real, donde dos vehículos se cruzaban usando comunicación V2V. Además hacen un estudio de cómo afecta el tamaño de los paquetes al rendimiento de esta tecnología.

El desarrollo de pruebas experimentales en escenarios reales es importante, pero lo es también seguir las directrices establecidas por los organismos de estandarización, y en este caso, la arquitectura de referencia definida por el ISO/ETSI. En nuestra propuesta tenemos muy en cuenta esta circunstancia, a diferencia de las primeras propuestas que aparecieron en este segmento, como en Pinart et al. [39] (2008), donde se desarrolla un entorno de pruebas experimental para validar una solución para proveer comunicaciones vehiculares a través de un *Car Gateway*, concepto muy similar al de *Mobile Router* usado en la presente Tesis. El trabajo realizado en Cespedes et al. [40] (2011) también adolece del mismo problema y no tiene en consideración los esfuerzos de estandarización que se están llevando a cabo. Sin embargo, este último trabajo sí coincide con nosotros en destacar las bondades del uso de IPv6 en redes ITS, evaluando además el uso de NEMO como servicio de movilidad. También podemos constatar esta importancia de IPv6 en otros trabajos como [41] (2011), donde se presenta *CarGeo6*, una implementación abierta de una arquitectura de red para C-ITS basada en IPv6 y GeoNetworking [42], respetando, al igual que nosotros, la arquitectura de referencia definida por el ISO/ETSI. Sin embargo, según los propios autores, su rendimiento es mejorable.

MIPv6 [43] ha demostrado ser un protocolo válido para aportar movilidad en redes vehiculares, a pesar de no estar diseñado específicamente para este tipo de redes. Su evolución, llamada *Network Mobility* (NEMO), encaminada a dotar de movilidad a redes enteras, también fue evaluado en Tsukada et al. [44] (2010), aunque los resultados obtenidos son ligeramente peores a los nuestros. Esto es debido a que en dichas pruebas fueron usadas las tecnologías 3G y WiFi, y en el nuestro se usó una versión mejorada de

3G (3.75G) y WiFi se sustituyó por la tecnología 802.11p, más apropiada para entornos vehiculares. En Haossain et al. [45] (2012) también se evaluó el comportamiento de NEMO en un entorno real usando WiFi como tecnología. Los resultados mostrados son sorprendentemente buenos (sin perdidas durante los traspasos), que atribuimos al uso de un espacio para pruebas muy limitado y poco realista, donde un remolque era usado para mover el equipamiento que se supone debía ir dentro del vehículo. Esto queda lejos de ser una evaluación realista, como sí se ha hecho en la presente Tesis, donde se han usado carreteras y vehículos reales. El comportamiento tanto de MIPv6 como de NEMO durante los traspasos es mejorable en muchas de las contribuciones anteriores. Por ello no tardaron en aparecer en la literatura científica nuevos acercamientos basados en MIPv6 pero tratando de mejorar sus puntos débiles. Sus variantes *Proxy Mobile IP version 6* [46] (PMIPv6) y *Fast Mobile IP version 6* [47] (FMIPv6), mejoran en gran medida los tiempos de traspaso, pero no consiguen un traspaso totalmente limpio. Esto es debido a que estas mejoras están enfocadas a reducir el tiempo de latencia de los mensajes que notifican el cambio de IP. Aunque consiguen reducirla en gran medida, siempre quedará algo de latencia que no se puede eliminar. La presente Tesis sigue una línea diferente no basada en reducir dicha latencia, sino en solaparlas mediante el uso de más de una interfaz de red al mismo tiempo, lo que conlleva el uso simultáneo de varias direcciones IP, que conseguimos gracias a la extensión *Multiple Care-of Address Registration* [48] (MCoA). Los primeros estudios en esta línea fueron llevados a cabo en simuladores, como indica por ejemplo Sousa et al. [49] (2011) para el caso de OMNeT++, que constató que MCoA mejoraba el rendimiento de las aplicaciones multimedia [50] (2011). Chen et al. [51] (2009) realiza una implementación de MCoA sobre NEMO, acercamiento parecido a la propuesta de esta Tesis, pero dejando a un lado tecnologías adaptadas a las redes vehiculares como es 802.11p, y limitándose a pruebas de *Round-Trip Time* (RTT) mediante un simple *ping* cada segundo. Además, no usa IEEE 802.21 ni ningún otro mecanismo estándar para recoger información y realizar la toma de decisión a la hora de elegir una interfaz u otra. De lo mismo adolece el trabajo de Shin et al [52] (2009), que mejora el algoritmo de selección de interfaz con respecto al trabajo de Chen teniendo en cuenta también la preferencia de los usuarios, pero sigue sin usar ningún método estandar de recolección de datos y toma de decisiones. Sin embargo, Flétscher et al. [53] (2010) realiza un estudio del uso de IEEE 802.21 en VANETs, usando el simulador *Network Simulator* (NS), donde se constata la necesidad de un estándar y un sistema de políticas que permita definir tanto las bondades de cada tecnología como las prioridades del usuario, hecho que se materializa en Marquez-Barja et al. [54] (2014), donde se presenta un resumen de las diferentes técnicas de traspaso, proponiendo un algoritmo de selección de tecnología sobre el estándar IEEE 802.21 que considera las particularidades de la red, el contexto del entorno, los requisitos de las aplicaciones y las preferencias del usuario para ofrecer una experiencia de calidad en las comunicaciones vehiculares. En este último caso también se empleó el simulador NS en su versión 2, dejando de lado los trabajos en entornos reales, lo que conlleva el no poder usar nuevas tecnologías como 802.11p. Nuestro planteamiento es diferente desde su origen, partiendo de una implementación inicial y abierta de 802.21 llamada ODTONE [33], sobre el que hemos construido un

sistema de obtención de valores del entorno, como es la fuerza de la señal, y un algoritmo de decisión basado en dicha fuerza de señal. La integración de esta tecnología en nuestra propuesta es un gran paso dentro de una línea de investigación en curso, que será origen de numerosos trabajos futuros.

En cuanto a la seguridad se refiere, la solución presentada en esta Tesis donde se usa un esquema criptográfico simétrico basado en IPsec para proteger el tráfico, es poco común en entornos vehiculares, donde los trabajos suelen centrarse más en aportar seguridad y privacidad a nivel de mensaje usando esquemas criptográficos asimétricos o de clave pública. Tanto es así que en trabajos como Raya et al. [55] (2006) y más recientemente Bassem Mokhtar et al. [56] (2015), se hacen estudios de las vulnerabilidades de las redes vehiculares y las posibles soluciones que se han propuesto hasta la fecha por distintos autores, pero no se ocupan del tema de la eficiencia a la hora de enviar grandes flujos de información, como puede ser el necesario para transmitir contenido multimedia. Como hemos adelantado antes, en nuestra propuesta se presenta IPsec con la ayuda de IKEv2 para el establecimiento de sesiones, ya que usa esquemas criptográficos simétricos más rápidos para proteger las transmisiones.

Los servicios de seguridad y movilidad han sido extensamente estudiados en la literatura, pero desplegarlos a la vez conlleva problemas que pocos autores han cubierto. Sin embargo, hay algunos trabajos que, al igual que nosotros, han detectado la necesidad de establecer un canal de comunicación entre ambas implementaciones de los servicios de movilidad y seguridad. Sugimoto et al. [57] (2006) estableció una interfaz de comunicación entre MIPv6 e IKEv2 que llamó *MIGRATE*. Está basado en PF_KEY [58], interfaz estándar para comunicarse con IPsec normalmente localizado en el núcleo del sistema operativo. Ambos servicios, seguridad y movilidad, deben implementar esta interfaz para estar pendiente de los cambios de direccionamiento que se puedan producir y hacer los cambios oportunos. Xu et al. [59] (2008) van un paso más allá y crean una base de datos intermedia llamada *Mobile Security Reference Database* (MSRD), a la cual ambos servicios acceden a través de PF_KEY, con una entrada por cada dirección IP usada. En nuestra propuesta se ha seguido un acercamiento diferente, evitando pasar por las interioridades del sistema operativo y realizando dicha comunicación entre los servicios de movilidad y seguridad de forma directa. En definitiva se ha visto que el problema es que el servicio de seguridad debería tener acceso a la *Binging Caché* donde quedan establecidas las direcciones IP que estamos usando en cada momento en la movilidad. Por este motivo, se ha implementado una vía directa mediante comunicación entre procesos para que IKEv2 y NEMO estén comunicados, resolviendo el problema que existe entre ambos servicios sin pasar por el núcleo del sistema operativo. Además, en nuestra propuesta se da un paso más allá para dar soporte al uso simultáneo de varias direcciones IP (MCoA), además de hacer un pequeño análisis entre varias alternativas de comunicación inter-proceso, haciendo énfasis en la correcta distribución de las responsabilidades entre dichos servicios.

1.6. Organización del Documento

La presente Tesis doctoral se encuentra dividida en un conjunto de capítulos que siguen una secuencia lógica de un proceso de investigación científica de rama técnica, presentando en primer lugar las tecnologías que van a formar parte de nuestra solución, desarrollando después la solución en sí, y terminando con las pruebas realizadas y sus resultados, que nos llevan directamente a las conclusiones. De esta forma, el **Capítulo 1** (dentro del cual está la presente sección) realiza una contextualización del trabajo presentado encuadrándolo dentro del mundo de los sistemas inteligentes de transporte (ITS).Se hace un recorrido a través de otros trabajos relacionados con el presentado aquí para dar una idea del estado del arte en las diferentes líneas que sigue este trabajo. Además, se marca los objetivos y las aportaciones realizadas, sirviendo como esquema estructurador de la Tesis, facilitando así su lectura y manteniendo al lector contextualizado en todo momento.

El **Capítulo 2** está destinado a presentar la base tecnológica necesaria para entender correctamente el grueso de la Tesis doctoral. Para tal fin, se presenta la arquitectura de referencia ISO/ETSI, los tipos de estaciones que identificamos y cómo se conforma la pila de comunicaciones en cada una de ellas. También se exponen las tecnologías inalámbricas que serán incluidas en nuestra propuesta, así como tecnologías de soporte relacionadas con los traspasos, tan presentes en estos entornos vehiculares. También se exponen los protocolos y servicios utilizados, dando gran protagonismo a IPv6 y sus tecnologías asociadas, como aquellas en los ámbitos de la seguridad y la movilidad de redes.

En el **Capítulo 3** desarrollamos nuestra propuesta de arquitectura de red, especificando en detalle nuestras estaciones ITS y qué protocolos se incluyen en los distintos niveles, dependiendo de en qué servicios participen. Se trata pues de unir las diferentes piezas de una arquitectura de red que siga las pautas de estandarización establecidas por las organizaciones ISO y ETSI, pero incorporando protocolos de comunicación de Internet, explotando la sinergia entre la estandarización ISO/ETSI e IETF. Además, dichas pautas nos ayudarán a situar los distintos módulos software donde corresponda, estableciendo así cómo se comunicarán dichos módulos entre sí de forma estandarizada.

En el **Capítulo 4** se detalla la experiencia de crear una nueva implementación de IKEv2 desde cero que, junto a IPsec, conforman el servicio de seguridad que incluimos en nuestra propuesta. Además, se expone una comparativa con otras soluciones IKEv2 que también han ido apareciendo a lo largo de los años. También hacemos especial mención a las aportaciones que realizamos para completar el estándar mediante las incidencias detectadas en el proceso de implementación que nos llevaron a estados indefinidos y no contemplados.

En el **Capítulo 5** desarrollamos la problemática de integrar los servicios de movilidad y seguridad. Exponemos las diferentes alternativas para llegar a una solución, valorándolas y eligiendo una de ellas para incorporarla a nuestra propuesta. También, a la hora de llevar a cabo esta integración, aparecieron detalles que los estándares

no contemplaban y, por tanto, tuvimos que realizar modificaciones a nivel de diseño que se materializaron en una implementación funcional, siendo esta una de nuestras principales aportaciones a esta Tesis.

En el **Capítulo 6** nos centramos en el proceso de traspaso entre tecnologías, proceso muy frecuente en entornos vehiculares, tratando de identificar las posibles vías de mejora, haciendo que el tiempo invertido en dicho proceso sea mínimo e incluso nulo. Esto no habría sido posible sin identificar primero las etapas de las que se compone un traspaso y cuáles de ellas son susceptibles de mejora.

En el **Capítulo 7** se desarrolla todo el plan de pruebas que hemos llevado a cabo sobre nuestra propuesta de arquitectura, teniendo especial cuidado de establecer una metodología rigurosa y exhaustiva que nos proporcione resultados confiables que sean extrapolables a escenarios de mayor envergadura. Para ello, se han definido las métricas, los procedimientos a seguir, el procesado de los datos obtenidos, el despliegue de la red, el equipamiento a utilizar, qué pruebas realizar y en qué orden y, por último, se exponen los resultados de donde se extraen conclusiones que nos permiten validar nuestra propuesta en términos de corrección y rendimiento, además de ayudar a identificar puntos débiles, localizar la causa y poder proponer soluciones.

Finalmente, en el **Capítulo 8** se exponen las conclusiones generales que hemos extraído en el desarrollo de la Tesis y de su valoración global, identificando además las vías futuras de investigación tanto de trabajo directo a partir de lo realizado en la Tesis, como tendencias futuras.

1.7. Publicaciones Derivadas de la Tesis Doctoral

Durante el periodo en el que se han realizado los trabajos de investigación en la Universidad de Murcia conducentes a la obtención del título de Doctor, se han realizado diversas aportaciones científicas que son resumidas en la presente sección. Éstas se organizan en función de su medio de divulgación: artículos en revistas, capítulos en libros y aportaciones en congresos del área.

1.7.1. Artículos en Revistas

- Alejandro Pérez-Méndez, Pedro J. Fernández Ruiz, Rafael Marín López, Gregorio Martínez Pérez, Antonio F. Gómez Skarmeta, Kenichi Taniuchi. **OpenIKEv2: Design and Implementation of an IKEv2 Solution** *IEICE Transactions* pp. 1319-1329, DOI: 10.1093/ietisy/e91-d.5.1319, 2008 **(Impacto: 0.396/Q4/JCR-2008)**.

  En este artículo [60] se expuso como se había llevado a cabo la implementación del protocolo IKEv2, implementación a la que se llamó OpenIKEv2 precisamente por ser de código abierto. Se introduce de lleno en la ingeniería del software empleada para obtener una implementación escalable, sencilla y fácilmente portable a otras plataformas. Por eso se hizo en forma de librería. Esta implementación de IKEv2

sería de mucha utilidad en los futuros trabajos y proyectos de investigación, siendo el germen de nuestra aportación de seguridad sobre IPv6 en redes vehiculares.

- Pedro J. Fernández Ruiz, Fernando Bernal Hidalgo, Cristian A. Nieto Guerra, Antonio F. Gómez Skarmeta, **Mobility and security in a real VANET deployed in a heterogeneous networks** - *Security and Communication Networks*, John Wiley & Sons Ltd, DOI:10.1002/sec.518, 2012. **(Impacto: 0.311/Q4/JCR-2012)**.

  En este trabajo [61] se presentan numerosas líneas de investigación: en primer lugar se muestra un despliegue de una red vehicular real basada en las tecnologías WiFi y WiMAX, dentro del Campus de Espinardo de la Universidad de Murcia. En segundo lugar, se establecen las bases para la integración de los servicios de movilidad y seguridad, definiendo dos estrategias diferentes para acometerlo, haciendo una comparativa entre ambos casos. En tercer lugar se hace una introducción a las posibles formas de mejorar el rendimiento de los traspasos, destacando los diferentes métodos de autenticación usando EAP, el protocolo IEEE 802.21 de asistencia al traspaso y la extensión de movilidad MCoA que permite el uso simultáneo de interfaces de red, todo ello de forma conceptual.

- José Santa, Fernando Perenéguez-Garcia, Fernando Bernal Hidalgo, Pedro J. Fernández, Rafael Marin-Lopez, Antonio F. Gómez Skarmeta, **A Framework for Supporting Network Continuity in Vehicular IPv6 Communications** - *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 17-34, ISSN:1939-1390, DOI:10.1109/MITS.2013.2274876, 2014. **(Impacto: 0.821/Q3/JCR-2014)**.

  El trabajo presentado en este artículo [62] introduce las bases de la que será nuestra propuesta de arquitectura de comunicaciones basada en los estándares definidos para entornos ITS por los organismos de estandarización ISO y ETSI. También aporta una solución completa basada en IEEE 802.21 para la asistencia al traspaso entre tecnologías en entornos ITS, aunque solamente de forma teórica. Se muestran también los resultados de las pruebas de rendimiento efectuadas usando las tecnologías 3G y WiFi.

- José Santa, Fernando Pereñíguez, Antonio Moragón, Pedro J. Fernández, Fernando Bernal, Antonio F. Gómez Skarmeta. **IPv6 Communication Stack for Deploying Cooperative Vehicular Services** - *International Journal of Intelligent Transportation System Research*, vol. 12, no. 2, pp. 48-60, Springer, DOI: 10.1007/s13177-013-0068-6, 2014. **(Impacto: 1.377/Q3/SJR-2014)**.

  Este artículo [63] continúa refinando lo que será la arquitectura de comunicaciones propuesta en la presente Tesis, definiendo las entidades que finalmente formarán parte de la arquitectura. Se hace mención al uso de *IP Multimedia Subsystem* (IMS), un entorno de trabajo común para la provisión de servicios multimedia, que puede encajar también en las redes ITS. También se destaca el trabajo realizado en proyectos como OASIS [64], donde se ha usado otro entorno, esta

vez de aplicaciones, llamado OSGi, con el que se implementan facilidades en Java que permiten la implementación de servicios basados en la posición geográfica, como son el rastreo de vehículos y la diseminación de mensajes. Además, en este trabajo se realizan unas primeras pruebas de rendimiento, esta vez usando las tecnologías 3G y 802.11p.

- Pedro J. Fernández, Jose Santa, Fernando Bernal, Antonio F. Gómez Skarmeta, **Securing Vehicular IPv6 Communications** - *IEEE Transactions on Dependable and Secure Computing* DOI: 10.1109/TDSC.2015.2399300, 2015. **(Impacto: 1.351/Q2/JCR-2014)**.

  Este artículo [65] es fundamental para la presente Tesis. Sigue las mismas líneas que los artículos anteriores, estableciendo una propuesta de arquitectura para ITS, y refinando la integración de servicios de movilidad y seguridad, proponiendo soluciones a lagunas de implementación no tenidos en cuenta por los estándares. Además, se introduce el concepto de flujo de paquetes y se establece una novedosa estrategia de controlar el tráfico a partir de la definición de este concepto, estableciendo una serie de reglas para cada flujo que determinan qué interfaz usará cada flujo y qué tratamiento se le harán a sus paquetes. También aporta una evaluación experimental, más ambiciosa que las anteriores, en un escenario de movilidad usando las tecnologías 3G y 802.11p. Los resultados fueron determinantes para marcar la dirección a seguir y mejorar, por ejemplo, los tiempos y la calidad de los traspasos.

- Jose Santa, Pedro J. Fernández, Fernando Pereñiguez, Antonio F. Gomez Skarmeta, **Real Experience with IPv6 Communications in Highways** - *Journal of Wireless Mobile Networks, Ubiquitous Computing and Dependable Applications (JoWUA)* Voulmen 6, Número 3, Sep 2015

  En el articulo [66] se recoge las experiencias vividas en las pruebas realizadas en la A2, a la altura de Madinaceli (Provincia de Sória), y se exponen interesantes resultados. Estas pruebas estan enmarcadas dentro del proyecto FOTsis [20]. Los resultados arrojan la necesidad de incorporar algún mecanismo que mejore el comportamiento de los traspasos. Como sugieren otros autores en trabajos anteriores, IEEE 802.21 es una alternativa válida en la que estamos empezando a trabajar activamente.

- José Santa, Pedro J. Fernández, Fernando Pereñéguez, Antonio F. Gómez Skarmeta, **Deployment of Vehicular Networks in Highways using 802.11p and IPv6 Technologies** - *International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC)* DOI: To Be Published, 2015. **(Impacto: 0.554/Q4/JCR-2014)**.

  En este articulo [67] se continua la línea de pruebas sobre la arquitectura propuesta a lo largo de trabajos anteriores, esta vez con el apoyo del proyecto FOTsis [20] que nos permite tener acceso a entornos de pruebas reales en distintos puntos de la geografía europea. Se exponen los resultados de dichas pruebas,

extrayendo conclusiones que nos ayudarán a mejorar el comportamiento de la arquitectura en general. También se recogen experiencias vividas a lo largo de todas estas pruebas que tenemos por seguro que serán de gran ayuda a futuros investigadores que sigan por esta línea.

1.7.2. Capítulos en Libros

- Pedro J. Fernández, Cristian A. Nieto, José Santa, Antonio F. Gómez-Skarmeta, Johann Márquez-Barja, Pietro Manzoni. **Experience Developing a Vehicular Network Based on Heterogeneous Communication Technologies, Wireless Technologies in Vehicular Ad Hoc Networks: Present and Future Challenges** (Raul Aquino Santos, Arthur Edwards, Victor Rangel Licea, eds.) - *Wireless Technologies in Vehicular Ad Hoc Networks: Present and Future Challenges*, Editorial IGI Global, Capítulo 14, DOI: 10.4018/978-1-4666-0209-0.ch014, 2012.

  En este primer capítulo [68] aportado a la Editorial IGI Global se realiza una revisión de las tecnologías inalámbricas vehiculares, tales como WiMAX y WiFi, y cómo hacer traspasos entre una y otra, contemplando IEEE 802.21 como estándar apropiado para asistir dichos traspasos, aunque en esta ocasión sólo de forma teórica. También se enumeran tecnologías IPv6 como son la movilidad y la seguridad, donde se empiezan a ver los primeros pasos hacia la integración de los mismos en redes vehiculares. También se hace mención a una posible solución en la unidad de abordo (OBU) para disponer de una interfaz de red virtual que enmascare las físicas, pero sólo de forma conceptual.

- Pedro Javier Fernández Ruiz, Fernando Bernal Hidalgo, José Santa Lozano, Antonio F. Skarmeta. **Deploying ITS Scenarios Providing Security and Mobility Services Based on IEEE 802.11p Technology, Vehicular Technologies - Deployment and Applications** (Lorenzo Galati Giordano and Luca Reggiani, eds.). - *Vehicular Technologies - Deployment and Applications*, Editorial InTech, Capítulo 4, DOI 10.5772/55285, 2013.

  En este segundo capítulo [69] se observa claramente que sigue la dirección establecida en el capítulo anterior, apareciendo numerosos avances significativos. En primer lugar, la integración de los servicios de movilidad y seguridad ya se establecen de forma más acorde con los estándares establecidos. En segundo lugar, se empieza a contemplar la posibilidad de llevar nuestra propuesta basada en IPv6 a entornos móviles vehiculares ITS, siguiendo la arquitectura de referencia establecida por los organismos de estandarización ISO/ETSI. En tercer y último lugar, aparece una nueva tecnología específica para vehículos llamada 802.11p, que junto con la ya establecida 3G, compondrían nuestras dos principales tecnologías a usar en un entorno ITS. Sin embargo, la tecnología WiFi se deja para establecer una red interna en cada vehículo.

- Fernando Pereñiguez, José Santa, Pedro Javier Fernández Ruiz, Fernando

Bernal Hidalgo, Antonio F. Skarmeta, Thierry Ernst. **The Use of IPv6 in Cooperative ITS: Standarization Viewpoint** (C. Campolo et al., eds.). - *Vehicular Ad-Hoc Networks*, Editorial Springer, Capítulo 9, DOI 10.1007/978-3-319-15497-8_9, 2015.

El libro que contiene a este capítulo es una referencia bibliográfica destacable en la literatura ITS, ya que sus capítulos han sido escritos por los investigadores más relevantes en el área. Nuestro equipo de investigación también puso su grano de arena aportando este capítulo [70], que se centra en ver cómo IPv6 y sus servicios asociados (movilidad y seguridad) pueden ser integrados en sistemas cooperativos ITS, un nuevo concepto que está revolucionando los estándares en entornos vehiculares ITS. Se hace mención a los diferentes roles que juegan las diferentes entidades necesarias para desplegar una red cooperativa ITS que siga, además, la arquitectura de referencia establecida por ISO y ETSI, dejando clara la función de cada capa y plano de dicha arquitectura. Este capítulo se extiende también por los servicios típicos que se pueden desplegar en este tipo de redes, como el acceso a Internet o servicios de eficiencia del tráfico y seguridad vial. En último lugar se muestran los resultados de unas pruebas de referencia sobre el rendimiento efectuadas sobre las tecnologías 802.11p y 3G, y el proceso de traspaso entre ellas.

1.7.3. Congresos

- Pedro J. Fernandez Ruiz, Cristian A. Nieto Guerra, Antonio F. Gómez-Skarmeta. **Deployment of a Secure Wireless Infrastructure Oriented to Vehicular Networks** - *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference, Perth, Australia*, IEEE; p1108-1114, ISSN:1550-445X, ISBN:978-1-4244-6695-5, DOI: 10.1109/AINA.2010.177, 2010

  El trabajo presentado en este artículo [71] se puede considerar como el primer paso hacia la consecución de la presente Tesis, pues se establece en él el interés de los autores por determinadas líneas de investigación bien definidas, como las redes vehiculares y los servicios de seguridad y movilidad. En él se comenta el interés creciente por las redes vehiculares y la necesidad de incorporar a dichas redes servicios de movilidad y seguridad. Se crea un escenario usando tecnología WiMAX, del cual se comenta su despliegue. También se aborda otra línea de investigación relacionada con las unidades de abordo (OBU) y sus interfaces de red. Se exponen los principios de una propuesta donde se establece una interfaz única virtual que use la interfaz física que en cada momento sea conveniente.

- Pedro J. Fernández Ruiz, Cristian A. Nieto Guerra, Antonio F. Gómez Skarmeta. **Autenticación basada en IKEv2 y EAP para Escenarios de Redes Vehiculares** - *IX Jornadas de Ingeniería Telemática JITEL, Valladolid, España*, ISBN: 978-84-693-5398-1, 2010

  Este trabajo [72] fue el primero que se presentó en castellano y en un congreso

nacional. Sigue las mismas líneas que en el trabajo anterior, pero con algunos progresos, como el reportado en el servicio de seguridad. Se abre el abanico de alternativas para llegar a un mismo objetivo, proteger el tráfico de movilidad mediante IPsec e IKEv2. Se realiza un estudio de las encontradas y se valoran para seleccionar una con la que seguir desarrollando nuestra arquitectura de red.

- Pedro J. Fernandez, Antonio F. Gomez-Skarmeta. **Providing Security using IKEv2 in a Vehicular Network based on WiMAX Technology** - *3rd IEEE Intelligent Vehicular Communications System Workshop/Consumer Communication and Networking Conference (IVCS/CCNC), Las Vegas, Nevada, USA*, ISBN: 978-1-4244-8788-2, IEEE, p1001-1006, 2011

  Este artículo [73] presenta un salto cualitativo en las líneas investigadas hasta la fecha. En primer lugar, se introduce la tecnología WiFi al escenario vehicular donde antes sólo había WiMAX, lo que permite estudiar otros tipos de traspasos. En segundo lugar, se profundiza en la integración de los servicios de seguridad y movilidad, exponiendo las diferentes alternativas. Por último, se identifican las formas de mejorar un traspaso. Hasta la fecha sólo contemplábamos reducir los tiempos de autenticación o usar IEEE 802.21 para asistir el traspaso. A estas se le añade una más llamada MCoA, extensión de la movilidad que permite el uso simultáneo de varias interfaces de red.

- Pedro J. Fernandez, Cristian A. Nieto, Antonio F. Gomez-Skarmeta **Seguridad y movilidad en una VANET real desplegada con diferentes tecnologías inalámbricas** - *X Jornadas de Ingeniería Telemática JITEL, Santander, España*, ISBN: 978-84-694-5948-5, p128-135., 2011

  Con este trabajo [74] se participa de nuevo en otro congreso nacional, donde se siguen las mismas líneas que en trabajos anteriores, pero éste especialmente centrado en la autenticación y cómo reducir sus tiempos para mejorar el traspaso de una estación WiMAX a otra. A pesar de que el escenario sólo contempla WiMAX como tecnología, se empiezan a proponer tecnologías como WiFi, 3G o la específica para redes vehiculares llamada 802.11p, aunque esto sólo de forma conceptual por el momento.

- José Santa, Manuel Mora, Antonio Moragón, Andrés S. García, Pedro J. Fernández, Fernando Bernal, Antonio F. Skarmeta, Jaime Arrazola. **Arquitectura de comunicaciones en OASIS: desarrollo de una pila de comunicación ITS siguiendo los conceptos de CALM y Arquitectura Europea de Comunicación ITS** - *XII Congreso Español sobre ITS. Madrid, España*, 2012

  El trabajo presentado en este artículo [75] introduce las bases de la que será nuestra propuesta de arquitectura de comunicaciones basada en los estándares definidos para entornos ITS por los organismos de estandarización ISO y ETSI. Se muestra un escenario desplegado con tecnologías WiFi y 3G, esta última como novedad, en donde ya se hace distinción de las diferentes estaciones ITS.

- José Santa, Pedro J. Fernández, Antonio Moragón, Andrés S. García, Fernando Bernal, Antonio F. Gómez-Skarmeta. **Architecture and development of a networking stack for secure and continuous service access in vehicular environments** - *19th ITS World Congress (ITSWC 2012). Viena, Austria*, 2012

  El trabajo [76] sigue las líneas presentadas en el artículo anterior. Podemos destacar la mención a la capa software intermedia utilizada para posibilitar el acceso a servicios multimedia basados en IMS, solución introducida en el proyecto OASIS [64], trabajo más centrado en la capa de facilidades.

- José Santa, Fernando Bernal, Pedro J. Fernández, Antonio Moragón, Andrés S. García, Antonio F. Gómez-Skarmeta. **Continuous IPv6 Communications in a Vehicular Networking Stack for Current and Future ITS Services** - *First International Workshop on IPv6-based Vehicular Networks (Vehi6) 2012 - IEEE Intelligent Vehicles Symposium (IV'2012). Alcalá de Henares, Spain*, 2012

  El trabajo [77] reafirma las bases de la que será nuestra propuesta de arquitectura de comunicaciones basada en los estándares definidos para entornos ITS por los organismos de estandarización ISO y ETSI. Se muestra además un escenario desplegado con tecnologías WiFi, WiMAX y 3G, en donde ya se hace distinción de las diferentes estaciones ITS. También aporta un principio de solución basada en IEEE 802.21 para la asistencia al traspaso entre tecnologías en entornos ITS, aunque solamente de forma teórica.

- José Santa, Pedro J. Fernández, Fernando Pereñiguez, Fernando Bernal, Antonio F. Gómez-Skarmeta. **A Vehicle Network Mobility Framework: Architecture, Deployment and Evaluation** - *IEEE INFOCOM International Workshop on Mobility Management in the Networks of the Future World (MobiWorld 2015). Hong Kong, China*, 2015

  Este artículo [78], la idea de seguir una arquitectura de referencia ISO/ETSI está más madura y refleja mayor claridad a la hora de localizar los diferentes módulos en dicha arquitectura. Para los escenarios de pruebas se introduce definitivamente la tecnología 802.11p, más apropiada para entornos ITS. Se aborda el despliegue de un entorno de pruebas real en uno de los puntos de prueba disponibles en el proyecto FOTsis [20]. En concreto fue el localizado en la A2 entre los puntos kilométricos 139 y 146, que resultaron en experiencias valiosas que recogimos para que otros investigadores puedan tenerlas en cuenta.

Capítulo 2

Estándares y Tecnologías basadas en IPv6 para entornos ITS

En este capítulo enumera las diferentes tecnologias, estándares y protocolos que vamos a utilizar como piezas para la creación de nuestra propuesta de red vehicular. El origen de éstas es dispar, ya que disponemos desde una arquitectura de referencia específica para redes vehiculares de la mano de organismos como ISO y ETSI, hasta protocolos y servicios procedentes del mundo de Internet, al amparo del organismo IETF, pasando por las tecnologías inalámbricas procedentes de 3GPP e IEEE, entre otros. Se trata aquí de que el lector adquiera un conocimiento resumido pero suficiente de dichas tecnologías para, de esta manera, ayudar en la comprensión de los siguientes capítulos.

2.1. Arquitectura de Referencia ISO/ETSI

En una apuesta clara hacia la armonización de las distintas arquitecturas de red, la comunidad internacional ITS ha definido una arquitectura de comunicaciones de referencia común para todos que satisfaga las necesidades de los posibles escenarios ITS. Para ello se han incluido gran variedad de tecnologías inalámbricas (802.11p, infrarrojos, 2G/3G/4G, satélite, etc.) pensadas para un conjunto de aplicaciones orientadas a entornos móviles (seguridad vial, eficiencia del tráfico, comodidad y entretenimiento). En la definición de estos estándares intervienen organismos de diferentes continentes y países, cada uno con regulaciones y políticas diferentes.

En lo que concierne a Europa, las especificaciones de esta arquitectura de referencia han sido definidas, como hemos visto en el capítulo introductorio, por el ISO en el documento ISO_21217 [8] y por el ETSI en el documento ETSI_EN_302_665 [9]. Se le denominó *ITS station reference architecture* (arquitectura de referencia para estaciones ITS). Como puede apreciarse en la Figura 2.1, esta arquitectura de referencia sigue un modelo similar al modelo de capas definido por el organismo de estandarización *Open Systems Interconnection* (OSI): capa de acceso, de red y transporte, de facilidades y de aplicaciones. Estas capas están acompañadas por dos planos verticales adicionales

para las tareas de gestión y seguridad, como podemos ver simplificado en la Figura 2.1. Todos estos bloques funcionales están interconectados a través de los *Service Access Points* (SAP), o Puntos de Acceso al Servicio, que permiten intercambiar información entre entidades de diferentes capas.

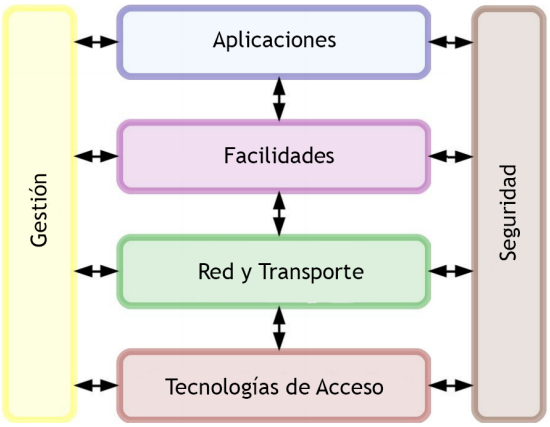


Figura 2.1: Visión simplificada de la pila de comunicaciones de referencia del ISO/ETSI

2.1.1. Tipos de estaciones ITS

Existen diferentes tipos de estaciones ITS (ITS-S):

- *Vehicle ITS-S* (Estación Vehicular ITS), que representa a todo tipo de vehículos, como pueden ser los coches, camiones, etc.
- *Roadside ITS-S* (Estación de Carretera ITS), representando a la infraestructura necesaria para comunicarse con los vehículos, como pueden ser los puestos de peaje, puntos de acceso de la carretera, etc.
- *Central ITS-S* (Estación Central ITS), que normalmente representa al centro de control del operador de la carretera.

Cada uno de los tipos de estación ITS implementa un subconjunto de funcionalidades de todas las posibles reflejadas en la estación de referencia ITS genérica. Este subconjunto se establece acorde con el rol que juegue en cada caso. Por ejemplo, en el caso de la Estación Vehicular ITS se requiere un conjunto más amplio de tipos de tecnologías, especialmente de naturaleza inalámbrica. Aparte de la tecnología 802.11b/g para conectar con los dispositivos de abordo, necesita de tecnologías inalámbricas como 802.11p para conectar con las Estaciones de Carretera ITS, y 3G para conectar directamente con la Estación Central ITS. Sin embargo, la Estación Central ITS sólo requiere de tecnologías de naturaleza cableada con 802.3 para conectarse a las Estaciones de Carretera ITS y a Internet.

En la mayoría de los casos, la funcionalidad de una Estación ITS está dividida entre un *router* y un *host*. El *router* es un nodo dedicado a funciones de enrutamiento de paquetes normalmente usado para conectar dos redes y reenviar paquetes que no vayan dirigidas a su Estación ITS. En contraste, el *host* es un nodo encargado de ejecutar aplicaciones específicas para ITS. Estas dos entidades están conectadas entre sí mediante una red interna de la propia Estación ITS. Sin embargo, en algunas ocasiones, ambas funcionalidades podrían aparecer mezcladas en una sola entidad.

2.1.2. Capas de la Pila de la Arquitectura de Referencia

Como hemos adelantado antes y contemplado en la Figura 2.1, la pila de comunicaciones usada como referencia para cada nodo de una Estación ITS sigue el esquema de capas ISO/ETSI que considera, de abajo a arriba, las siguientes capas: tecnologías de acceso, red y transporte, facilidades y aplicaciones. Estas capas están acompañadas por dos planos transversales para las tareas de gestión y seguridad, accesibles desde cualquiera de las otras capas.

2.1.2.1. Capa de Acceso

La capa de tecnologías de acceso incluye una gran variedad de tecnologías de telecomunicaciones, en su mayoría de naturaleza inalámbrica (WiFi, WiMAX, 802.11p, 3G), que nos permiten comunicarnos con otras estaciones ITS. Estas tecnologías pueden usarse simultáneamente y varían dependiendo del cometido de cada tipo de estación ITS. Independientemente de las tecnologías soportadas por el estándar, la incorporación de nuevas tecnologías no supondrá un impacto en el resto de la arquitectura gracias a la estratificación por capas.

2.1.2.2. Capa de Red y Transporte

La capa de red y transporte agrupa a todos los protocolos de red y transporte encargados de transmitir los mensajes que se intercambian los diferentes nodos ITS. Aunque ya existen protocolos como FNTP (*Fast Networking and Transport layer Protocol*) especificado por el grupo de trabajo ISO TC204 en el documento ISO 21210 [79,80], o *GeoNetworking* [11,81], mencionado en el capítulo anterior, el estándar es suficientemente flexible para aceptar otros protocolos de red y transporte en el futuro. De hecho, IPv6 está tomando cada vez más fuerza dentro de los grupos de estandarización, que trabajan para poder aprovechar todas las ventajas que puede ofrecer a las redes C-ITS.

2.1.2.3. Capa de Facilidades

La capa de facilidades hace de intermediario entre la capa de red y transporte y la capa de aplicaciones, ofreciendo a estas capas acceso a información intercambiada entre las distintas estaciones ITS y liberando a las aplicaciones de realizar esta tarea. El beneficio inmediato es el poder compartir dicha información con todas las aplicaciones.

El intercambio de esta información debe realizarse de forma estandarizada. Ejemplos de facilidades son el *Cooperative Awareness Message* (CAM) [82] y el *Decentralized Environmental Notification Basic Service* (DENM) [83]. Con estos servicios la estación ITS puede emitir o recibir información de interés para ella y para las estaciones que la rodean (mensajes CAM) y también notificar eventos de interés para la seguridad y la eficiencia del tráfico basadas en la posición geográfica de las Estaciones Vehiculares ITS (mensajes DENM).

2.1.2.4. Capa de Aplicaciones

En la capa de aplicación residen las aplicaciones ITS, todas aquellas que ayuden a conseguir los objetivos de aportar seguridad, eficiencia del tráfico e informar y ofrecer servicios de valor añadido a los ocupantes de los vehículos. Estas aplicaciones pueden hacer uso de los servicios ofrecidos por la capa de facilidades.

2.1.2.5. Planos de Gestión y Seguridad

Como hemos adelantado antes, estas capas están rodeadas por dos capas verticales, lo que quiere decir que pueden ser accedidas desde cualquiera de las capas horizontales mencionadas. Una de las capas verticales se dedica a tareas de gestión, encargada de funciones como la selección de la mejor interfaz al transmitir datos, basándose para ello en los requisitos de las aplicaciones, las características de las tecnologías de acceso y de las condiciones de la red. La segunda capa vertical desempeña tareas de seguridad, encargada de proporcionar funcionalidades atómicas tales como la generación de números pseudo-aleatorios, resumen digital, firma digital, cifrado y descifrado a las capas anteriores, pudiendo actuar además de repositorio de credenciales, certificados y material criptográfico.

2.2. Tecnologías Inalámbricas Apropiadas para Entornos Vehiculares

En los últimos años se han desarrollado múltiples tecnologías de comunicación gracias a la popularización de los terminales móviles. Estos terminales, en su mayoría *Smartphones*, tienen requisitos similares a los que podría tener un *ordenador de abordo* de un vehículo. Por tanto podría usar las mismas tecnologías en principio diseñadas para terminales móviles, como pueden ser GSM, UMTS, o el más reciente LTE. A pesar de ello, los institutos de estandarización como el IEEE han estado trabajando en tecnologías que se adapten a los requisitos específicos que tienen las redes vehiculares, apareciendo así la tecnología IEEE 802.11p. Podríamos decir que se trata de una variante más del estándar WiFi, pero especialmente adaptada a las redes vehiculares.

Independientemente de lo anterior, la tendencia en dispositivos móviles y unidades de abordo es a la hibridación de las comunicaciones, es decir, disponer del máximo número de tecnologías inalámbricas, teniendo al menos un interfaz por tecnología.

Esto permite que el terminal esté conectado siempre a través de la mejor tecnología atendiendo a condicionantes como son el ancho de banda, precio de la transmisión, ocupación del canal, etc.

Tecnología	Tasa de subida	Tasa de bajada
2.5G / GPRS	20 Kbps	114 Kbps
2.5G+ / EDGE	60 Kbps	384 Kbps
3G / UMTS	1.8 Mbps	3.1 Mbps
3.5G / HSDPA	1.8 Mbps	14.4 Mbps
3.5G+ / USUPA	5.8 Mbps	14.4 Mbps
3.9G / HSPA+	22 Mbps	84 Mbps
Pre-4G / LTE	50 Mbps	100 Mbps
4G / LTE Advanced	60 Mbps	+100 Mbps

Tabla 2.1: Distintas tasas de transferencia teóricas de las tecnologías inalámbricas dependientes del 3GPP

2.2.1. Tecnologías 3GPP

El *3rd Generation Partnership Project* (3GPP), es una colaboración de grupos de asociaciones de telecomunicaciones, conocidos como Miembros Organizativos, donde el ETSI es uno de ellos. Su objetivo inicial era asentar las especificaciones de un sistema global de comunicaciones de tercera generación (3G) para móviles basándose en las especificaciones del sistema evolucionado *Global System for Mobile Communications* (GSM) dentro del marco del proyecto internacional de telecomunicaciones móviles 2000 del *International Telecommunication Union* (ITU).

3G es la abreviatura de tercera generación de transmisión de voz y datos a través de telefonía móvil mediante *Universal Mobile Telecommunications System* (UMTS) que está gestionado por la organización 3GPP. Ésta ha ido incrementando las velocidades de transferencia máximas, apareciendo la tecnología *High Speed Downlink Packet Access* (HSDPA), también denominada 3.5G, que mejora significativamente la capacidad máxima de transferencia de información, pudiéndose alcanzar tasas de bajada de hasta 14.4 Mbps. Después apareció la tecnología HSUPA, también llamada 3.5G+, con velocidades de subida de hasta 5,8 Mbps, y finalmente HSPA+ (3.9G) con velocidades de hasta 84 Mbps de bajada y 22 Mbps en la subida. En la Tabla 2.1 aparecen resumidas las distintas tasas de transferencia de las diferentes tecnologías que han ido apareciendo a lo largo de los años. Como podemos ver, la siguiente generación de tecnologías 4G está empezando a asentarse, sobre todo en núcleos urbanos, con velocidades muy superiores a sus generaciones predecesoras.

2.2.2. IEEE 802.11p y ETSI G5

Para dar soporte a las comunicaciones de corta distancia en entornos ITS, o también llamadas *Dedicated Short Range Communication* (DSRC), y debido a la ausencia de interfaces homogéneas de comunicaciones entre los distintos fabricantes de vehículos, se definió la familia de estándares IEEE 1609 [16], también llamada *Wireless Access for Vehicular Environments* (WAVE), que es el fruto de las investigaciones llevadas a cabo principalmente por Estados Unidos, Europa y Japón. Cubren la definición de la arquitectura de comunicaciones, la definición y formato de mensajes para diferentes servicios de datos y gestión, los servicios de seguridad aplicables sobre dichos mensajes, entre otros. En los inicios del desarrollo de DSRC, la tecnología utilizada para este fín se basaba en la variante del estándar WiFi IEEE 802.11a. Sin embargo, para dar soporte a comunicaciones en movimiento a altas velocidades, lo que es muy común en entornos vehiculares, IEEE se basó en IEEE 802.11p para definir las capas físicas y de acceso (MAC) de las comunicaciones dentro del marco de WAVE, que se trata de la adaptación del extendido y afamado IEEE 802.11 a entornos vehiculares. Una de las aportaciones más importantes de esta adaptación es la eliminación del intercambio de tramas necesario para el establecimiento del canal, haciendo que el enlace se establezca en cuanto emisor y receptor estén dentro del mismo rango de cobertura. El inconveniente es que no hay posibilidad de realizar ningún proceso de autenticación y/o cifrado a este nivel, delegándolo a capas superiores de la pila de comunicaciones.

En Europa, 802.11p es también considerado dentro del estándar ITS-G5 [15] definido por el ETSI para establecer las capas físicas y de acceso al medio en entornos ITS. Las frecuencias utilizadas para realizar estas comunicaciones de corta distancia rondan los 5.9 Ghz, que variará dependiendo del país en concreto. Esta frecuencia tan elevada, a pesar de aportar grandes anchos de banda para la transmisión de datos, implica que emisor y receptor deben estar en línea de visión directa para un obtener un mejor rendimiento y, por tanto, se ve afectada en gran medida por los obstáculos y elementos orográficos que se puedan interponer. A pesar de ello, IEEE 802.11p se considera un medio adecuado para implementar aplicaciones como tele-pago en peajes y estaciones de repostaje, servicios de notificación de emergencia para vehículos, soporte a la conducción, control de aparcamiento, y por supuesto también los contemplados para entornos C-ITS, como hemos visto en el Apartado 1.2.

Las comunicaciones físicas están basadas en la Multiplexación por División en Frecuencias Ortogonales (OFDM), con unas tasas de transferencia de entre 3 y 27 Mbps. Las distancias que de media alcanzan las ondas emitidas son de aproximadamente un kilómetro, usando una potencia limitada y sin presencia de obstáculos. Sin embargo, efectos como la reflexión, difracción, dispersión o *Doppler*, pueden reducir considerablemente tanto la tasa de transferencia como la distancia. Sólo en casos específicos de gran emergencia puede incrementarse la potencia de la señal emitida para aumentar así el radio de cobertura.

En el mercado actual existen varios dispositivos compatibles con ITS-G5. En nuestras pruebas hemos tenido la oportunidad de usar los dispositivos de la mano de Sztaki (Instituto de Ciencias de la Computación en Hungría), que colaboraron con

Figura 2.2: Dispositivo Laguna de Commsignia compatibles con ITS-G5

nosotros dentro del marco del proyecto ITSSv6 [19]. El modelo concreto se denomina "Laguna" (ver Figura 2.2), que podía hacer las veces tanto de Estación Vehicular ITS como Estación de Carretera ITS.

2.2.3. Asistencia al Traspaso mediante IEEE 802.21

Un concepto que siempre ha ido ligado a las tecnologías inalámbricas es el traspaso, del inglés *handover*. Se produce cuando un nodo móvil cambia de una tecnología a otra debido a diferentes motivos, como pueden ser la calidad de la señal, coste, o simple preferencia. Estos traspasos por sí solos originan desconexiones momentáneas en el momento de realizarlos. Por mejorar esto, ya en 2004 se empezó a investigar en un protocolo que asistiera a los traspasos en redes heterogéneas. Este se denominó IEEE 802.21, estándar donde se especifican mecanismos independientes a las tecnologías de acceso utilizados para optimizar los traspasos entre dichas tecnologías. Se materializa como un protocolo que describe intercambios de mensajes entre varias entidades que cooperan para llevar a cabo dicha función de asistencia al traspaso. Trabajan a diferentes niveles:

- En el nivel más bajo se encuentra una entidad software que implementa el "Media Independent Event Service" (MIES), servicio encargado de estar pendiente de eventos, como posibles cambios que se produzcan en capas inferiores de la pila de red (ej. nivel de enlace), e informar de ello a capas superiores. Esta entidad también recibe comandos de estas capas superiores y se encarga de hacerlos efectivos.

- En el nivel central, que podríamos denominar de red, nos encontramos con el "Media Independent Handover Function" (MIHF), entidad que se encarga de recibir los eventos de nivel de enlace y reenviarlos a capas superiores. También

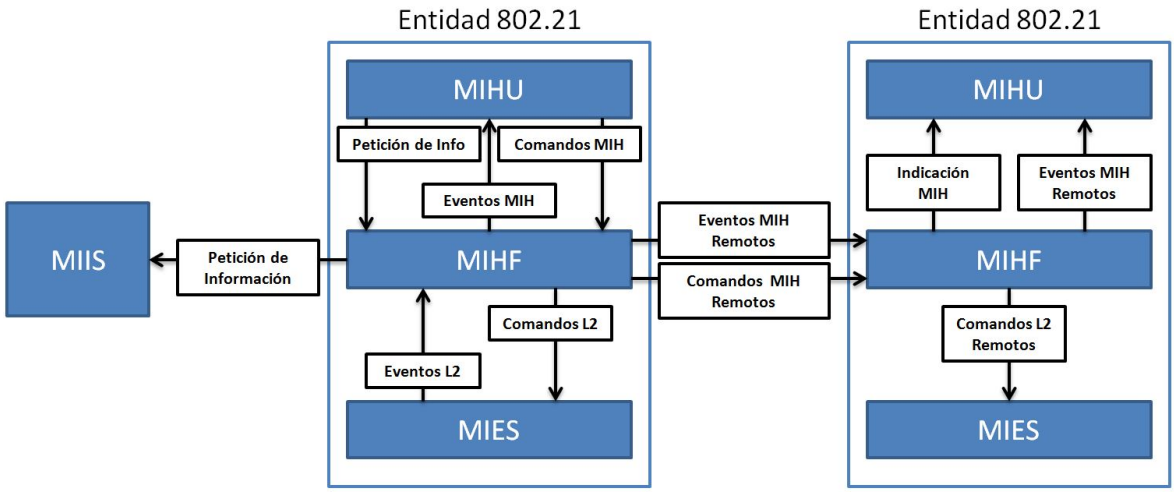


Figura 2.3: Esquema de entidades IEEE 802.21 y sus mensajes

recibe comandos de dichas capas superiores. Además, dichos eventos y comandos también pueden ser intercambiados con otras entidades 802.21 a través de la propia red mediante eventos y comandos remotos. En definitiva, esta entidad mantiene conectadas todas las entidades 802.21 presentes en la red.

- Como último nivel, se encuentra el "Media Independent Handover User" (MIHU), entidad encargada de implementar el "Media Independent Command Service", servicio encargado de transmitir comandos de las capas superiores a las inferiores. Además esta entidad recibe los eventos de las capas inferiores, toma decisiones en cuanto a qué interfaz de red utilizar en cada momento, y genera los comandos para hacer efectivas estas decisiones. También puede interactuar con otras aplicaciones para notificarlas de cuándo se desea que se realicen los traspasos.

Otra entidad aparte es el "Media Independent Information Service" (MIIS), que actuará como repositorio actualizado de la información recogida por las entidades de la infraestructura 802.21. De esta forma, dicha información puede ayudar mucho a la toma de decisiones y a tener una mejor instantánea global del estado de la red en todos sus puntos, consiguiendo traspasos más suaves.

2.3. Tecnologías IPv6 de Interés para C-ITS

El *Internet Protocol version 6* (IPv6) es una versión del protocolo *Internet Protocol* (IP) definido en el RFC 2460 [12] y diseñado para reemplazar al *Internet Protocol version 4* (IPv4) [84], que actualmente está implementado en la gran mayoría de dispositivos que acceden a Internet. La gran motivación que ha originado su aparición es debido a que el espacio de direccionamiento de IPv4 está prácticamente agotado,

imposibilitando a la red Internet seguir creciendo. Con la aparición de terminales móviles con posibilidades para conectarse a Internet, las necesidades de direcciones se han disparado, haciendo inevitable el paso al nuevo protocolo IPv6, que pasa de un direccionamiento de 32 bits a otro de 128 bits.

IPv4 e IPv6 son incompatibles y, por tanto, se debe funcionar con uno u otro. Por este motivo no se puede realizar un cambio súbito a IPv6, teniendo que implantar mecanismos de transición que posibiliten ir paulatinamente migrando de IPv4 a IPv6. Uno de los mecanismos de transición más extendidos es *Dual Stack* (doble pila), implementando ambas pilas IPv4 e IPv6 en los terminales, de tal forma que si algún servicio no ha sido migrado a IPv6, poder usar IPv4 para acceder a él. En la propuesta que se presenta en la Tesis se ha optado, no obstante, por otro mecanismo de transición llamado *Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers* (NAT64) [85], cuya idea principal es hacer que la red Internet quede enmascarada en una red de prefijo de la forma **64:ff9b::/96**, haciendo corresponder cada numero IPv4 a una dirección IPv6 de esta red con dicho prefijo, y traduciendo directamente sus cuatro números a valores hexadecimales que ocupan los 32 últimos bits de la dirección final, dejando el resto a cero. Este mecanismo debe ir apoyado por otro de resolución de nombres llamado *DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers* (DNS64) [86], que hace transparente para el usuario dicho enmascaramiento.

Aprovechando la necesidad de la creación de un nuevo protocolo, se han resuelto algunos problemas que tenía IPv4 y se han añadido interesantes mejoras que hacen de IPv6 un digno protocolo para el futuro Internet. Por ejemplo, el agotamiento del direccionamiento está propiciando el excesivo uso de *Network Address Translation* (NAT), tecnología que permite usar una dirección IP global por un número indeterminado de nodos, ahorrando así direcciones. El uso de NAT implica una larga lista de inconvenientes, como la imposibilidad de usar aplicaciones P2P, impide el desarrollo de nuevas tecnologías y compromete el rendimiento de la red y su facilidad de administración. Entre las mejoras que incluye además IPv6, se encuentran la simplificación de la cabecera para optimizar el procesamiento en los enrutadores, la evolución del concepto de *broadcast* al de *multicast*, mejorando también las funcionalidades de movilidad y seguridad. Además se le añaden mecanismos de auto-configuración de direcciones que facilita la configuración de los equipos conectados. De estas mejoras hablaremos con más detalle en los siguientes apartados, ya que son de vital importancia a la hora de dar soporte a una red vehicular C-ITS.

2.3.1. Asignación Automática de Direcciones

Los nodos IPv6 pueden configurarse a sí mismos automáticamente cuando son conectados a una red usando los mensajes de descubrimiento de enrutadores del protocolo *Internet Control Message Protocol version 6* ICMPv6. Un nodo IPv6, nada más levantar el interfaz de red, ya dispone de una dirección IP de tipo *link-local* que permite conectarse a otros nodos que estén conectados al mismo canal de comunicaciones. Es decir, es de ámbito local. En la Figura 2.4 puede verse cómo generar

dicha dirección en base a la dirección MAC de la interfaz física. Primero se transforma dicha MAC que sigue el formato EUI-48 de 48 bits al formato EUI-64 de 64 bits para así tener cubierta la mitad derecha de la dirección IPv6. Solamente quedaría añadirle el prefijo **fe80::** que ocuparía los restantes 64 bits para obtener así la dirección IP de ámbito local.

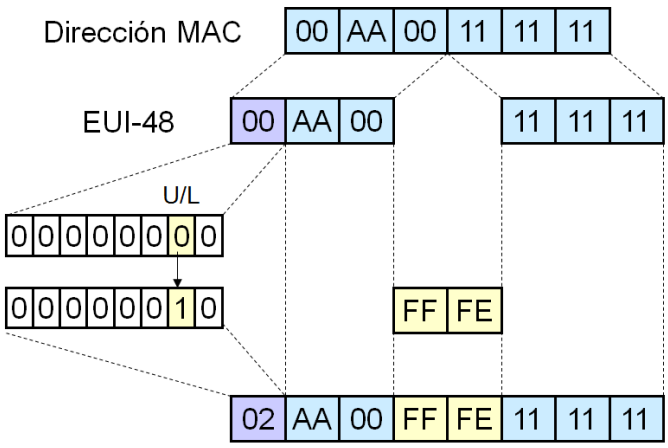


Figura 2.4: Paso de una dirección MAC EUI-48 a EUI-64

Usando esta dirección IP de ámbito local, el nodo envía una solicitud llamada *Router Solicitation* usando una dirección *multicast* de destino predefinida (ff01::2) donde todos los enrutadores al alcance están registrados. Si los enrutadores que estén presentes están configurados para esto, responderán con un anuncio de enrutador *Router Advertisement* con los parámetros de configuración de capa de red, tales como el prefijo de red a utilizar, la dirección del propio enrutador o si dentro de la red desplegada por el router hay presente un servidor HA dando soporte al servicio de movilidad. Este mensaje RA también suele ser transmitido periódicamente por los enrutadores, lo que puede originar que los nodos móviles lo reciban sin haberlo solicitado. Se genera pues una nueva dirección IPv6, esta vez de ámbito global, usando el prefijo especificado y los 64 bits extraídos de la dirección MAC (*Media Access Control*). Se comprueba que dicha dirección no esté siendo usada, usando para ello un mecanismo de detección de direcciones duplicadas (DAD) propio de IPv6. Llegados a este punto el nodo ya está capacitado para enviar paquetes a cualquier nodo de la red.

Este tipo de auto-configuración se denomina "sin estado", ya que las asignaciones de IP no se controlan por ningún ente regulador. Si se desea tener un mayor control sobre qué direcciones IPv6 asignar a qué nodos, es posible utilizar un mecanismo "con estado" como *Dynamic Host Configuration Protocol* para IPv6 (DHCPv6) o bien configurar de forma estática los nodos. Los enrutadores presentan un caso especial de requerimientos para la configuración de direcciones, ya que muchas veces son la fuente de información de autoconfiguración, como anuncios de prefijos de red y anuncios de enrutador. Por ello su configuración suele ser estática.

Este mecanismo de auto-configuración es muy interesante para las redes vehiculares dentro de un entorno C-ITS, ya que los nodos no paran de cambiar de una red a otra debido a su carácter móvil. Este mecanismo nos permite tener de forma rápida conectividad con otras estaciones ITS.

2.3.2. Servicio de Movilidad Basado en *Mobile IP* (MIPv6)

En escenarios ITS es una necesidad permanecer conectado a la red constantemente, incluso cuando los vehículos se mueven y cambian de red o de tecnología de acceso. Sin embargo, durante estos cambios no es extraño que el vehículo use diferentes direcciones IP, cada una perteneciente a las distintas redes por las que va pasando. A estas direcciones se las denomina *Care-of Address* (CoA). Esto supone un problema grave, ya que las aplicaciones que estén usando la red se ven afectadas por esos cambios, teniendo en ocasiones que restablecer las conexiones en cada cambio, con la interrupción del servicio de conectividad que eso supone. El Servicio de Movilidad viene a solucionar este problema, aportando una única dirección IP, llamada *Home Address* (HoA) que las aplicaciones usarán sin ser conscientes de los cambios de CoA.

El servicio de movilidad ha sido uno de los servicios más beneficiados de la incorporación de IPv6, ya que solventa algunas desventajas que tenía el uso del servicio de movilidad en IPv4, como por ejemplo el enrutamiento triangular, que explicamos en el siguiente apartado.

2.3.2.1. Principios de Funcionamiento de MIPv6

La idea fundamental que sigue la movilidad como servicio es la de conseguir que un nodo de una red que llamaremos la *red home (Home Network)* pueda convertirse en un nodo móvil (MN) y conectarse a otras redes, que llamaremos *redes visitadas*, de tal forma que cualquier nodo de la red pueda comunicarse con él sin ser consciente de que se ha movido de su *red home*. Esto se consigue dedicando un equipo conectado a esta *red home* que intercepte los paquetes que vayan dirigidos al MN y reenviárselos a través de un túnel IP allá donde esté conectado. Para que este equipo, denominado *Home Agent* (HA), pueda realizar su cometido necesita saber la CoA actual que tiene asignado el MN para así poder dirigirle el tráfico a través del túnel. Esto se consigue manteniendo una caché de asociaciones HoA-CoA en el HA. Para ello, cada vez que el MN se conecte a una nueva red y le sea asignada una nueva CoA, éste debe informar al HA de este cambio para actualizar dicha caché. Esto se realiza con los mensajes *Binding Update* (BU) y *Binding Ack* (BA). De esta forma, el tráfico siempre pasa por el HA como intermediario mientras el MN esté fuera de su *red home*. Esto supone el denominado **enrutamiento triangular**, que se puede evitar mediante optimizaciones del protocolo. En este caso, el MN puede enviar un BU notificando al otro extremo de la comunicación, que llamaremos *Correspondent Node* (CN), cuál es la CoA que tiene asignada el MN para no tener que pasar por el HA y comunicarse a partir de ese momento directamente usando la CoA.

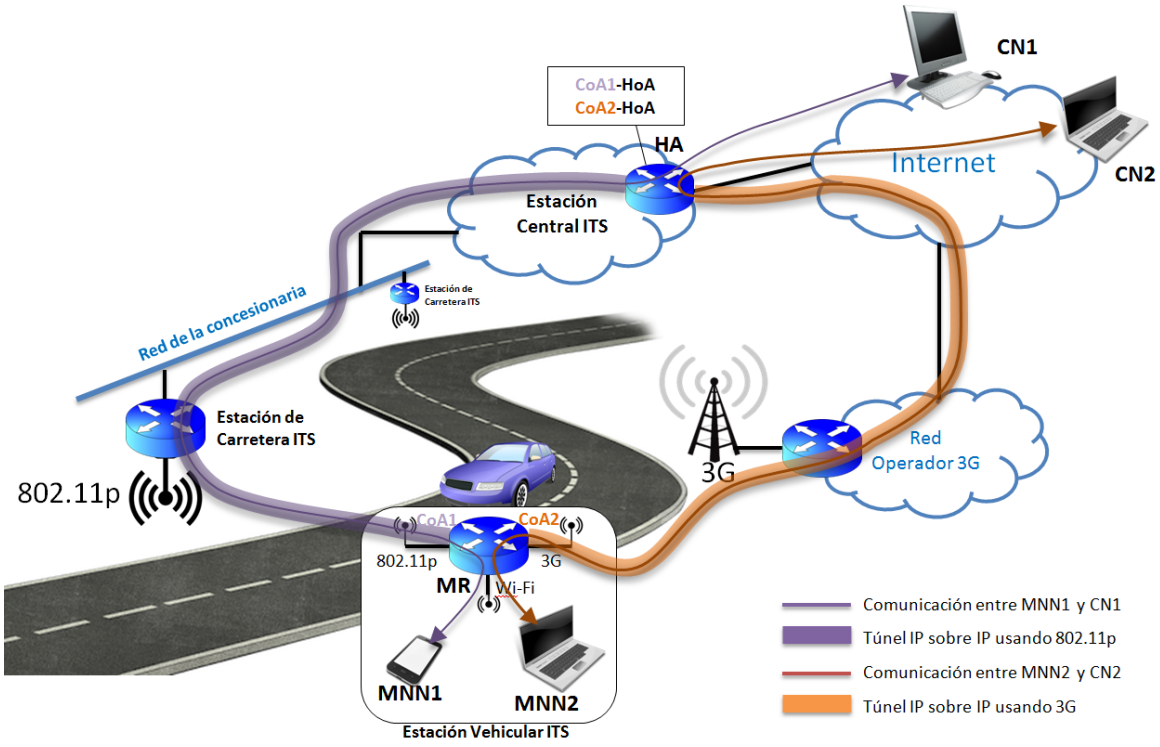


Figura 2.5: Escenario típico de uso de NEMO

2.3.2.2. Movilidad de Redes con *Network Mobility* (NEMO)

El servicio de movilidad, inicialmente implementado por MIPv6, ha seguido evolucionando y dando lugar a nuevos protocolos, como es el caso de *Network Mobility Basic Support* (NEMO), que permite al MN convertirse en un *Mobile Router* (MR) para una red móvil. Es decir, cambiamos de movilidad de nodos a movilidad de redes enteras. Los dispositivos conectados a estas redes móviles no serán conscientes de estar en una red móvil como tal pues, gracias al servicio de movilidad desplegado con NEMO, no se requerirá de ningún cambio de direccionamiento cuando el MR se mueva de una red a otra.

Esta funcionalidad que NEMO aporta es de mucha utilidad en escenarios C-ITS para dotar de conectividad a todos los nodos que viajan a bordo de las Estaciones Vehiculares ITS, instalando en cada uno de ellos un MR. Se hace especialmente recomendable en vehículos grandes con gran capacidad de pasajeros (trenes, autobuses, etc.), ya que permite unificar múltiples conexiones en una sola conexión al exterior.

La Figura 2.5 refleja la forma de trabajar de NEMO. La red *home* (en la Estación Central ITS) puede delegar parte de su direccionamiento de red a los MRs gracias a los llamados *Mobile Network Prefix* (MNP), que son prefijos de red IPv6 que determinan sub-rangos de direcciones que pasan a ser gestionadas por cada uno de los MR. Por tanto, en NEMO, aparte de tener registradas las asociaciones HoA-CoA en la caché del

HA, a cada HoA habrá que adjuntar el prefijo delegado y así saber a qué MR enviar tráfico dirigido a estos sub-rangos delegados. Cada MR asignará direcciones a los nodos de su red móvil, llamados *Mobile Network Nodes* (MNN), usando para ello el MNP que se le ha asignado.

En un estado inicial donde todavía no hay establecida ninguna conexión ni asociación (HoA/MNP)-CoA, los MR se mueven siguiendo el trayecto de los vehículos en donde van instalados. Aquí entran en juego las Estaciones de Carretera ITS desplegadas a lo largo de la carretera, que son los puntos de acceso por donde las Estaciones Vehiculares ITS pueden alcanzar a la Estación Central ITS y sus servicios. Para advertir la presencia de una Estación de Carretera ITS, éstas envían periódicamente un mensaje de anuncio llamado *Router Advertisement* (RA) en modo *multicast* para que cualquier MR que lo reciba perciba su presencia y decida conectarse. Como determina el modelo de movilidad, los MR tienen asignadas cada uno una HoA, dirección fija perteneciente al rango de direcciones de la *red home* de la Estación Central ITS. Sin embargo, cuando el MR se conecta a una Estación de Carretera, dicha HoA no nos permite alcanzar la Estación Central ITS. Por eso, gracias al prefijo de red anunciado en el mensaje RA, el MR puede autogenerar una nueva dirección (CoA) que sí le permita llegar a la Estación Central ITS y sus servicios. Esta CoA es notificada inmediatamente al HA mediante el par de mensajes BU y BA. A partir de este momento, cualquier tráfico que llegue a la *red home* dirigida a alguna HoA o a alguna subred delegada con prefijo MNP, será reenviado mediante un túnel IP al MR. De igual forma, el MR redirigirá todo el tráfico que reciba de sus nodos MNNs al HA a través de otro túnel IP.

En definitiva, mediante el proceso anterior, NEMO simula la presencia física del MR como si estuviera directamente conectado a la *red home* junto al HA. Precisamente es el HA el que se hace pasar por ellos y los sustituye, haciendo pensar a cualquier nodo de Internet, que llamaremos *Correspondent Node* (CN), que siempre están en el mismo sitio y no se están moviendo. Todas estas tareas de encapsulamiento, mantenimiento de la caché de asociaciones HoA-CoA y envío de mensajes BU y BA se lleva a cabo por entidades software que se ejecutan en la capa de aplicación y complementan la funcionalidad proporcionada por el núcleo del sistema operativo. Un ejemplo de implementación de código abierto es la proporcionada por el proyecto UMIP [32], que es la que se ha utilizado en la presente Tesis para la implementación de nuestra propuesta de arquitectura de red para C-ITS descrita en el Capítulo 3.

2.3.2.3. Uso Simultáneo de Múltiples Interfaces de Red mediante *Multiple Care-of Address* (MCoA)

Como hemos visto en anteriores apartados, la tendencia en el mundo de los dispositivos móviles camina hacia una hibridación de las comunicaciones, es decir, el uso simultáneo de diferentes tecnologías de red. Hasta ahora, la movilidad desplegada mediante NEMO sólo permitía tener asociado a cada MR una CoA en su caché de asociaciones. Esto hacía que, aunque hubiera varias interfaces disponibles con diferentes CoAs, sólo una de ellas fuera usada para el envío del tráfico, descartando las otras. Para

aprovechar esta conectividad múltiple, a NEMO se le añadió una extensión llamada *Multiple Care-of Addresses Registration* (MCoA) definida en el RFC 5648 [48]. Dicha extensión permite proveer movilidad mediante el uso de múltiples CoAs, usando para cada una de ellas un túnel IP distinto. Para poder distinguir entre un túnel u otro, cada CoA deberá asociarse, aparte de con una HoA y un MNP, con un identificador llamado *Binding Identification number* (BID). Este identificador sienta las bases de la asociación de flujos de tráfico (*Flow Binding* [87]), permitiendo determinar qué tráfico enviar por cada interfaz en función de unas políticas pre-establecidas. Esto se realiza a través del marcado de paquetes, usando el identificador BID asociado a la interfaz por donde se quiere transmitir cada paquete. Si un paquete viene sin marcar, se usará un túnel por defecto (el de mayor prioridad).

El disponer de varias vías de comunicación simultáneas supone una gran mejora en los procesos de traspaso de una tecnología a otra, ya que el cambio es inmediato. Esto hace que esta extensión sea ideal para entornos vehiculares C-ITS donde se producen traspasos constantemente. Por ejemplo, en la Figura 2.5, la cobertura intermitente proporcionada por 802.11p puede ser complementada sin apenas experimentar cortes a través de la conectividad 3G/4G.

La implementación proporcionada por el proyecto UMIP [32] no tenía la extensión MCoA implementada, pero gracias a los esfuerzos realizados en el proyecto de investigación ITSSv6 [19] en el que ha participado la Universidad de Murcia y el doctorando, se consiguió incorporar dicha extensión y poder así utilizar sus ventajas. En nuestra propuesta de arquitectura que desarrollamos en el Caputulo 3, vamos a utilizar precisamente dicha implementación resultante del proyecto ITSSv6, implementación que lleva asociados también algunos cambios importantes en el núcleo del sistema operativo, sin los cuales, el servicio de movilidad no funcionaría correctamente.

2.3.3. Servicio de Seguridad Basado en IPsec

Internet Protocol security (IPsec), definido por el IETF en el documento RFC 4301 [88], es un conjunto estándar de protocolos cuya función es la de proteger las comunicaciones llevadas a cabo a través del protocolo IP. Para ello cifra la información transmitida mediante una serie de algoritmos criptográficos que nos permiten, en definitiva, aportar autenticidad y confidencialidad a las comunicaciones. Estos algoritmos pueden ser establecidos manualmente, pero el estándar también define un protocolo para poder negociar dichos algoritmos y el material criptográfico que usarán para cifrar la información transmitida.

La principal característica de estos protocolos definidos en IPsec es que actúan en la capa de red, a nivel IP. La gran ventaja que introduce IPsec frente a otros protocolos de seguridad utilizados ampliamente como pueden ser *Secure Socket Layer* (SSL), *Transport Layer Security* TLS y *Secure Shell* (SSH), es la posibilidad de proteger los protocolos de transporte como son *Transmission Control Protocol* (TCP) y *User Datagram Protocol* (UDP) de forma transparente, sin que entidades que residen en capas superiores de la pila IP se percaten de ello. De este modo, las aplicaciones que requieren de este servicio de seguridad no necesitan modificar su implementación, como

sí lo hacen en el caso de protocolos mencionados antes que operan en la capa de transporte y superiores.

La arquitectura de seguridad que define IPsec utiliza el concepto de "Asociación de Seguridad" (SA) como base para construir funciones de seguridad sobre IP. Dicha SA consiste en una selección de algoritmos criptográficos y una serie de parámetros para hacerlos funcionar. Además, en ella se define a qué tráfico se le aplicarán dichos algoritmos, entre otros parámetros de menor importancia.

La implementación de IPsec está recomendada para IPv6 y es opcional para IPv4. Originalmente, el estándar IPsec fue definido por los RFC 1825 [89] y siguientes, publicados en 1995. En 1998 fueron sustituidos por una nueva versión incompatible con la anterior, definida en los RFC 2401 [90] y siguientes. En el año 2005 se publicaron nuevos RFCs, concretamente el 4301 [88] y siguientes, donde se actualizaba el estándar IPsec al que se le añadía, en esta ocasión, una segunda versión del protocolo "Internte Keying Exchange version 2" (IKEv2). Este protocolo es esencial en la propuesta de seguridad presentada en esta Tesis, tal y como se verá más adelante, pero antes hay que dejar definidos algunos conceptos relacionados con IPsec.

2.3.3.1. Motivación del uso de IPsec

El uso de IPsec tiene su origen en la necesidad de luchar contra una serie de amenazas que comprometen la información transmitida en múltiples aspectos. Estas amenazas se materializan en ataques contra la transmisión de datos que estemos realizando. A continuación vamos a mostrar tres ejemplos típicos de ataques, que nos llevaran a descubrir las tres características deseables en un sistema de protección de datos como es IPsec:

- *Sniffing*: Consiste en la interceptación de tramas de un determinado enlace de comunicaciones, ya sea cableado o inalámbrico, por una tercera parte ajena a la transmisión. Se puede realizar de una forma fácil cambiando a "modo promiscuo" una determinada interfaz de red para poder recepcionar todas las tramas que se transmiten por el canal. Podemos evitar la captura de estas tramas evitando que el tercero no tenga acceso al canal. Esto se puede hacer de forma física (protegiendo cables, etc.) o de forma lógica (cifrando la transmisión). A veces es imposible proteger físicamente un enlace, como los de naturaleza inalámbrica. Cifrando los datos no evitamos la captura de tramas, sino el poder obtener los datos transmitidos de ellas. IPsec también resuelve esta amenaza mediante cifrado simétrico, lo que conlleva la aportación de la primera de las características: **la confidencialidad**.

- *IP Spoofing*: Consiste básicamente en la posibilidad de suplantar la dirección IP en una transmisión por un tercero, permitiendo a un atacante enviar un paquete cambiando la IP de origen para hacerla coincidir con una de las IP de los extremos de una transmisión dada. Además permitiría dificultar dicha transmisión, o incluso alterar los datos transmitidos. Esto puede resolverse gracias a la "firma digital", que permite generar y adjuntar un pequeño resumen de los

datos que se transmiten que permita verificar en destino si dichos datos han sido alterados o no. Para ello se genera de nuevo el resumen de los datos en destino y se compara con el que viene adjunto. Si son iguales, podemos asegurar que los datos no han sido modificados. Para la creación de estos resúmenes se emplean algoritmos que además de los datos, necesitan un secreto compartido para evitar que dichos resúmenes puedan ser creados por terceros. Este mecanismo nos aporta la segunda de las características: **la integridad**.

- *TCP Hijacking*: Se trata de que un tercero pueda secuestrar una conexión TCP/IP que tengamos establecida ante un servidor simplemente mandando un paquete TCP falso pero con los números se secuencia correctos, que pueden ser averiguados si el atacante puede esnifar los paquetes. Una vez que el servidor recibe dicho paquete, el servidor da por hecho que es correcto y manda el siguiente paquete a la dirección origen del atacante. El cliente lícito se queda sin la conexión. Protocolos como FTP o telnet no comprueban el origen de los paquetes que recibe. Para evitar esto, los extremos de la comunicación tienen que ser capaces de autenticar el origen de los paquetes. Esto se hace usando de nuevo el resumen digital (HMAC). IPsec también incorpora estas técnicas para evitar ataques de este tipo. Esto nos lleva a la tercera de las características: **la autenticidad del origen**.

2.3.3.2. Protocolos

Como habíamos adelantado, IPsec se compone de varios protocolos. A continuación describimos brevemente cada uno de ellos:

- ***Authentication Header* (AH)**, que proporciona integridad, autenticación del origen y no repudio. Protege los datos y los campos invariables de la cabecera IP, como son los números IP origen y destino.
- ***Encapsulated Security Payload* (ESP)**, que proporciona confidencialidad, integridad, autenticación y no repudio. En esta ocasión no incluye protección a los campos de la cabecera IP.
- ***Internet Keying Exchange* (IKE)**, que se emplea para establecer un secreto compartido y negociar los algoritmos criptográficos a usar. De este protocolo hablaremos en profundidad más adelante.

2.3.3.3. Modos de funcionamiento y escenarios típicos

IPsec tiene dos modos de funcionamiento, usándose uno u otro dependiendo del escenario en el que nos encontremos. Estos modos son:

- **Modo Túnel**, en donde cada paquete es encapsulado en otro paquete IP, al que se le aplica la cabecera IPsec. Este modo de funcionar protege el paquete completo original, cabecera incluida, ocultando además el origen y el destino de dicho

paquete (para el caso de ESP). Este modo se utiliza generalmente entre puertas de enlace que conectan redes entre sí, como vemos en la Figura 2.7a. Los nodos de dichas redes no serían conscientes de dicha protección. Existe otro escenario llamado "Roadwarrior", o más conocido como "Virtual Private Network" (VPN), donde se utiliza este mismo modo de funcionamiento. Consiste, como vemos en la Figura 2.7b, en que un nodo de una red se conecta a ella remotamente a través de una pasarela de seguridad y se le asigna una dirección IP de una red virtual que se establece entre el host y la pasarela. De esta forma puede tener acceso a su red a través de la pasarela.

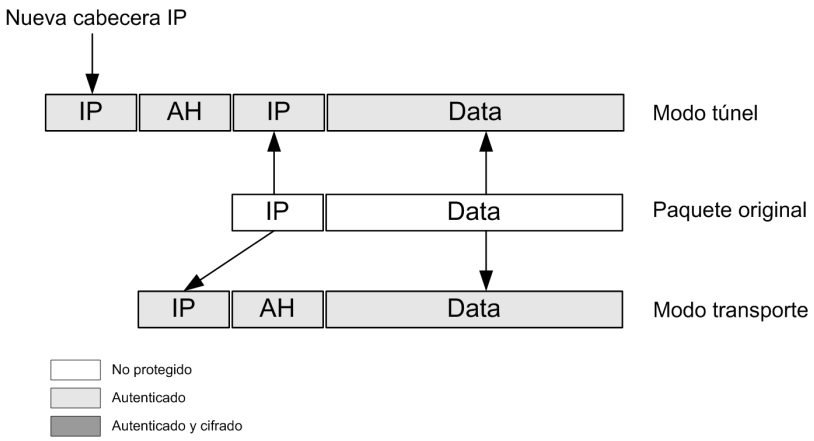
- **Modo Transporte**, en donde las cabeceras IPsec son aplicadas por los mismos extremos de la comunicación. Este modo es ideal para los casos donde los extremos del túnel coinciden con los extremos de la comunicación, como vemos en la Figura 2.7c. Supone menos sobrecarga que el modo túnel, además de evitar la redundancia de direcciones IP que se produciría. Este modo no oculta el origen y destino de los paquetes (tanto con AH como con ESP).

2.3.3.4. Asociaciones y Políticas de Seguridad

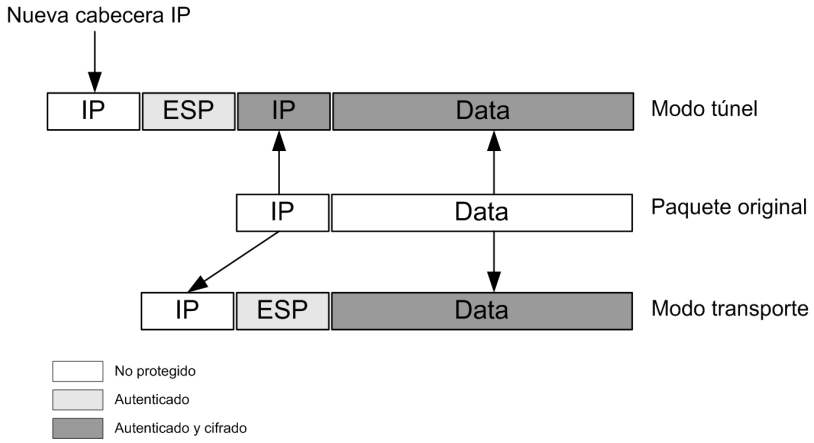
La arquitectura de seguridad IPsec funciona en torno al concepto de *Asociación de Seguridad* o *Security Association* (SA), donde se establecen todos los parámetros necesarios para proteger un determinado tráfico mediante IPsec. Una SA se compone de los siguientes campos:

- Security Parameter Index (SPI), que hace las veces de identificador de la SA.
- IP destino del paquete.
- IP origen del paquete.
- Protocolo IPsec: AH o ESP.
- Modo de operación: Túnel o transporte.
- Selección de algoritmos criptográficos a utilizar.
- Material criptográfico necesario para los algoritmos.

Las SAs activas que se aplican en un momento dado están almacenadas en una base de datos interna llamada "Security Association Database" (SAD). La definición de estas SAs está dentro de las responsabilidades del administrador de red. Sin embargo, hacerlo de forma manual puede ser tedioso e incluso no aplicable a redes muy grandes con multitud de nodos. Para poder crearlas bajo demanda y de forma automática, se define otro concepto importante dentro de la arquitectura de seguridad IPsec llamado *Politica de Seguridad* o *Security Policy* (SP). Estas políticas definen el tratamiento que se le desea dar a determinados tráficos. Entre las opciones disponibles aplicables a un determinado tipo de tráfico están:



(a) Cabecera AH



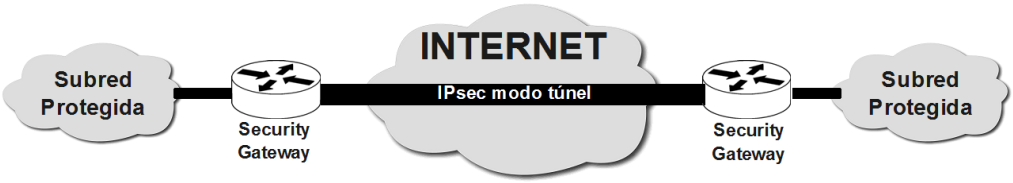
(b) Cabecera ESP

Figura 2.6: Aplicación de las cabeceras AH y ESP

- Bloquear el tráfico.
- Permitir el tráfico.
- Permitir el tráfico sólo si es protegido mediante IPsec.

Estas políticas se establecen por el administrador y se almacenan en otra base de datos interna llamada *Security Policy Database* (SPD). Una SP está compuesta principalmente por:

- **Selector del origen del tráfico**: Reglas que definen qué orígenes tienen que tener los paquetes para aplicarle la política.
- **Selector del destino del tráfico**: Reglas que definen de la misma forma los destinos del tráfico al que se les aplicará la política.



(a) Protección entre pasarelas (Modo túnel)

(b) Protección entre un host y una pasarela (Módo túnel)

(c) Protección extremo a extremo (Modo transporte)

Figura 2.7: Escenarios típicos de uso de IPsec

- **Protocolos a proteger**: Aquí se especifican los protocolos de niveles superiores al de red, además de los puertos de origen y destino (caso de TCP y UDP), tipos y subtipos (caso de ICMP), etc.
- **Propuesta de algoritmos criptográficos**: Lista con las posibles configuraciones en cuanto a algoritmos criptográficos se refiere. Serán los posibles algoritmos que la política acepta para la protección de los datos.
- **Prioridad de la política**: Valor de prioridad que será aplicable cuando más de una política coincida con un determinado tráfico. Se aplicará la de mayor prioridad, no todas las que coincidan.
- **Sentido del tráfico**: Este puede ser tráfico que llega (IN), que sale (OUT) o que se reenvía (FWD).

Normalmente las políticas se establecen por pares, ya que cada política solo define el tratamiento en un sentido. Cuando un determinado tráfico cumple con alguna de

las políticas establecidas y en ella se establece que debe protegerse mediante IPsec, se consulta la base de datos SAD para ver si hay alguna SA instanciada que cumpla los requisitos de esta política. Si la encuentra la usa para proteger el tráfico. Si no, entonces entra en juego el protocolo IKE para la negociación y establecimiento automático de SAs. Para comprender mejor el funcionamiento de la arquitectura de seguridad IPsec, observe la Figura 2.8, donde se muestran los pasos que se producen para crear o borrar SAs a partir de las SPs.

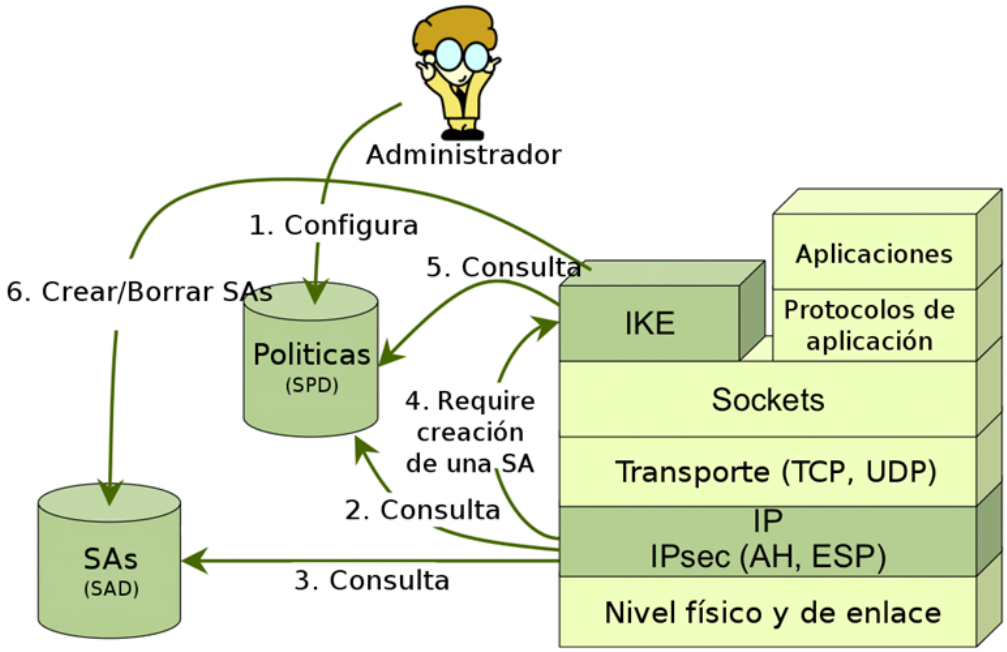


Figura 2.8: Funcionamiento de la arquitectura de seguridad IPsec

Como puede verse en la Figura 2.8, IKE está situado en el nivel de aplicación de la pila IP, es decir, que debe existir una aplicación que implemente el protocolo, pues el sistema operativo por sí solo no suele implementarlo. En la siguiente sección se explican las principales características de este protocolo, en el que se han centrado principalmente nuestros esfuerzos en investigación.

2.3.3.5. *Internet Keying Exchange* versión 2 (IKEv2)

El protocolo *Internet Keying Exchange* (IKE) fue diseñado, como ya hemos adelantado en la sección anterior, para automatizar el establecimiento de las asociaciones de seguridad IPsec (IPsec SAs). Su primera versión vio la luz en 1998 y fue definido en tres documentos RFC del IETF (2407 [91], 2408 [92] y 2409 [93]). Esta primera versión (IKEv1) sufría de varias debilidades y adolecía de ser demasiado complejo, por lo que el IETF decidió proponer una segunda versión (IKEv2) en el año 2005 con el documento RFC 4306 [94], que resolvía dichas debilidades aparte de

simplificar el protocolo notablemente. Además, como veremos, se le añaden nuevas funcionalidades para hacer de IKEv2 un protocolo más versátil. Después, en 2010 y en 2014 el estándar se actualiza con los documentos RFC 5996 [30] y 7296 [31] respectivamente. Parte de las actualizaciones incorporadas en el RFC 5996 fueron producto de los descubrimientos que se hicieron en este trabajo a la hora de llevar a cabo una implementación del mismo. Concretamente, las aportaciones se incluyeron en el RFC 4718 [29], que describía guías y clarificaciones útiles para implementar IKEv2, que posteriormente fueron incorporadas en dicho RFC 5996.

2.3.3.5.1 Detalles del Protocolo

Como hemos visto en la sección anterior, IKEv2 se implementa en la capa de aplicación de la pila IP. La aplicación que lo implemente usará el protocolo UDP para transportarlo, usando los puertos 500 y 4500. Esta aplicación podrá tomar uno de los siguientes roles a la hora de iniciar una negociación: El *iniciador* y el *respondedor*. Como sus nombres indican, el iniciador actúa como cliente y el respondedor actua como servidor durante la negociación. El protocolo está compuesto de un conjunto bien definido de cuatro tipos de intercambio de mensajes, entendiendo como intercambio un par de mensajes petición-respuesta. Estos son: IKE_SA_INIT, IKE_AUTH, CREATE_CHILD_SA e INFORMATIONAL. La Figura 2.9 muestra de forma gráfica dichos intercambios. Gracias a este mecanismo basado en intercambios se puede aportar confiabilidad al protocolo, ya que UDP no dispone de mecanismos para ello.

El intercambio IKE_SA_INIT, establece una primera asociación de seguridad (IKE SA) a nivel de protocolo IKE, cuyo cometido será proteger todos los intercambios que se produzcan a partir de ese momento. Es importante no confundir esta IKE SA con las asociaciones de seguridad IPsec (IPsec SA) que serán establecidas en intercambios posteriores. En concreto, el siguiente intercambio que se produce, el llamado IKE_AUTH, autentica a las partes y crea la primera IPsec SA entre ellas. Estos dos primeros intercambios son llamados *Intercambios Iniciales* ya que solo se usan una vez y siempre en el mismo orden.

Existen dos intercambios más que mencionar, que son los responsables de gestionar las IPsec SAs. El intercambio CREATE_IKE_SA permite crear nuevas IPsec SAs. El intercambio INFORMATIONAL permite destruir IPsec SAs, y una combinación de ambos intercambios permite renovar una IPsec SA con nuevo material criptográfico. Además, el intercambio INFORMATIONAL permite la implementación de funcionalidades adicionales como la notificación de eventos, o la configuración de direcciones IP, entre otras.

2.3.3.5.2 Nuevas Características Incorporadas

IKEv2 tiene una especial relevancia con respecto a IKEv1 por la inclusión de un conjunto de nuevas funcionalidades:

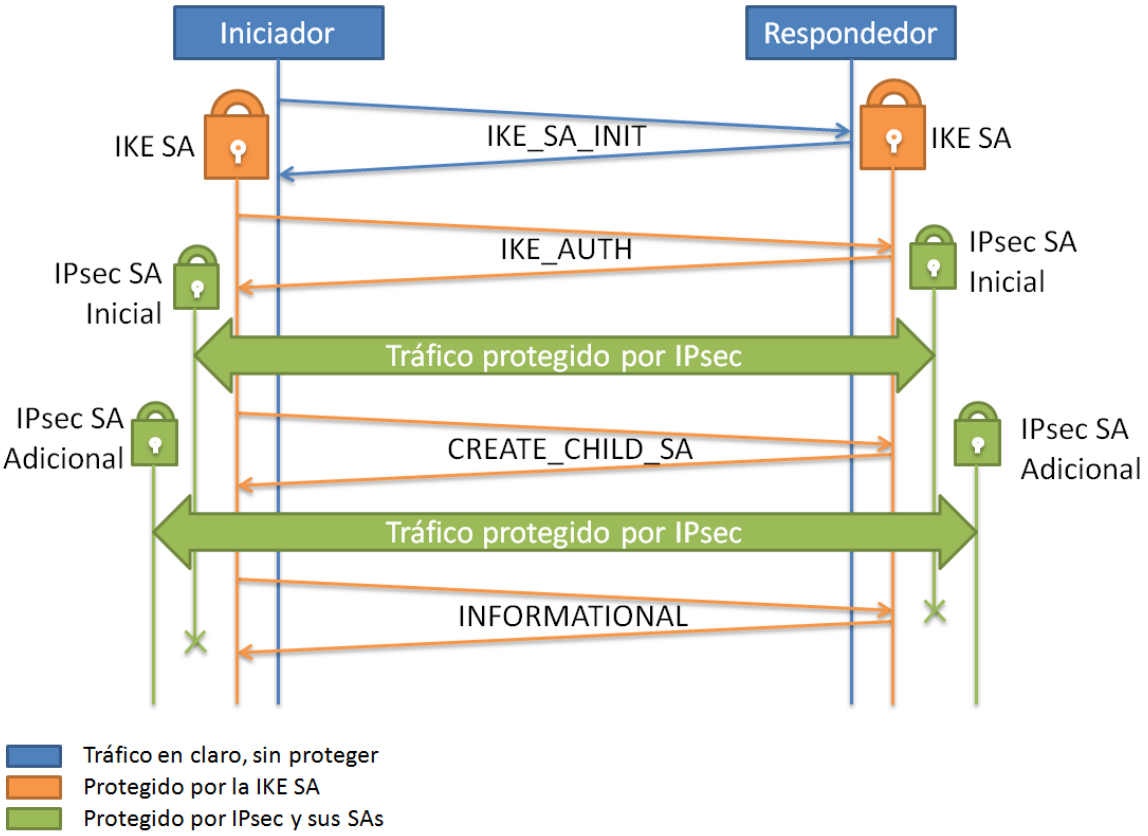


Figura 2.9: Intercambios IKEv2 y sus respectivas asociaciones

- ***NAT trasversal***: Permite a IPsec establecer asociaciones de seguridad cuando hay un servidor NAT por medio.

- ***Extensible Authentication Protocol (EAP)***: El transporte de mensajes adicionales para extender los métodos de autenticacion por medio del protocolo EAP [95].

- ***Configuración remota de direcciones IP***: Permite establecer escenarios de tipo VPN donde un equipo puede conectarse remotamente a una red de forma segura.

No podemos olvidar que IKEv1 tiene debilidades en su diseño que la versión 2 solventa, por lo que tanto por las nuevas características como por las debilidades encontradas, se hace muy recomendable el uso generalizado de IKEv2 como protocolo de intercambio de material criptográfico y negociación de asociaciones de seguridad.

2.3.3.6. Necesidad de IPsec e IKEv2 en Redes Vehiculares

Hasta la fecha, la mayoría de aproximaciones para aportar seguridad a las redes vehiculares, y más concretamente a las C-ITS, establecen soluciones orientadas a la protección de mensajes individuales. Debido a que los servicios más habituales desplegados en C-ITS suelen generar poca cantidad de mensajes, el uso de la criptografía asimétrica ha demostrado ser la más adecuada para este tipo de protección. Sin embargo, para los nuevos servicios de *infotaiment* de alto consumo de ancho de banda, servicios multimedia en su mayoría, este enfoque de protección resulta ineficiente y altamente demandante de recursos, lo que lo convierte en un cuello de botella cuando el tráfico es grande, sobre todo cuando hablamos de dispositivos incorporados a los vehículos especialmente diseñados para consumir poca energía. Es aquí donde aparece la necesidad de un esquema de protección diferente, que esté basado en la protección de flujos de paquetes, es decir, seguridad orientada a sesión. Por tanto proponemos IPsec como candidato para definir este nuevo esquema de protección, basado en criptografía simétrica, menos demandante de recursos. Este sistema de protección de tráfico necesita estableces las sesiones, y en su caso concreto, las asociaciones de seguridad. Para ello existe el protocolo IKEv2, que como hemos visto en apartados anteriores, es un complemento fundamental para IPsec como sistema de creación y gestión de asociaciones de seguridad. Como veremos en el Capítulo 7 donde probamos el rendimiento y la validez de nuestra propuesta, la aplicación de IPsec como protección del tráfico apenas supone sobrecarga y consumo de recursos, llegando a registrar prácticamente las mismas tasas de transferencia que en pruebas donde no se aplicó IPsec.

2.4. Conclusiones

Las organizaciones de estandarización ISO y ETSI han comprendido que la necesidad de una arquitectura de referencia única para todos es real, y por ello ya disponemos de la definición de un marco de trabajo en el que se ha basado nuestra propuesta de red vehicular. También hemos constatado que el mundo de Internet y los estándares definidos en el seno del IETF pueden ser muy útiles también en el mundo de las redes vehiculares, y en concreto en las C-ITS, ya que la incorporación de IP como protocolo de red lleva consigo el poder usar tecnologías asociadas, sobre todo a nivel de red y transporte para la implementación de servicios, como los de autoconfiguración, seguridad y movilidad. Sin embargo, en cuanto a las tecnologías de comunicaciones inalámbricas, siguen siendo IEEE, 3GPP y ETSI las que establecen los estándares a nivel físico y de enlace.

Capítulo 3

Propuesta de Arquitectura de Red

Establecido el marco en el que se desarrolla la Tesis en el Capítulo 1 y presentadas las tecnologías que formarán parte de este desarrollo en el Capítulo 2, se presenta en este y próximos capítulos nuestra propuesta de red vehicular. Veremos cómo, desde un principio, se apuesta por seguir los estándares establecidos, ya que de esa manera los fabricantes tendrán la posibilidad de desarrollar y desplegar sus servicios sobre una misma arquitectura. Por esta razón, la arquitectura de comunicaciones que proponemos está basada en la arquitectura de referencia establecida por el ISO/ETSI. También veremos que IPv6 es el pilar donde se apoya principalmente nuestra propuesta de arquitectura, dejando los protocolos específicos para ITS en el nivel de red a un lado. Además, IPv6 viene acompañado de otras tecnologías y protocolos estandarizados por el IETF procedentes del mundo de Internet y que hemos integrado en nuestra arquitectura siguiendo el trabajo marcado por el ISO 21210, donde se establece cuáles de ellos combinar y utilizar en entornos ITS. Gracias a estos protocolos se completa una pila capaz de dar soporte al despliegue de los distintos servicios y aplicaciones, teniendo en cuenta las peculiaridades de la misma en los distintos tipos de Estaciones ITS.

Sobre esta pila de comunicaciones, se muestran además los primeros servicios que consideramos fundamentales en entornos móviles, como son los servicios de seguridad y movilidad. Ambos servicios, implementados con tecnologías y protocolos ya conocidos en el mundo Internet, al ser llevados a este entorno vehicular descubrimos ciertas incompatibilidades entre ellos. Afortunadamente, el RFC 4877 [96] establece las adaptaciones que deben realizarse en ambos servicios para que puedan interoperar entre ellos. Sin embargo, al seguir este proceso se han descubierto ciertas situaciones inconsistentes no contempladas por el estándar, situaciones a las que proponemos distintas soluciones.

A lo largo de la Tesis se constata mediante grandes campañas de pruebas que el comportamiento de la comunicación durante los traspasos es mejorable. Proponemos por tanto distintos frentes por donde atacar a este problema, consiguiendo finalmente traspasos limpios y sin cortes. En este capítulo se introduce este trabajo, que se detalla en el Capítulo 6.

3.1. Nuestra Propuesta de Arquitectura de Comunicaciones

La arquitectura de comunicaciones propuesta en la presente Tesis usa como base el protocolo IPv6 y varias de sus múltiples tecnologías. En la Figura 3.1 se muestra una vista esquemática simplificada de nuestra propuesta de arquitectura de red, que sigue las especificaciones de la arquitectura de referencia definida por el ISO/ETSI [8,9]. La nomenclatura de las entidades que componen dicha arquitectura tiene dos vertientes: por un lado las derivadas del los estándares establecidos por el IETF en sus RFCs, y por otro lado las establecidas por ISO/ETSI en su arquitectura de referencia para entornos ITS. Intentaremos hacer referencia a ambas nomenclaturas y así establecer ese paralelismo entre ambos estándares. En el diagrama presentado en la Figura 3.1 podemos distinguir tres entidades principales:

- Estación Vehicular ITS (*Vehicle ITS-S*), formada por un *ITS-S router* que hace las veces de Router Móvil (MR), y también por uno o varios *ITS-S hosts* que son los dispositivos de abordo conectados a la red móvil que despliega.
- Estación de Carretera ITS (*Roadside ITS-S*), que provee conectividad inalámbrica a los vehículos a lo largo de la carretera para poder conectarse a la red de la concesionaria, aparte de tener ciertas capacidades para procesamiento de datos y de enrutamiento de tráfico. En el diagrama viene condensado en una sola entidad de tipo *ITS-S router*.
- Estación Central ITS (*Central ITS-S*), que incluye los *ITS-S hosts* necesarios para proveer de servicios a los vehículos. Además incluye un *ITS-S router* haciendo las veces de *Home Agent*, necesario para el servicio de movilidad, seguridad y enrutamiento del tráfico, como veremos.

La pila de comunicaciones mostrada para cada nodo sigue el esquema de capas ISO/ETSI que considera, de abajo a arriba, las siguientes capas: tecnologías de acceso, red y transporte, facilidades y aplicaciones. Estas capas están acompañadas por dos planos transversales para las tareas de gestión y seguridad, como vimos simplificado en la Figura 2.1 del Capítulo 2.

La capa de tecnologías de acceso incluye una gran variedad de tecnologías de telecomunicaciones, en su mayoría de naturaleza inalámbrica (Wi-Fi, WiMAX, 802.11p, 3G), que nos permiten comunicarnos con otras estaciones ITS. Estas tecnologías pueden usarse simultáneamente y varían dependiendo del cometido de cada tipo de estación ITS. Independientemente de las tecnologías soportadas por el estándar, la incorporación de nuevas tecnologías no supondrá un impacto en el resto de la arquitectura gracias a la estratificación por capas.

La capa de red y transporte agrupa a todos los protocolos de red y transporte encargados de transmitir los mensajes que se intercambian las diferentes Estaciones ITS. Aunque ya existen protocolos específicos para ITS como FNTP (*Fast Networking*

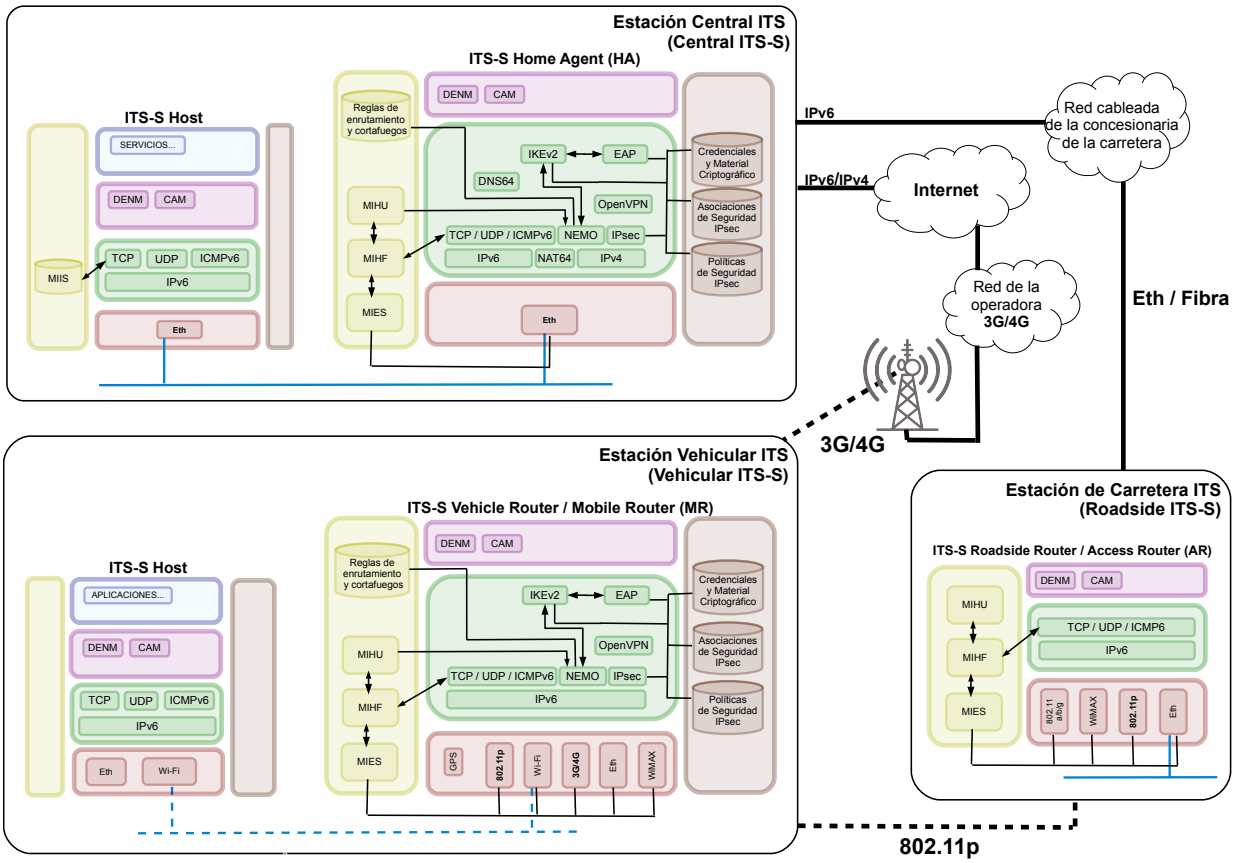


Figura 3.1: Arquitectura de comunicaciones basada en tecnologías IPv6

and Transport layer Protocol) definidos por el grupo de trabajo ISO TC204 en el documento ISO 29281 [97], o el ya mencionado *GeoNetworking* [11, 81], el estándar es suficientemente flexible para aceptar otros protocolos de red y transporte en el futuro. De hecho, IPv6 se está contemplando ya como una alternativa real a los protocolos antes mencionados. Prueba de ello es el trabajo realizado por el mismo grupo de trabajo ISO TC204 en el documento ISO 21210 [79] donde se especifica cómo combinar IPv6 y sus protocolos asociados para dar soporte de red en entornos C-ITS. El interés de las redes vehiculares por IPv6 es precisamente por la madurez del protocolo y sus tecnologías asociadas, como los servicios de movilidad y seguridad, ya ampliamente desarrollados en el mundo de Internet. También es evidente la gran ventaja que supone el uso de IPv6 a la hora de que dichas redes vehiculares puedan ser incorporadas al Internet del futuro, donde IPv6 es uno de los pilares fundamentales.

La capa de facilidades hace de intermediario entre la capa de red y transporte y la capa de aplicaciones, ofreciendo a estas capas acceso a la información intercambiada

entre las distintas estaciones ITS, implementando servicios de soporte a las aplicaciones y liberando a las mismas de realizar esta tarea. El beneficio inmediato que encontramos es el poder compartir dicha información y servicios con todas las aplicaciones. El intercambio de esta información debe realizarse de forma estandarizada. Ejemplos de facilidades son el *Cooperative Awareness Message* (CAM) [82] y el *Decentralized Environmental Notification Basic Service* (DENM) [83]. Con estos servicios la estación ITS puede emitir o recibir información de interés para ella y para las estaciones que la rodean (mensajes CAM) y también intercambiar notificaciones basadas en la posición geográfica de las Estaciones Vehiculares ITS (mensajes DENM).

En la capa de aplicación residen las aplicaciones ITS, todas aquellas que ayuden a conseguir los objetivos de aportar seguridad, eficiencia del tráfico e informar y ofrecer servicios de valor añadido a los ocupantes de los vehículos. Estas aplicaciones pueden hacer uso, como acabamos de decir, de los servicios ofrecidos por la capa de facilidades.

Como hemos adelantado antes, estas capas están rodeadas por dos planos transversales, lo que quiere decir que pueden ser accedidas desde cualquiera de las capas mencionadas. Uno de los planos se dedica a tareas de gestión, encargado de funciones como agregador de reglas de filtrado y ruteo, además de la selección de la mejor interfaz al transmitir datos (basándose para ello en los requisitos de las aplicaciones, las características de las tecnologías de acceso y de las condiciones de la red). El segundo plano transversal desempeña tareas de seguridad, encargado de proporcionar funcionalidades de seguridad a las capas anteriores, sirviendo de repositorio de credenciales, certificados y material criptográfico. También es capaz de aportar funcionalidades atómicas de seguridad tales como la generación de números pseudo-aleatorios, resumen digital, firma digital, cifrado y descifrado.

Todas estas capas y planos están comunicados entre sí a través de los *Service Access Points* o Puntos de Acceso al Servicio (SAPs), puntos bien definidos por donde interactúan los módulos residentes en diferentes planos y capas del estándar. Los planos tienen SAPs con todas las capas, sin embargo, las capas sólo tienen SAPs con sus adyacentes.

A continuación vamos a ver con más detalle nuestra propuesta de arquitectura, analizando cada una de las estaciones ITS presentadas.

3.1.1. Estación Vehicular ITS

En el vehículo, la funcionalidad de la pila está dividida en dos nodos: *ITS-S host* y *ITS-S router*, este último conocido también como *Mobile Router* (MR). Éste es capaz de desplegar dentro del vehículo una red inalámbrica usando tecnología WiFi (802.11b/g/n) para ofrecer acceso a los dispositivos (*ITS-S hosts*) que pudieran estar conectados dentro. Estos dispositivos no son conscientes de los cambios de interfaz y traspasos que efectúa el MR cuando el vehículo está en movimiento. Para la comunicación exterior con la Estación Central ITS, el MR viene provisto con distintas tecnologías inalámbricas, como son la tecnología 3G que nos conecta a la estación central ITS vía Internet, y la tecnología IEEE 802.11p (compatible con ITS-G5) que permite conectarnos con las Estaciones de Carretera ITS que, a través de ellas, nos

darán acceso a la Estación Central ITS.

Para ofrecer conectividad IPv6 nos hace falta un conjunto de elementos incluidos en las capas de red y transporte del MR. Por un lado, el protocolo *Network Mobility Basic Support* [98], en adelante NEMO, es el encargado de mantener alcanzable y dotar de conectividad la red móvil IPv6 interna que se despliega en el vehículo. Adicionalmente, para ser capaces de usar varios interfaces del MR al mismo tiempo (del inglés *multi-homed*), se le ha añadido a NEMO la extensión llamada *Multiple Care-of Address Registration* [48], en adelante MCoA. Esta extensión nos permite utilizar más de una dirección IP para dar un servicio de movilidad más continuo. Por otro lado, el MR está equipado con los elementos necesarios para proteger todo el tráfico que el servicio de movilidad genere, ya sea de control o los datos en sí. Esta protección se realiza por medio del protocolo de seguridad IPsec [88], gracias a la creación automática de asociaciones de seguridad por parte del servicio de seguridad *Internet Keying Exchange version 2* [31], en adelante IKEv2.

Los *ITS-S hosts* conectados a la red desplegada por el *ITS-S router* de la Estación Vehicular ITS se encargan de ejecutar aplicaciones finales localizadas en la capa destinada para ello. Estas aplicaciones pueden colaborar remotamente con otras aplicaciones que estarían ejecutándose en otros *ITS-S hosts* dentro de la Estación Central ITS. Como se puede observar, en las dos posibles entidades que conforman la Estación Vehicular ITS, la pila de comunicaciones incluye una capa de transporte típica de Internet, donde coinciden los protocolos Transport Control Protocol (TCP) y User Datagram Protocol (UDP). La capa de facilidades incluye módulos de envío y recepción de mensajes CAM y DENM, aparte de otras funcionalidades, para hacer más fácil la implementación de aplicaciones y servicios.

3.1.2. Estación de Carretera ITS

La pila de comunicación instanciada en el *ITS-S router*, o también llamado *Access Router* (AR), de la Estación de Carretera ITS, actúa como punto de acceso para los vehículos, usando para ello tecnologías de corto y medio alcance. En este caso, las tecnologías inalámbricas disponibles son WiFi (802.11b/g/n) y 802.11p ITS-G5 compatible.

Estas estaciones contribuyen en la implementación de los servicios aportados por la capa de facilidades. Entre ellos destacamos el soporte de los mensajes CAM y DENM, ya que estas estaciones son intermediarias entre las Estaciones Vehiculares ITS y la Estación Central ITS, ya que todos los mensajes CAM y DENM pasan a través de ellas.

3.1.3. Estación Central ITS

Finalmente, en la parte alta de la Figura 3.1 se puede apreciar la Estación Central ITS y, dentro de ella, las pilas de comunicación de un *ITS-S host* ejecutando aplicaciones que dan soporte a los servicios ofrecidos a los vehículos, y un *ITS-S router* haciendo las veces de *Home Agent* (HA), entidad necesaria para ofrecer el servicio de

movilidad. Por esta razón, los módulos incluidos en la capa de red del HA son los mismos que los incluidos en el MR del vehículo, ya que son entre estas dos entidades donde, aparte de la movilidad, además se aplica el servicio de seguridad por medio de IPsec sobre IPv6.

3.2. Distribución de Protocolos IPv6 sobre la Arquitectura

Puede apreciarse que el principal objetivo en esta arquitectura que proponemos es obtener una comunicación segura de vehículo a infraestructura (V2I), usando para ello una solución basada en IPv6 y sus servicios asociados de seguridad y movilidad. También, como cualquiera podría concluir, las comunicaciones vehículo-a-vehículo (V2V), pese a ser de forma indirecta, son completamente posibles ya que todos los nodos están globalmente accesibles a través de IPv6. A continuación se va a determinar dónde va a ir localizado cada servicio IPv6 dentro de nuestra propuesta de arquitectura.

3.2.1. Servicio de Autoconfiguración

Para la señalización necesaria para establecer las conexiones entre Estaciones ITS sobre IPv6, disponemos de protocolos que realizan ese cometido. Los más importantes son *IPv6 Internet Control Message Protocol* [99] (ICMPv6) y *IPv6 Neighbour Discovery Protocol* [100] (ND). Estos protocolos, específicamente diseñados para dar soporte a IPv6, vienen a realizar las funciones relacionadas con la comunicación entre los dispositivos conectados a una misma red local, funciones antes realizadas por los protocolos *Address Resolution Protocol* [101] (ARP) e ICMP sobre IPv4. Es por esto que ND e ICMPv6 son considerados como protocolos de soporte para IPv6, además de tener una relación muy cercana entre ellos, ya que se utiliza un subconjunto de tipos de mensajes de ICMPv6 para implementar las funciones que ofrece ND. Entre estas funciones, destacamos las siguientes por ser determinantes para nuestra propuesta:

- *Router Discovery* (descubrimiento de puertas de enlace), que mediante los mensajes *Router Solicitation* y *Router Advertisement*, permiten a las Estaciones Vehiculares ITS descubrir la presencia de Estaciones de Carretera ITS.

- *Mobile Prefix Discovery* (descubrimiento de prefijo móvil), necesario para implementar el servicio de movilidad, permite a las Estaciones Vehiculares ITS obtener información de su *red home* que le permita establecer su *home address* (HoA).

- *Address Resolution* (resolución de direcciones), que permite a cualquier Estación ITS obtener la dirección de enlace de cualquier otra Estación ITS conectada a la misma red local y permitir así comunicarse entre sí. Se realiza mediante los mensajes *Neighbour Solicitation* y *Neighbour Advertisement*.

- *Neighbour Unreachability Detection* (detección de vecino inalcanzable), función que previene el envío de mensajes a Estaciones ITS vecinas que se sabe de antemano que no están alcanzables, evitando así hacer un uso inútil de la red. Para ello, todas las Estaciones ITS mantendrán una caché de vecinos, que será actualizada según se reciban mensajes *Neighbour Advertisement* o *Router Advertisement*, informándonos de las Estaciones ITS vecinas que nos rodean y están alcanzables, directa o indirectamente, en este último caso a través de la puerta de enlace por defecto.

- *Stateless Address Autoconfiguration* (autoconfiguración de direcciones), función que implementan todas las Estaciones ITS nada más conectarse a la red. En IPv6 se establecen dos tipos de direcciones: locales y globales. Las locales se generan inmediatamente para cada interfaz de red, normalmente a partir de la dirección MAC de la tarjeta de red. Mediante los mensajes *Neighbour Solicitation* y *Neighbour Advertisement* se determina si la dirección autogenerada está ya en uso o está libre. A este proceso se le denomina *Duplicated Address Detection* (DaD). Si se recibe un mensaje *Router Advertisement* con información sobre el enrutador de la red y el prefijo a utilizar, el proceso de autoconfiguración genera una dirección, esta vez de caracter global, a partir de este prefijo ofrecido por el enrutador de la red. De nuevo se lanza el proceso de detección de duplicados que, de encontrarlos, se propondría una nueva dirección hasta que dicha prueba DaD devuelva negativo. Una vez establecida la dirección global, se configurará automáticamente al enrutador que envió el mensaje *Router Advertisement* como puerta de enlace por defecto, completando así la configuración de la Estación ITS. Esta función es especialmente interesante para implementar soluciones de movilidad, ya que permite a las Estaciones Vehiculares ITS moverse entre distintas redes y obtener direcciones válidas sin tener ningún conocimiento previo del esquema de direccionamiento utilizado por cada una de ellas.

Estos protocolos suelen estar presentes en todos los tipos de Estaciones ITS existentes, ya que son fundamentales para el correcto funcionamiento del protocolo IPv6. Lo localizaremos dentro de cada Estación ITS en la capa de red y transporte, al igual que IPv6. En la Figura 3.2 podemos ver la presencia de estos protocolos en nuestra propuesta de arquitectura.

3.2.2. Servicio de Movilidad

El servicio de movilidad es aquel que permite a las Estaciones Vehiculares ITS usar una misma dirección de red, a pesar de poder estar conectado a través de diferentes redes de acceso con diferentes rangos de direccionamiento. Este servicio es gestionado por el protocolo NEMO, que permite la movilidad de redes enteras mediante el uso de *Mobile Routers* (MRs), que están localizados en cada Estación Vehicular ITS según nuestra propuesta. Con esto se consigue hacer que la red interna desplegada en cada vehículo pueda beneficiarse del mismo servicio de movilidad, no afectando así a los

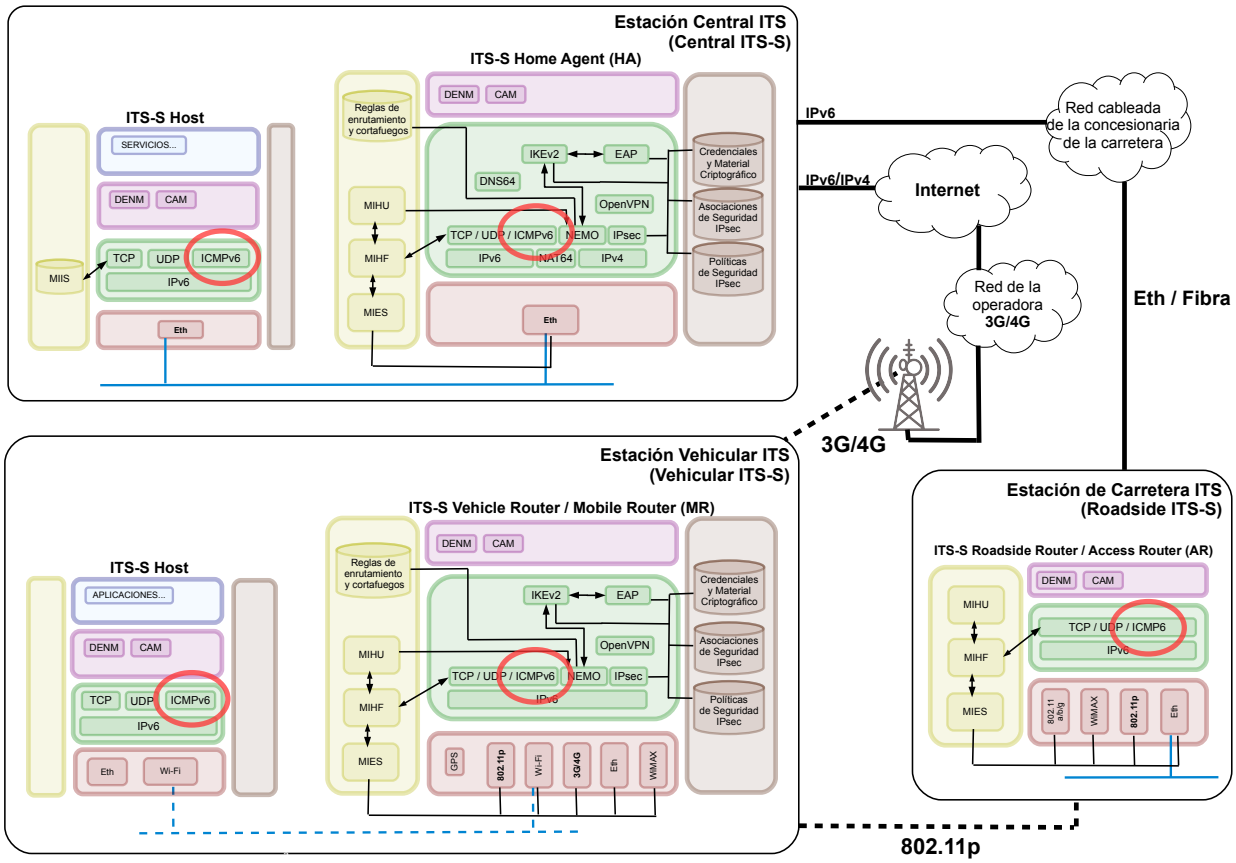


Figura 3.2: Presencia de ICMPv6 en nuestra propuesta de arquitectura

posibles nodos de dicha red los posibles cambios de direccionamiento que se producen cuando la Estación Vehicular ITS va cambiando de una red a otra.

Para ello también nos hace falta otra entidad localizada en la Estación Central ITS, que llamaremos *Home Agent* (HA), que es el responsable de enviar el tráfico a los MRs registrados. Por la forma de funcionar de la movilidad a base de túneles IP establecidos entre HA y MRs, el módulo NEMO que implementa dicha movilidad sólo necesita estar presente en estas entidades, localizado en ambos en la capa de red y transporte, como podemos observar en la Figura 3.3

Los mensajes utilizados por este protocolo para el establecimiento de la movilidad son los llamados Binding Update (BU) y Binding Acknowledgement (BA). Su cometido es permitir al MR registrar en el HA una nueva CoA en el momento en el que le es asignada. El mensaje BU es un paquete IPv6 al que se le añade una cabecera de extensión llamada *Mobility Options*, donde se añadirán las opciones necesarias. En esta ocasión necesitaremos incluir la opción llamada *Home Address destination option*,

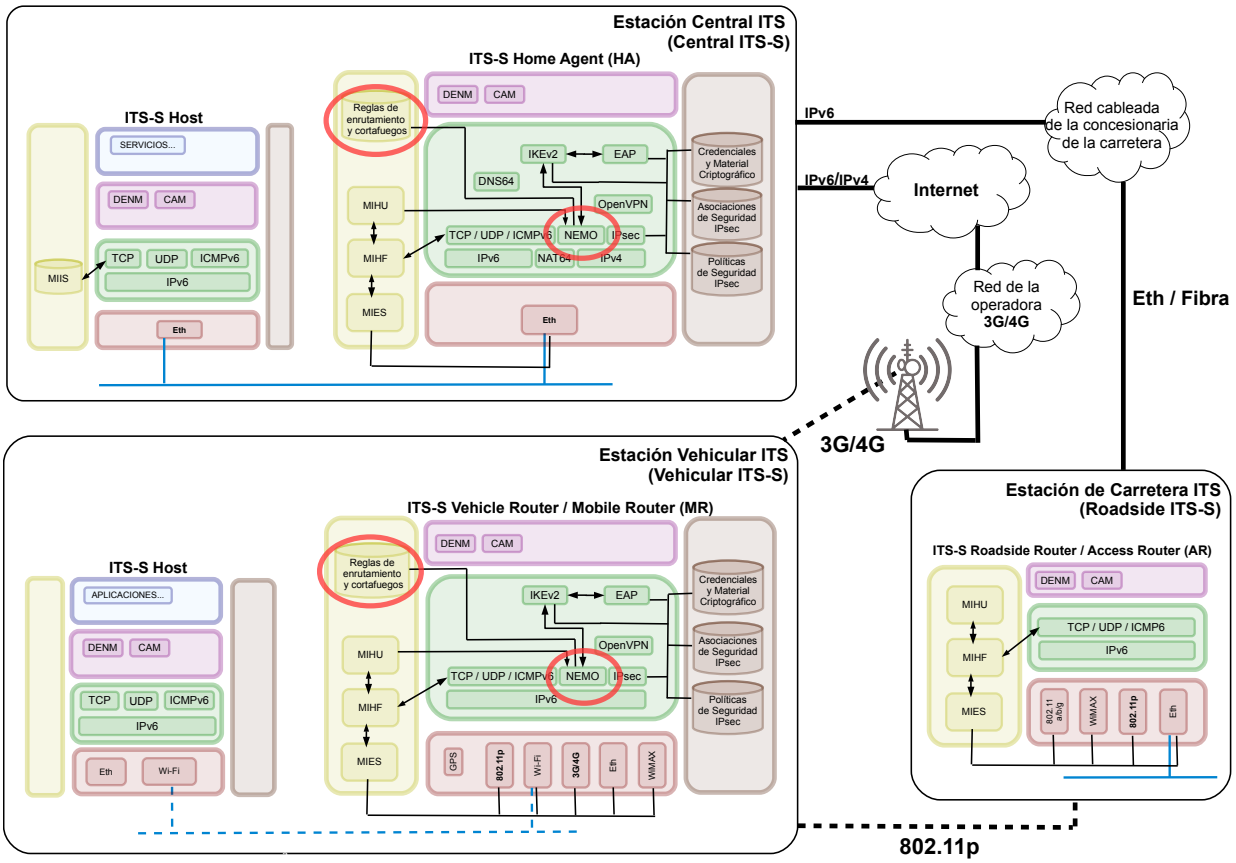


Figura 3.3: Presencia de NEMO y MCoA en nuestra propuesta de arquitectura

que permite usar la CoA como origen del paquete IP y añadir en esta opción la HoA a la que quiere ir asociada dicha CoA. Esto evita problemas con los cortafuegos de las redes que va visitando la Estación Vehicular ITS, ya que colocar como dirección de origen la HoA puede ocasionar un filtrado del tráfico debido al conocido *Ingress Filtering* [102].

En estos mensajes BU y BA además hay una serie de bits reservados para informar de ciertas circunstancias que pueden hacer cambiar el comportamiento del protocolo y realizar distintas acciones. Sin querer profundizar demasiado por el momento en cada uno de ellos, es necesario destacar los siguientes:

- El bit K, presente tanto en BU como en BA, está destinado a informar al otro extremo de la capacidad de poder establecer asociaciones de seguridad IPsec para la protección del tráfico, así como establecer la CoA en cuestión como IP con la que negociar dichas asociaciones de seguridad mediante el protocolo IKEv2.

- El bit R, presente tanto en BU como en BA, indica que cada extremo implementa las extensiones NEMO, es decir, que no se está usando movilidad básica sino movilidad de redes. En el caso de ir desactivado estaría funcionando como lo haría el protocolo MIPv6, antecesor de NEMO, donde sólo hay movilidad de nodos individuales, no de redes.

Como hemos visto, incorporar el bit R indica que existe movilidad no sólo de un nodo (MR con una determinada CoA), sino de una red entera. Por tanto, aparte de la CoA, también debemos de establecer un prefijo para definir dicha red, e informar de ella al HA a través del mismo mensaje BU, añadiendo en esta ocasión otra opción de movilidad en la cabecera de extensión llamada *Mobile Network Prefix option*.

Además, nuestra implementación de NEMO lleva incorporada la extensión MCoA, que nos permite el uso simultáneo de diferentes CoAs al mismo tiempo, tantas como interfaces de red tengamos en el MR de la Estación Vehicular ITS. En nuestro caso, de todas las tecnologías disponibles, hemos podido disponer de dos de ellas: 3G y 802.11p. Por tanto, se trata de poder usar ambas tecnologías simultáneamente mientras estén disponibles. Esto introduce algunos cambios en la forma de funcionar de NEMO, pues para identificar cada CoA asociada a cada interfaz, necesitará asociar además un identificador único para cada uno llamado BID. Por tanto, este BID debe ser incorporado a la estructura de la caché del HA y también tiene que ser notificado en el mensaje BU y BA. Para ello, se añade una opción más de movilidad llamada *Binding Identifier mobility option*. El MR establecerá el valor del mismo. En cambio, el HA responderá con el mismo valor si soporta la extensión MCoA. En caso contrario, el mensaje BA enviado por el HA no incorporaría la opción con el BID, dando entender al MR que no dispone de dicha extensión.

Al disponer de varias vías simultáneas por las que comunicarnos, nos surge el problema de elegir cuál de ellas usar en cada momento. Para determinar qué tráfico enviar a través de una interfaz u otra, NEMO debe marcar cada paquete del tráfico del servicio de movilidad con el BID correspondiente a la interfaz deseada, y establecer también unas reglas en el cortafuegos que examinen esas marcas y dirijan cada paquete a su interfaz correspondiente. En el caso de no ir marcado un paquete, se establece una interfaz por defecto a la que redirigir todo el tráfico no marcado. Estas reglas son almacenadas dentro del plano de gestión según la arquitectura de referencia ISO/ETSI, ya que precisamente una de las funciones de este plano es la de decidir qué interfaz usar en cada momento dado. Todo esto se verá con más detalle más adelante en el Capítulo 5.

3.2.3. Servicio de Seguridad

El servicio de seguridad es entendido aquí como el que otorga la capacidad de dotar a las comunicaciones de tres elementos clave: confidencialidad, integridad y autenticidad. Estos pueden ofrecerse fundamentalmente a los niveles siguientes:

- a **nivel de enlace**, donde la tecnología cifra los datos transmitidos con algoritmos propios donde además se proporciona una fase de autenticación para acceder a la

tecnología de comunicaciones en cuestión. En WiFi, son conocidos los esquemas de cifrado WEP, WPA y WPA2, y el sistema de autenticación 802.1x. Para el caso de 802.11p los datos se transmiten en abierto y no hay fase de autenticación debido a su diseño de establecimiento rápido de conexiones. En cambio, para el caso de 3G sí se dispone de esquemas de cifrado y autenticación tanto del terminal como de la estación base.

- a **nivel de red**, donde se permite o no al nodo acceder a la red IP. Para poder pertenecer a una red IP hace falta tener una dirección IP que no esté filtrada por el enrutador de la red. En redes IPv6 suelen darse sistemas de auto-configuración que otorgan direcciones IP sin hacer ningún tipo de comprobación. Sin embargo no tendrán acceso a la red si no es a través de los servicios combinados de seguridad y movilidad. En el caso de IPv4 e IPv6 se provee de confidencialidad e integridad a los datos mediante IPsec, que gracias a sus dos cabeceras AH y ESP protegen los datos transmitidos. En cuanto a la autenticación, esta se realiza en el momento de establecer las asociaciones de seguridad IPsec mediante el servicio IKEv2. Este servicio dispone de métodos propios para autenticarse en ambas direcciones basados en firma digital, aunque también soporta el transporte de EAP [103] (*Extensible Authentication Protocol*), que posibilita la utilización de algoritmos compatibles con EAP, como EAP-TLS [104].

- a **nivel de mensaje**, muy común en redes que no ofrecen protección a nivel de enlace o de red, la protección suele darse en cada mensaje, mediante la aplicación de esquemas criptográficos asimétricos. En redes vehiculares es muy común encontrar este tipo de protección.

- a **nivel de aplicación**, donde serán las propias aplicaciones las que decidan qué esquemas criptográficos utilizar para proteger los datos de se intercambian entre ellas. Estas suelen ser orientadas a sesión, como SSL, TLS, etc.

Nuestra propuesta de arquitectura viene a proporcionar un tipo de protección a nivel de red orientada a sesión mediante el uso de criptografía simétrica gracias a IPsec e IKEv2, muy diferente a lo visto hasta ahora en redes vehiculares, normalmente basados en protección orientada a mensajes, donde la criptografía asimétrica se ha demostrado ineficiente a la hora de enviar grandes cantidades de datos, como los necesarios en aplicaciones multimedia, por ejemplo. Nuestro principal objetivo aquí es proteger todo el tráfico generado por el servicio de movilidad entre el HA y los diferentes MRs. Por tanto, los protocolos asociados a la seguridad sólo estarán presentes en dichas entidades. Como hemos dicho antes, estos protocolos están localizados en la capa de red y transporte de la arquitectura de referencia ISO/ETSI. Sin embargo, las bases de datos donde se almacenan políticas y asociaciones de seguridad, así como el material criptográfico utilizado se sitúan en el plano de seguridad, como hemos destacado en la Figura 3.4.

Como es de esperar por lo visto anteriormente, ambos servicios de seguridad y movilidad deberán cooperar para poder desempeñar correctamente sus funciones, hecho que abarcamos en profundidad en el Capítulo 5.

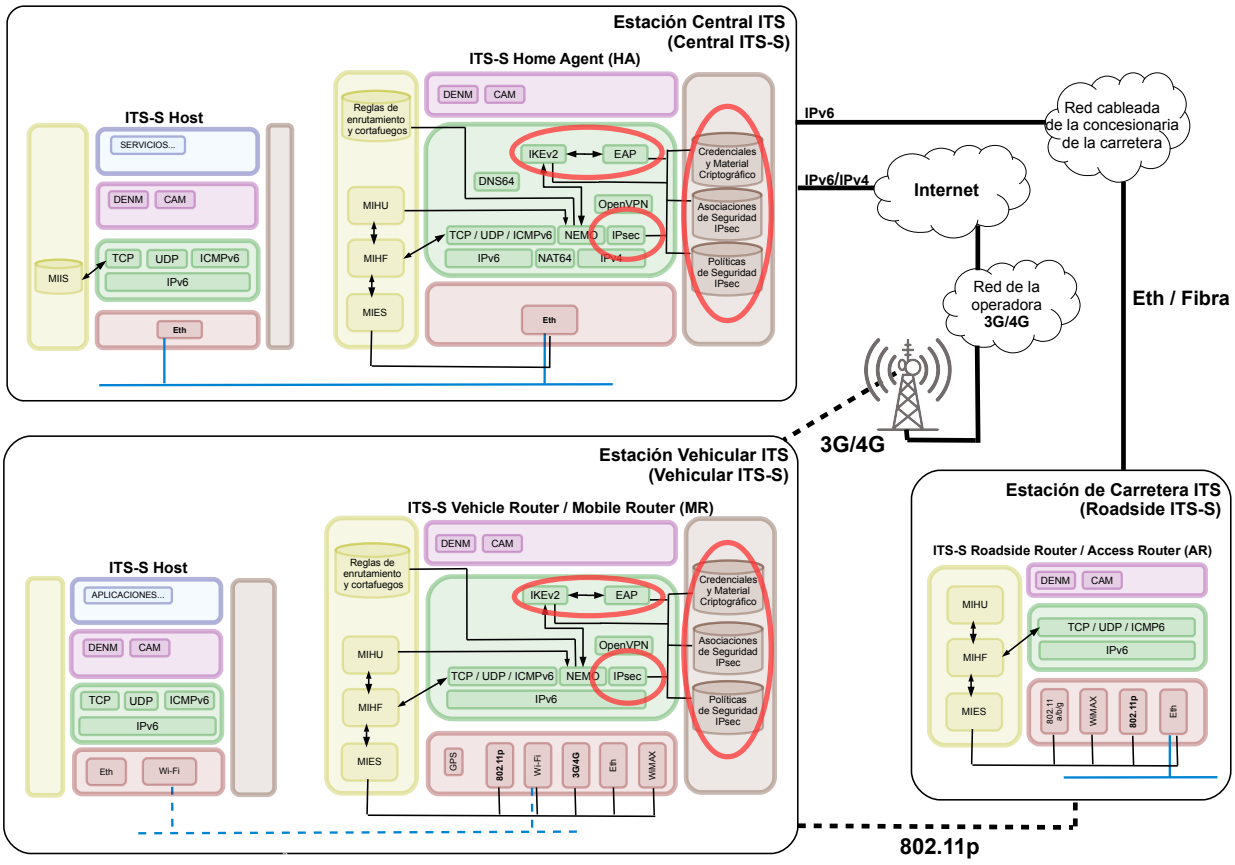


Figura 3.4: Presencia de NEMO y MCoA en nuestra propuesta de arquitectura

Centrándonos ahora en la seguridad, para que IPsec pueda funcionar correctamente, hay que establecer previamente los siguientes puntos dentro de estas entidades:

- Determinar qué tráfico deseamos proteger. Esto se hace a través del establecimiento de políticas de seguridad IPsec, las cuales nos dan una gran flexibilidad para determinar qué tráfico debe ir seguro y cual no. En nuestro caso, queremos que el tráfico tanto de control como de datos del servicio de movilidad vaya protegido. Por tanto habrá que establecer las políticas necesarias para ello, e impedir que otro tipo de tráfico no deseado pueda fluir. Además en estas políticas se establecen también parámetros tales como los algoritmos a utilizar, tipos de cabecera IPsec (AH o ESP), entre otros. Estas políticas son almacenadas en una base de datos (SPD) situada en el plano de seguridad según la arquitectura de referencia ISO/ETSI, tanto en el HA como en cada MR.

- Determinar qué material criptográfico utilizar para hacer funcionar a los

algoritmos de cifrado e integridad y establecer su caducidad, ya sea por tiempo o por cantidad de datos transmitidos. Esta información viene recogida en las llamadas asociaciones de seguridad IPsec, que normalmente vienen ligadas a una determinada política IPsec. Son almacenadas en otra base de datos (SAD), también situada en el plano de seguridad de la arquitectura de referencia ISO/ETSI, tanto en el HA como en cada MR. Estas asociaciones pueden establecerse a mano, pero en nuestro caso utilizamos un mecanismo automático de establecimiento de asociaciones bajo demanda, gracias al protocolo de seguridad IKEv2. Este protocolo nos permite establecer y actualizar las asociaciones de seguridad necesarias para proteger el tráfico de movilidad.

La autenticidad, uno de los tres elementos indispensables que un servicio de seguridad debe aportar, la conseguimos en nuestra propuesta mediante el propio IKEv2, que posee en su definición diferentes mecanismos de autenticación de los extremos de la negociación, que en este caso son el HA y todos los posibles MRs. Esta segunda versión de IKE soporta el transporte de EAP, protocolo que permite utilizar cualquier método de autenticación basado en él, de tal forma que podemos elegir el más conveniente para escenarios de tipo ITS. Sin embargo, a pesar de estar abierta esta posibilidad, en las pruebas de nuestra implementación de la arquitectura que proponemos, no hemos usado estos tipos de métodos de autenticación, eligiendo sin embargo el más rápido y sencillo disponible en IKEv2 basado en firma digital y claves pre-compartidas. La habilidad de cambiar de protocolo de autenticación es uno de los frentes a la hora de mejorar los tiempos de traspaso, que desarrollaremos en el Capítulo 6.

3.2.4. Asistencia al Traspaso

El traspaso, del inglés *handover*, es un concepto asociado a tecnologías inalámbricas, y se produce en nuestro caso cuando la Estación Vehicular ITS cambia de tecnología para comunicarse con la Estación Central ITS. Recordamos que disponemos para nuestro escenario de pruebas de dos tecnologías: 3G y 802.11p. Como profundizaremos más adelante en el Capítulo 6, las pruebas que hemos efectuado han demostrado que los traspasos de una tecnología a otra eran deficientes, con pérdida de conectividad momentánea. Esto nos llevó a la conclusión de que este proceso de traspaso debía ser asistido para, en definitiva, realizarlos en el mejor momento posible y evitar las pérdidas de conectividad comentadas. Para nuestra propuesta hemos elegido el estándar IEEE 802.21, que define un protocolo entre diferentes entidades que están distribuidas en distintas Estaciones ITS de la red, dependiendo de la funcionalidad que desempeñen. Estas colaboran para asistir el proceso de traspaso. En particular, se definen:

- El MIH *Mobile Node* (MIH-MN), que en nuestra propuesta estará localizado en la Estación Vehicular ITS, más concretamente en el MR. Todos los hosts conectados a la red móvil desplegada por el MR no necesitan ser conscientes de los procesos que acontecen en los traspasos, y por tanto no requieren de capacidades MIH.

- Los MIH-MN se comunicarán con los MIH *Point of Service* (MIH-PoS) a través de los llamados MIH Point of Attachment (MIH-PoA). Ambos estarán implementados por las Estaciones de Carretera ITS localizadas a lo largo de la carretera formando parte de la infraestructura de red.
- Otra entidad importante que define el estandar es el servidor *Media Independent Information Server* (MIIS), que estará localizado en la Estación Central ITS, en cualquier *ITS-S host*, almacenando información actualizada del estado de la red recogida a través de las entidades MIH.

En cuanto a la integración de IEEE 802.21 en nuestra pila de comunicaciones, la decisión más importante que se ha tomado al respecto ha sido la colocación de los módulos *Media Independent Event Service* (MIES), *Media Independent Handover Function* (MIHF) y *Media Independent Handover User* (MIHU) presentados en el Capítulo 2 en el plano de gestión según la arquitectura de referencia ISO/ETSI, como podemos apreciar en la Figura 3.5. Este plano se encarga de controlar los enlaces de las comunicaciones y, aparte de otras funcionalidades, también del proceso de los traspasos. Por esta razón parece razonable que dichos módulos pertenezcan a este plano ya que es precisamente la función que acometen. Esto además facilita el acceso a dicha funcionalidad desde las distintas capas (acceso a red, red, facilidades y aplicaciones) a través de los SAPs. Esta localización también favorece las funciones MIH, ya que se tendrá acceso directo a los SAPs de la capa de acceso y de red que nos permitirán intercambiar mensajes MIH entre entidades remotas usando para ello la red IPv6 y sus protocolos de transporte, o directamente a través del enlace interactuando con las diferentes tecnologías de acceso. De hecho, las especificaciones establecidas en ISO ya conciben que los módulos localizados en este plano de gestión pueden usar servicios de transporte con el propósito de comunicar diferentes planos de gestión de las distintas Estaciones ITS distribuidas por la red.

Otro aspecto importante es el relacionado con la interacción existente entre MIHF y las diferentes tecnologías de acceso. El módulo MIHF es una capa de abstracción que permite a usuarios del servicio MIH solicitar servicios de traspaso (como por ejemplo activar y desactivar interfaces físicas) sin tener que conocer los detalles particulares de cada tecnología. Es precisamente en el SAP que une el plano de gestión y la capa de tecnologías de acceso donde se realiza esa correspondencia entre los servicios genéricos MIH y las funcionalidades específicas ofrecidas por cada tecnología.

3.2.5. Elementos de Transición de IPv4 a IPv6

Aunque la presencia de IPv6 en este tipo de redes vehiculares no se pone en duda, todavía siguen existiendo redes basadas en IPv4 donde pueden estar alojados algunos de los servicios de los que podamos hacer uso desde las redes C-ITS. Esto supone el uso de alguno de los mecanismos de transición existentes para habilitar el alcance de redes IPv4 desde las estaciones ITS basadas en IPv6. En nuestra propuesta hemos optado por las tecnologías NAT64 [85] y DNS64 [86], ya presentadas en el capítulo anterior. Localizamos estos módulos dentro de nuestra arquitectura en la Figura 3.6.

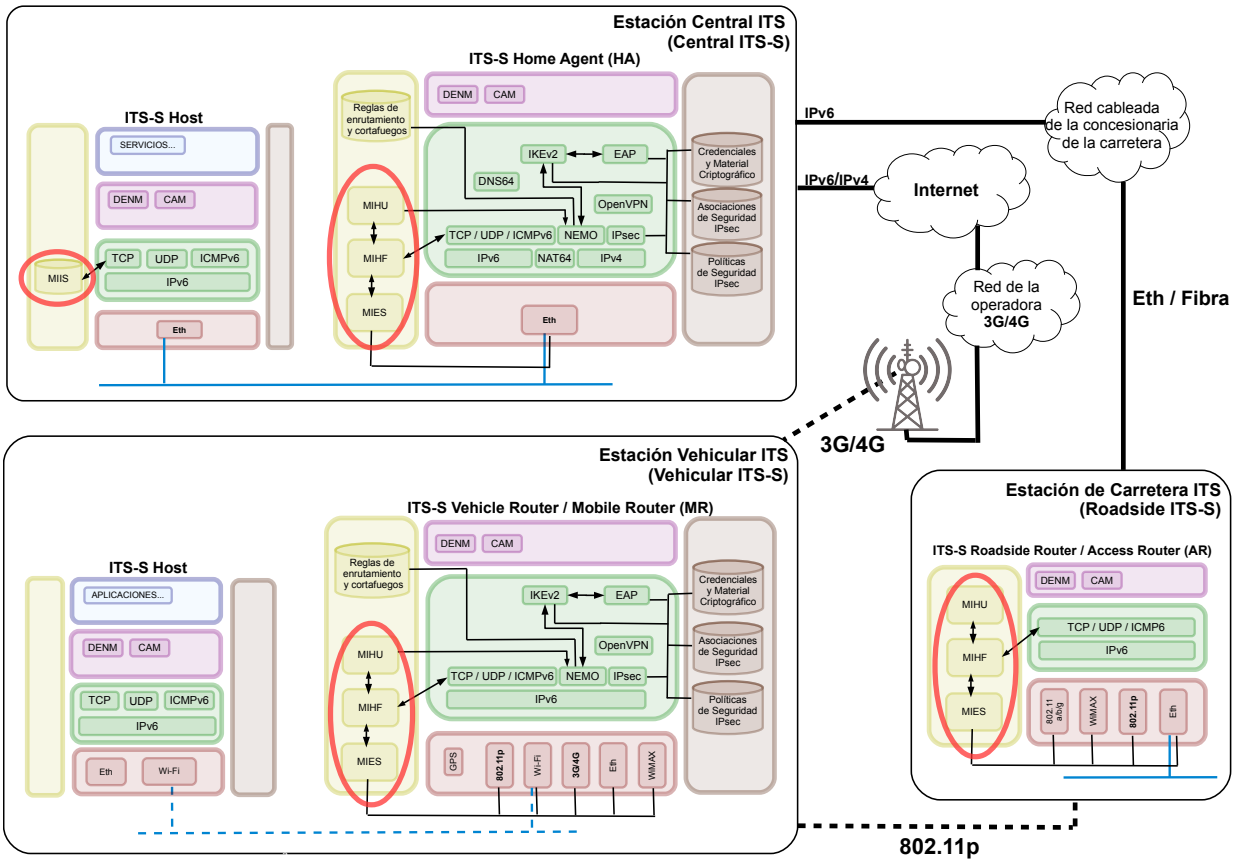


Figura 3.5: Incorporación de los servicios MIH al plano de gestión de la arquitectura de referencia OSI/ETSI

Además, para permitir el uso de las redes públicas basadas en IPv4 mediante la tecnología 3G/4G, se establecen túneles OpenVPN [105] entre la Estación Vehicular ITS y la Estación Central ITS. Esto permitirá al tráfico IPv6 fluir a través de la interfaz 3G de forma transparente.

3.3. Conclusiones

Como hemos visto a lo largo de este capítulo, son muchos los servicios que tenemos que integrar para establecer nuestra propuesta de red vehicular. Es necesario por tanto seguir unas directrices pre-establecidas para poder realizar dicha integración de forma ordenada y consensuada. En consecuencia, nuestra propuesta de red vehicular basada en IPv6, donde se han provisto servicios de movilidad, seguridad y asistencia

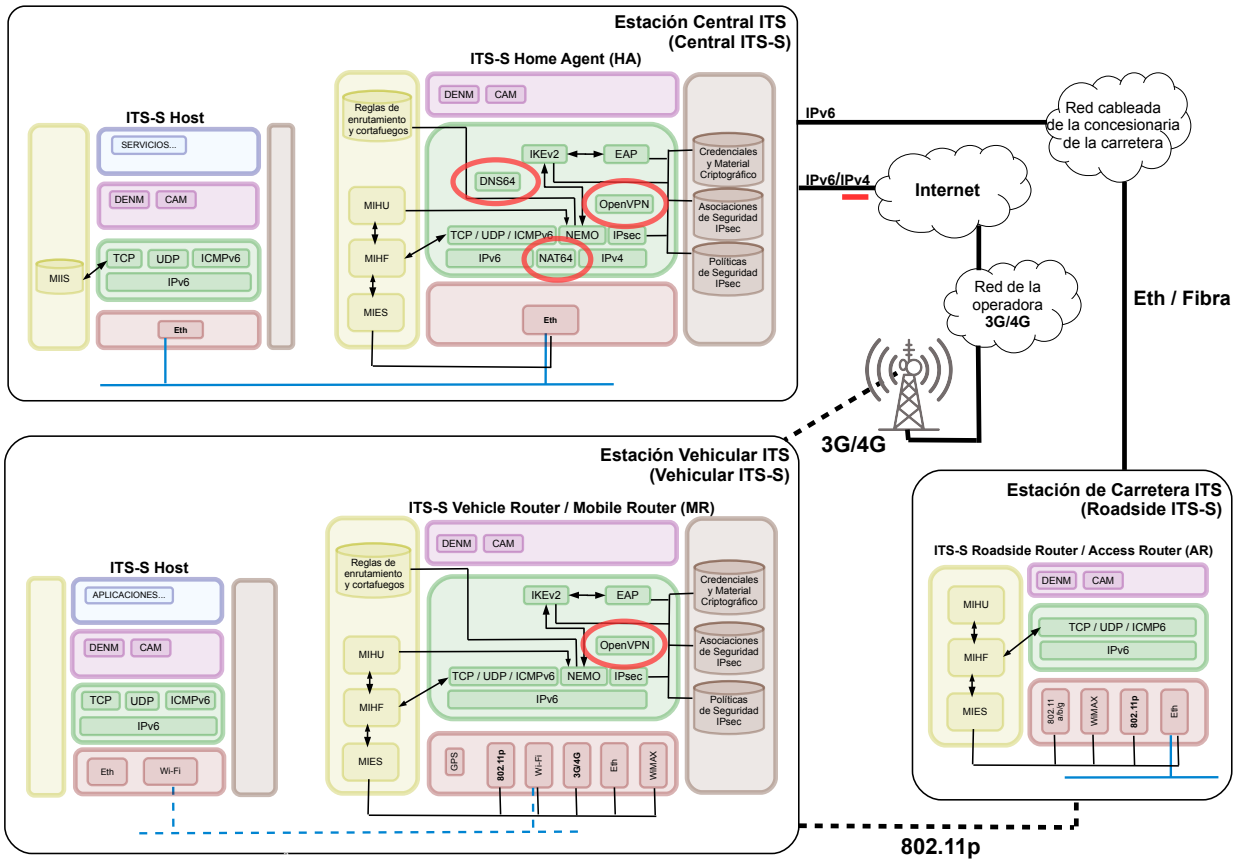


Figura 3.6: Elementos que permiten el uso de IPv4 en nuestra arquitectura basada en IPv6

al traspaso, entre otros, sigue la arquitectura de referencia establecida por los órganos de estandarización ISO/ETSI. Aparte de establecer un orden y vías de comunicación estándar entre los distintos módulos software, una arquitectura de red estándar aporta numerosas ventajas de cara al futuro, ya que asegura una compatibilidad entre propuestas que sigan la misma arquitectura, pudiendo funcionar todas ellas sobre un mismo dispositivo, evitando duplicidades y acumulación de los mismos en el vehículo, que conllevaría a un mayor coste de energía, seguridad y confort. En los siguientes capítulos nos adentramos en los detalles de los principales módulos de comunicación de nuestra propuesta, como son los de seguridad, movilidad y asistencia al traspaso.

Capítulo 4

IKEv2 y Nuestra Implementación OpenIKEv2

Hasta el momento nos hemos adentrado en el mundo de las redes ITS y la propuesta de arquitectura de comunicaciones estándar que proponen los organismos de estandarización ISO y ETSI. Hemos visto el importante papel que juega el protocolo IPv6 y sus tecnologías asociadas a la hora de dar soporte a esta arquitectura, implementando servicios de auto-configuración, seguridad y movilidad. También hemos visto las tecnologías inalámbricas más comunes que nos podemos encontrar en este tipo de redes C-ITS. También hemos constatado la necesidad de un nuevo esquema de seguridad en redes vehiculares orientada a flujos de datos, y no a mensajes individuales. Es aquí donde IPsec e IKEv2 tienen vital importancia para satisfacer esta necesidad, que integramos en nuestra propuesta de arquitectura de red vehicular. Esta integración ha requerido la implementación de algunos de los servicios. En este capítulo nos centraremos en IKEv2 y su implementación de código abierto OpenIKEv2, así como las interesantes aportaciones que se han podido incorporar a los estándares fruto del trabajo dedicado a dicha implementación. Gracias a esta integración podemos realizar pruebas de rendimiento y validación de nuestra propuesta en un entorno real, obteniendo resultados no simulados. En primer lugar presentaremos OpenIKEv2, realizando una comparativa en cuanto a rendimiento con otras implementaciones libres de IKEv2 que también aparecieron en el panorama de software libre. Después expondremos las incidencias encontradas en dicho proceso de implementación y como fueron resueltas, aportando una experiencia útil que permitió complementar la propia definición del estándar de IKEv2.

4.1. Implementaciones de Código Abierto: OpenIKEv2

En el momento de comenzar este trabajo de investigación no existían alternativas de implementaciones que fueran de código abierto, con lo que se decidió aportar a esta comunidad una nueva implementación de IKEv2 realizada íntegramente en

la Universidad de Murcia. La llamamos "OpenIKEv2" [106]. Además surgió como necesidad dentro del Proyecto de Investigación Europeo IST ENABLE [28], donde se demandaba una implementación IKEv2 basada en Linux para ser desplegada en los escenarios donde se requería de sus nuevas funcionalidades. Después fue igualmente utilizado en otros proyectos como ITSSv6 [19]. Todas estas circunstancias permitieron el desarrollo de una implementación de IKEv2 libre basada en Linux desde cero, programada en C++.

Sin embargo, no podemos dejar de mencionar que otros proyectos de implementaciones de IKEv2 de código abierto han ido apareciendo con el tiempo, lo que da valor al propio protocolo IKEv2 y lo establece como protocolo por defecto a la hora de establecer asociaciones de seguridad IPsec. Podemos destacar *Racoon2* [107],*StrongSwan* [108] e IKEv2 [109].

En el Apéndice (B) desarrollamos los detalles que supuso diseñar e implementar nuestra propia implementación IKEv2 (OpenIKEv2). Sin embargo, sí mostraremos a continuación una comparativa entre esta y las implementaciones de código abierto mencionadas antes. Además expondremos también los importantes detalles de implementación que no fueron cubiertos por el documento IETF RFC 4306 en el momento que fue redactado. Estos fueron detectados y solucionados durante la implementación de OpenIKEv2, aportando una información valiosa que sirvió para redactar una serie de aclaraciones y recomendaciones en el documento RFC 4718, que permitió dotar de más rigor a la definición del protocolo IKEv2 para futuras actualizaciones.

4.1.1. Comparativa de Implementaciones de IKEv2 de Código Abierto

En el momento en el que teníamos una primera versión de nuestra implementación OpenIKEv2, otras implementaciones también surgieron: Racoon2 [107], StrongSwan [108] e IKEv2 [109]. Se nos presentó la oportunidad de poder comparar nuestra implementación con la del resto para tener una idea del rendimiento que estábamos obteniendo y si dicho rendimiento era comparable con el del resto de implementaciones. Todas ellas tenían una forma similar de funcionar, es decir, como servicio ejecutado en segundo plano. A pesar de esta similitud, cada implementación tenía diferentes funcionalidades y particularidades. En la Tabla 4.1 se muestran las diferentes características que incluía cada implementación en el momento de ser analizada. Las versiones analizadas estaban en un estado de madurez alto y solamente algunas características faltaban para tener una implementación completa de IKEv2. Las siguientes características estaban soportadas por todas las implementaciones:

- **Cookies**: Mecanismo de protección contra ataques de denegación de servicio (DoS).
- **Negociación de grupo DH, propuestas y selectores de tráfico**: Los participantes pueden negociar el grupo Diffie-Hellman a usar para generar el

secreto compartido, los algoritmos criptográficos y los selectores de tráfico que determinan de todo el tráfico cuál es el que hay que proteger.

- **Métodos de autenticación**: estos están basados en secreto pre-compartido (PSK) y en certificados (PKI).
- **Creación, renovación y eliminación de IKE SAs y IPsec SAs.**
- **Soporte de los modos "túnel" y "transporte" de IPsec.**
- **Soporte de IPv4 e IPv6.**

Sin embargo, había otras características que no estaban implementadas por todas, como son:

- **Estrechamiento de selectores de tráfico (TS narrowing)**: Este es el mecanismo de negociación de selectores de tráfico que busca la parte común de lo que proponen los participantes, haciendo una selección del tráfico lo más precisa posible.
- **Configuración remota de direcciones IP**: Capacidad de los participantes de auto-configurar una dirección IP en el lado del iniciador. Esto suele utilizarse en escenarios "RoadWarrior", donde el iniciador necesita una IP perteneciente a la red donde se encuentra el respondedor para usarla en los túneles IPsec. El respondedor puede proporcionar una dirección IP configurada estáticamente o por medio de una consulta a un servicio DHCP.
- **NAT trasversal**: Mecanismo que permite usar IKEv2 a través de mecanismos de traducción de direcciones (NAT).
- **Soporte de autenticación extensible (EAP)**: Soporte del transporte de un protocolo capaz de proporcionar una forma uniforme y extensible de soportar diferentes métodos de autenticación.
- **Seguridad perfecta hacia adelante (*Perfect Forward Secrecy*)**: En una renovación del material criptográfico se puede usar parte del material que ya se tiene para general el nuevo, o bien, generar un nuevo secreto compartido a través de un intercambio Diffie-Hellman incluido en el intercambio CREATE_CHILD_SA, y usar éste en su lugar para la generación del nuevo material criptográfico. Esto hace que la seguridad perfecta hacia adelante (PFS) se respete, lo que proporciona una renovación del material criptográfico sin depender del antiguo. No usarlo sería solamente excusable si se disponen de pocos recursos de procesamiento y se desea que la renovación sea lo más rápida posible.
- **API para su uso por otros programas**: Esta característica proporciona una vía para que otros programas puedan adquirir la funcionalidad que ofrece IKEv2. Sólo OpenIKEv2 ofrece esta posibilidad, que es uno de sus puntos fuertes.

Característica	1	2	3	4
Cookies	√	√	√	√
Negociación de grupo DH	√	√	√	√
Negociación de algoritmos	√	√	√	√
Negociación de selectores de tráfico	√	√	√	√
Estrechamiento de selección (TS narrowing)	√		√	√
Autneticación basada en PSK	√	√	√	√
Autenticación basada en Certificados	√	√	√	√
Renovación de IPsec SAs	√	√	√	√
Renovación de IKE SAs	√	√	√	√
Eliminación de IPsec SAs	√	√	√	√
Eliminación de IKE SAs	√	√	√	√
Configuración remota de direcciones	√	?		√
NAT Transveral		√	√	√
Transporte de EAP	√		√	√
Modo túnel	√	√	√	√
Modo transporte	√	√	√	√
Perfect Forward Secrecy (PFS)	√	√		√
Soporte IPv6	√	√	√	√
API	√			

(1) OpenIKEv2 v0.94.
(2) racoon2 20070720a.
(3) IKEv2 v2.0 alpha2.
(4) strongSwan v4.1.8.

Tabla 4.1: Tabla comparativa de características IKEv2

La similitud mostrada por todas las implementaciones (con respecto a su nivel de madurez) nos ha permitido analizar con más detalle otros aspectos. Este análisis nos ha llevado a descubrir otras diferencias destacables entre las implementaciones analizadas. En particular, uno de nuestros principales objetivos era mostrar cómo cada implementación de IKEv2 realizaba cada operación, y ver cómo influían ciertas decisiones de diseño e implementación que cada una había tomado por separado. Para conseguir esto, aparte de las conclusiones a las que podamos llegar mediante la observación directa del código fuente, se ha configurado un pequeño escenario de pruebas del que sacar mediciones de tiempo en las negociaciones IKEv2 y comprobar así de una manera empírica el impacto de ciertas decisiones de implementación. Los resultados se muestran con detalle en la siguiente sección.

4.1.2. Escenario de Pruebas para las Diferentes Implementaciones de IKEv2

El escenario de pruebas se ha confeccionado con dos máquinas exactamente iguales con las características especificadas en la Tabla 4.2.

Para cada implementación la hemos instalado en ambas máquinas, una haciendo las veces de iniciador y la otra de respondedor. Se realizan cien negociaciones IKEv2

Tabla 4.2: Características Hardware y Software de los equipos usados en las pruebas

Software	
Sistema Operativo	Linux (Ubuntu Dapper 6.06)
Kernel	2.6.15-27-386 #1 PREEMPT
Hardware	
Constructor de la CPU	CentaurHauls
Modelo de la CPU	VIA Nehemiah
Velocidad de la CPU	1200 Mhz (2401 bogomips)
Caché L2 de la CPU	64 KB
Memoria principal	512 MB
Red	
Tecnología	Ethernet (IEEE 803.3)
Velocidad del enlace	100 Mbps
Configuración IKEv2	
Grupo D-H	2 (MODP1024)
Algoritmo de Cifrado	AES_CBC 128 bits
Algoritmo de Integridad	HMAC_SHA1 96 bits
Algoritmo PRF	HMAC_SHA1
Método de Autenticación	PSK (Secreto pre-compartido)
Configuración IPsec	
Protocolo IPsec	ESP
Modo	Túnel
Algoritmo de Cifrado	AES_CBC 128 bits
Algoritmo de Integridad	HMAC_SHA1 96 bits

(intercambios IKE_SA_INIT e IKE_AUTH) y de ellas se saca la media de las medidas de tiempo para cada una de las implementaciones. Como podremos apreciar en la Tabla 4.3, el número de negociaciones realizadas para obtener los resultados se considera suficiente para obtener unos intervalos de confianza realmente pequeños.

Para poder analizar por separado las diferentes operaciones que se realizan en una negociación inicial IKEv2, se ha dividido ésta en cinco pasos que enumeramos a continuación y mostramos también gráficamente en la Figura 4.1:

- **Paso 1**. Comienza cuando IPsec o cualquier aplicación con capacidades IKEv2 solicita una nueva IPsec SA, y concluye cuando la petición del intercambio IKE_SA_INIT sale del iniciador con destino al respondedor.
- **Paso 2**. Comienza cuando la petición del intercambio IKE_SA_INIT llega al respondedor, y termina cuando la respuesta del intercambio IKE_SA_INIT sale del respondedor con destino al iniciador.

- **Paso 3**. Comienza cuando la respuesta del intercambio IKE_SA_INIT llega al iniciador, y termina cuando la petición del intercambio IKE_AUTH sale del iniciador con destino al respondedor.
- **Paso 4**. Comienza cuando la petición del intercambio IKE_AUTH llega al respondedor, y termina cuando la respuesta del intercambio IKE_AUTH sale del respondedor con destino al iniciador.
- **Paso 5**. Comienza cuando la respuesta del intercambio IKE_AUTH llega al iniciador, y termina cuando la IPsec SA es creada en el iniciador.

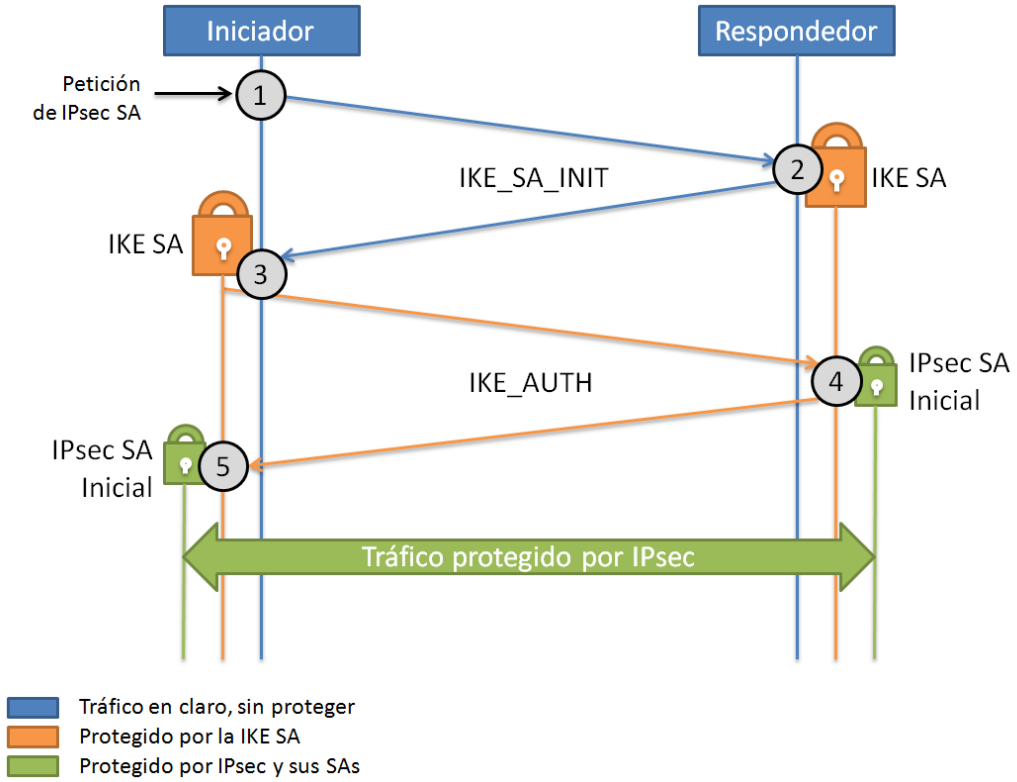


Figura 4.1: Diferentes pasos de la negociación IKEv2

Para cada paso, se ha tomado el tiempo transcurrido (en milisegundos). Esto es, el tiempo entre la recepción del mensaje, procesamiento del mismo y preparación del siguiente hasta que ha sido mandado. Este tiempo ha sido extraído gracias a la salida de depuración que ofrece cada implementación. La Tabla 4.3 muestra los diferentes valores medios de cada paso (filas) de cada implementación (columnas). Adicionalmente se han incluido los intervalos de confianza de los datos (con un nivel de confianza del 95 %). Como podemos apreciar, se ofrecen datos de OpenIKEv2 tanto cuando se usa XFRM como PFKEYv2, que son las dos interfaces disponibles para interactuar con

el núcleo del sistema. OpenIKEv2 permite usar ambas interfaces (ver Sección B.1.1.4 del Apéndice para más detalles sobre estos dos interfaces). Sin embargo, las otras implementaciones no disponen de esa posibilidad. La Figura 4.2 muestra un diagrama de barras donde se resumen los tiempos empleados en cada implementación.

Tabla 4.3: Comparativa de tiempos de negociación medios de diferentes implementaciones IKEv2 con intervalos de confianza al 95 %

	OpenIKEv2 XFRM	**StrongSwan XFRM**	**OpenIKEv2 PFKEYv2**	**IKEv2 PFKEYv2**	**Racoon2 PFKEYv2**
PASO 1	36.47 ±0.10	33.39 ±0.20	36.46 ±0.10	39.88 ±0.44	37.91 ±0.15
PASO 2	72.58 ±0.12	64.46 ±0.20	73.26 ±0.09	40.54 ±0.16	36.16 ±0.07
PASO 3	37.49 ±0.10	49.30 ±0.38	37.40 ±0.10	45.22 ±0.22	38.82 ±0.47
PASO 4	2.79 ±0.08	10.58 ±0.11	2.53 ±0.10	10.50 ±0.09	743.87 ±0.22
PASO 5	2.15 ±0.07	9.65 ±0.26	743.32 ±0.09	754.32 ±1.27	742.64 ±0.24
TOTAL	**151.48 ±0.12**	**167.38 ±0.48**	**892.97 ±0.13**	**887.07 ±1.62**	**1599.40 ±0.64**

A la luz de estos resultados, se aprecian grandes diferencias entre implementaciones en el tiempo empleado en algunos de los pasos, siendo similar en otros. Dichas diferencias han venido motivadas por tres razones:

- el uso de diferentes librerías criptográficas.
- el establecimiento de IPsec SAs a través de diferentes interfaces (PFKEYv2 o XFRM).
- diferentes momentos elegidos para procesar los mensajes.

Con respecto a la primera razón, la elección de librería criptográfica, *StrongSwan* usa su propia librería, mientras que *OpenIKEv2*, *IKEv2* y *Racoon* usan OpenSSL como tal. Esta elección ha supuesto que *StrongSwan* emplee menos tiempo en operaciones de criptografía asimétrica como es el intercambio *Diffie-Hellman*, mejorando en 3 segundos los tiempos del resto de implementaciones en los pasos 1 y 2 que hemos definido antes. Sin embargo, en los pasos 3, 4 y 5 donde se cifran y descifran los mensajes mediante criptografía simétrica, StrongSwan ha obtenido peores resultados empleando 8 milisegundos más que *OpenIKEv2* y *Racoon.*

En cuanto a la segunda razón, usando PFKEYv2 (que es el más estandar de los interfaces IPsec) para crear IPsec SAS tiene dramáticas consecuencias, especialmente con el tiempo consumido en los intercambios. La implementación PFKEYv2 implementada en el núcleo de Linux no parece estar muy bien optimizada, ya que le lleva sobre los ≈700 milisegundos crear una IPsec SA. En contraste con esto vemos que XFRM sólo emplea 2 milisegundos para crear IPsec SAs. Por tanto, a pesar de que PFKEYv2 tiene la ventaja de poderse usar en distintos sistemas operativos,

su rendimiento en términos de tiempo de procesamiento es bajo. Este hecho repercute muy negativamente en el tiempo total de la negociación IKEv2.

Finalmente, asociada a la tercera razón, se han destacado dos aspectos principales. En primer lugar, hemos observado que, con *IKEv2* y *Racoon2*, el respondedor envía la respuesta del intercambio IKE_SA_INIT (paso 2) antes de calcular el secreto compartido (usando Diffie-Hellman). De esta forma, mientras el iniciador recibe y procesa la respuesta, el respondedor puede generar mientras tanto el secreto compartido. Esta forma de proceder permite ahorrar tiempo en la negociación, ya que solapamos tiempo de cálculo de secreto compartido con el tiempo de transmisión de los mensajes. Adicionalmente, ya que el respondedor no usa el secreto compartido hasta recibir el siguiente mensaje del intercambio IKE_AUTH que el iniciador le tiene que enviar, esta decisión de diseño no afecta al protocolo IKEv2 en absoluto. Como hemos dicho, esta optimización ha sido incorporada sólo por las implementaciones de *racoon2* e *IKEv2*, ahorrando sobre unos ≈35 milisegundos en el paso 2, como puede verse en la Tabla 4.3.

El segundo aspecto es la creación de la IPsec SA en el paso 4. En general, las implementaciones pueden seleccionar dos maneras de llevar esto a cabo. Todo depende de si se crea la SA antes o después de que el respondedor envíe la respuesta del intercambio IKE_AUTH. Si se crea después, parte del tiempo empleado en la creación de las IPsec SAs se solapará con el envío de la respuesta al iniciador y la creación de las IPsec SAs correspondientes. Un problema potencial aparece aquí, cuando la creación de las IPsec SAs en el lado del respondedor falla, el iniciador será informado erróneamente como si todo hubiera ido bien. Sin embargo, este problema solo implicaría un intercambio adicional de tipo INFORMATIONAL iniciado por el respondedor para eliminar dicha IPsec SA en el iniciador. Como podemos observar en la Figura 4.2, todas las implementaciones excepto *Racoon2* usan esta mejora y por tanto el paso 4 es finalizado en un corto espacio de tiempo (sobre unos 10 milisegundos en el peor caso). Sin embargo, *Racoon2* espera hasta el establecimiento de la IPsec SA antes de mandar la respuesta del intercambio IKE_AUTH. Además, como esta implementación usa la interfaz PFKEYv2 para la creación de la IPsec SA, el paso 4 tarda en completarse del orden de 700 milisegundos, incrementando dramáticamente el tiempo total de la negociación IKEv2. Merece la pena destacar que la Figura 4.2 también revela que la duración del paso 5 es similar en todas las implementaciones que usan PFKEYv2, ya que estos pasos finalizan cuando la IPsec SA ha sido establecida.

Adicionalmente, estos resultados han permitido elaborar recomendaciones a la hora de implementar IKEv2 u otros protocolos de seguridad similares. En primer lugar, la elección de la librería criptográfica a usar influye en el tiempo total de la negociación. Sin embargo, considerando que no hay una gran diferencia entre usar una librería criptográfica particular o usar OpenSSL, recomendamos el uso de este último pues es una buena librería de código abierto, ampliamente conocida y revisada permanentemente por una gran comunidad de usuarios.

Por otro lado, ya que PFKEYv2 es un estándar, su portabilidad es mayor que XFRM. Sin embargo, XFRM ha mostrado un mejor rendimiento en sistemas Linux. Por lo tanto, la decisión que debe tomar el desarrollador la hará pensando si lo que quiere beneficiar es la portabilidad o el rendimiento. Por ejemplo, ya que nuestra

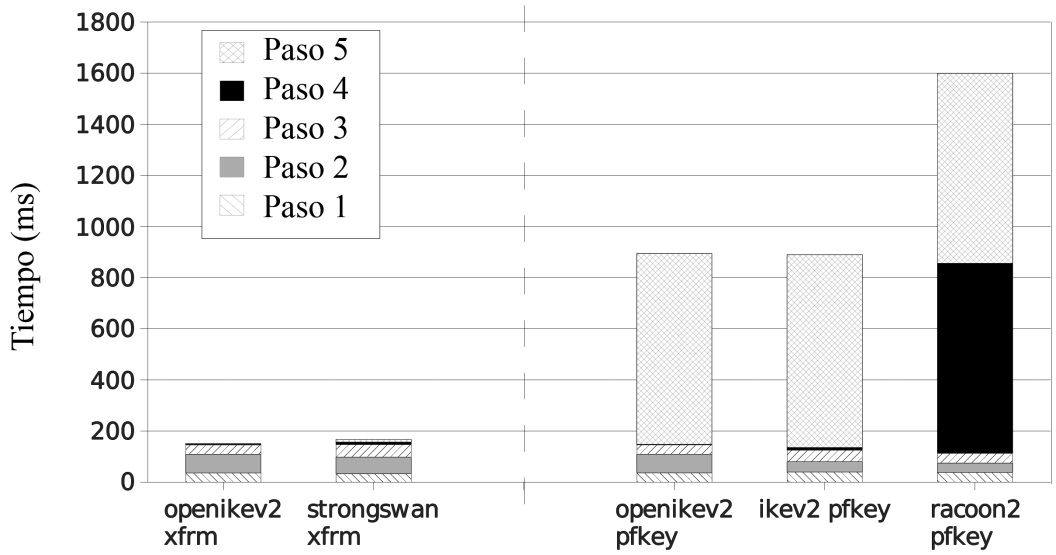


Figura 4.2: Tiempos de negociaión IKEv2 en milisegundos agrupados por tipo de interfaz IPsec.

implementación fue inicialmente concebida para ser usada en proyectos de investigación como ENABLE [28] o ITSsV6 [19] que demandan un tiempo de negociación IKEv2 lo más reducido posible en sistemas Linux, se tomó la decisión de usar XFRM (a pesar de que PFKEYv2 estaba también soportado). Dependiendo de los requisitos del escenario en cuestión, el desarrollador debe tomar diferentes decisiones en cada caso. De la experiencia también se ha aprendido que se debe simultanear ciertas operaciones de procesamiento intenso con los tiempos de envío para así reducir más el tiempo total de negociación. Finalmente, como hemos podido observar, el uso del lenguaje C o C++ no ha supuesto diferencias notables en los resultados. De hecho, OpenIKEv2 que está implementada en C++ ha mostrado ser una implementación competitiva en comparación con otras implementaciones hechas en C. Esto es comprensible pues donde más uso computacional se realiza es con las operaciones criptográficas, y estas están implementadas en su mayoría por librerías programadas en C. Ganamos todas las ventajas de la orientación a objetos sin apenas contrapartidas.

4.2. Incidencias Encontradas en la Implementación de OpenIKEv2

Durante la fase de implementación de OpenIKEv2 se han encontrado varias situaciones que no estaban bien definidas y resueltas en las especificaciones del protocolo IKEv2 (RFC 4306). Se han propuesto diversas soluciones frente a dichas situaciones, justificando en todo caso los motivos que nos han llevado a introducir nuestras propias mejoras que vienen a aclarar situaciones que no quedan claras en las

especificaciones. Como resultado de este trabajo, parte de nuestras mejoras han sido añadidas al documento de clarificaciones del protocolo IKEv2, el RFC 4718 [29], como hemos adelantado anteriormente. Pasamos a continuación a enumerar esos problemas de implementación que fueron subsanados y reportados:

4.2.1. Cookie inválida

Durante la implementación del mecanismo de cookies (ver Apéndice A.8), se encontró un problema del cual la especificación de IKEv2 no trataba: si el iniciador enviaba una cookie inválida en un intercambio IKE_SA_INIT, la cuestión que no queda clara es cómo debe actuar el respondedor en dicho caso. Se han propuesto tres soluciones:

1. **Descartar la petición**. Supongamos la situación en la que un atacante envía un número indiscriminado de peticiones con la cookie inválida. IKEv2 se puede defender descartando dicha petición para no hacer uso innecesario de procesamiento en el respondedor. Sin embargo, un iniciador lícito podría enviar una cookie inválida también cuando, antes de mandar una petición donde se incluye la cookie, el respondedor cambia la cookie correcta. En este caso, el iniciador que envía la petición no es un atacante, pero su petición será silenciosamente descartada y el iniciador pensará que el respondedor no está activo.

2. **Enviar el cookie correcto**. En este caso, el cookie correcto es enviado al iniciador en la respuesta para permitirle reenviarlo en una nueva petición. A pesar de que generar una nueva cookie consume recursos del servidor, esta sobrecarga no supone un riesgo frente a un ataque de denegación de servicio (DoS), cuando se reciben un número ingente de peticiones inválidas.

3. **Bloquear la IP del iniciador en el respondedor**. Aquí se opta por bloquear la IP de un iniciador si se detecta un número de peticiones anormal por parte de un iniciador. Este bloqueo sería por un corto espacio de tiempo. Este método no otorga mayor protección contra ataques de denegación de servicio. Peor aún, permite al atacante enviar peticiones con cookies inválidas para provocar que el respondedor bloquee la IP a un iniciador lícito, falsificando la IP de origen de la petición.

Se han valorado las tres opciones y hemos elegido la segunda opción ya que las otras dos opciones no ofrecen protección adicional.

4.2.2. Refresco de Material Criptográfico en una IPsec SA Doble

Recordamos que las asociaciones de seguridad se suelen presentar por pares, una para cada dirección ("inbound" para la del tráfico que llega, y "outbound" para

el trafico que sale), y cada una se identifica con su número de identificación SPI. Para hacer referencia a este par de asociaciones unidireccionales, normalmente se usa el SPI de la de tráfico entrante (inbound). En el documento RFC2401 [90] donde se especifica la versión anterior de IPsec, existía el concepto "SA bundle", una asociación IPsec donde se usa tanto la cabecera AH (integridad) como la cabecera ESP (integridad y confidencialidad), haciendo así una asociación doble. Por el contrario, en la nueva versión de IPsec definida en el documento RFC4301 [88], este concepto ha sido eliminado. Sin embargo, las versiones del núcleo del sistema operativo Linux usadas en el momento de llevar a cabo la implementación de OpenIKEv2 se utilizaba todavía la versión antigua, así que teníamos la posibilidad de dar soporte a dichas asociaciones con doble cabecera. Cuando implementamos dicho soporte, en concreto la parte de renovación de material criptográfico de una asociación doble, se vio que las especificaciones estaban incompletas. Esto es así porque cualquiera de las dos partes, ya sea iniciador o respondedor, que quiera renovar el material criptográfico de una asociación doble tiene que especificar el número (SPI) que identifica la asociación y enviarlo en un campo llamado REKEY_SA mediante un intercambio CREATE_CHILD_SA. El problema es que dicha asociación doble tiene dos SPIs de asociaciones "inbound", una por cada cabecera. Frente a este problema se han propuesto estas alternativas:

1. **Añadir un campo REKEY_SA adicional en el intercambio CREATE_CHILD_SA**. Sin embargo, esta solución entra en conflicto con las especificaciones de IKEv2, ya que el respondedor sólo espera la recepción de un único campo REKEY_SA.

2. **Realizar dos procesos de renovacion de material criptográfico**. En este caso, resultarían dos asociaciones de seguridad independientes, lo que rompería el concepto de "SA bundle".

3. **Indicar sólo uno de los dos posibles SPIs**. Ya que ambos participantes saben que se trata de una asociación doble, es posible indicar sólo una de las SPIs que la identifican, quedando así identificada la asociación doble completa.

Inicialmente se optó por la tercera y última opción ya que así no se contradecían las especificaciones y permitíamos identificar de forma sencilla la asociación doble. Sin embargo, después de discutir este problema en las listas de correo del grupo de trabajo sobre IPsec del IETF, se decidió eliminar finalmente este tipo de asociaciones dobles en la implementación de OpenIKEv2 ya que el grupo de trabajo la consideraba obsoleta. Como resultado de estas discusiones, se añadió una nueva sección (7.13) en el RFC 4718 para aclarar este punto para que futuros implementadores no se encontraran con los mismos problemas.

4.2.3. SPI Inválido en el Campo DELETE

Bajo determinadas circunstancias, uno de los participantes puede recibir una petición para eliminar una asociación de seguridad. El problema se dá cuando el SPI

indicado en el campo DELETE del intercambio INFORMATIONAL no pertenece a ninguna de las asociaciones existentes. Esta situación puede ocurrir o bien por un reinicio en uno de los participantes, fallo de sincronización o pérdida de algún paquete. Esta condición no está cubierta por las especificaciones del protocolo IKEv2. Se proponen cuatro posibilidades de actuación:

1. **Enviar en la respuesta una notificación de error**. Esta solución no parece ser apropiada debido a que no hay ningún código de error que encaje con la situación y podemos confundir al otro participante sobre las razones del error.

2. **Enviar en la respuesta un campo DELETE vacío**. El problema con esta solución es que puede ser también interpretado como que ha ocurrido un proceso de eliminación simultánea de una IPsec SA, hecho documentado en [94], dejando como resultado una IPsec SA en estado de eliminación incompleta.

3. **Enviar en la respuesta una notificación de error por SPI inválido**. Sin embargo esta solución fue diseñada para otra situación distinta y por tanto entra en conflicto con las especificaciones de IKEv2.

4. **Omitir el campo DELETE**. En este caso el participante se queda reenviando por un tiempo el paquete pensando que se ha perdido, hasta que desiste y cierra la IKE SA entera ya que supone que el otro extremo ha caído.

A pesar de que la segunda opción puede crear cierta falta de sincronización, se ha decidido tomar esta opción porque el protocolo IKEv2 ya define un mecanismo para resolver dicha falta de sincronización (cerrando la IKE_SA después de un cierto tiempo). Fruto de este planteamiento, se ha añadido una nueva sección (5.11.6) en el RFC 4718 de aclaraciones sobre las especificaciones de IKEv2.

4.2.4. Colisión de Intercambios

Ya que ambos extremos de la negociación pueden tomar indistintamente los roles de iniciador o respondedor, podría ocurrir el caso de que ambos decidieran comenzar un intercambio al mismo tiempo y jugar el rol de iniciadores. Si se da este caso, es posible que ambos intercambios se solapen. A esto se le denomina colisión de intercambios. Algunas de dichas posibles colisiones están cubiertas por las especificaciones de IKEv2, pero no todas. Cuando estresábamos la implementación OpenIKEv2, nos dimos cuenta de que cuando se solicitaba la renovación de la IKE SA por un lado, y una IPsec SA por otro lado, se podía desembocar en un estado de desincronización.

Por ejemplo, como se aprecia en la Figura 4.3, cuando los dos procesos de renovación finalizan, el participante que ha comenzado la renovación de la IKE SA tiene una nueva IKE SA con SPI=2 de las que dependen las IPsec SAs con SPI=A y SPI=B, la primera heredada de la IKE SA antigua, y la segunda recién creada solicitada por el otro participante. La IKE SA antigua con SPI=1 ya no tiene IPsec SAs a su cargo, así que puede ser eliminada sin problemas, que es además el comportamiento esperado.

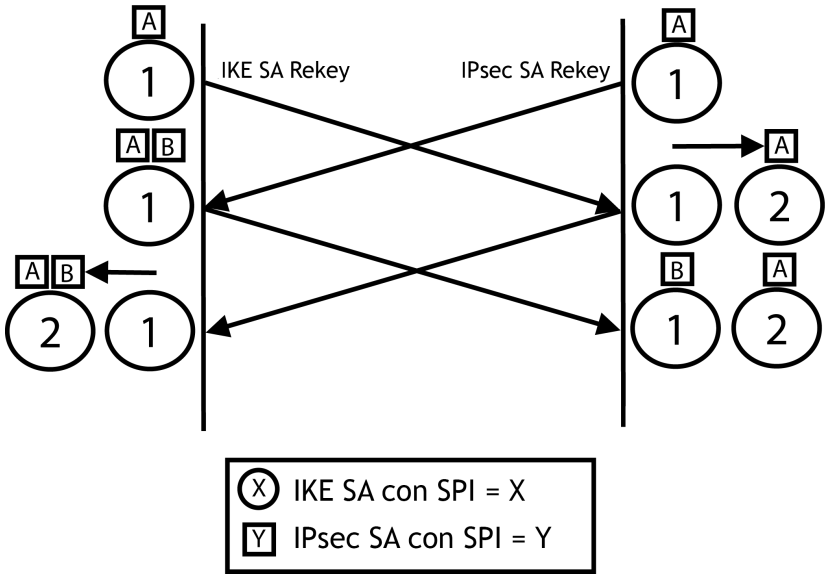


Figura 4.3: Ejemplo de una colisión de intercambios

Sin embargo, el participante que comenzó la renovación de la IPsec SA con SPI=A ahora tiene una IKE SA nueva con SPI=2 que ha heredado la IPsec SA con SPI=A. La nueva IPsec SA con SPI=B se queda en la IKE SA con SPI=1 que fue quien la solicitó, pero la herencia de IPsec SAs entre las IKE SAs con SPI=1 y SPI=2 ya ha ocurrido, con lo que la IKE SA con SPI=1 no queda vacía, sino asociada a la IPsec SA con SPI=B. Por tanto, dicha IKE SA no puede ser eliminada ya que eliminaríamos también la IPsec SA con SPI=2, que no es precisamente nuestra intención. Ha ocurrido un error de sincronización ya que en ambos extremos las asociaciones entre IKE SAs e IPsec SAs deben ser idénticas. Este tipo de situaciones deben ser evitados a toda costa.

Al igual que en los problemas anteriores, fueron discutidos ampliamente en los grupos de trabajo del IETF encargados de la definición de IPsec e IKEv2 y finalmente se determinó que se necesitaba de algún mecanismo que evitara creaciones o renovaciones de IPsec SAs mientras se esté dando una renovación de la IKE SA asociada. Para lograr este objetivo, cada petición del intercambio CREATE_CHILD_SA recibido durante la renovación de una IKE SA tiene que ser respondida con una notificación de error de tipo NO_ADDITIONAL_SAS. Se añade una nueva sección (5.11.8) en el documento RFC 4718 de clarificaciones sobre IPsec e IKEv2 para documentar este y otros tipos de posibles colisiones, con las indicaciones de cómo actuar en cada caso.

4.3. Conclusiones

Podemos considerar OpenIKEv2 como implementación válida, eficiente y de fácil manejo, como candidata perfecta para ser incorporada en nuestra propuesta de

arquitectura de red y satisfacer así la necesidad de ofrecer un tipo de protección distinta hasta la fecha, orientada hacia los flujos de datos y no hacia la protección de mensajes individuales. En la fase de diseño de un estándar como IKEv2 es muy difícil llegar a definirlo con todo detalle. Esta afimación se confirma en la fase de implementación, donde suelen aparecen detalles no contemplados hasta el momento, que implican un refinamiento del estándar. Este proceso es el que hemos vivido y constatado en la implementación de IKEv2, aportando una experiencia valiosa que ha ayudado a perfeccionar la definición de dicho estándar, hecho que hemos querido dejar constancia con este capítulo.

Capítulo 5

Integración de los servicios de Movilidad y Seguridad

En los capítulos anteriores hemos desarrollado nuestra propuesta de arquitectura que cumple con los estándares ISO/ETSI para entornos ITS, primero desde el punto de vista de las tecnologías de acceso que se iban a utilizar sobre dicha arquitectura y segundo incorporando tecnologías al nivel de red basadas en IPv6, como los servicios de movilidad y seguridad. Cabe mencionar aquí la aportación realizada en este sentido, incorporando el servicio de seguridad mediante nuestra implementación de IKEv2, llamada OpenIKEv2 [106] por su carácter de código abierto. Dicha implementación también nos ha proporcionado una experiencia útil que ha permitido complementar los estándares que lo definen. En este capítulo exponemos otro problema que surge de la propia experiencia de uso de los servicios de seguridad, pero también de movilidad, en este caso implementado por el proyecto UMIP [32]. Ambos servicios, y por tanto sus implementaciones, son inicialmente incompatibles entre sí, lo que impide usarlos simultáneamente. Esto provoca la aparición del documento RFC 4877 [96] donde se detalla cómo deben interoperar ambos servicios para funcionar correctamente. Sin embargo, a raiz de nuestras primeras etapas de implementación preliminar, pudimos comprobar deficiencias no tratadas a nivel de diseño, detalles que hemos analizado y resuelto en el proceso de integrar las implementaciones de dichos servicios.

5.1. Aproximaciones al Problema

Como se ha visto en los capítulos anteriores, los servicios de movilidad y seguridad pueden ser aplicados sin problemas por separado, es decir, usando uno u otro servicio. El problema llega cuando queremos usar ambos servicios a la vez. Queremos en definitiva un servicio de movilidad seguro, donde todo nuestro tráfico esté protegido. Hemos encontrado dos formas de integrar ambos servicios que analizaremos para ver cuál de ellas es la más conveniente.

La seguridad y la movilidad utilizan diferentes sistemas de encapsulamiento de tráfico para realizar su función, por lo que veremos que existen túneles de movilidad

y túneles de seguridad. Pensando en cómo resolver el problema, se nos plantean dos enfoques diferentes dependiendo de cómo anidemos dichos túneles, es decir, qué servicio encapsula al otro.

Supongamos un escenario ITS de red vehicular, donde hay una red central que une las diferentes entidades de la Estación Central ITS, como por ejemplo el HA; por otro lado tenemos otra red que interconecta a las Estaciones de Carretera ITS entre sí y con la Estación Central ITS, desplegadas éstas a lo largo de la carretera. Las Estaciones Vehiculares ITS y sus *Móbile Routers* (MR) usarán las Estaciones de Carretera ITS para llegar a los servicios ofrecidos por la Estación Central ITS, como son la movilidad y seguridad. Como queremos un servicio combinado que nos proporcione un servicio de movilidad seguro, ambos servicios deben interactuar para poderse aplicar de forma simultánea. Como hemos adelantado antes, hemos identificado dos acercamientos a la solución del problema:

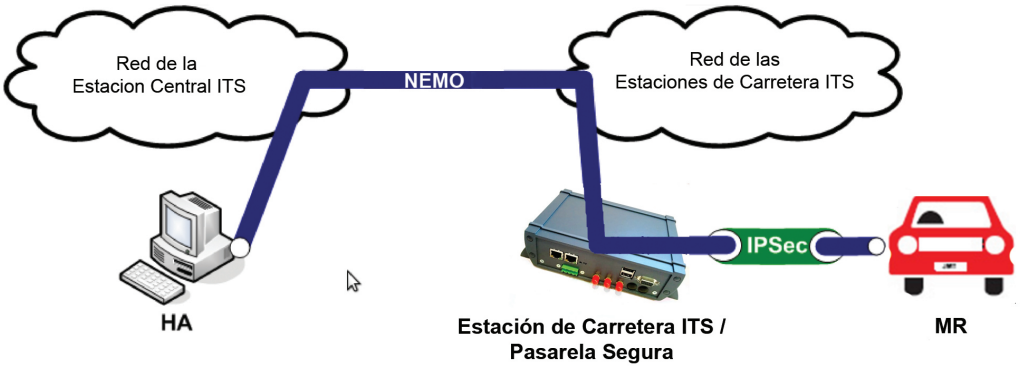
- Caso A: Se establece primero el túnel de seguridad entre el MR y la Estación de Carretera ITS, que en este caso hará las funciones de "Pasarela Segura" o también llamado *Security Gateway* (SG). La movilidad se establecerá después entre MR y HA sin ningún tipo de mecanismo de protección, ya que IPsec lo protege desde el MR hasta la Estación de Carretera ITS, que es el enlace más expuesto debido a su naturaleza inalámbrica (tecnología 802.11p). El resto de tramo entre la Estación de Carretera ITS y el HA, de naturaleza cableada, queda sin proteger. En este caso se puede ver que la movilidad NEMO va encapsulada dentro del túnel IPsec.

- Caso B: Se establece primero el túnel de seguridad entre el MR y el HA, de tal forma que las asociaciones de seguridad queden establecidas usando como extremo la HoA del MR. Esto permite que el tráfico IPsec generado por dichas asociaciones de seguridad pueda serle aplicado el servicio de movilidad y, de esta forma, dichas asociaciones sobrevivan a un posible cambio de CoA. La seguridad IPsec es aplicada en todo el recorrido desde el MR hasta el HA, independientemente de la naturaleza de los enlaces. En este caso, el tráfico IPsec va encapsulado dentro del túnel de movilidad NEMO.

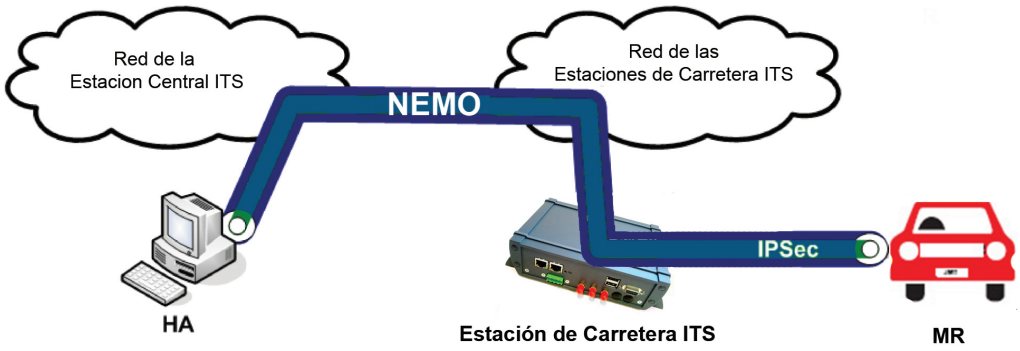
Ambos casos los podemos ver representados gráficamente en la Figura 5.1. En los siguientes apartados veremos en detalle ambos acercamientos y analizaremos las ventajas y desventajas de cada uno. Al final seleccionaremos uno de ellos para ser incorporado a nuestra solución final de arquitectura de comunicaciones orientada a entornos ITS.

5.1.1. Caso A: Movilidad Encapsulada en la Seguridad

En este primer enfoque, para que el MR pueda detectar la presencia de una Estación de Carretera ITS utilizamos los mensajes *Router Advertisements* (RAs) ofreciendo información del prefijo de red, como podemos apreciar en la Figura 5.2. Como queremos que todo el tráfico, incluido el del establecimiento de la movilidad, vaya protegido por



(a) Caso A: IPsec encapsula a NEMO



(b) Caso B: NEMO encapsula a IPsec

Figura 5.1: Diferentes enfoques en aplicación simultánea de servicios de movilidad y seguridad

IPsec, la seguridad deberá negociarse en primera instancia. Para ello, los demonios de los servicios IKEv2 y NEMO se comunican entre ellos para que NEMO pueda pedirle a IKEv2 el establecimiento de la seguridad (ya que el demonio de IKEv2 no es capaz de escuchar los mensajes RAs), y una vez establecida, IKEv2 pueda notificar a NEMO si la seguridad ha podido establecerse y qué CoA se ha asignado en el proceso de negociación de IKEv2. Esto es así ya que el servicio IKEv2 permite la asignación remota de direcciones en el establecimiento de un túnel IPsec, tal y como se hace en una VPN común.

Esta negociación IKEv2 se realiza utilizando las direcciones IPv6 *link-local* del MR y de la Estación de Carretera ITS, esta última extraída de la cabecera del mensaje RA. IKEv2 establece así un mecanismo de autenticación y autorización de acceso a la red, ya que la Estación de Carretera ITS hace las veces de Pasarela de Seguridad, que solamente reenviará el tráfico intercambiado con los MRs que tengan permitido establecer túneles IPsec para proteger sus datos. Cualquier otro tipo de tráfico de cualquier otro origen será descartado, haciendo imposible la comunicación a MRs no

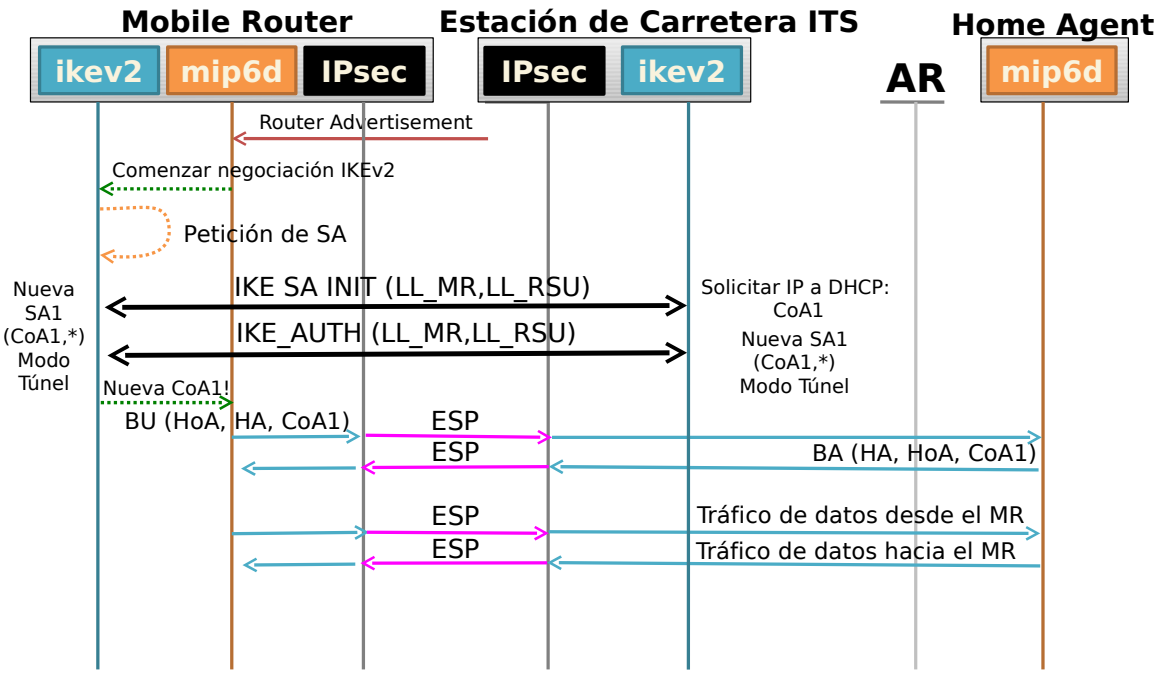


Figura 5.2: Diagrama de secuencia del establecimiento de la seguridad y movilidad en el caso A

autorizados. Se establece para ello una asociación de seguridad y política IPsec en modo túnel que proteja a todo el tráfico con origen o destino dicha dirección IP recién asignada, usando para ello la cabecera ESP de IPsec. El resto de tráfico no podrá fluir al no encontrar ninguna política de seguridad que se lo permita. Merece la pena destacar aquí que este mecanismo de autenticación/autorización es independiente de la tecnología del medio que se esté utilizando, ya que no se realiza a nivel de enlace, sino a nivel de red.

Una vez establecida la seguridad entre MR y la Estación de Carretera ITS, la movilidad puede establecerse notificando la dirección IP que le ha sido asignada (CoA) mediante los mensajes BU y BA. El servicio de movilidad no será consciente de que su tráfico está siendo protegido por IPsec entre el MR y la Estación de Carretera ITS.

En esta aproximación, el funcionamiento de IKEv2 se ha modificado ligeramente para poder provocar la creación de una asociación de seguridad en un momento dado, y no bajo demanda por la coincidencia de alguna política IPsec como se hace habitualmente. Esto es así porque el tráfico que provocaría la creación del túnel IPsec

no puede producirse hasta que la negociación IKEv2 haya terminado. El tráfico que provocaría la creación del túnel sería el mensaje BU con la CoA asignada. Sin embargo, esta CoA se asigna en el proceso de negociación IKEv2 y no a partir del mensaje *Router Advertisement* (RA) inicial con el que los enrutadores anuncian periódicamente su presencia, ya que sólo la CoA negociada con IKEv2 tendrá acceso a la red y no cualquier otra que pueda asignarse automáticamente el MR con mecanismos de auto-configuración. Por ello vemos en la Figura 5.2 que el demonio IKEv2 del MR genera su propia solicitud de creación de asociación de seguridad.

Este tipo de solución tiene que venir apoyada por las operadoras que ofrecen el servicio de conectividad. Si por ejemplo añadimos la tecnología 3G a nuestro ejemplo y las operadoras 3G no ofrecen una *Pasarela de Seguridad IPsec*, no se podría establecer estos mecanismos de seguridad. Esto es un inconveniente importante que se debe tener en cuenta.

El proceso de traspaso en este enfoque requiere de una renegociación completa de la seguridad debido al cambio de Estación de Carretera ITS. De este proceso de traspaso podría haber un cambio de IP también, lo que forzaría a la movilidad a actualizar sus asociaciones HoA-CoA en el Home Agent. Es decir, prácticamente es el mismo proceso que conectarse por primera vez. Ninguna asociación de seguridad sobrevive a un traspaso.

Una forma de permitir que las asociaciones de seguridad sobrevivan es utilizando soluciones como la "Trasferencia de Contexto IPsec" (*IPsec Context Transfer* [110]) que permita a través de la red que interconecta las Estaciones de Carretera ITS compartir el material criptográfico y algoritmos (contexto IPsec) entre todas las Estaciones de Carretera ITS. De esta forma, las renegociaciones serían innecesarias. La aplicación de esta solución queda fuera del ámbito de este trabajo, proponiéndolo para posibles trabajos futuros.

5.1.2. Caso B: Seguridad Encapsulada en la Movilidad

La idea general de este segundo acercamiento al problema es hacer que a la seguridad y todo su tráfico se le aplique la movilidad, es decir, vaya encapsulado por la movilidad. De esta forma las asociaciones de seguridad se establecerán en términos de HoA, que no cambia a lo largo del tiempo, generando políticas y asociaciones independientes de la CoA que se esté usando en cada momento. El gran beneficio que esto aporta es que dichas asociaciones sobrevivirán a los cambios de CoA en los traspasos y, por tanto, no es necesario renegociarlas.

Este acercamiento está descrito en el documento IETF RFC 4877 [96], que hemos seguido concienzudamente para su implementación. A pesar de ello, se han encontrado detalles importantes que no han sido tratados en él y que impiden llevar a cabo una implementación completamente funcional. Más adelante profundizaremos en estos detalles, pero antes vamos a describir como se combinan los servicios de movilidad y seguridad en este acercamiento.

Todo el tráfico generado por el servicio de movilidad es intercambiado entre el MR, situado dentro de la Estación Vehicular ITS, y el HA, situado este en la Estación

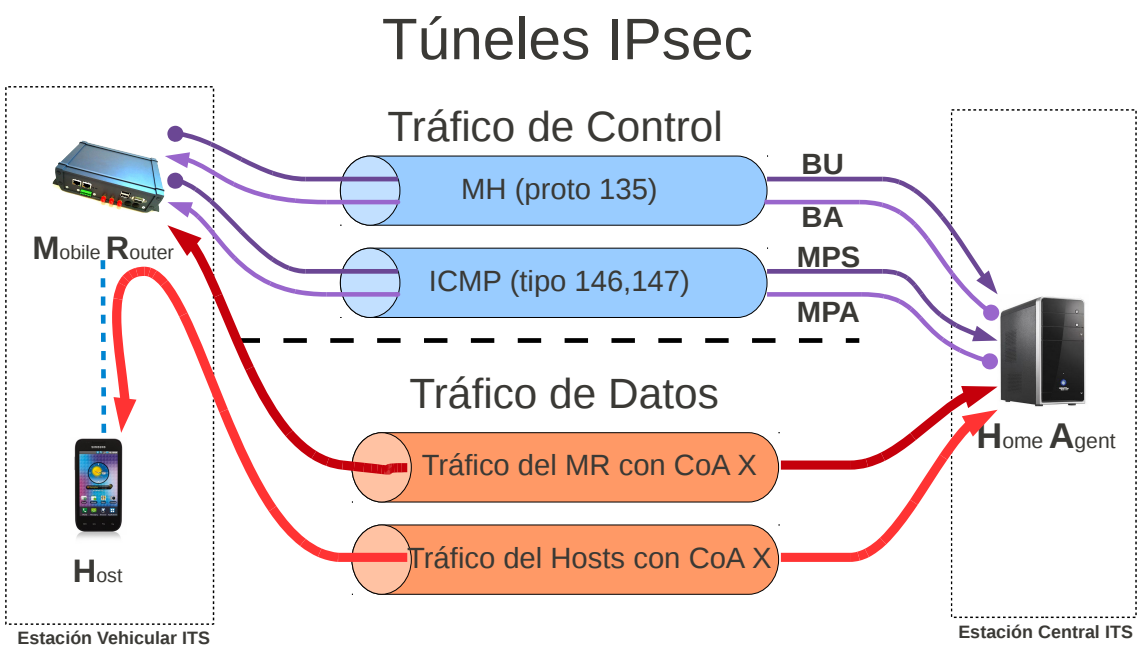


Figura 5.3: Túneles IPsec establecidos entre MR y HA para proporcionar seguridad al tráfico de movilidad

Central ITS. La idea básica es proteger este canal de extremo a extremo (entre el MR y el HA), tanto los mensajes de control, como pueden ser los mensajes BU y BA, como el tráfico de datos en sí. Es importante tener en cuenta aquí que antes de que IPsec pueda ser usado, deben acordarse los parámetros para las asociaciones de seguridad IPsec que necesitamos. Para ello, en ambas entidades MR y HA está presente el servicio IKEv2 que, junto con una serie de políticas recomendadas en el estándar [96], creará las IPsec SAs necesarias para mantener todo el tráfico de movilidad protegido. La Figura 5.3 resume las IPsec SAs resultantes. Para el tráfico de control se establecen IPsec SAs específicas, como son las necesarias para proteger el protocolo Mobility Header (MH) utilizado por NEMO para actualizar las IPs asociadas en el HA, con los mensajes BU y BA. También tienen sus IPsec SAs particulares dos subtipos de mensajes ICMPv6: Mobile Prefix Solicitations (MPS) y Mobile Prefix Advertisement (MPA), también necesarios para un correcto funcionamiento de NEMO.

Empezando desde cero en un escenario de movilidad con NEMO, como se puede ver en la Figura 5.4, el primer mensaje que aparece es el denominado *Router Advertisement*

(RA), que recordamos que son mensajes generados por los enrutadores para anunciar periódicamente su presencia, en el que se especifica el prefijo de la red. Supongamos que dicho enrutador es el que está presente en una Estación de Carretera ITS cercana, que nos sirve como punto de acceso a la red. El MR examina el prefijo de red anunciado en dicho mensaje y, a partir de él, crea una nueva IP que en términos de movilidad será una nueva *Care-of Address* (CoA) asignada a la interfaz de red por donde se ha recibido el RA. Este comportamiento es parte del funcionamiento normal de sistema de autoconfiguración sin estado disponible en IPv6, el cual queda fuera de este estudio.

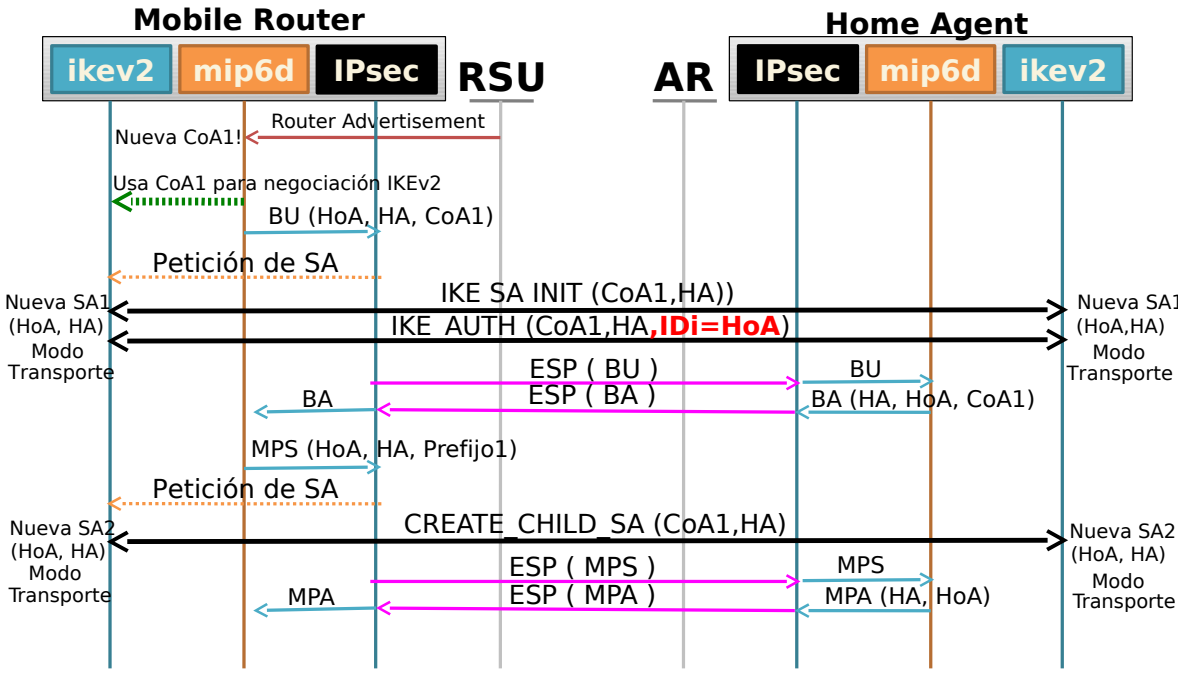


Figura 5.4: Diagrama de secuencia del establecimiento de la movilidad y seguridad en el caso B

Después de tener ya disponible nuestra nueva CoA, NEMO reacciona notificando dicha IP al HA usando el mensaje BU. Antes de poder ser enviado por la pila de comunicaciones, este mensaje concuerda con una de las políticas IPsec establecidas en el MR y, por tanto, es un tráfico que debe ser protegido. En acuerdo a lo establecido en [96], la política concordante establece que la IPsec SA necesaria debe estar establecida entre la dirección fija del MR, también llamada Home Address (HoA), y la dirección del HA (HAaddr) en modo transporte, usando para ello la cabecera de

extensión ESP. Sin embargo, se cambia aquí la forma normal de actuar de IKEv2 y, en vez de usar las IPs de los extremos, se sustituye la HoA por la CoA, ya que si no sería imposible establecer la comunicación con el HA. Hay que tener en cuenta aquí que la movilidad todavía no se ha establecido.

Cabe destacar que también se crea una IPsec SA similar en modo transporte y cabecera ESP para proteger los mensajes ICMPv6 MPS y MPA, que también forman parte del tráfico de control de la movilidad, como aparece en la Figura 5.4. El modo transporte usado en estos casos es debido a que el origen y el destino de los mensajes son los mismos extremos de la IPsec SA y, por lo tanto, establecer el modo túnel supondría tener dos cabeceras exactamente iguales. El modo transporte evita dicha redundancia.

Sin embargo, para el resto del tráfico, se utiliza un conjunto diferente de IPsec SAs, usando esta vez para todas ellas la cabecera ESP en modo túnel. Dichas IPsec SAs son creadas bajo demanda con la aparición del tráfico de datos, que dependerán del origen y destino de esos datos. Como puede verse en la Figura 5.3, la primera de las asociaciones IPsec se usa para proteger el tráfico intercambiado entre el propio MR y cualquier destino. Esta primera asociación no incluye el tráfico procedente de los diferentes dispositivos conectados a la red móvil desplegada en el vehículo, ya que dicho tráfico, a pesar de pasar a través del MR, irá protegido por una segunda IPsec SA. Por lo tanto, todos esos dispositivos comparten esta misma IPsec SA. El proceso de creación de estas dos asociaciones de seguridad lo podemos contemplar en la Figura 5.6, que muestra un diagrama de secuencia donde vemos cómo interactúan ambos servicios. El modo túnel sí es necesario aquí ya que los selectores de las IPsec SAs están establecidos en términos de HoA, pero para el túnel asociado se establece la CoA como extremo en vez de la HoA. Como resultado, dichas IPsec SAs que protegen el tráfico de datos son claramente dependientes de esta CoA. Esto origina que dichas asociaciones de seguridad no sobrevivan a un cambio de CoA. A pesar de ello, se pueden emplear algunas técnicas que permitan conservarlas cambiando la CoA de la asociación sin necesidad de renegociar. Las enumeramos a continuación:

- Una de estas técnicas es ***Migrate*** [111] propuesta por Sugimoto et al. [57], que implementa un nuevo tipo de mensaje para la interfaz PF-KEY que nos permite cambiar la dirección IP de una asociación de seguridad y sus políticas asociadas. También avisará a toda implementación IKE que esté escuchando por dicha interfaz a la espera de ser notificado por algún cambio de dirección IP. Este mensaje lo crea la implementación de la movilidad y lo envía al núcleo del sistema (donde reside IPsec), en el momento que envía o recibe un BU con el bit K activado. Este bit denota la habilidad de MR de cambiar las direcciones IP de los extremos de las asociaciones establecidas y, por tanto, hacer sobrevivir sus asociaciones después de un cambio de CoA. En este método, es la implementación de la movilidad quien trata directamente con IPsec, dejando a la implementación de IKEv2 como mera espectadora de los cambios que se realizan, obligando a IKEv2 a estar pendiente de los cambios para así actualizar las réplicas de las bases de datos de políticas y asociaciones que pueda tener la propia implementación de IKEv2. Desde nuestro punto de vista, debería ser IKEv2 el encargado de

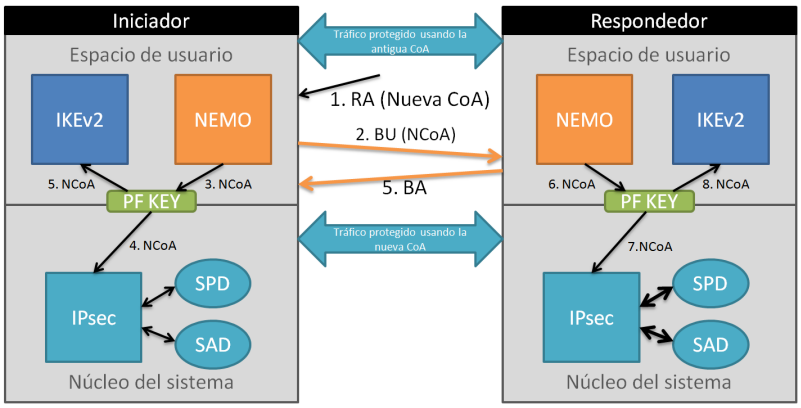
realizar cualquier modificación que tenga que ver con asociaciones o políticas de seguridad, en detrimento del servicio de movilidad. En la Figura 5.5a podemos ver un diagráma de secuencia resumiendo cómo funciona este mecanismo de forma esquemática.

- Otra técnica es mediante el uso de ***Mobike*** [112, 113], que permite igualmente cambiar las direcciones IP de los extremos de las asociaciones de seguridad y políticas sin necesidad de reconstruirlas. En esta ocasión, la implementación de movilidad debe solicitar este cambio de IP a la implementación de IKEv2 (directamente o mediante PF_KEY) en el MR cuando envía el mensaje BU con el bit K activado. Mediante un nuevo tipo de intercambio de mensajes llamado *MOBIKE Address Update*, la implementación de IKEv2 realiza el cambio de dirección IP tanto en las políticas como en las asociaciones de seguridad y actualiza sus bases de datos internas. Podría observarse aquí que el intercambio de Mobike es redundante, pues en el mensaje BU ya lleva la nueva CoA que hay que cambiar. Sin embargo no es así, ya que en el HA la movilidad no informará a IKEv2 del cambio de IP cuando recibe el BU, ya que lo habrá recibido antes mediante su intercambio de mensajes específico de Mobike. En el caso anterior y en este, es la movilidad la encargada de estar pendiente del cambio de CoA, pero en este segundo caso es la implementación de IKEv2 la que modifica las asociaciones y políticas, cosa que nos parece más razonable, pues esa es la misión de IKEv2. En la Figura 5.5b podemos también ver de forma esquemática esta aproximación.

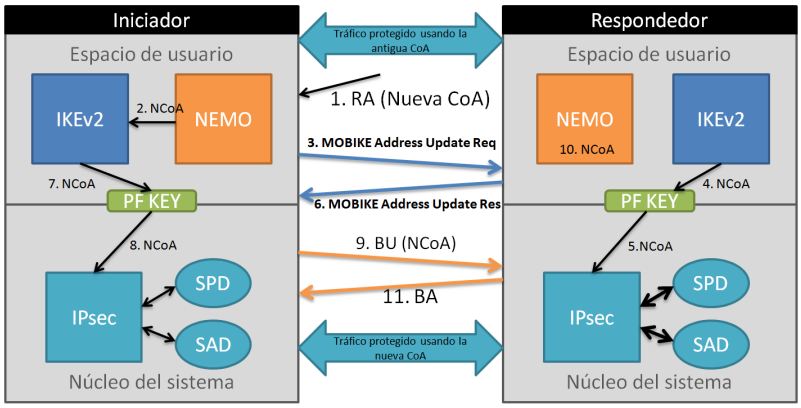
En nuestra propuesta hemos optado por una solución intermedia, condicionada por el uso de *Migrate* que ya hace la implementación de movilidad del proyecto UMIP [32]. Para evitar tener que implementar *Migrate* en la implementación de IKEv2, se ha establecido un canal directo inter-proceso entre ambas implementaciones y definido un pequeño protocolo para intercambiar información que permita mantener sincronizado el estado de las mismas. Esta estrategia es debida a que el uso de *Migrate* supone usar la interfaz PF_KEY, que como vimos en el Capítulo 4 donde hablamos de OpenIKEv2 [106], se decidió descartar su uso para comunicarnos con el sistema, en favor de la interfaz nativa del sistema operativo Linux (XFRM), por su mayor eficiencia y flexibilidad.

A pesar de que no todas las políticas y asociaciones de seguridad son independientes de la CoA utilizada, al menos aquellas que protegen el tráfico de control de la movilidad como son los mensajes BU, BA, MPS y MPA, sobreviven a los cambios de direccionamiento IP. En un supuesto caso de que sólo se quisiera proteger dicho tráfico de control y no el de datos, no habría que utilizar ninguno de los mecanismos vistos anteriormente para posibilitar cambios de direccionamiento IP de los extremos de las asociaciones y sus políticas.

Como podemos constatar, este "Caso B" supone un nivel de integración mayor entre los servicios de movilidad y seguridad que en la alternativa presentada en el apartado anterior ("Caso A"). La implementación de IKEv2 tiene que ser consciente de que va a



(a) Aproximación usando MIGRATE



(b) Aproximación usando MOBIKE

Figura 5.5: Distintas estrategias de cooperación entre los servicios de movilidad y seguridad ante el cambio de direccionamiento IP

proteger un servicio de movilidad y tratar a sus mensajes de control BU y BA de forma diferente a como lo haría con cualquier tráfico convencional. También encontramos que en esta ocasión el control de acceso se hace en el *Home Agent* (HA), en la red central de la operadora de carretera, debiendo proporcionar otro mecanismo de autenticación y autorización para limitar el acceso a la red que interconecta las Estaciones de Carretera ITS. En el caso de usar tecnología 802.11p, no es posible añadir ningún mecanismo de nivel de enlace, permitiendo el acceso a dicha red a todos los MRs. Sin embargo se pueden añadir reglas de filtrado de tráfico para que sólo circule el dirigido o procedente del HA, y así evitar tráfico indeseado.

Incorporar el control de acceso en el HA es una ventaja, ya que no requiere de las operadoras para que realicen esta tarea. Otra gran ventaja de este acercamiento es la protección extremo a extremo que se ofrece, protegiendo el tráfico en todo su recorrido desde el MR hasta el HA, pasando por enlaces tanto cableados como inalámbricos. Se

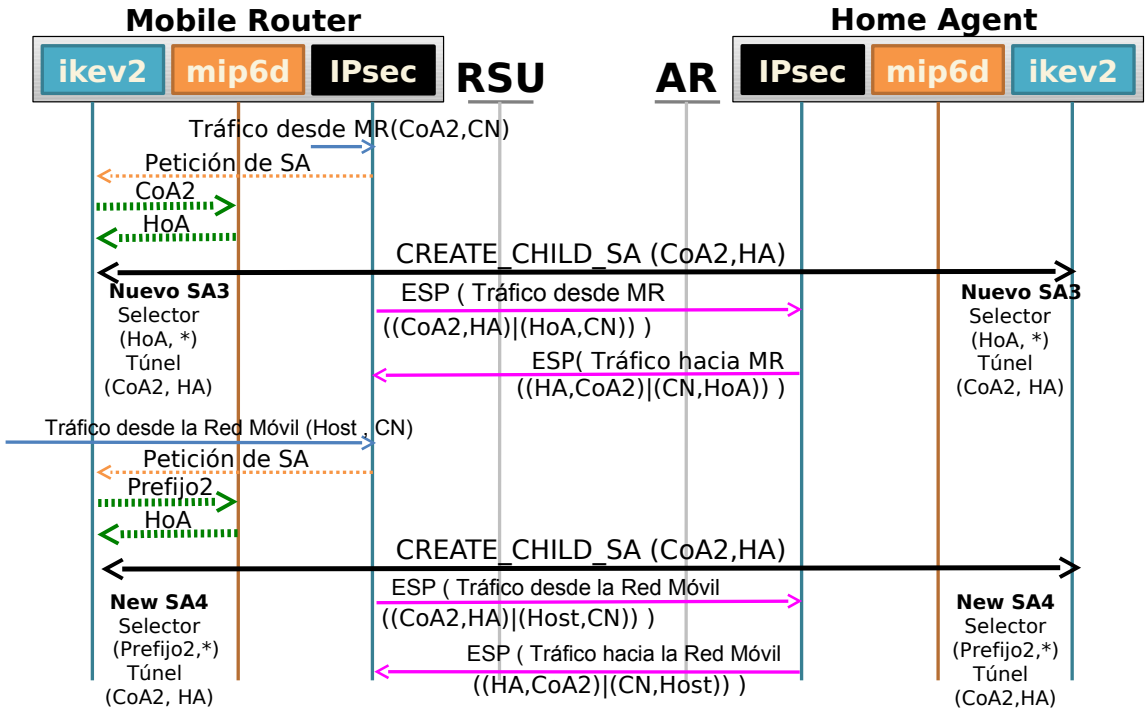


Figura 5.6: Diagrama de secuencia de las negociaciones necesarias para crear las IPsec SAs que protegen el tráfico de datos

hace recaer la labor de protección de los datos en los extremos, descargando de esa responsabilidad a las posibles operadoras de red que transporten el tráfico, que además no tendrán acceso a los datos transmitidos.

5.1.3. Estudio y Comparativa de los Acercamientos Planteados

Ambos acercamientos planteados en los apartados anteriores solucionan el problema de integración de los servicios de movilidad y seguridad, cada uno con sus pros y sus contras. Se trata aquí de valorar cuál de ellos es el más adecuado para un entorno de comunicaciones orientado a ITS. Para ello nos centraremos en varios puntos importantes:

- Grado de integración de las implementaciones de los servicios de movilidad y seguridad.
- Grado de protección del tráfico.

- Comportamiento en los traspasos
- Mecanismos de control de acceso: autenticación y autorización
- Grado de estandarización.

En los siguientes sub-apartados desarrollaremos estos puntos para determinar cuál de las dos soluciones es la más idónea.

5.1.3.1. Grado de Integración de las Implementaciones

En esta cuestión el caso A destaca claramente, pues no hay que aplicar a penas cambios en las implementaciones de la movilidad y la seguridad actuales. Simplemente requiere de una ligera interconexión entre los demonios para transmitir el evento de creación de la IPsec SA cuando se recibe un RA, y otro para la recepción de la confirmación de la creación de la IPsec SA y la CoA asociada. Puede decirse que el grado de dependencia entre las implementaciones de ambos servicios es pequeña.

Sin embargo, en el caso B el número de adaptaciones que hay que realizar para integrar ambos servicios es mucho mayor. La implementación de la seguridad tiene que hacer distinción entre tráfico de control de la movilidad y tráfico de datos, y tratar cada tipo de tráfico de una forma distinta. Además, es necesario forzar el uso de la HoA en las políticas y asociaciones de seguridad que se establecen, cuando ese no sería el comportamiento habitual. También usar la CoA en vez de la HoA para comunicarse en los momentos iniciales cuando la movilidad todavía no se ha establecido.

En este punto está claro que el nivel de dependencia entre las implementaciones es mayor en el caso B que en el A, y por tanto lleva a mayores esfuerzos para adaptar ambas implementaciones.

5.1.3.2. Grado de Protección del Tráfico

En este punto, el caso B es claramente el destacado, pues la protección se aplica extremo a extremo, protegiendo el tráfico de movilidad en todo su recorrido, desde el MR hasta el HA. Esto tiene la gran ventaja de que las operadoras que nos proveen de conectividad simplemente se limiten a la transmisión de datos cifrados, no teniendo acceso a los mismos y tampoco la obligación de desempeñar ningún proceso de autenticación/autorización para acceder al servicio de movilidad.

Por contra, en el caso A, la protección se establece a nivel de red de acceso y, por tanto, el grado de implicación de la operadora en las tareas de protección de los datos es mucho mayor. Además, sólo se protege el tramo inalámbrico existente entre el MR y la Estación de Carretera ITS, dejando al descubierto el tráfico en los tramos cableados dentro de la operadora de acceso a la red.

5.1.3.3. Comportamiento en los Traspasos

Como hemos visto, en el caso A se requiere de un proceso completo de negociación de las asociaciones de seguridad que pudiera haber establecidas, ya que uno de los

extremos pasa a ser una máquina distinta (otra Estación de Carretera ITS) y ésta no dispone de las asociaciones necesarias para proteger el tráfico.

Por otro lado, en el caso B, las asociaciones de seguridad están establecidas entre el MR y el HA, lo que hace que solo sea necesaria la renegociación de las mismas cuando se cambia de CoA. Además, no todas las asociaciones son dependientes de la CoA usada. Las hay que sólo están basadas en la HoA y no requieren ninguna renegociación.

En este punto vemos que el caso B tiene más bondades que el caso A y, por tanto, creemos que destaca más en este apartado.

5.1.3.4. Control de Acceso: Autenticación y Autorización

En el caso B se puede realizar un control de acceso a dos niveles. El primero a nivel de enlace para acceder al servicio de conectividad a la red, y el segundo a nivel de red que se realizará frente al Home Agent (HA) para acceder al servicio de movilidad y seguridad combinado.

En el caso A, estos dos controles de acceso se realizan en un sólo punto, frente a las Estaciones de Carretera ITS. Estas decidirán si nos dejan acceder tanto a la conectividad a nivel de enlace como a los servicios de movilidad y seguridad. La parte negativa de este enfoque es que se requiere que las operadoras dispongan de estos mecanismos, lo que normalmente no es posible, aparte de que estemos implicando a la operadora en tareas de autorización que no le competen.

Una ventaja que podemos destacar del caso A es que la autenticación a la red de acceso puede realizarse a nivel de red y, por tanto, de forma independiente de los mecanismos que dispongan las tecnologías de acceso utilizadas. Además, la inclusión de la tecnología 802.11p, la cual no dispone de ningún mecanismo de autenticación a nivel de enlace, hace de esta ventaja una necesidad.

Sin embargo, el caso B tiene una ventaja de mayor peso, donde las tareas de autorización y autenticación recaen en el HA, una entidad dentro de nuestro control y dominio, dejando fuera a la operadora de esta responsabilidad.

5.1.3.5. Grado de Estandarización

En el caso A, al existir una gran independencia entre las implementaciones, no se requieren de apenas cambios y, por tanto, sería la aplicación de dos estándares que funcionan correctamente cada una por su lado. Sin embargo, las interacciones a realizar entre las implementaciones de seguridad y movilidad no están reflejadas en ningún documento de estandarización.

El caso B está apoyado por el documento IETF RFC 4877 [96], que a pesar de tener algunas lagunas que discutiremos más adelante, sienta unas bases fuertes para que las implementaciones de movilidad y seguridad puedan interoperar incluso cuando los extremos de ambos servicios coinciden, hecho que el caso A no soporta.

5.1.4. Conclusiones y Selección de una de las Soluciones

No es fácil decidirse por una de las dos alternativas que hemos desarrollado en los apartados anteriores. Sin embargo, algunas de las características analizadas son claves en un entorno ITS. Primero se trata de un entorno móvil y con numerosos traspasos y, por tanto, el caso B es el más adecuado pues los traspasos son menos traumáticos. Además, el caso B ofrece una protección extremo a extremo, descargando la responsabilidad de la red de acceso a meramente un transmisor de tráfico cifrado, con las ventajas en seguridad que eso supone. A pesar del alto grado de dependencia entre las implementaciones de seguridad y movilidad de este caso B, todos los cambios y adaptaciones a realizar están apoyados por el documento estándar IETF RFC 4877 [96], que da mayores garantías a la hora de integrarlo en una arquitectura que sigue un marco de referencia también estándar, esta vez por el ISO/ETSI. Por estas razones decidimos optar por el caso B para incorporarlo a nuestra propuesta de arquitectura. De aquí en adelante dicho caso B pasaremos a considerarlo como nuestra propuesta de integración, que analizaremos a fondo en las siguientes secciones.

5.2. Incorporación de la Solución Seleccionada a Nuestra Arquitectura

En nuestra propuesta de arquitectura de comunicaciones ya teníamos definidos los roles de cada entidad, la pila estándar a construir según ISO/ETSI y las tecnologías IPv6 a utilizar. Precisamente el decidir cómo integrar dos de estas tecnologías ha sido el fruto del trabajo expuesto hasta aquí en este capítulo. Seleccionado el **Caso B**, retomamos el desarrollo de este acercamiento. Vimos con anterioridad en la Figura 5.3 que esta forma de integrar los servicios de movilidad y seguridad exigía la creación de cuatro asociaciones de seguridad, dos específicas para el tráfico de control de la movilidad (modo transporte, independientes de la CoA) y otras dos para el tráfico de datos (modo túnel, dependientes de la CoA).

Como en nuestra propuesta queremos incorporar la extensión MCoA de NEMO, tenemos que tener en cuenta que se puede usar más de una CoA simultáneamente. Por esta razón, las dos asociaciones en modo túnel que protegen el tráfico de datos deberán también crearse para cada una de las CoAs de las que dispongamos, ya que, como hemos dicho, dependen de ellas. Un ejemplo de esto lo podemos ver reflejado en la Figura 5.7, donde aparece una Estación Vehicular ITS con un MR con tres interfaces, cada una con una tecnología diferente (estas tecnologías son ejemplos hipotéticos y sólo tienen importancia para la comprensión del ejemplo). Cada interfaz tendrá asignada una CoA diferente.

Esta forma de funcionar sienta las bases para el control del tráfico basado en flujos, dando la posibilidad de decidir por qué interfaz se desea transmitir cada flujo en cada momento. Para poder entender con más claridad esto, necesitamos desarrollar el concepto de "flujo" y tener una idea clara de lo que es un "selector" o un "túnel", es decir, conocer las diferentes partes de una asociación de seguridad IPsec.

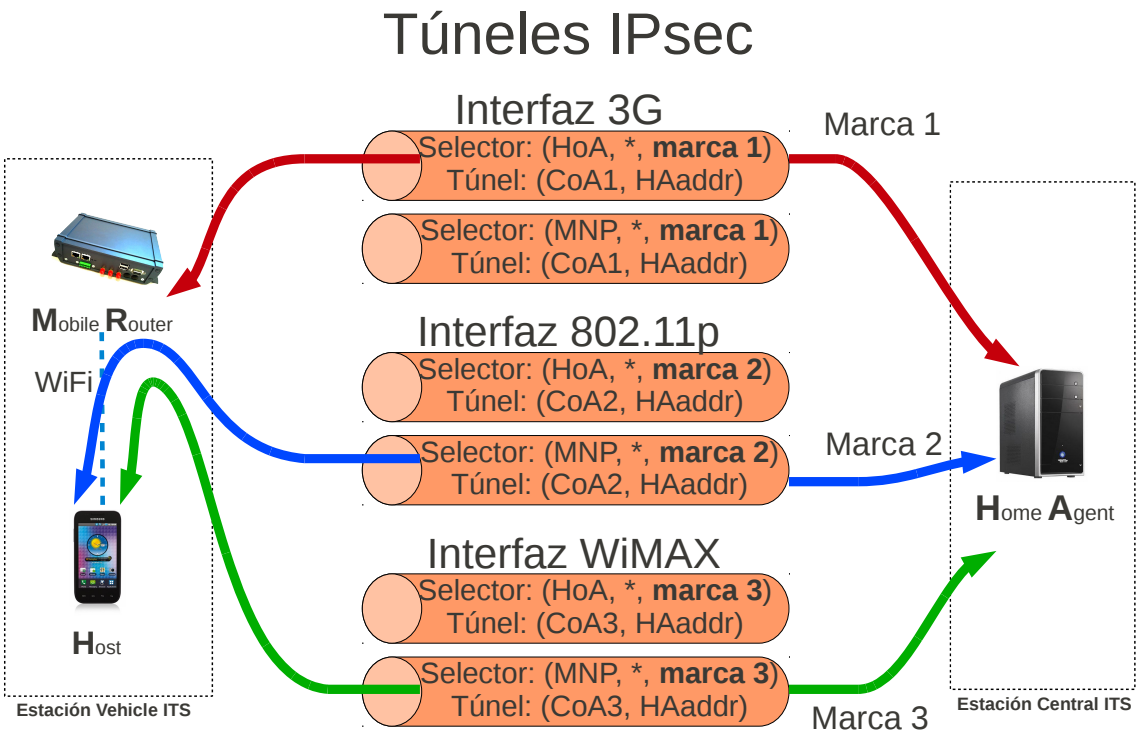


Figura 5.7: Selección de interfaz basado en flujos de datos

5.2.1. Selección de Flujos de Tráfico de Datos

Antes de nada se debe dejar definido el concepto de flujo, fundamental para comprender este apartado. Como flujo de datos denominamos a aquel subconjunto del total del tráfico que va a ser tratado de una misma forma para su envío. Por tanto, en un mismo enlace puede haber un número indeterminado de flujos, tantos como tratamientos diferenciados se hagan.

Para comprender con detalle cómo el tráfico de datos es reenviado por el MR y el HA a través de diferentes flujos, primero se van a exponer las partes de las que se compone una asociación de seguridad IPsec (IPsec SA). A pesar que en la Sección 2.3.3.4 del Capítulo 2 ya expusimos dichas partes, merece la pena recordarlas a continuación:

- El tipo de cabecera IPsec que se usará (ESP o AH).
- El tipo de tráfico (flujo) definido por un selector de tráfico, compuesto de una serie de reglas que definen a qué tráfico afectara esta IPsec SA (rangos de IPs, protocolos, puertos, marcas,...).

- Una selección de algoritmos de autenticación y de encriptado deseados para proteger los datos.
- El material criptográfico, es decir, las contraseñas para el funcionamiento de los algoritmos.
- Tipo de asociación, es decir, modo túnel o transporte. Además habrá que incluir las direcciones IP de los extremos del túnel en el caso que se use el modo túnel. En modo transporte se obtienen de los extremos de la negociación.

Por tanto, si nos damos cuenta, cada IPsec SA está definiendo un flujo de datos si nos atenemos a la definición anterior. Sin embargo, volviendo al ejemplo de la Figura 5.7, a pesar de que las IPsec SAs resultantes para proteger el tráfico de datos tienen diferentes CoAs en los extremos de sus túneles, todas ellas tienen los mismos selectores de tráfico. Es decir, definen un único flujo. Esto hace que el tráfico por sí mismo no sea suficiente para poder elegir cual de las IPsec SAs utilizar. Necesitamos añadir un elemento más al selector para poder diferenciar flujos distintos. Para ello es necesario acudir a la marcación de paquetes. Así pues, si añadimos en cada uno de estos selectores que, además el tráfico debe de estar marcado con un identificador diferente para cada IPsec SA, entonces sólo habrá que marcar el tráfico con el identificador deseado para considerar que un determinado paquete pertenezca a un flujo o a otro. De nuevo volvemos a la Figura 5.7, donde puede apreciarse un ejemplo de esto precisamente. Vemos un MR con tres interfaces disponibles, cada una de ellas con una CoA y su correspondiente par de IPsec SAs. Cada par usa en sus selectores un determinado identificador. Por tanto realmente hay seis flujos, dos flujos por interfaz. Uno llega hasta el MR y el otro hasta la red móvil desplegada en el vehículo. La elección de uno u otro viene dada por las direcciones IP origen y destino del tráfico. Sin embargo, para poder elegir la interfaz por donde enviarlo, es necesario marcar cada paquete para que concuerde con uno de los tres selectores disponibles. El responsable de realizar dicho marcado será el servicio de movilidad, que decidirá por qué interfaz enviar el tráfico en cada momento. La elección de la interfaz suele realizarse estableciendo una relación de prioridad entre ellas. En el ejemplo de la Figura 5.7 aparecen tres flujos de tráfico, cada uno marcado con el identificador de cada interfaz, y así poder elegir la interfaz por donde mandar los datos, siempre de forma segura gracias a IPsec.

5.2.2. Ajustes de Diseño en nuestra Propuesta de Solución

En nuestra propuesta de arquitectura de red hemos visto que son fundamentales los servicios de movilidad y seguridad. Sin embargo vimos que cuando ambos servicios son usados a la vez para disponer de un servicio conjunto de seguridad y movilidad, había que realizar algunos ajustes para que ambos servicios fueran compatibles. Dichos ajustes, como ya hemos visto en el Caso B, vienen especificados en el documento estándar IETF RFC 4877 [96], donde se explica cómo proteger con IPsec e IKEv2 el tráfico de movilidad entre el HA y el MR. A la hora de llevar a cabo esta solución se han encontrado problemas importantes en las especificaciones redactadas en este documento

que son mucho más fáciles de identificar una vez empezada la fase de implementación de la solución propuesta. Las desarrollamos a continuación:

5.2.2.1. CoA vs HoA

Problema: *IKEv2 usa la HoA en vez de una CoA del MR para la primera negociación, pero ésta no es alcanzable por el HA cuando se está en una red distinta a la red "home" donde el propio HA reside.*

Causa: Cuando el MR consigue conectividad por una de sus interfaces y el tráfico de control de la movilidad debe ser protegido con una nueva IPsec SA. Lo que suele hacer IKEv2 en un caso normal es usar la HoA del MR para realizar la negociación, pero la HoA no está accesible directamente cuando el MR está fuera de su *red home*. El servicio de movilidad es el encargado de permitir esta comunicación usando la CoA recién asignada para ello, pero no está establecido este servicio de movilidad todavía.

Solución: Usar la CoA en vez de la HoA para hacer la negociación IKEv2. El servicio NEMO en el MR informará de la CoA que ha lanzado el proceso de negociación, ya que puede haber más de una CoA presente debido a las capacidades de MCoA. En la Figura 5.4 se muestra esta interacción entre los servicios NEMO e IKEv2 en el MR. Todas las negociaciones IKEv2 posteriores seguirán utilizando dicha CoA elegida. Si el servicio NEMO tuviera la necesidad de cambiar a otra CoA diferente como podemos ver en la Figura 5.8 (con la aparición de una nueva interfaz), se activa el bit *K* [114] en los mensajes de movilidad BU y BA para notificar que se desea que la CoA que se está notificando, se use a partir de ese momento para futuras negociaciones IKEv2. Es decir, actualizar los extremos de la IKE SA. Para ello, NEMO debe informar de esta circunstancia a IKEv2 mediante una comunicación inter-proceso, como se puede apreciar en la Figura 5.8.

5.2.2.2. Imposible Identificar al MR a través de la CoA

Problema: *En la primera negociación IKEv2, el HA no puede identificar qué MR se está comunicando con él, ya que usa una CoA para ello que todavía no ha sido anunciada.*

Causa: En la primera negociación IKEv2 con la finalidad de proteger los mensajes de movilidad BU/BA, la IPsec SA resultante tiene que crearse en términos de HoA entre el MR y el HA. Sin embargo, el servicio de IKEv2 presente en el HA, una vez que recibe el primer mensaje de dicha negociación, no tiene la posibilidad de saber qué MR se lo está mandando, ya que la dirección IP de origen utilizada ha sido una CoA recien generada y que el HA todavía no sabe a qué HoA va a estar asociada. El mensaje BU que notifica esa asociación CoA-HoA al HA es precisamente el mensaje que se quiere proteger.

Solución: El MR envía su HoA a través del campo *Initiator Identification payload* (IDi) que viaja en el intercambio de mensajes IKE_AUTH, concretamente en el mensaje de petición. Este campo no viaja en claro, ya que está protegido por la IKE SA establecida previamente por el intercambio de mensajes IKE_SA_INIT, por lo tanto no

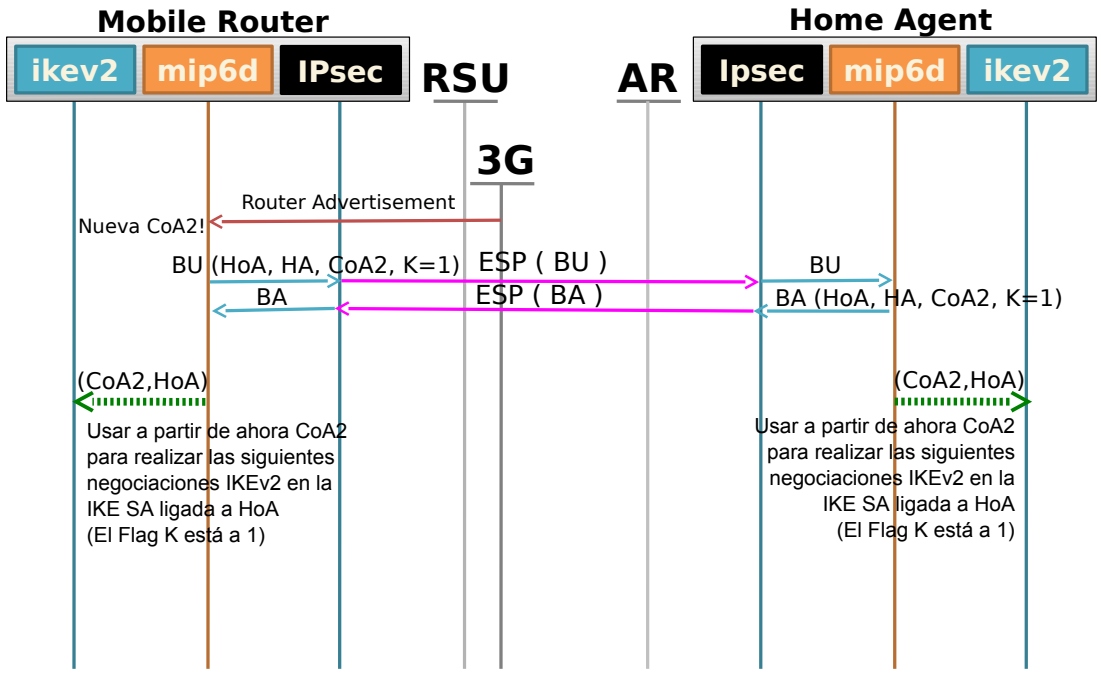


Figura 5.8: Diagrama de secuencia de la creación de una CoA adicional

supone ningún riesgo para la seguridad. En la Figura 5.4, este campo IDi se ha marcado en negrita y rojo en el intercambio IKE_AUTH. Sin embargo, toda negociación para crear IPsec SAs adicionales no tendrán este problema debido a que IKEv2 usará la misma IKE SA para negociarlas, la cual está ya asociada a la HoA.

5.2.2.3. Identificación de las IKE SAs mediante la HoA

Problema: *IKEv2 necesita la HoA para buscar la IKE SA cuando se necesita crear una nueva IPsec SA.*

Causa: La creación de nuevas IPsec SAs para proteger el tráfico de datos puede verse en detalle en la Figura 5.6. El servicio IKEv2 tiene una base de datos interna con las IKE SAs que actualmente tiene establecidas. Dichas IKE SA pueden ser recuperadas de esa base de datos por medio de las direcciones IP de los extremos entre los que están establecidas. En este caso, la HoA del MR y la dirección IP del HA (HAaddr). Sin embargo, la instancia de IKEv2 que inicia una nueva negociación para crear una IPsec SA sobre una de las IKE SA ya establecidas sólo conoce la CoA del MR y la HAaddr.

Solución: El servicio IKEv2 debe preguntar al servicio NEMO por la HoA asociada a la CoA que conoce, ya que las HoAs son exclusivamente conocidas por dicho servicio de movilidad, el cual dispone de una base de datos con las asociaciones CoA-HoA actualizadas. En el caso del MR como iniciador de dicha negociación, la HoA y la HAaddr pueden ser consideradas estáticas, ya que sólo habrá una HoA y por tanto una sola IKE SA en este lado de la negociación. Sin embargo, eso no es así cuando el iniciador de la negociación es el HA, ya que el HA tiene que mantener las conexiones seguras con todos los MR a los que esté dando servicio y, por tanto, en la base de datos de IKE SAs habrá una por cada MR conectado, donde se hace necesaria esta colaboración entre servicios de seguridad y movilidad.

5.3. Conclusiones

En este capítulo se ha demostrado que hay múltiples caminos para conseguir una integración de los servicios de movilidad y seguridad. Se ha estudiado cuál de ellos es el más ventajoso para los entornos C-ITS donde se quieren utilizar. Independientemente del seleccionado, también se ha visto que hay una necesidad evidente de definir una estrategia de comunicación entre las implementaciones de NEMO e IKEv2 para conseguir una mejor cooperación entre ellas. También hemos visto que las asociaciones de seguridad tienen una alta dependencia de la CoA por lo general, teniendo que acudir a soluciones que permitan conservar las asociaciones a pesar de los cambios de CoA.

La implementación de la movilidad del proyecto UMIP, a modo informativo, usa la solución *Migrate* para el cambio de las CoAs, no necesitando a la implementación de IKEv2 para ello. Sin embargo, hemos visto que IKEv2 debe ser notificado en cualquier caso, barajando varias alternativas. Una estrategia razonable para evitar esta comunicación inter-proceso podría ser la de fusionar ambas implementaciones en una sola. OpenIKEv2, nuestra implementación de IKEv2, separó claramente la funcionalidad IKEv2 en librerías para su fácil integración en otras aplicaciones. Se deja como propuesta para el futuro evaluar si dicha integración de IKEv2 dentro de la implementación de NEMO es razonable.

Capítulo 6

Mejoras en el Traspaso

En los capítulos anteriores hemos desarrollado nuestra propuesta de arquitectura, primero desde el punto de vista de las tecnologías que se iban a utilizar, para después describir la incorporación de los servicios de movilidad y seguridad por separado. Hemos realizado nuestra propia implementación de IKEv2 para aportar dicho servicio de seguridad. Finalmente, se ha realizado un análisis de cómo integrarlos y hacerlos funcionar juntos, llevando a cabo una implementación del acercamiento propuesto. En el Capítulo 7 de resultados, después de una extensa batería de pruebas, aparecerá de forma evidente la necesidad de mejorar los procesos de cambio de red, también llamados traspasos. En el presente capítulo nos vamos a centrar en este proceso, realizando inicialmente un análisis del problema en las redes vehiculares, para después proponer soluciones a sus principales inconvenientes.

6.1. El Traspaso (*Handover*)

El traspaso es un procedimiento muy común en escenarios donde las tecnologías inalámbricas son las protagonistas. Este procedimiento consiste en un cambio de punto de acceso de red debido al cambio de posición de un elemento móvil como son los vehículos.

6.1.1. Tipos de Traspasos

Los puntos de acceso, habitualmente estáticos, pueden ser de la misma o diferentes tecnologías, como pueden ser WiFi en todas sus variantes, 3G, 802.11p y Wi-MAX. Estos traspasos entre tecnologías suelen realizarse a nivel de enlace en la pila de referencia OSI. Sin embargo, otros mecanismos en la capa de red pueden darse si el cambio requiere modificar la dirección de red del dispositivo móvil. Se dice entonces que se ha cambiado de red o dominio. Dependiendo de estas dos circunstancias, podemos distinguir estos tipos de traspasos. Si tenemos en cuenta las tecnologías, encontramos:

- **Traspaso horizontal**. Este traspaso es aquel que se realiza entre la misma tecnología y por lo tanto sólo a nivel de enlace, haciendo que sea totalmente

transparente para el nivel de red. Este tipo de traspaso además puede ser **inter** o **intra-dominio** si requerimos hacer un cambio de direccionamiento IP o no, respectivamente.

- **Traspaso vertical**. Este traspaso sin embargo se realiza entre diferentes tecnologías, y por tanto la capa de red es consciente de ello pues suele suponer el uso de interfaces de red distintas. Es por esto que en este tipo de traspasos es habitual un cambio en el direccionamiento de red, y por tanto suele ser **inter-dominio**.

Independientemente del tipo de traspaso, la finalidad de éste es seguir dando servicio de conectividad de forma continua. Sin embargo, esto veremos que no siempre se consigue y que es necesario realizar cambios y aplicar nuevas tecnologías para conseguir que verdaderamente el servicio no se interrumpa en ningún momento.

6.1.2. Fases en el Traspaso

Como acabamos de ver, se trata de reducir ese tiempo de desconexión al mínimo posible en el proceso de traspaso. Para ello, primero vamos a identificar qué fases y procesos pueden formar parte de un traspaso y así poder identificar elementos mejorables:

1. **Fase de inicialización de la tecnología**: En esta fase se produce la inicialización de la tecnología a nivel físico y de enlace. Esta tecnología implementa la interfaz de red a la que queremos pasar. Dependiendo de la tecnología puede conllevar o no una negociación de establecimiento. En el caso de 802.11p no se requiere esta negociación.

2. **Fase de inicialización del nivel de red**: En esta fase se produce la configuración del nuevo interfaz de red. Entran en juego los mecanismos de auto-configuración, como por ejemplo los propios de IPv6: descubrimiento automático de red, auto-asignación de dirección IP y la detección de dirección IP duplicada.

3. **Fase de establecimiento de la seguridad**: Esta fase se produce cuando se necesita aplicar seguridad en las comunicaciones. Para establecer dicha seguridad es habitual realizar una negociación previa, donde se autentica a las partes y se negocian algoritmos y material criptográfico. En nuestro caso la seguridad nos la aporta IPsec. El establecimiento y negociaciones de las asociaciones de seguridad necesarias se llevan a cabo mediante IKEv2.

4. **Fase de autenticación**: Esta fase puede producirse a diferentes niveles: o bien en la fase 1 de negociación del establecimiento del enlace (ej. EAP sobre 802.1x); o bien mediante IKEv2 en la fase 3 de establecimiento de la seguridad (ej. EAP sobre IKEv2). En nuestra propuesta, al usar 802.11p, protocolo caracterizado

entre otras cosas por la ausencia de negociación de establecimiento, será realizada en la fase 3 de establecimiento de la seguridad mediante IKEv2.

5. **Fase de establecimiento de la movilidad**: El servicio de movilidad requiere para establecerse actualizar qué dirección IP está utilizando como *Care-of Address* (CoA), dirección que el *Home Agent* (HA) necesita conocer para comunicarse con el Router Móvil (MR) del vehículo. Para ello se utiliza el intercambio de mensajes *Binding Update* (BU) y *Binding Acknoledgement* (BA). Esta fase se repetirá por cada interfaz disponible, ya que se le asigna a cada una de ellas una CoA distinta, gracias a la extensión del servicio de movilidad llamada *Multiple Care-of Address Registration* (MCoA). El tráfico usará una u otra CoA dependiendo de por qué interfaz quiera ser transmitido.

6.1.3. Traspasos sin Cortes

Cada una de las fases anteriormente enumeradas tiene un consumo de tiempo concreto. Para acortar y/o solapar estas fases con la idea final de hacer que el traspaso sea inapreciable para el usuario, no vale simplemente con mejorar una de estas fases, sino aplicar un conjunto de cambios que agilicen el proceso de manera completa. Por ello se han identificado varias líneas de mejora:

- En la fase de cambio de tecnología podemos mejorar en el tiempo empleado usando el estándar 802.11p cuyo punto fuerte es la ausencia de negociación. El enlace se establece espontáneamente cuando hay cobertura, pudiendo empezar a enviar y recibir paquetes inmediatamente. Esto permite ahorrarnos el tiempo necesario de asociación como pasa en otras tecnologías que sí lo requieren (ej. WiFi). En contrapartida carece de procesos de autenticación y autorización, que obliga a realizarlos en capas superiores de la pila de comunicaciones.

- En la fase de autenticación y establecimiento de la seguridad, fases que en nuestra propuesta se produce de forma anidada, el tiempo empleado dependerá de qué tipo de autenticación estemos realizando. Es habitual realizar autenticaciones sobre EAP basadas en certificados, como puede ser el método de autenticación EAP-TLS, sin embargo no es considerado de los más rápidos. Si queremos ganar velocidad, podemos simplificar esta fase utilizando la autenticación propia que usa IKEv2 mediante claves pre-compartidas (PSK) basado en firma digital. Ganamos así en velocidad pero perdemos en escalabilidad y versatilidad propia de EAP y sus métodos. En la literatura han aparecido métodos que han intentado situarse a mitad de camino entre estos dos métodos anteriores, intentando ganar en tiempo sin perder las ventajas que nos aportan los certificados. Este es el caso del método EAP-FRM (Fast Reauthentication Method [115]).

- Otra línea de mejora la encontramos a la hora de encontrar el momento adecuado para realizar los traspasos. En numerosas ocasiones, elegir bien o mal el momento en el que se realiza un traspaso puede ser determinante en el consumo de tiempo.

El ejemplo más habitual es realizar el traspaso cuando todavía no hay suficiente señal de la tecnología a la que vamos a pasar. Esto supone que las fases anteriores de las que estamos hablando en esta sección tarden más de lo habitual pues en las negociaciones pueden producirse perdidas de paquetes, que supone a su vez realizar retransmisiones. Para disponer de esta característica de anticipar o retrasar los traspasos, en nuestra propuesta de arquitectura de red se ha incorporado el estándar IEEE 802.21, que será el encargado de asistir los traspasos para encontrar el momento más adecuado para realizarlos. Incluso nos permitiría anticipar alguna fase en particular, como por ejemplo la autenticación, antes de que el traspaso se produjera (escenario de pre-autenticación), descrito en el RFC 5836 [95]. IEEE 802.21 utiliza información del medio que le rodea, niveles de señal, posiciones GPS, todo para intentar predecir el mejor momento para realizar dichos traspasos y, por tanto, evitar en lo posible transmitir en zonas donde la cobertura no sea lo suficientemente buena para comunicarse con ciertas garantías de calidad.

- Y por último, otra forma de mejorar los tiempos de traspaso es poder simultanear las fases mencionadas anteriormente con la utilización de la interfaz actual, es decir, poder disponer de conectividad simultánea a través de más de un interfaz de red. Esta nueva característica puede ser incorporada a nivel de movilidad, haciendo que el dispositivo móvil disponga de tantas direcciones IP como interfaces disponga, y usar una u otra dependiendo de ciertas políticas preestablecidas de preferencia. Esto se consigue, como ya establecimos en la Seción 2.3.2.3 del Capítulo 2, incorporando la extensión a la movilidad NEMO llamada *Multiple Care-of Address Registration* (MCoA). Este importante aporte consigue realizar las negociaciones necesarias en las distintas fases de un traspaso mientras se sigue utilizando la conectividad ofrecida por otra interfaz ya inicializada. Conseguimos así reducir prácticamente a cero el tiempo de un traspaso ya que el cambio de usar una interfaz u otra es instantáneo. Sólo depende de las reglas de enrutamiento y políticas presentes en el sistema en cada momento. El tiempo empleado en cada fase no influye en este caso, siempre y cuando se trate de un traspaso vertical. En traspasos horizontales, donde se usa la misma tecnología, no hay posibilidad de usar dos interfaces (en casos con sólo un interfaz por tecnología). En este caso el resto de mejoras recuperan su importancia, ya que ahora no hay posibilidad de simultanear varias conexiones.

En los siguientes apartados veremos en profundidad alguna de las mejoras planteadas anteriormente y que hemos introducido a nuestra arquitectura de red.

6.2. Optimización de la Fase de Autenticación y Autorización

El proceso de autenticación es un paso deseable si queremos restringir el uso de la red sólo a quienes tengan permiso para hacerlo. En nuestra propuesta este proceso

de autenticación y autorización lo realiza IKEv2 en el momento de inicializar la seguridad IPsec de una determinada interfaz. IKEv2, aparte de disponer de un método de autenticación propio basado en firma digital, entre sus características se encuentra la de poder transportar paquetes EAP [1] (RFC 3748) [103], y por tanto soportar los métodos de autenticación existentes basados en este protocolo y los futuros que puedan aparecer. A continuación veremos con más detalle tanto el método de autenticación propio de IKEv2 como algunos ejemplos de métodos basados en EAP.

6.2.1. Método de Autenticación mediante Firma Digital

Este es el método propio que IKEv2 tiene para autenticar ambos extremos en la negociación de una asociación de seguridad IPsec. Cuando se usa este método, se firma digitalmente un bloque de datos que incluirá la petición o la respuesta del primer intercambio IKE_SA_INIT de forma íntegra (según sea el iniciador o respondedor respectivamente), concatenado con un pequeño bloque de datos aleatorios llamado *nonce* intercambiado en ese mismo intercambio, y concatenado a su vez con un resumen digital (o también llamado *hash*) aplicado sobre el propio *payload ID* de identificación.

```
AUTHi = firma/MAC ( MSGi | Nr | hash (IDi) )
AUTHr = firma/MAC ( MSGr | Ni | hash (IDr) )
```

Esto genera el contenido de los payloads AUTHi y AUTHr, y serán intercambiados entre los extremos mediante el segundo intercambio IKE_AUTH, que además pueden ir acompañados de certificados que acrediten que la identificación usada para el cálculo de la firma realmente pertenece a la entidad indicada en el *payload ID* de identidad. Si son muy grandes estos certificados, tanto que el mensaje ya construido exceda el máximo tamaño de transmisión del canal (MTU), se puede usar el método *Hash & URL*, que permite reducir el tamaño del certificado a un pequeño resumen digital (*hash*) del propio certificado y una URL (ej. http://www.servidor.es/certificado.crt) de donde obtener el certificado completo. Simplemente para verificar que es el certificado correcto se repite la operación de *hash* y se compara el obtenido con el enviado junto a la URL. Si son iguales podemos estar seguros de que es el certificado correcto.

En la Figura 6.1 podemos ver el intercambio de mensajes que se producen en este método de autenticación. Como podemos apreciar sólo son dos intercambios, lo que hace que este método sea de los más rápidos soportados por IKEv2.

6.2.2. Métodos de Autenticación mediante EAP

IKEv2 ofrece soporte para el protocolo extensible de autenticación (EAP) [103] que abre la puerta al soporte de múltiples métodos ya implementados basados en este protocolo y la deja abierta a los nuevos métodos de autenticación que puedan aparecer en un futuro. Generalmente estos métodos son asimétricos (un usuario autenticándose contra un servidor AAA) y se usan para autenticar al iniciador. Para autenticar al

[1] Protocolo de Autenticación Extensible

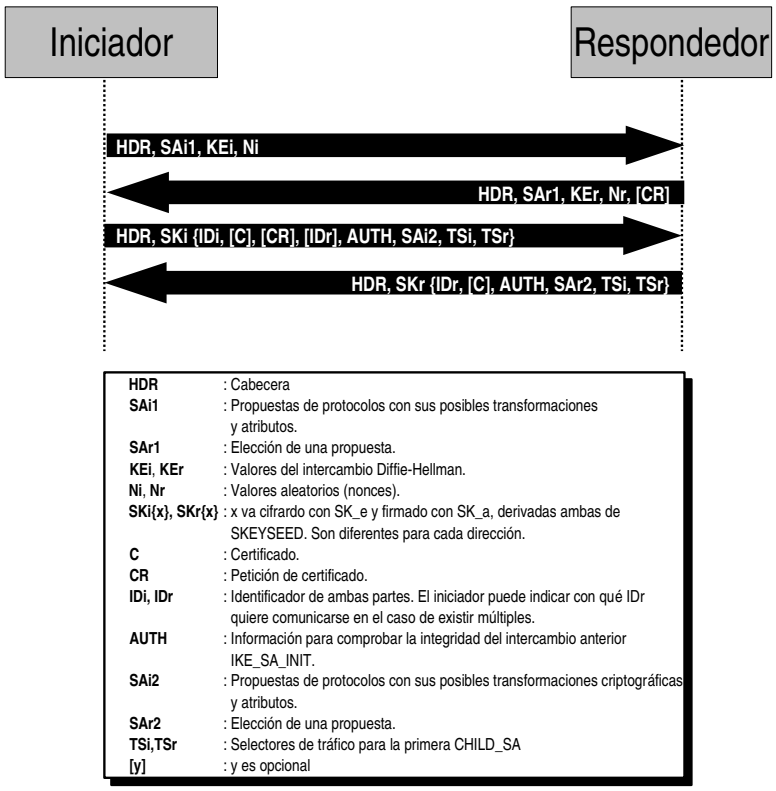


Figura 6.1: Intercambios en la autenticación basada en firma digital en IKEv2

respondedor frente al iniciador se suele usar, sin embargo, el modo de firma digital que hemos visto antes en el apartado anterior.

EAP requiere un número indeterminado de mensajes dependiendo del método de autenticación que transporte, por lo que se implementa en IKEv2 como una serie de intercambios extra de tipo IKE_AUTH que deben ser completados antes de poder dejar creadas las asociaciones de seguridad. Para indicar que se desea usar EAP, el iniciador no incluirá el *payload AUTH* dentro de la petición del intercambio IKE_AUTH. De esta forma se declara una identidad (*payload ID*) pero no se aporta nada para demostrar su autenticidad, trabajo que se delega al protocolo EAP. El receptor de la petición debe retrasar el envío de SAr2 (propuesta de algoritmos con los que proteger los datos transmitidos), TSi y TSr (selectores de tráfico que determinan el tráfico a proteger) hasta que la autenticación tenga éxito. En la Figura 6.2 se muestra un ejemplo del uso de EAP en los intercambios iniciales.

Un método de autenticación basado en EAP utilizado extendidamente es el llamado EAP-TLS [104]. Al estar basado en certificados, éste necesita intercambiarlos entre los extremos, con lo que requiere de un gran número de intercambios que dependerá del tamaño de los mismos. Esto hará incrementar el tiempo empleado en la autenticación.

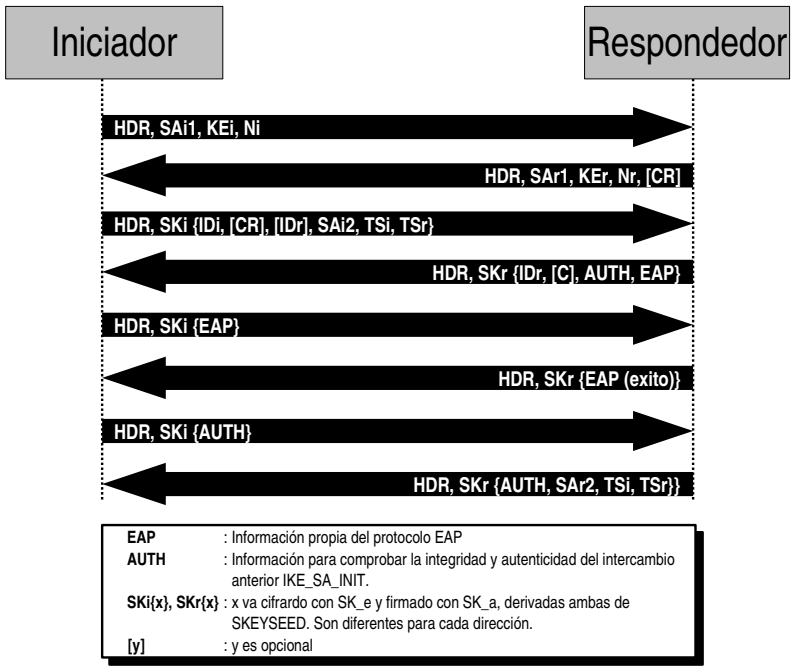


Figura 6.2: Intercambios en la autenticación IKEv2 con EAP

Además influirá decisivamente el retardo que exista hasta el servidor de autenticación AAA. Otro método interesante es EAP-FRM [115], propuesto por investigadores del mismo grupo de investigación al que pertenece el autor de esta Tesis, que persigue encontrar un método que en una primera negociación inicial fuera equivalente a un EAP-TLS, pero que las re-autenticaciones consiguieran ser más rápidas. Sin querer entrar en más detalle, queremos hacer una comparativa de tiempos de autenticación entre los métodos mencionados. Para ello se ha repetido y medido cada método de autenticación 150 veces y después calculado la media de los tiempos. Los resultados pueden apreciarse en la Figura 6.3, donde se ve claramente que el método de firma digital es el más rápido, seguido de las re-autenticaciones de EAP-FRM (2 veces más lento) y por último EAP-TLS (aproximadamente 10 veces más lento).

Por tanto, podemos concluir que el método de autenticación utilizado será determinante en el tiempo empleado por los traspasos. Por eso, dependiendo de nuestras necesidades, deberemos elegir el que más nos convenga.

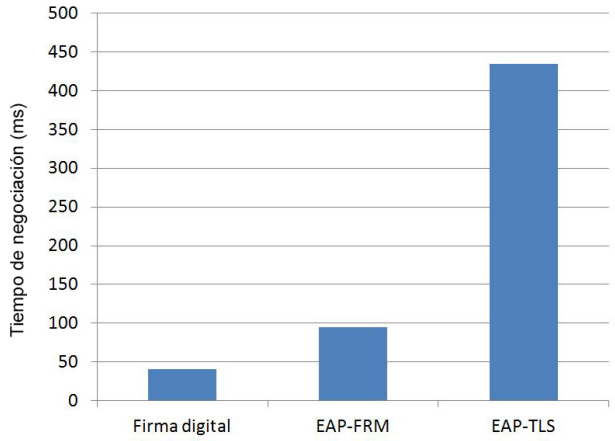


Figura 6.3: Comparativa de tiempos de autenticación con diferentes métodos

6.3. Uso Simultáneo de más de una Interfaz: MCoA

Los dispositivos móviles actuales siguen una tendencia clara a la hibridación de las comunicaciones, es decir, disponer de varias interfaces con distintas tecnologías inalámbricas para poder usar la que más convenga en cada momento. Ejemplos de estas tecnologías son 3G/4G, WiFi, Wi-MAX, y las específicas para el mundo vehicular como 802.11p.

El disponer de más de una interfaz con distintas tecnologías hace que se pueda dar el traspaso vertical, pues ya no es sólo una tecnología con la que trabajamos. Este tipo de traspasos tiene una gran ventaja, ya que mientras se está inicializando una interfaz en un traspaso, la interfaz que actualmente estuviéramos usando sigue estando disponible, pudiendo usar las dos interfaces simultáneamente. Esto precisamente es aprovechado por la extensión "Multiple Care-of Address" (MCoA) descrita en el RFC 5648 [48], que sirve tanto para MIPv6 como para NEMO. Esta extensión modifica el protocolo de movilidad y sus estructuras internas para posibilitar el hecho de que un dispositivo móvil disponga de más de una CoA, una por interfaz disponible en un determinado momento. Como el "Home Agent" (HA) tendrá que asociar ahora varias CoAs a una misma HoA, es necesario añadir una identificación a cada una para poder distinguirlas entre sí. Esta identificación se denomina *Binding ID* (BID) y deberá incorporarse a los mensajes *Binding Update* (BU) y *Binding Ack* (BA). También en la caché de asociaciones HoA-CoA tendremos que incluir este identificador y así, a través de la HoA y un BID podremos obtener una única CoA:

```
{HOA BID}  COA   MNP
----------------------
 HoA BID1  CoA1  MNP1
 HoA BID2  CoA2  MNP2
 HoA BID3  CoA3  MNP3
```

Nótese que en este caso, al ser NEMO el protocolo que estamos utilizando, también se asociará el prefijo de red IPv6 que define la red interna de cada nodo móvil, que al tener una red a su cargo y disponer de capacidades de enrutamiento son considerados como enrutadores móviles (o *Mobile Routers / MR*), formando parte fundamental de cada Estación Vehicular ITS.

Este identificador BID sienta las bases de la definición de "flujo de datos" que introduce el RFC 6089 [87] y que permite que un determinado tráfico utilice una u otra interfaz. Esta identificación de flujos de tráfico se realiza mediante marcado de paquetes, con el mismo valor de BID que el asociado a la CoA que se quiera utilizar. Por el momento, el servicio de movilidad que utilizamos sólo tiene definida una relación de preferencia entre interfaces, y usará solamente aquella con más prioridad a pesar de tener más de una interfaz disponible.

De esta forma, si por ejemplo estamos conectados a través de tecnología 3G y descubrimos que tenemos cobertura 802.11p disponible, mientras se inicializa esta nueva interfaz, la otra seguirá disponible y no se interrumpirá el servicio. Una vez que el servicio de movilidad actualiza la caché de asociaciones con un nuevo BID y una nueva CoA, entonces se usará la de mayor prioridad, que en este caso establecemos que sea 802.11p. Este cambio es prácticamente instantáneo y no produce ninguna pérdida ni interrupción de la conectividad ya que sólo conlleva una actualización en el conjunto de políticas de enrutamiento.

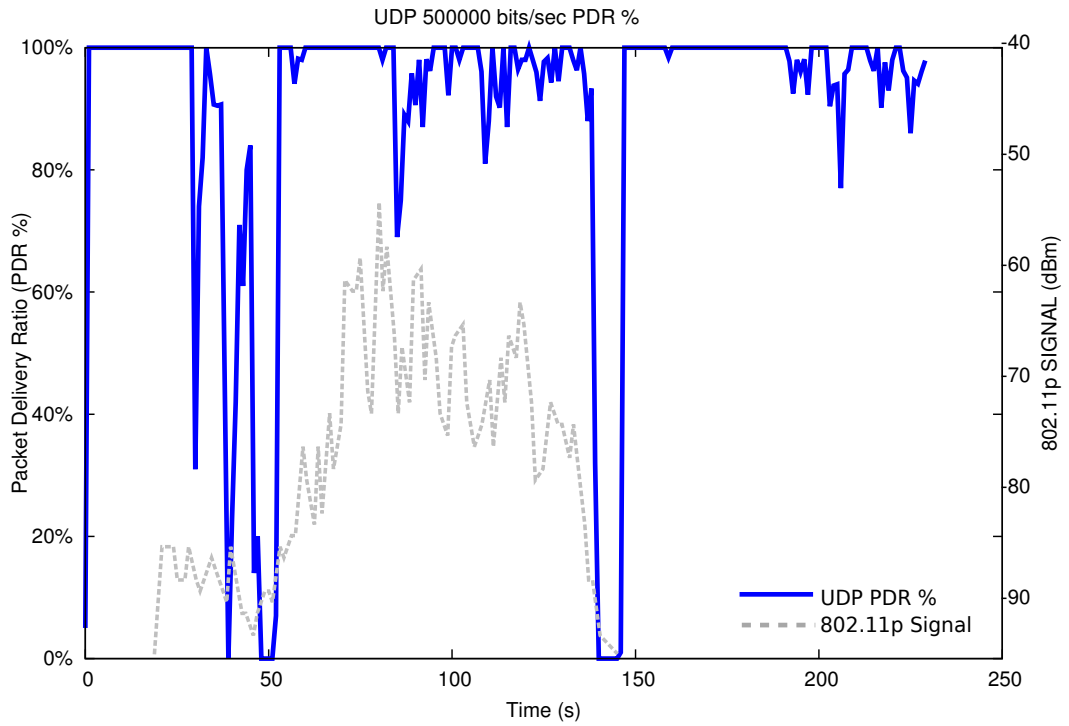


Figura 6.4: PDR en una vuelta con tecnologías 3G y 802.11p usando MCoA

Sin embargo, cuando se pierde la cobertura 802.11p, no se cambia a tecnología 3G hasta que la entrada en la cache de asociaciones HoA-CoA no caduque. Esta es la forma de actuar habitual de la movilidad NEMO. Este tiempo dependerá del tiempo de vida que tengan dichas asociaciones, normalmente entre 10 y 60 segundos. Esto supone que habrá un espacio de tiempo en el que se podría estar utilizando la interfaz 3G para enviar el tráfico pero no se hará porque se sigue intentando hacer por 802.11p. Esta desconexión temporal puede apreciarse perfectamente en la Figura 6.4 (figura adelantada del siguiente Capítulo 7 de pruebas y resultados) en donde se refleja el porcentaje de paquetes entregados (PDR) de un flujo de datos UDP constante durante una vuelta al circuito de pruebas, justo cuando la cobertura 802.11p desaparece.

Esto podría ser solucionado haciendo que la movilidad cancelara dicha asociación en cuanto determinara que la comunicación ya no es viable. El decidir la viabilidad de un interfaz no está definido en las especificaciones de MIPv6 o NEMO, con lo que nos hace falta algo más que nos ayude a determinar cuándo una interfaz es apta para el uso y cuándo no. La solución a esto tiene que ver con la siguiente mejora que tratamos a continuación, es decir, la incorporación del estándar IEEE 802.21 a la arquitectura para determinar cuándo es el mejor momento para realizar los traspasos utilizando para ello información del entorno.

6.4. Asistencia al Traspaso mediante IEEE 802.21

Como ya presentamos en el Capítulo 2, el estándar IEEE 802.11 establece unas entidades funcionales y un protocolo entre ellas que colaboran para recoger información del entorno y, según unas preferencias establecidas, escoger cual es la mejor interfaz para comunicarse en cada momento y establecer el mejor momento para realizar el traspaso de una interfaz a otra en caso de ser necesario. Para incorporarlo a nuestro prototipo hemos usado un software llamado ODTONE, del que hablaremos a continuación.

6.4.1. Implementación de IEEE 802.21

Como hemos adelantado, existe una implementación de código abierto del protocolo 802.21 llamada ODTONE, la cual, a pesar de estar en un estado poco avanzado, es suficiente para nuestros requerimientos y puede ser considerado como buen punto de partida para incorporar 802.21 a nuestra infraestructura. Este primer paso se ha centrado en conseguir compilar las entidades para que funcionen en nuestros dispositivos y dotarlas de una funcionalidad mínima.

Hemos incorporado inicialmente la entidad MIH-MN en el *Router Movil* (MR). El traspaso será llevado a cabo solamente por esta entidad, dejando a un lado entidades como MIH-PoS, MIP-PoA o MIIS-Server. En la Figura 6.5 podemos ver estas entidades de forma gráfica, resaltando las que usaremos. La toma de decisiones se va a realizar usando información que pueda ser obtenida dentro del mismo MR, como es el nivel de fuerza de la señal 802.11p. Dicha fuerza de señal será obtenida periódicamente del sistema por el MIES, y será transmitida al MIHF como un evento, que será

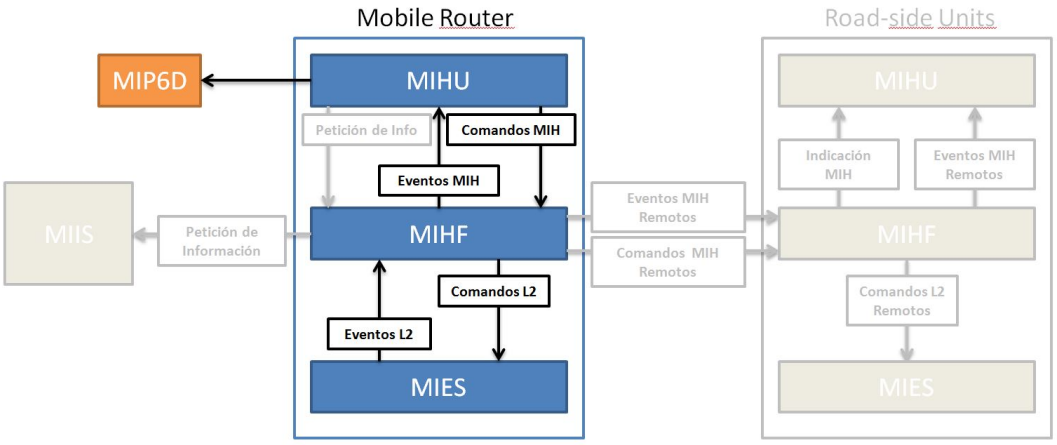


Figura 6.5: Implementación mínima de IEEE 802.21

retransmitido a su vez al MIHU. Esta última entidad será la que conozca el valor de los umbrales y tome la decisión de cuándo realizar los traspasos, generando comandos e interactuando con otras aplicaciones, como en este caso con el servicio de movilidad NEMO implementado por "Mip6d".

La idea que hemos llevado a cabo para controlar que la movilidad realice el cambio de tecnología cuando 802.21 lo determine ha sido la de inhibir la recepción de los mensajes *Router Advertisements* (RA) de un determinado interfaz, mensajes que señalan la presencia de un enrutador de acceso IPv6. Será 802.21, y en concreto el módulo MIHU el que decida en cada momento activar o desactivar esa inhibición en cada una de las interfaces disponibles. En nuestro caso disponemos de las tecnologías 3G y 802.11p.

El algoritmo de elección de red que toma las decisiones de traspasar de una tecnología a otra puede hacerse muy complejo y usar mucha información del entorno y la que pueda obtener de la entidad MIIS-Server. Sin embargo, para esta ocasión y como primer paso hemos usado el nivel de señal inalámbrica y definido umbrales para decidir entre el uso de una u otra tecnología. Hemos añadido algo más de inteligencia a esta toma de decisiones definiendo dos umbrales distintos en vez de uno sólo. Con ello evitaremos el efecto ping-pong que provoca un traspaso intermitente entre dos tecnologías cuando el nivel de señal está cercano a un único umbral. La Figura 6.6, representación gráfica de una de las vueltas extraídas de las pruebas realizadas que serán mostradas con detalle en el próximo capítulo, se pueden observar ambos umbrales. Podemos distinguir qué tecnología se está usando en cada momento basándonos en los valores de RTT entre el MR y el HA, mucho mayores en el caso de 3G que de 802.11p. Sin embargo puede apreciarse que hay una pequeña vuelta a 3G por una caída muy breve de señal 802.11p.

Para evitar estos micro traspasos podemos ser más conservativos y esperar hasta que, por ejemplo, tres lecturas consecutivas de la señal 802.11p sobrepasen los umbrales

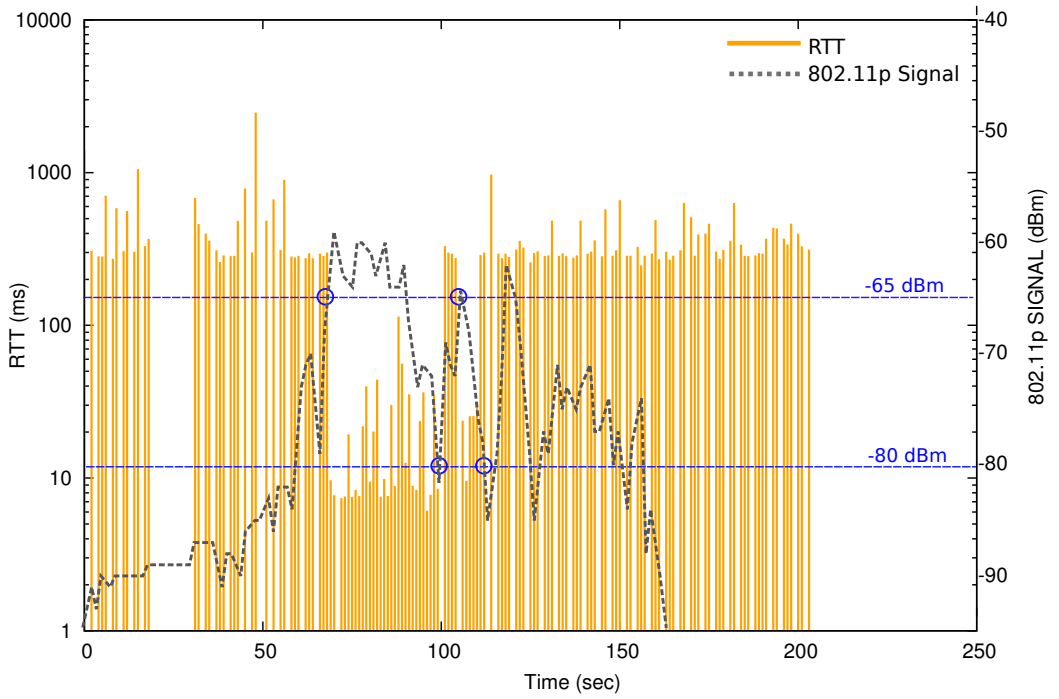


Figura 6.6: Valor de RTT durande la realización de traspasos entre 3G y 802.11p

marcados, en vez de solamente una. También elegir bien el valor de estos umbrales es determinante, y por eso previamente se ha realizado un estudio de la señal recibida a lo largo del circuito y, en función de esto, hemos definido unos umbrales que nos han parecido razonables después de realizar unas vueltas preliminares.

6.4.2. Resultados de la Incorporación de IEEE 802.21 a Nuestra Propuesta

Con la incorporación de IEEE 802.21, en conjunción con la extensión de la movilidad MCoA que hemos visto en uno de los apartados anteriores, conseguimos reducir a prácticamente cero el tiempo de traspaso entre una tecnología a otra, consiguiendo resolver el problema que teníamos de incomunicación al volver a la tecnología 3G desde 802.11p. Los resultados mostrados en la Figura 6.7 refleja que los valores porcentuales de PDR en otra de las vueltas realizadas en las pruebas adicionales donde sí incorporamos 802.21. A pesar de que la comunicación no es perfecta y se pierden algunos paquetes, no encontramos ninguna ruptura de comunicaciones temporal como ocurría en la Figura 6.4, ni tampoco unos valores de PDR muy bajos en zonas de traspaso. Esto es así porque ahora los traspasos se realizan cuando la señal 802.11p es más alta, evitando así pérdidas innecesarias. En concreto, el primer traspaso desde tecnología 3G hacia 802.11p se retrasa hasta que el nivel de señal suba a niveles

aceptables (umbral de conexión).

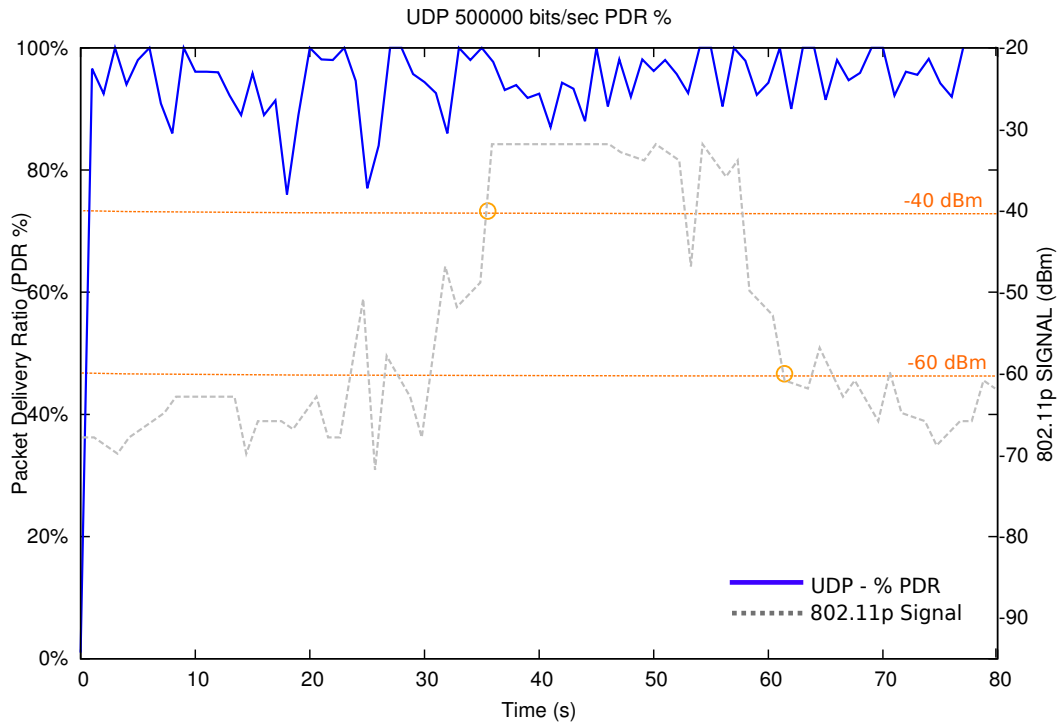


Figura 6.7: PDR en una vuelta con tecnologías 3G y 802.11p usando MCoA asistido con IEEE 802.21

En el segundo traspaso, desde tecnología 802.11p hacia 3G, se adelanta antes de que la señal 802.11p se extinga (umbral de desconexión) y no permita cancelar la asociación de movilidad HoA-CoA. Esta cancelación se produce mediante el envío de un menaje de *Binding Update* al HA con tiempo de vida cero. Es importante señalar aquí que aunque el servicio de movilidad sea el que mande este mensaje, es 802.21, y en concreto el módulo MIHU, el que ha tomado la decisión de realizar el traspaso y provocar esa cancelación temprana de la CoA.

6.5. Conclusiones

Los traspasos han resultado ser uno de los principales problemas de las redes vehiculares, ya que los servicios básicos de movilidad que ofrece NEMO no son capaces de gestionarlos adecuadamente. Para ello, se ha propuesto añadir distintas tecnologías que contribuyan a la desaparición de los cortes en las comunicaciones que se vienen produciendo durante los traspasos. Hemos constatado que tecnologías como IEEE 802.21 y mejoras del servicio de movilidad como el aportado por MCoA, son decisivos para conseguir eliminar dichos cortes y por tanto, aumentar la calidad de

las comunicaciones percibida por los usuarios. También hemos distinguido otro frente en el que trabajar para recudir aún más el tiempo empleado en los traspasos, que es mejorando los métodos de autenticación. En esta línea, el protocolo EAP cobra un protagonismo importante, ya que permite el intercambio de métodos de autenticación de una forma fácil y extensible, permitiendo elegir aquellos compatibles con EAP que beneficien a los traspasos, es decir, los más rápidos.

Capítulo 7

Pruebas y Análisis de Resultados

Establecida nuestra propuesta es necesario llevar a cabo una comprobación empírica de que las mejoras propuestas realmente lo son y en qué medida. Para ello hemos tenido la suerte de haber podido realizar un despliegue real de red vehicular, localizado en el Campus de la Universidad de Murcia. Como se explica en este capítulo, se trata de un espacio real de conducción que se acerca a una situación cotidiana y realista. Entre otras, estas pruebas pretenden dar respuesta a las siguientes cuestiones:

- Comportamiento del retardo de los mensajes.
- Capacidad de la red de entregar los paquetes que se envían.
- Encontrar los valores máximos de ancho de banda.
- Valorar el impacto que supone la aplicación de la seguridad.
- Comparar las tecnologías de transmisión de datos utilizadas (3G y 802.11p).
- Estudiar el comportamiento de las comunicaciones durante los traspasos entre las tecnologías anteriores (*handovers*), tanto con como sin el uso de 802.21.
- Comprobar si la velocidad del vehículo influye en los resultados obtenidos.

La configuración del escenario de pruebas, planificación y resultados obtenidos son comentados con detalle en los siguientes apartados.

7.1. Configuración del Escenario de Pruebas

Se trata de desplegar un escenario que nos permita probar nuestra propuesta de aplicación de los protocolos NEMO con su extensión MCoA, IPsec e IKEv2 para obtener una red vehicular con servicios de movilidad y seguridad, así como traspasos suaves. Se han tenido en cuenta dos tecnologías distintas para poder dar lugar a dichos traspasos verticales y poder así estudiarlos en detalle. Estas tecnologías son 3G, como tecnología más presente, y 802.11p, tecnología específica para redes vehiculares.

7.1.1. Despliegue de la Arquitectura de Red

El escenario de pruebas mostrado en la Figura 7.1 ha sido desplegado dentro de las infraestructuras de la Universidad de Murcia. Como puede apreciarse en la figura, los tres tipos de estaciones ITS están presentes, que ya tratamos en profundidad en la sección 3.1. En el diagrama se distingue perfectamente las dos vías de comunicación disponibles, una por medio de la tecnología 802.11p, usando para ello el canal de control definido en el perfil ETSI G5 [15], y la otra por medio de tecnología 3G desplegada por operadoras comerciales. Se muestra también el esquema de direccionamiento IP, usando direcciones IPv6 dentro de un rango de IPs específico para pruebas según los estándares. Debido a que las operadoras de 3G no ofrecen conectividad IPv6, hemos utilizado túneles OpenVPN sobre IPv4 para salvar este problema de conectividad. Además hemos montado una estación de carretera ITS conectada mediante red cableada a la estación central ITS, donde están a su vez instalados el servidor de movilidad HA y el enrutador que nos conecta a la red de la Universidad y por lo tanto a Internet. Hemos utilizado este enrutador como destino para el envío y recepción de paquetes en las pruebas que hemos realizado y así no depender del estado de la red de la Universidad, que podría influir negativamente en la fiabilidad de los resultados.

También se han desplegado distintos escenarios de pruebas dentro del marco del proyecto FOTsis [20], usando para ello tramos de carreteras más importantes, como por ejemplo, las autovías A2 y M12. Estas pruebas ayudaron en gran medida a mejorar el funcionamiento de nuestra propuesta.

7.1.2. Equipamiento de la Red Vehicular

La Figura 7.2 muestra parte del equipamiento usado. La antena de la estación de carretera ITS ha sido colocada en una ventana de la Facultad de Informática, mientras que la estación central ha sido instalada en un laboratorio cercano dentro del mismo edificio. La estación vehicular ITS ha sido instalada en uno de los coches de la flota de vehículos de la Universidad de Murcia, en el que está incluido el *Router* Móvil (MR) ofreciendo conectividad WiFi dentro del coche. Las antenas usadas, tanto la instalada en la ventana del edificio (Figura 7.2a) como la montada en el techo del coche (Figura 7.2c) están especialmente diseñadas para mejorar la recepción de la señal 802.11p en términos de ganancia.

Para favorecer su lectura y claridad, todos los componentes usados en este escenario de pruebas se muestran listados en la Tabla 7.1. Como puede apreciarse en la tabla, el mismo equipamiento hardware se utiliza tanto para el AR de la Estación de Carretera ITS como para el MR de la Estación Vehicular ITS. La pila de comunicaciones ITSSv6 está presente tanto en el HA de la central como en el MR del vehículo. Esto es así pues dicha pila da soporte a los servicios de movilidad y seguridad, que incluyen cambios y mejoras en el núcleo del sistema que permiten hacer funcionar correctamente dichos servicios. Dentro de la estación de carretera ITS, el AR también usa la pila ITSSv6, y a pesar de que en este caso no es necesario dicho soporte ofrecido por la pila, ésta viene instalada por defecto en dicho dispositivo usado como AR (Laguna). En el resto

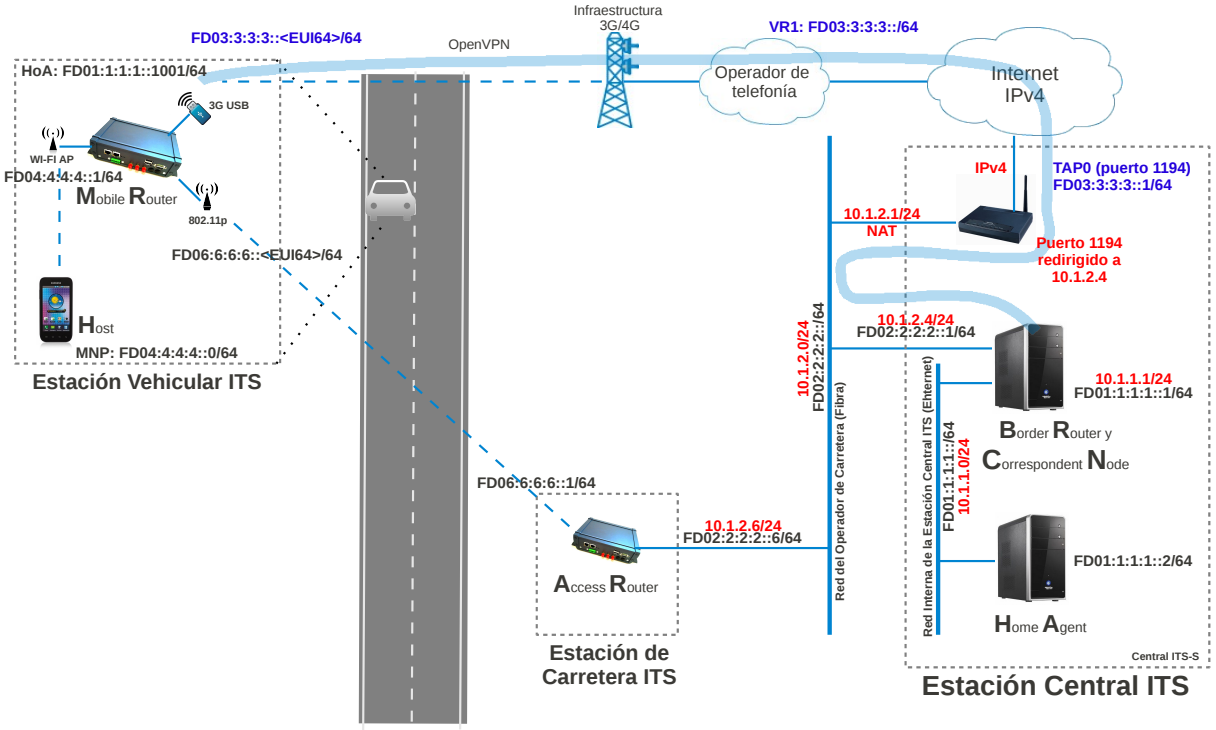


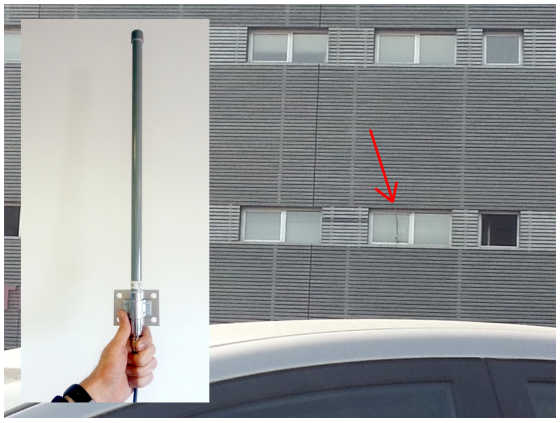
Figura 7.1: Escenario ITS desplegado en la Universidad de Murcia

de nodos se usa una distribución Linux habitual llamada Ubuntu en su versión 12.04 LTS.

Adicionalmente, se muestra una lista de los parámetros más importantes usados en los componentes anteriores en la Tabla 7.2. Solo cabe destacar aquí que el periodo de envío de mensajes *Router Advertisements* (RA) se ha reducido en el caso de la interfaz 802.11p con el propósito de minimizar el tiempo empleado en detectar la presencia de un AR de una estación de carretera ITS. El resto de valores pueden considerarse normales.

7.2. Plan de Pruebas

El propósito de esta campaña de pruebas es comprobar el funcionamiento de los mecanismos de movilidad y seguridad y observar el rendimiento de la transmisión de datos durante los traspasos de una tecnología a otra (*handovers* verticales). Otro objetivo es comparar las tecnologías de comunicación 3G y 802.11p, analizando la



(a) Antena omni-direccional de la Estación de Carretera ITS

(b) Vehículo de la flota de la Universidad de Murcia

(c) Antena combinada del vehículo con soporte magnético

(d) Router Móvil de la Estación Vehicular ITS

Figura 7.2: Equipamiento usado en el escenario de pruebas

incidencia en el rendimiento cuando la seguridad es aplicada conjuntamente con la movilidad, comparándolo con la aplicación solamente de movilidad.

7.2.1. Metodología

Dicha campaña consiste en realizar una serie repetida de vueltas con un vehículo siguiendo un recorrido fijo a través del Campus de Espinardo que cruza una zona de cobertura 802.11p limitada solamente a una parte del circuito. Dicho recorrido y niveles de cobertura son mostrados en la Figura 7.3. Sin embargo, la cobertura 3G está siempre presente a lo largo de todo el recorrido. Cada vuelta empezará fuera de la cobertura 802.11p (flecha amarilla) y, por tanto, sólo dispondrá de 3G para comunicarse. Al entrar a la zona de cobertura 802.11p se produce el primer traspaso, este de 3G a 802.11p. Pasados unos trescientos metros, el vehículo vuelve a salir de dicha cobertura 802.11p,

Tabla 7.1: Componentes usados en el escenario de pruebas

Entidades de la red ITS		
Nodo	**Hardware**	**Software**
MR	Laguna LGN-0011	Pila ITSSv6
Host	Portatil, Intel i7, 4GB	Ubuntu 12.4
HA	PC Via 532Mhz, 476MB	Pila ITSSv6
BR/CN	PC Intel i5, 3.1Ghz, 3GB	Ubuntu 10.4
AR	Laguna LGN-0011	Pila ITSSv6
Componentes de comunicaciones		
Artículo	**Modelo**	
3G USB	TP-LINK MA-180	
Emisor/Receptor 802.11p	Unex DCMA-86P2 mini-PCI en AR/MR	
Antena de vehículo	Omni-combinada 3G/ 11p/ GPS 7dBi	
Antena de carretera	Omni 12dBi	

volviendo otra vez a usar la tecnología 3G en un segundo traspaso.

Los pasos que se dan en una de las vueltas los detallamos a continuación:

1. En el caso donde la seguridad es aplicada, el MR usa IKEv2 para establecer la IPsec SA que protegerá el tráfico intercambiado con el HA a través de la tecnología 3G.

2. El MR que va incorporado al vehículo y ya conectado a la red 3G entra en la zona de cobertura 802.11p, perteneciente a una red distinta (traspaso vertical entre dominios diferentes). Mientras la conexión establecida anteriormente mediante 3G se mantiene en un segundo plano, el MR se conecta con el AR una vez se reciba un mensaje de *Router Advertisement* (RA), generándose una nueva CoA para poder acceder a esta nueva red. En este momento es cuando se activa el procedimiento de movilidad, haciendo registrar dicha CoA en el HA, utilizando para ello la misma conexión 802.11p recién establecida. Destacar aquí que 802.11p no requiere de ningún proceso de asociación a nivel de enlace. El mero hecho de recibir un RA implica que el enlace se considera establecido. A partir de este momento y debido a la extensión *Multiple Care-of Address* (MCoA) de NEMO, disponemos de dos CoAs con las que poder comunicarnos. Como la movilidad tiene establecida una relación de prioridad entre las interfaces de red y en este caso se le da más prioridad a la interfaz 802.11p, el tráfico empieza a marcarse con el identificador BID (*Binding ID*) de la CoA asociada a la interfaz 802.11p. Esto provoca una nueva negociación IKEv2 para establecer la IPsec SA necesaria

Tabla 7.2: Parámetros de configuración usados

Nodo	Servicio	Parámetro	Valor
AR	radvd	Perido de Router Adv. (3G)	3-4 s
AR	radvd	Perido de Router Adv. (11p)	0.2-0.6 s
MR	mip6d	Tiempo de vida del *binding*	12 s
MR y HA	openikev2	Encriptación ESP	3DES
MR y HA	openikev2	Algoritmo AH	SHA1
MR y HA	openikev2	Grupo DH	2

para proteger dicho tráfico de datos. Esta negociación sólo se da en el primer contacto con la cobertura 802.11p, pero para el resto de vueltas al circuito ya no será necesario rehacer dicha asociación, pues todavía estará disponible.

3. El coche permanece en movimiento con velocidad constante circulando por zonas con diferentes niveles de cobertura 802.11p, pero finalmente termina saliendo de ella, y por tanto, cuando los mecanismos de movilidad son conscientes de que no se ha podido renovar la CoA de la interfaz 802.11p, se vuelve a marcar el tráfico con el identificador asociado a la CoA asignada a la interfaz 3G para que el tráfico pueda ser de nuevo intercambiado entre el MR y el HA, pero a través de 3G (de nuevo un traspaso vertical entre dominios diferentes). El coche continúa bajo cobertura 3G hasta el final de la vuelta al circuito.

En total se han realizado 200 vueltas al circuito, 100 aplicando sólo movilidad y otras 100 aplicando además seguridad. Además habría que añadir otras 25 vueltas más que fueron necesarias realizar para el estudio de la incorporación de 802.21. Este gran número de vueltas nos proporciona unos resultados estadísticamente confiables, permitiendo alcanzar conclusiones sobre el impacto de la movilidad y la seguridad en un escenario real, y por tanto minimizar las incidencias de congestión puntuales en las redes 3G, atascos en el circuito, etc.

Cada una de las vueltas ha sido llevada a cabo a diferentes velocidades del vehículo con la intención de analizar si este factor influye en el rendimiento de las comunicaciones. En particular, las velocidades elegidas han sido de 10, 20, 30 y 40 Km/h y, además, para cada una de estas velocidades, las pruebas se han repetido cinco veces.

7.2.2. Métricas

Las pruebas han sido realizadas con diferentes protocolos que nos han permitido medir determinadas características de las comunicaciones. Estos han sido usados por separado en vueltas independientes para evitar interferencias entre los mismos. El

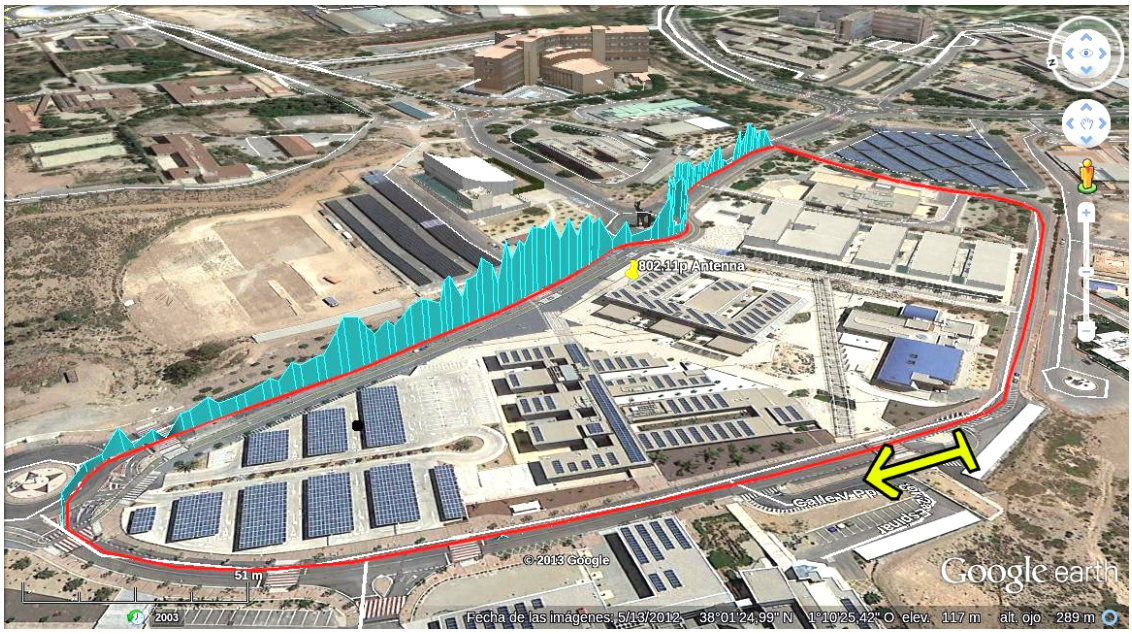


Figura 7.3: Recorrido de cada prueba y el nivel de señal 802.11p a lo largo de ella

tráfico era generado y recibido, dependiendo del caso, desde un ordenador portátil (HOST de la Estación Vehicular ITS) conectado a la red interna del vehículo desplegada por el MR. En una prueba completa (5 vueltas) a una cierta velocidad de circulación del vehículo, se han usado las siguientes métricas:

- Máximo ancho de banda del canal: consiste en averiguar cuáles son las máximas capacidades del canal de comunicaciones en cuestión. Nos servirá para ver que cada tecnología tiene un máximo ancho de banda disponible. Para ello se da una vuelta al circuito usando el protocolo TCP. Este protocolo de transporte incrementa la tasa de envío mediante el mecanismo llamado *Slow-Start*, hasta que algún paquete se pierda, momento en el que reduce de nuevo la tasa de transferencia. No da una lectura de cuanta información es capaz de transmitir corrigiendo los errores que pudieran producirse. El tráfico es generado desde el Host de la Estación Vehicular ITS y enviado al enrutador de la Estacion Central ITS.

- Retardo de la red en segundos: es medido mediante el Round-Trip Time"(RTT), que consiste en averiguar el tiempo que emplea un paquete ICMPv6 en ir de su origen al destino y volver. Es deseable que este tiempo sea cuanto más bajo mejor. Se envía un paquete ICMPv6 de 56 bytes y se espera la contestación antes de enviar el siguiente, donde se obtiene el tiempo que tarda un paquete en ir al HA de la central y volver al Host dentro de la red móvil del MR. Se ha elegido el HA como objetivo de esta prueba porque el interés está en el retardo que hay entre el MR y el HA, extremos de los túneles de seguridad y movilidad.

Tabla 7.3: Valores medios obtenidos en las pruebas con ICMPv6 con intervalo de confianza al 95 %

Velocidad (Km/h)	**Media 3G (ms)**	**Media 3G con IPsec (ms)**	**Media 11p (ms)**	**Media 11p con IPsec (ms)**
10	309.43 ±29.82 (601 muestras)	191.68 ±9.19 (780 muestras)	12.4 ±0.73 (659 muestras)	10.85 ±0.37 (759 muestras)
20	277.96 ±30.70 (489 muestras)	242.17 ±28.76 (385 muestras)	13.3 ±1.19 (425 muestras)	14.15 ±1.09 (359 muestras)
30	187.09 ±16.44 (349 muestras)	222.59 ±22.12 (467 muestras)	11.0 ±0.91 (302 muestras)	13.76 ±1.12 (361 muestras)
40	173.42 ±13.31 (258 muestras)	174.88 ±12.02 (294 muestras)	11.1 ±1.02 (233 muestras)	15.7 ±1.68 (199 muestras)

- Tasa de paquetes entregados: también llamado "Packet Delivery Ratio" (PDR), es el porcentaje de paquetes que son enviados y llegan a su destino. Está claro que cuantos menos paquetes se pierdan por el camino, mejores serán las comunicaciones, ya que no se perderá tiempo con reenvíos. Para ello se dan tres vueltas con el vehículo usando el protocolo UDP, generando tráfico a una tasa de transferencia constante en cada vuelta: 500, 1000 y 2000 Kbps. El tráfico es generado en el enrutador de la Estación Central ITS y enviada al Host de la Estación Vehicular ITS conectado a la red móvil del vehículo, enviando 1230 bytes por paquete UDP.

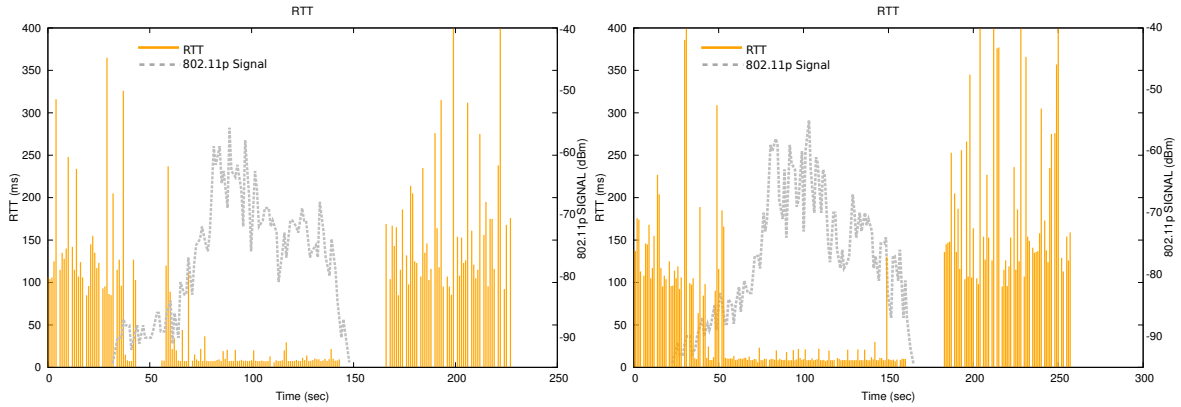
Se ha utilizado la herramienta "iperf" para generar el tráfico UDP y TCP. Sin embargo, para ICMPv6 se ha usado el habitual comando "ping6". Para más detalles de cómo se ha generado el tráfico, como se ha capturado y como se han obtenido los resultados que a continuación mostramos, consulte el Apendice C.

7.3. Resultados

7.3.1. Estudio del Retardo en el Envío de Paquetes

Para tener una visión más gráfica de cómo han transcurrido estas pruebas, nos fijaremos en una de ellas realizada a 20 Km/h, con y sin seguridad, cuya representaciones gráficas las podemos encontrar en la Figura 7.4. En ellas podemos apreciar los valores de RTT en milisegundos obtenidos a lo largo de esta vuelta en concreto. Los valores de RTT cuando se usa la tecnología 802.11p son muy bajos, comparados con los valores obtenidos mediante la tecnología 3G.

Estos valores son resumidos en la Tabla 7.3, donde puede apreciarse esa gran diferencia entre los valores medios de RTT obtenidos sobre 802.11p y sobre 3G. Además, puede apreciarse mediante los intervalos de confianza aportados, que la variabilidad de los valores obtenidos en 3G es mucho más grande que los obtenidos con 802.11p, hecho que puede observarse también en la Figura 7.4.



(a) RTT sin seguridad (b) RTT con seguridad

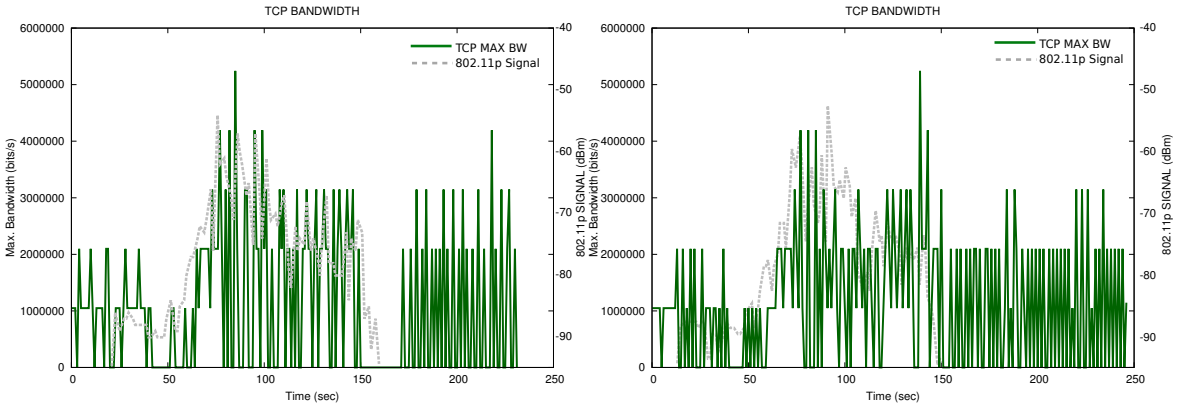
Figura 7.4: Evaluación del tráfico ICMPv6 en términos de RTT

Además, podemos constatar que cuando se incrementa la velocidad del vehículo, el valor medio del RTT baja. Una de las posibles causas es porque a velocidades más altas, el vehículo permanece menos tiempo en las zonas de traspaso de tecnologías, zonas donde la señal de 802.11p es demasiado débil y no se puede asegurar una buena transmisión y muchos de los paquetes se pierden. En cuanto a la aplicación de seguridad, podemos observar que los resultados son muy similares y no se puede afirmar que la aplicación de seguridad suponga un impacto significativo en el rendimiento de las comunicaciones a nivel de retardo. También hay que decir que esta prueba es poco exigente en cuanto a información transmitida, pues se transmiten bidireccionalmente 64 bytes por segundo aproximadamente.

7.3.2. Estudio del Ancho de Banda Máximo

Como en el tipo de prueba anterior, exponer de manera gráfica los resultados obtenidos nos ha ayudado bastante a comprenderlos y obtener conclusiones. En estas pruebas se quiere estudiar cuál es el máximo ancho de banda alcanzable en cada tecnología. Para ello usaremos tráfico TCP, que transmite información sin pérdida, es decir, recuperando fallos. En la Figura 7.5 se muestra pues los resultados de una de las vueltas a 20 Km/h, con y sin aplicar seguridad.

Como puede apreciarse, en zonas donde sólo estaba presente la cobertura 3G, el máximo ancho de banda alcanzado llegaba a 1-2 Mbps. Cuando la señal de 802.11p también estaba disponible, los valores máximos de ancho de banda subieron a los 4-5



(a) Ancho de banda máximo TCP (b) Ancho de banda máximo TCP con seguridad

Figura 7.5: Ancho de banda máximo TCP con y sin seguridad a 20 Km/h

Mbps. Estos valores son muy similares tanto con como sin seguridad. Atendiendo a los resultados mostrados en la Tabla 7.4, de nuevo, no se aprecia un impacto notable en el uso y aplicación de IPsec para proteger las comunicaciones. Sin embargo, existe una pequeña reducción del rendimiento, debido al mayor tiempo de procesamiento que es necesario. También esta merma del rendimiento tiene su origen en la inclusión de una nueva cabecera (ESP o AH) que se añade a las ya existentes (IPv6,...). En cada paquete que suele ser de un tamaño constante no mayor al MTU del enlace (1500 bytes), al usarse un poco más de espacio para las cabeceras, el espacio restante para datos se reduce y, por tanto, se necesitan más paquetes para transmitir la misma información.

Tabla 7.4: Valores medios de ancho de banda máximo TCP y sus intervalos de confianza al 95 %

Velocidad (Km/h)	**Max. TCP BW sin IPsec (Mbits/sec)**	**Max. TCP BW con IPsec (Mbits/sec)**
10	1.085 ±0.066 (1649 muestras)	0.898 ±0.087 (1976 muestras)
20	0.953 ±0.081 (1254 muestras)	0.895 ±0.065 (1076 muestras)
30	0.723 ±0.063 (653 muestras)	0.816 ±0.074 (1106 muestras)
40	0.820 ±0.071 (652 muestras)	0.820 ±0.073 (764 muestras)

7.3.3. Estudio de la Fiabilidad del Envío de Paquetes

Para entender y observar el comportamiento de la red en cuanto a fiabilidad de los envíos, hemos realizado pruebas usando tráfico UDP con el que podemos medir el porcentaje de paquetes que consiguen llegar a su destino respecto al número total de paquetes enviados, es decir, el *Packet Delivery Ratio* (PDR). Al igual que en pruebas anteriores, se muestra de forma gráfica en la Figura 7.6 los datos obtenidos de una de las vueltas a una velocidad determinada, en este caso a 20 Km/h. Así puede observarse a simple vista que los valores de PDR son más bajos cuando la tasa de transferencia sube. Esto se produce por la congestión de enlace inalámbrico que es afectado por las condiciones del entorno y el movimiento, especialmente notable en la zona donde se produce el primer traspaso, donde el vehículo apenas tiene cobertura 802.11p. Es por tanto la causa de numerosos paquetes perdidos.

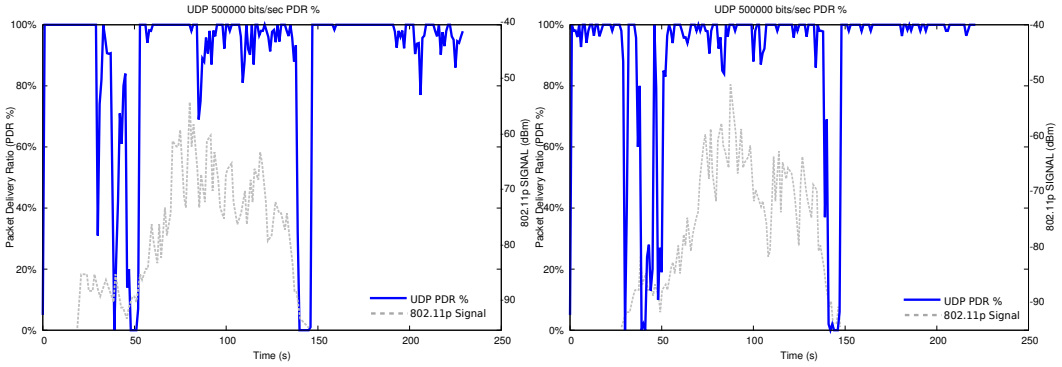
Atendiendo a los valores medios mostrados en la Tabla 7.5, no hemos observado una influencia significativa de la velocidad en el rendimiento de la red. También debe tenerse en cuenta que son velocidades relativamente bajas de entre 10 y 40 Km/h, lo que no nos permite tener una conclusión clara aquí. El vehículo circula por un entorno dentro del Campus Universitario de la Universidad de Murcia donde está en su mayoría limitado a 30-40 Km/h.

Estos resultados también pueden verse desde el punto de vista del comportamiento de la tasa de transferencia de los paquetes UDP y ver cómo evoluciona cuando vamos incrementando dicha tasa. Este comportamiento es mostrado en la Figura 7.7, donde se puede observar el PDR obtenido a diferentes tasas de transferencia, aplicando y sin aplicar seguridad IPsec. Podemos observar en línea gris punteada el nivel de señal 802.11p que se registraba en cada momento. Sabiendo esto, podemos apreciar cómo la comunicación es duramente afectada cuando el vehículo circula por zonas de baja cobertura 802.11p. En el primer traspaso, de 3G a 802.11p, el nivel de señal permanece demasiado tiempo en un nivel muy bajo, haciendo la comunicación casi imposible e impidiendo un traspaso limpio. Es de suponer que en estos momentos la elección de 3G y 802.11p no es clara ni duradera, lo que provoca numerosos traspasos intermitentes que degradan la calidad del canal. En el caso del segundo traspaso, de 802.11p a 3G, la señal 802.11p desaparece bruscamente, pero el MR continúa intentando comunicarse a través de 802.11p hasta que la asociación de movilidad CoA-HoA caduca. El tiempo de renovación de estas asociaciones está fijado en este caso a 12 segundos, como puede apreciarse en la Tabla 7.2. Esto significa que en el peor de los casos la comunicación será imposible debido a que no se realiza el traspaso a 3G hasta pasado ese tiempo. Pero este y otros detalles relativos a los traspasos de red serán discutidos más adelante cuando se analicen los resultados obtenidos concretamente en los traspasos.

En cuanto a la seguridad, se puede apreciar en la Tabla 7.5 que los valores de PDR obtenidos son, por lo general, ligeramente peores aplicando seguridad, pero no lo suficiente para decir que IPsec implica una sobrecarga significativa en el rendimiento de las comunicaciones. En las Figuras 7.6 y 7.7 las condiciones del canal afectan de una forma similar a los resultados tanto con como sin seguridad.

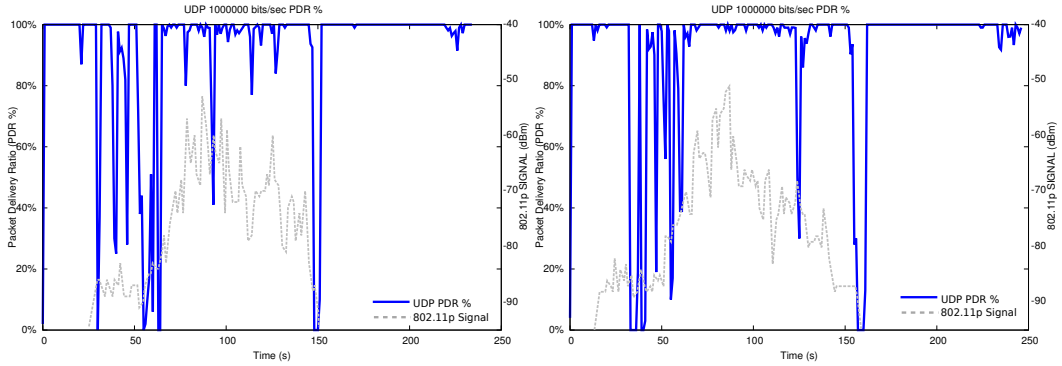
Tabla 7.5: Valores medios del PDR obtenidos en las pruebas con UDP y sus intervalos de confianza al 95 %

Velocidad (Km/h)	**500 Kbps PDR medio (%)**	**1000 Kbps PDR medio (%)**	**2000 Kbps PDR medio (%)**
Sin seguridad			
10	88.373 ±1.21 (1897 muestras)	88.542 ±1.30 (1735 muestras)	78.978 ±2.00 (1435 muestras)
20	87.997 ±1.71 (1353 muestras)	91.553 ±1.48 (1392 muestras)	86.109 ±1.95 (1147 muestras)
30	90.992 ±1.83 (782 muestras)	91.360 ±1.93 (823 muestras)	86.701 ±2.37 (604 muestras)
40	88.155 ±1.97 (687 muestras)	91.525 ±1.92 (692 muestras)	86.886 ±2.36 (693 muestras)
Con seguridad			
10	83.265 ±1.24 (1950 muestras)	78.261 ±1.27 (1992 muestras)	76.826 ±1.71 (1756 muestras)
20	80.189 ±1.67 (1175 muestras)	85.393 ±1.22 (1211 muestras)	78.185 ±1.95 (964 muestras)
30	88.364 ±2.03 (1121 muestras)	88.734 ±2.09 (1120 muestras)	82.326 ±2.36 (1101 muestras)
40	87.662 ±2.14 (695 muestras)	87.398 ±2.22 (712 muestras)	84.559 ±2.52 (688 muestras)



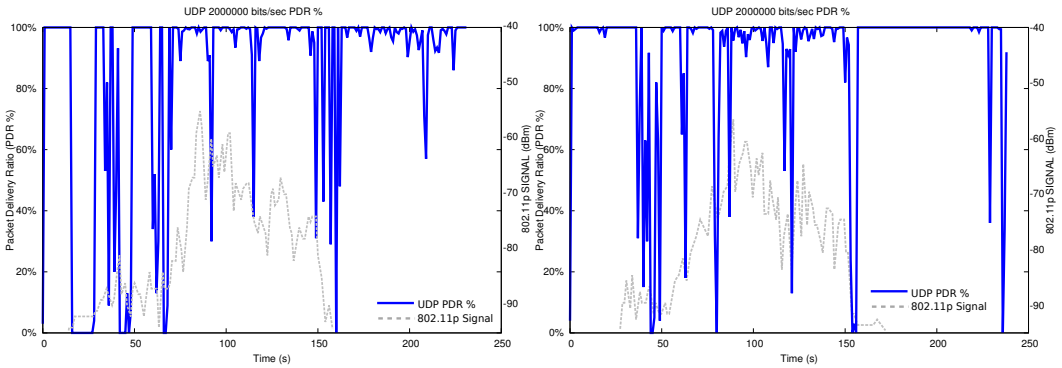
(a) 500 Kbps

(b) 500 Kbps con IPsec



(c) 1000 Kbps

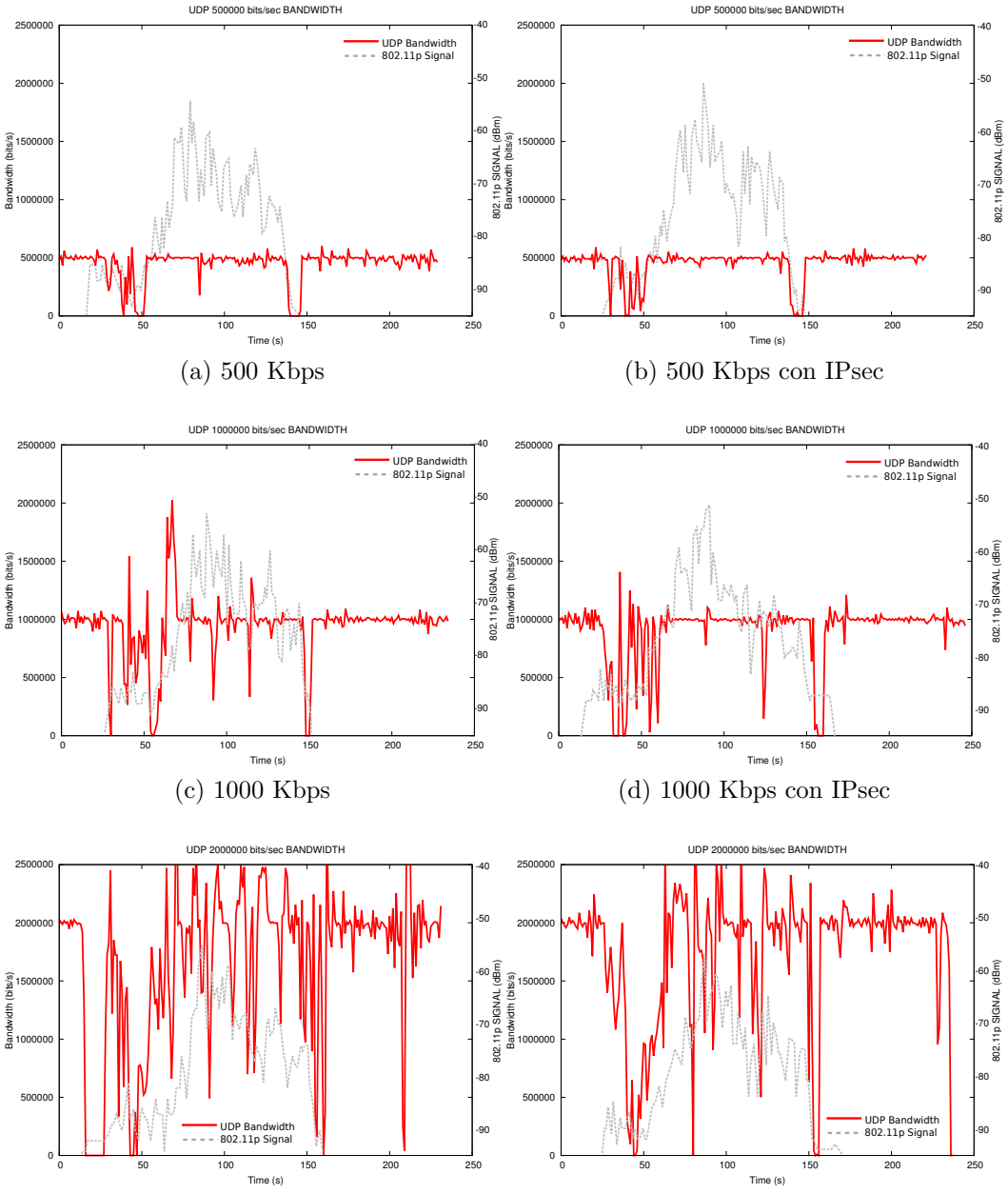
(d) 1000 Kbps con IPsec



(e) 2000 Kbps

(f) 2000 Kbps con IPsec

Figura 7.6: PDR (%) a diferentes tasas de transferencia de tráfico UDP a 20 Km/h



(a) 500 Kbps (b) 500 Kbps con IPsec

(c) 1000 Kbps (d) 1000 Kbps con IPsec

(e) 2000 Kbps (f) 2000 Kbps con IPsec

Figura 7.7: Comparación de anchos de banda de tráfico UDP a diferentes tasas de transferencia a 20 Km/h

7.3.4. Estudio del Impacto de la Aplicación de Seguridad

Aunque en cada apartado ya hemos analizado el impacto que se aprecia a la hora de aplicar seguridad a las comunicaciones mediante IPsec, hemos querido resumir aquí en un único apartado dicho análisis, teniendo ya una visión más amplia de las pruebas realizadas.

Los resultados en general muestran un ligero impacto de la seguridad cuando la carga de la red es importante. Sin embargo para pequeñas cantidades de datos es despreciable. Esto es debido a que cuando el estrés de la red es mayor (mayores tasas de transferencia) el procesamiento necesario de los paquetes en los extremos es mayor. El cifrado de los datos y el incremento del tamaño de la cabecera (por tanto menor espacio para datos por cada paquete), son dos de las causas de esta pequeña degradación del rendimiento.

En el caso de las pruebas de retardo, las pocas diferencias que encontramos entre aplicar o no IPsec son atípicas, ya que se espera que con esa tasa de trasferencia tan baja el retardo de los paquetes sea muy similar. Sin embargo, en el caso de las pruebas donde medimos el valor de PDR, la seguridad reduce el PDR en un 4.7 %, y en las pruebas de máximo ancho de banda, el rendimiento baja un 5.3 %. Además, se observa un incremento del 15.5 % en el tiempo de traspaso medio cuando aplicamos IPsec, que es atribuido de nuevo a dicha sobrecarga.

Con estos resultados se puede decir que aplicar seguridad mediante IPsec sobre IPv6 en un escenario vehicular donde usamos NEMO como protocolo de movilidad no supone un impacto destacable en el rendimiento de las comunicaciones, incluso cuando el canal es sobrecargado. Por lo tanto, merece la pena sufrir este pequeño impacto para beneficiarnos de la protección que nos aporta IPsec en cuanto a confidencialidad, integridad y autenticidad de los datos transmitidos.

7.3.5. Estudio del Comportamiento de la Red en los Traspasos

Uno de los principales objetivos de las pruebas realizadas es estudiar el comportamiento de las comunicaciones durante los traspasos que se producen durante el recorrido. En cada una de las vueltas se dan dos de ellos, ambos verticales (entre tecnologías diferentes) e inter-dominio (cambio de direccionamiento a nivel de red). Este análisis ha ayudado a mejorar los mecanismos de traspaso y formular unas conclusiones que han sido consideradas en este mismo trabajo y otras que lo serán en trabajos futuros. También aquí estudiaremos cual es la incidencia de la aplicación de la seguridad mediante IPsec en el proceso de traspaso. Para esto es importante tener en cuenta que las negociaciones IKEv2 para inicializar las IPsec SAs necesarias se realizan antes de empezar la prueba, y, por tanto, quedan fuera de este estudio. Se deja para un trabajo futuro el estudio de esta circunstancia que sólo afectaría al traspaso de 3G a 802.11p.

7.3.5.1. Traspaso de 3G a 802.11p

En cada vuelta, como se ha dicho antes, nos encontramos dos traspasos. El primero es provocado cuando el MR, que está usando la conectividad 3G, descubre la presencia de un AR de una estación de carretera ITS. En la tabla 7.6 analizamos el tiempo invertido en este traspaso sepárandolo en distintas fases:

1. RA-DAD: Desde la recepción de un mensaje *Router Advertisement* (RA) hasta el comienzo del proceso de detección de dirección duplicada (DAD). En esta fase se genera la nueva CoA en el Router Móvil (MR).

2. DAD-BU: Desde el comienzo del DAD hasta el envío del mensaje de movilidad *Binding Update* (BU) hacia el HA.

3. BU-BA: Desde el envío del BU hasta la recepción del mensaje de respuesta *Binding Acknoledgement* (BA), donde ya se da por concluido el traspaso.

Para obtener estos valores se han utilizado las capturas de tráfico realizadas en todas las vueltas efectuadas, que han sido procesadas por medio de *scripts* para agilizar los cálculos. Estos valores son resumidos en las tablas mostrando los valores medios teniendo en cuenta todas las pruebas realizadas para los protocolos ICMPv6, UDP y TCP, es decir, 25 vueltas para cada velocidad, 100 vueltas en total sin seguridad y otras 100 con seguridad. En estos datos no se han tenido en cuenta las 25 vueltas efectuadas para estudiar el soporte de 802.21.

Tabla 7.6: Valores medios del tiempo empleado en los traspasos de 3G a 802.11p con un intervalo de confianza al 95 %

Velocidad (Km/h)	**RA - DAD (seg)**	**DAD - BU (seg)**	**BU - BA (seg)**	**Total (seg)**
sin seguridad				
10 (26 muestras)	0.4224 ±0.043	3.6462 ±0.663	1.9738 ±0.474	6.0425 ±0.633
20 (27 muestras)	0.5330 ±0.055	3.6554 ±0.383	0.8172 ±0.282	5.0057 ±0.397
30 (21 muestras)	0.5239 ±0.051	2.6731 ±0.333	0.4461 ±0.182	3.6432 ±0.326
40 (20 muestras)	0.4196 ±0.059	2.2944 ±0.547	0.6524 ±0.282	3.3665 ±0.567
con seguridad				
10 (17 muestras)	0.4016 ±0.066	5.6353 ±0.798	1.4739 ±0.616	7.5109 ±0.802
20 (25 muestras)	0.3734 ±0.051	3.2836 ±0.351	1.8803 ±0.502	5.5373 ±0.568
30 (33 muestras)	0.4183 ±0.046	2.5722 ±0.277	0.9737 ±0.305	3.9643 ±0.381
40 (24 muestras)	0.5326 ±0.058	2.0787 ±0.281	1.2410 ±0.470	3.8525 ±0.560

A velocidades más altas, el tiempo empleado en realizar el traspaso es claramente menor, debido a que el tiempo que el MR permanece en zonas donde la señal 802.11p es baja es mucho menor, y por tanto se pierden menos paquetes. Comparando estos resultados con los valores obtenidos aplicando seguridad, también mostrados en la Tabla 7.6, se puede apreciar que estos últimos son ligeramente peores, como puede verse gráficamente en la Figura 7.8. Esto es debido a que el tiempo de procesado de los paquetes es mayor por la encriptación que se le aplica y la pequeña sobrecarga fruto de añadir una cabecera más a cada paquete.

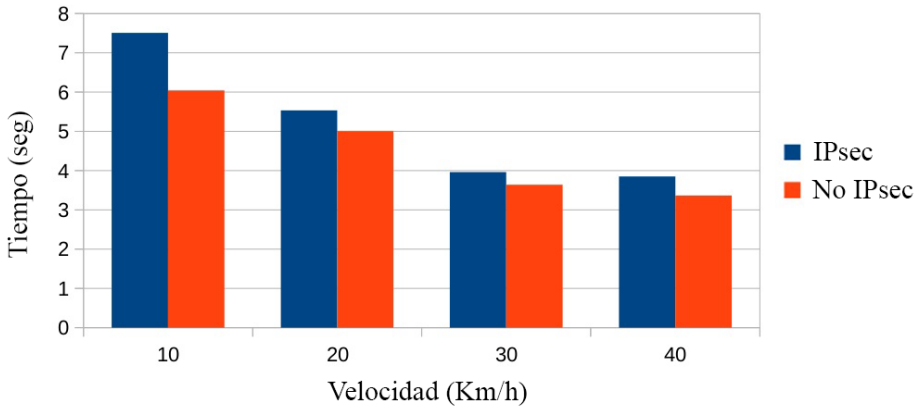


Figura 7.8: Valores medios de los traspasos de 3G a 802.11p, con y sin seguridad

También hay que destacar que durante este proceso de traspaso, la interfaz 3G sigue estando disponible y se sigue usando con normalidad. Sólo cuando el mensaje BA de confirmación llega al MR es cuando la movilidad puede utilizar esta nueva interfaz 802.11p. Esto implica que no debería haber demasiadas pérdidas de paquetes en este traspaso. Sin embargo vemos que se producen bastantes pérdidas, simplemente por haber realizado el traspaso demasiado pronto, cuando todavía la fuerza de la señal 802.11p es demasiado débil.

7.3.5.2. Traspaso de 802.11p a 3G

El segundo traspaso es provocado cuando la señal de 802.11p es tan baja que la transmisión de paquetes es casi imposible. A pesar de ello, se sigue intentando trasmitir por 802.11p hasta que el tiempo de vida del registro de su CoA asociada en el HA caduca. Será entonces cuando dejará de enviarse tráfico por esta interfaz 802.11p y pasará de nuevo a enviarse a través de 3G. Esto es así porque como la cobertura 3G ha estado presente siempre y disponemos ya de la CoA y la asociación en el HA para dicha CoA, podemos usarla inmediatamente. Esta ventaja se la debemos a la extensión MCoA en NEMO. El tiempo de vida del registro de una CoA en el HA aparece en la tabla 7.2, que son 12 segundos en este caso. Por tanto, desde que perdemos conectividad

hasta que nos demos cuenta y procedamos al cambio puede pasar un tiempo aleatorio no superior a ese tiempo y de media 6 segundos. Dicho tiempo podría reducirse hasta prácticamente cero si fuéramos capaces de anticiparnos al cambio de interfaz antes de que la señal fuera tan débil que no permitiese una comunicación suficientemente confiable. Para ello podría establecerse un mínimo nivel de fuerza de señal que asegurara una calidad aceptable y cambiar a otra interfaz lo antes posible en el caso de que la señal bajara de dicho umbral. Así evitaríamos perder tantos paquetes. Para este propósito el servicio NEMO podría cancelar inmediatamente una asociación CoA-HoA en el HA simplemente mandando desde el MR un mensaje BU con la CoA que se quiera cancelar y un valor cero de tiempo de vida, que vendría establecida en uno de sus campos. Sólo faltaría un mecanismo que provocara el traspaso cuando dicho umbral fuera alcanzado. Profundizamos en este punto en el siguiente apartado.

7.3.6. Estudio de la Aplicación de 802.21 al Proceso de Traspaso

Como hemos visto en los apartados anteriores, en ambos casos se adolece de no hacer el traspaso en el momento más adecuado y que la movilidad hasta ahora no ha usado mecanismos que le permitieran reaccionar más rápido ante las nuevas condiciones.

Es predecible suponer, viendo los resultados obtenidos, que el primer traspaso puede mejorarse, y la comunicación en general, retrasando el traspaso a un momento en el que la señal de la tecnología destino, en este caso 802.11p, sea lo suficientemente potente para evitar pérdida de paquetes. Esto haría que todo el proceso de traspaso fuera más rápido pues no se producirían tantas pérdidas y sus respectivos reenvíos.

En el segundo traspaso cuando el vehículo sale de la zona de cobertura 802.11p, en casi todas las figuras donde hemos mostrado el comportamiento de los valores obtenidos, puede apreciarse un espacio de tiempo de tamaño variable en donde la comunicación se interrumpe por completo. Esto es producido por este segundo traspaso, que como hemos visto antes puede durar entre 0 y 12 segundos, que es el tiempo de vida de la CoA en el HA. Está claro que se trata de demasiado tiempo y no puede considerarse como admisible. La movilidad debería reaccionar antes de que la señal 802.11p sea tan débil que fuera imposible la comunicación y no permitiera ni siquiera cancelar la CoA asociada en el HA mediante un mensaje BU con tiempo de vida con valor cero.

El servicio de movilidad por sí mismo no tiene mecanismos que le permitan tomar estas acciones en el momento más adecuado. Para ello hace falta el apoyo de un protocolo, preferiblemente estándar, que nos permitiera asistir los traspasos. Nos llamó la atención la ya existencia de un estándar desde 2004 que nació precisamente para este fin. Se trata del protocolo 802.21 [116], que aporta los mecanismos necesarios para asistir dichos traspasos y realizarlos cuando las condiciones sean las más idóneas. Sus características ya fueron analizadas en el Capítulo 6 dedicado a los traspasos. Nos centraremos aquí en analizar la incorporación de dicho protocolo a nuestra propuesta. Para obtener resultados se realizan una serie de pruebas extraordinarias idénticas a

las anteriores con la salvedad de que sí se ha implementado parte del estándar 802.21. Veamos si esta incorporación ha dado buenos resultados, centrándonos en cada uno de los traspasos.

7.3.6.1. Traspaso de 3G a 802.11p

Gracias al soporte ofrecido por 802.21 y como se ha explicado con profundidad en el Capítulo 6 dedicado a los traspasos, se determinan dos umbrales de niveles de señal 802.11p diferentes que provocarán los traspasos. Estos umbrales no deberían tomar valores muy cercanos, pues podría producirse un cambio intermitente entre una interfaz y otra demasiado rápida haciendo la conexión muy inestable (efecto "ping-pong"). Aún así, estas heurísticas pueden ser mejoradas, dejándose para futuros trabajos. Para el caso de pruebas dentro del laboratorio se han determinado los umbrales de -40 dbm y -60 dbm. En el caso de exteriores, donde la señal se recibe más débil, se han determinado los umbrales de -65 dbm y -80 dbm. En ambos casos, el umbral de mayor señal será el que provoque este traspaso de 3G a 802.11p. De esta forma, el protocolo 802.21 retrasa el traspaso hasta que el nivel de señal sea lo suficientemente bueno. De igual forma, el umbral de menor señal será el que provoque el traspaso de 802.11p a 3G, en el que profundizaremos más adelante.

Igual que se hizo anteriormente, para las pruebas se han dividido los tiempos en tres fases: Desde que se recibe el RA hasta que se inicia la detección de dirección duplicada (DAD); desde este proceso de detección hasta que se envía el BU; y desde este envío hasta la recepción del BA.

Tabla 7.7: Valores medios del tiempo empleado en los traspasos de 3G a 802.11p a 30 Km/h con y sin 802.21, con un valor de confianza al 95 %

Velocidad (Km/h)	**RA - DAD (seg)**	**DAD - BU (seg)**	**BU - BA (seg)**	**Total (seg)**
sin 802.21				
30 (24 muestras)	0.5239 ±0.049	2.6731 ±0.226	0.4461 ±0.242	3.6432 ±0.539
con 802.21				
30 (24 muestras)	0.23921 ±0.043	1.3683 ±0.197	0.10716 ±0.173	1.7147 ±0.324

Hemos realizado una serie de vueltas extra, en concreto 25, todas a 30 Km/h y sin seguridad, para probar el comportamiento de los traspasos usando la asistencia de 802.21. En la Tabla 7.7, y gráficamente en la Figura 7.9, se muestran los valores medios obtenidos, con y sin aplicar 802.21, para poder comparar resultados. Como se puede apreciar, hay una sustancial mejora reduciendo al 53 % el tiempo total del traspaso en relación al tiempo invertido sin usar 802.21, ya que las negociaciones se producen con niveles de señal aceptables, disminuyendo así el riesgo de que los paquetes se pierdan.

Además, como podrá apreciarse también en la Figura 7.10, no se producen tantas pérdidas localizadas en el momento en el que empieza a haber señal 802.11p como ocurría antes sin aplicar 802.21.

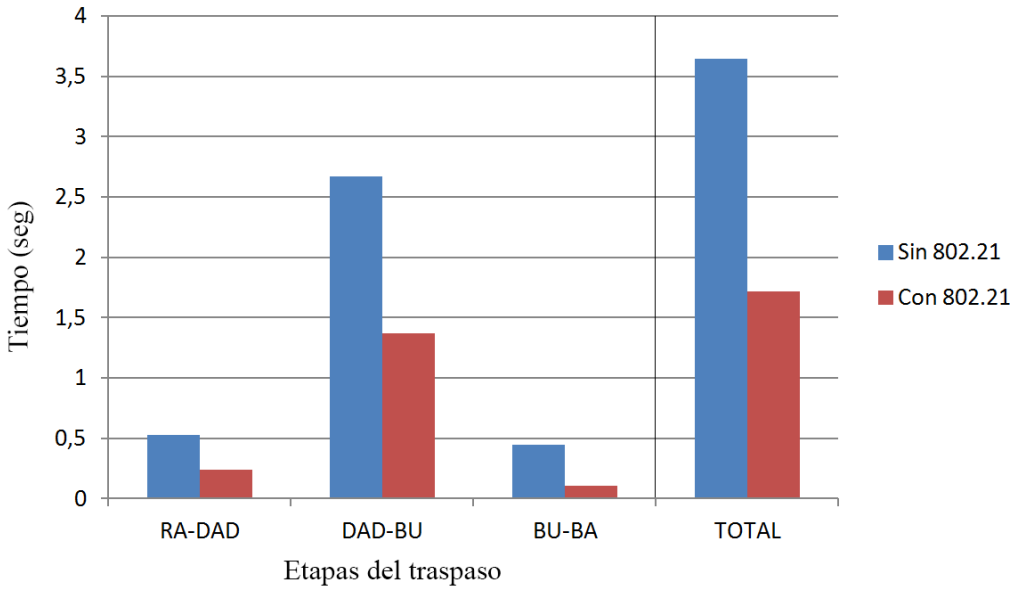


Figura 7.9: Representación gráfica de los valores medios del tiempo empleado en los traspasos de 3G a 802.11p a 30 Km/h con y sin 802.21

También observamos que el tiempo invertido por el proceso de detección de direcciones duplicadas de IPv6 (DAD) es del 73 % del total en el caso sin 802.21, y casi el 80 % en el caso de aplicar 802.21. Eliminar dicha fase de detección supondría reducir el tiempo del traspaso a 0.97 segundos (sin aplicar 802.21) y 0.347 segundos (aplicando 802.21), lo que supone una sustancial mejora a tener en cuenta para implementaciones futuras. No aplicando DAD en ninguno de los casos, la mejora que implica la presencia de 802.21 reduce el tiempo a un 35 % en relación a lo obtenido sin 802.21. Se deja como trabajo futuro realizar un estudio de las implicaciones que supondrían suprimir este mecanismos (DAD), para ver si merece o no la pena conservarlo, ya que la probabilidad de que dos direcciones IP coincidan en un entorno IPv6 se presupone prácticamente nula.

Por último, en apartados anteriores se estableció la causa de la gran influencia de la velocidad de los vehículos en los resultados. Se determinó que cuanto más tiempo permanecen los vehículos en zonas donde la cobertura 802.11p es insuficiente, mayores son las pérdidas. La mejora general observada en los resultados aplicando 802.21, y el hecho de que precisamente esta tecnología evita cambiar de interfaz en momentos donde la cobertura 802.11p no es buena, nos permite concluir que la velocidad del vehículo no debería de influir en los resultados, al menos a esas velocidades bajas (10-40 Km/h) donde no se compromete la eficacia de la tecnología usada. Se deja como trabajo futuro

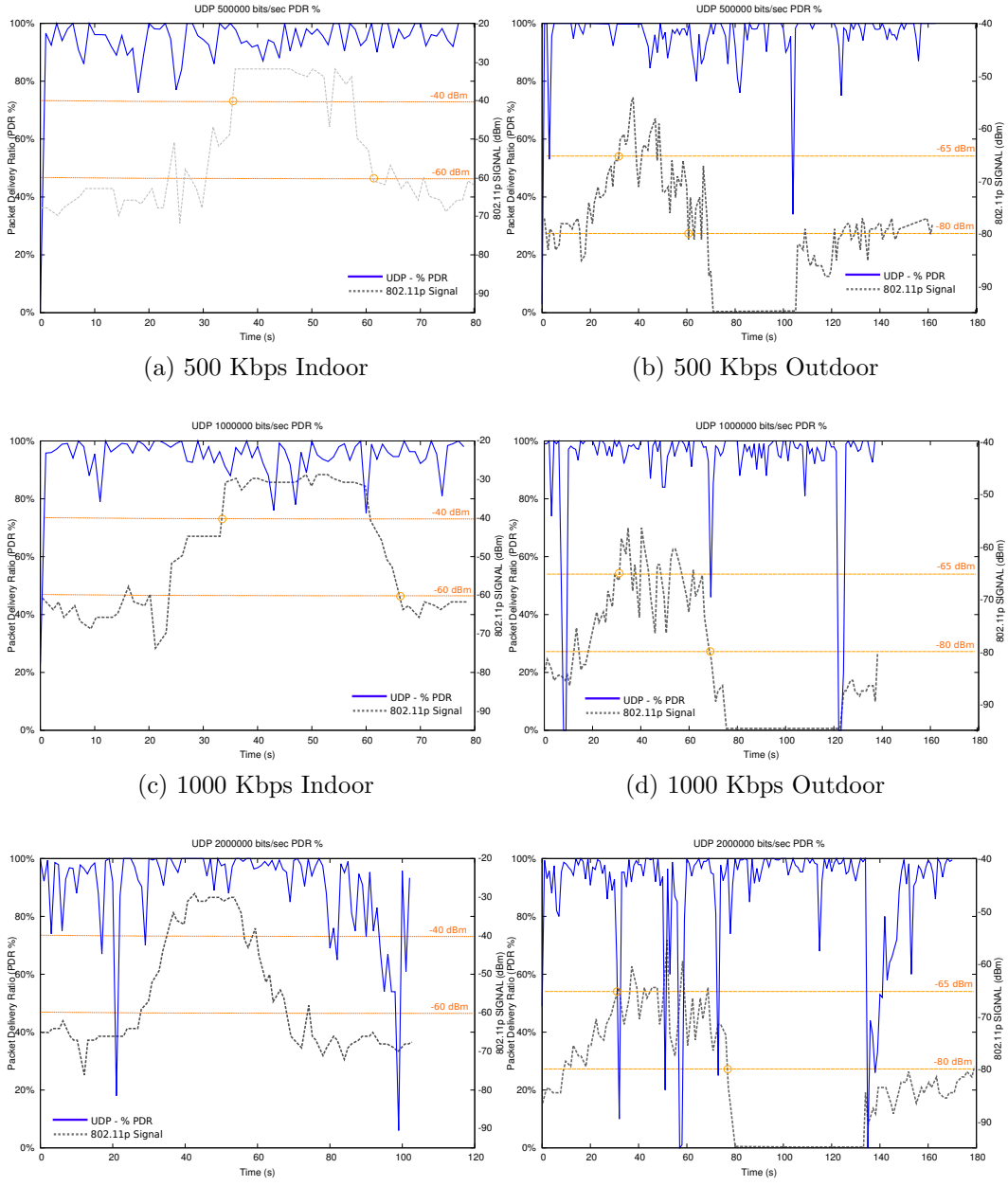
realizar una comprobación empírica usando varias velocidades en las pruebas aplicando 802.21.

7.3.6.2. Traspaso de 802.11p a 3G

Incorporado el soporte de 802.21 a nuestra arquitectura, el traspaso producido cuando cambiamos de tecnología 802.11p a 3G puede mejorar enormemente adelantándolo cuando todavía el nivel de señal 802.11p es lo suficientemente bueno como para realizar una negociación fluida. El mayor problema que encontrábamos aquí, tratado ya en profundidad en la Sección 6.4.1, era que el servicio de movilidad esperaba a que caducara el tiempo de vida de la CoA asociada a la interfaz 802.11p a pesar de que la señal 802.11p hubiera desaparecido y la comunicación fuera imposible. Además tampoco permitía cancelar dicha CoA pues el BU requerido con tiempo de vida con valor cero no llegaba al HA. Se trata, como hemos adelantado antes, de anticipar el traspaso cuando el nivel de señal 802.11p sea suficiente. Para ello se define otro umbral de señal 802.11p (el menor de los dos) a partir del cual el servicio de movilidad dejará de usar dicha tecnología siempre y cuando haya otra tecnología disponible, como en este caso la tecnología 3G. Siendo así, se producirá el traspaso a 3G siempre y cuando el valor de la fuerza de señal 802.11p baje de dicho umbral.

De las pruebas realizadas, en la Figura 7.10 se muestran los valores porcentuales de PDR para tráficos UDP de velocidades de transferencia constantes de 500, 1000 y 2000 Kbps. Además se muestra en esta ocasión el comportamiento en dos entornos diferentes: en el laboratorio, donde para provocar la caída de la señal se desenroscaban las antenas del *Router* Móvil (MR); y en el circuito con un vehículo, donde el nivel de señal 802.11p va variando dependiendo de la posición del vehículo como hemos visto en la Figura7.3. Se muestran además los umbrales de señal 802.11p utilizados, que como hemos visto antes tienen valores diferentes dependiendo del entorno.

En este tipo de pruebas con las que hemos obtenido valores de PDR es donde este traspaso se hacía más evidente en las pruebas anteriores como hemos visto en la Figura 7.6, donde se apreciaba un espacio de tiempo en el que la comunicación era nula. Este espacio, como vimos, era de duración aleatoria y podía ser de entre 0 y 12 segundos (tiempo de vida de las direcciones CoA). En estas pruebas, gracias al soporte 802.21, se anticipa el traspaso y además forzamos que la movilidad cancele la CoA mandando desde el MR un BU con tiempo de vida cero para la CoA asociada, consiguiendo así un cambio inmediato a 3G, ya que es una interfaz que siempre ha estado presente, pero de menor prioridad. Observando las gráficas de la Figura 7.10 podemos darnos cuenta de que cuando la señal 802.11p empieza a bajar y alcanza el umbral inferior no se produce ningún espacio de tiempo en donde la comunicación se interrumpa. Podemos considerar entonces que el traspaso es inmediato produciendo un número mínimo de paquetes perdidos. Para ver la duración aproximada de este traspaso hemos visto el tiempo que transcurre entre la recepción del último paquete UDP a través de 802.11p y la recepción del primero a través de 3G. Realizadas 15 vueltas a 30 Km/h, obtenemos un valor medio de 0.247 $\pm$0.054 segundos (intervalo de confianza al 95 %, 15 muestras), bastante mejor si lo comparamos con los 6 segundos de media que teníamos antes de



(a) 500 Kbps Indoor (b) 500 Kbps Outdoor

(c) 1000 Kbps Indoor (d) 1000 Kbps Outdoor

(e) 2000 Kbps Indoor (f) 2000 Kbps Outdoor

Figura 7.10: PDR (%) a diferentes tasas de transferencia de tráfico UDP en laboratorio y en circuito con asistencia en el traspaso mediante 802.21

aplicar 802.21, que es aproximadamente 24 veces más rápido.

7.3.7. Análisis Final de los Resultados

Efectuadas todas las pruebas, queremos resumir aquí las evidencias que hemos obtenido a través de ellas, que nos facilitan la formulación de conclusiones, como veremos en el próximo capítulo.

A través de las pruebas hemos podido analizar el rendimiento de las tecnologías utilizadas, apreciando que tanto 3G como 802.11p tienen sus pros y sus contras. En la Tabla 7.8 las hemos enumerado. Podemos destacar una diferencia importante entre estas tecnologías, que tiene que ver con las frecuencias de las señales utilizadas. En el caso de 3G, puede transmitir en las bandas de 900 Mhz y 2.1 Ghz. Sin embargo, 802.11p se restringe a la banda de los 5.9 Ghz. Esta diferencia influye notablemente en las comunicaciones, pues las señales a distintas frecuencias se comportan de forma diferente. En concreto, cuando subimos en frecuencia, las señales se vuelven más direccionales y menos penetrantes. A frecuencias tan altas como es el caso de 802.11p (5.9 Ghz), se requiere línea de visión con la antena. Cualquier obstáculo afecta en gran medida a la transmisión, lo que limita mucho el rango de cobertura. Con 3G, con frecuencias mucho más bajas, las señales pueden atravesar grandes obstáculos como edificios y pequeñas colinas. Esto hace que su rango de cobertura sea mayor. En contrapartida, 802.11p no requiere de ningún tipo de negociación a nivel de enlace, con lo que las comunicaciones son más rápidas y fluidas, especialmente pensado para entornos vehiculares donde los enlaces se crean y se destruyen constantemente.

Tabla 7.8: Pros y contras de las tecnologías 3G y 802.11p

-	Pros	Cons
3G	• Amplio rango de cobertura de varios kilómetros. • Señal capaz de penetrar edificios. • Valor de PDR muy alto. • Cobertura siempre disponible en la prueba.	• Requiere de una negociación de establecimiento del canal. • Tiempo de retardo alto de más de entre 170 y 300 milisegundos. • Valor de PDR sujeto a la congestión de la red que no controlamos que cambia según la hora del día. • Ancho de banda máximo de 1 a 2 Mbps.
802.11p	• No requiere de negociación para el establecimiento del canal. • Tiempo de retardo bajo de entre 11 y 15 milisegundos. • Ancho de banda máximo de 4 a 5 Mbps. • Potencia de la señal siempre inferior o igual a 1 Watio.	• Rango de cobertura reducido menor de 1 Km. • Señal direccional incapaz de salvar obstáculos. • Valor de PDR pobre, incluso en zonas de alta cobertura. • Cobertura no siempre disponible en la prueba.

En los resultados se ha comprobado que la extensión *Multiple Care-of Address*

Registration (MCoA) de NEMO ha sido un gran aporte a la hora de mejorar los traspasos, ya que nos permite disponer simultáneamente de varias interfaces de red operativas y, por tanto, traspasos con una interrupción del servicio nula. Además, disponer de esta tecnología hace que no tenga tanta influencia el tiempo de establecimiento de un nuevo canal, pues se puede simultanear este proceso con la utilización del canal que ya teníamos. Aunque hemos visto que se ha llegado a reducir el tiempo de la negociación a 1.71 segundos en el caso del traspaso de 3G a 802.11p, a éste habría que sumarle el tiempo de negociación IKEv2 en el caso de aplicar seguridad. Este tiempo de negociación depende de los métodos de autenticación empleados, como se ha visto en el Capítulo 6 dedicado a los traspasos. Sin embargo, este tiempo no influye en el caso de tener ya una interfaz activa, como podría ser 3G en nuestro caso. De ahí la importancia de esta extensión de NEMO. Independientemente de la aplicación de cualquier mejora, como 802.21 o MCoA, se ha visto que la supresión del mecanismo de detección de direcciones duplicadas supondría una gran mejora en los tiempos de negociación, reduciéndolos a 0.347 segundos. Sin embargo, dejamos como trabajo futuro averiguar las implicaciones de la supresión de dicho servicio que, apriori, parece ser factible dado al gran rango de direcciines IP del que dispone IPv6 y la dificultad de que dos direcciones IP colisionen.

Hemos constatado que el servicio de movilidad por sí solo no era capaz de efectuar un traspaso en el momento adecuado, lo que producía desconexiones y un número inadmisible de paquetes perdidos. La incorporación de 802.21 ha sido decisiva a la hora de mejorar este punto, ya que se trata de una entidad capaz de tomar decisiones dependiendo de las condiciones del entorno. En este primer paso hemos tenido en cuenta la potencia de la señal 802.11p a la hora de tomar las decisiones de cuándo realizar los traspasos. Sin embargo, cuanta más información utilicemos, mejores decisiones se pueden tomar. La incorporación de heurísticas más complejas que nos ayuden a predecir mejor el momento para provocar los traspasos se deja para trabajos futuros.

7.4. Conclusiones

Los resultados mostrados a lo largo de este capítulo nos han permitido llegar a diversas conclusiones. Con respecto a las tecnologías de comunicación usadas, los tiempos de retardo de 3G son mayores y más variables que los retardos obtenidos con 802.11p, mucho más pequeños y constantes. Sin embargo, es con 802.11p cuando perdemos la mayor cantidad de paquetes, en contraposición con 3G que resulta ser más efectiva y fiable. En cuanto a los servicios de movilidad y seguridad, hemos podido probar que la aplicación del servicio de seguridad supone una pequeña sobrecarga que apenas influye en el comportamiento de las comunicaciones. También hemos visto reducidos los tiempos de traspaso gracias a la incorporación de MCoA e IEEE 802.21 a nuestra propuesta.

Capítulo 8

Conclusiones y Trabajos Futuros

En este último capítulo resumimos los resultados que hemos ido obteniendo a lo largo de los diferentes capítulos, identificando a su vez las posibles vías futuras. Para ello seguimos el orden establecido por la propia Tesis, recapitulando los resultados de las propuestas presentadas, pero ahora desde un punto de vista global, con miras al futuro inmediato, para establecer las posibles líneas de investigación a las cuáles dirigir nuestros esfuerzos.

8.1. Conclusiones Generales

Dentro del mundo de las redes vehiculares, que están viviendo un momento de gran avance en los últimos años, los organismos dedicados a la definición y estandarización de dichas redes no quieren repetir errores del pasado no definiendo una arquitectura clara desde el principio, que tenga presentes temas tan importantes como la seguridad y la movilidad de sus nodos. El uso de este tipo de redes en C-ITS implica mucho mayor cuidado en el plano de la seguridad, pues en las carreteras hay vidas humanas en juego. Las organizaciones de estandarización ISO y ETSI no han tardado en detectar dicha necesidad, y por ello ya disponemos de la definición de dicha arquitectura en la que se ha basado nuestra propuesta de red vehicular, basada en IPv6, donde se han provisto servicios de movilidad, seguridad y asistencia al traspaso, entre otros. Aparte de establecer un orden y vías de comunicación normalizadas entre los distintos módulos software, una arquitectura de red estándar aporta numerosas ventajas de cara al futuro, ya que asegura una compatibilidad entre propuestas que sigan la misma arquitectura. Esto permite a las diferentes propuestas funcionar entre sí y, por tanto, no tener la necesidad de disponer de un dispositivo diferente por cada propuesta, ayudando a utilizar el menor número posible de éstos en el vehículo, lo que supone un ahorro de energía, y un aumento en seguridad y confort.

Una gran parte de los servicios que pueden incorporarse a una red vehicular pueden ser importados desde el mundo de Internet, protocolos y servicios ya maduros cuyos estándares se gestaron en el seno del IETF. Esto se acentúa con la incorporación de IP como protocolo de red en este tipo de redes, pues lleva consigo el poder usar tecnologías

asociadas, sobre todo a nivel de red y transporte, para la implementación de servicios básicos como los de auto-configuración, seguridad y movilidad, entre otros. Esto además facilita la integración de las redes vehiculares en el Internet del Futuro. Sin embargo, en cuanto a las tecnologías de comunicaciones inalámbricas siguen siendo IEEE, 3GPP y ETSI los que establecen los estándares a nivel físico y de enlace.

Nuestra iniciativa de construir un escenario vehicular de pruebas real, nos ha llevado a tener que implementar nosotros mismos parte de los protocolos involucrados en nuestra propuesta. Este es el caso de la implementación de IKEv2 [109], que nos ha aportado una experiencia valiosa que ha ayudado a perfeccionar la definición de su estándar. Como fruto de ello se ha presentado OpenIKEv2 [106] como implementación válida para entornos de pruebas, donde destaca su facilidad de uso y el hecho de estar diseñado como una librería fácil de incorporar por otros procesos que requieran las capacidades de IKEv2.

Integrar todos los servicios que aglutina nuestra propuesta no es fácil, como es el caso de la movilidad y la seguridad. Existen múltiples caminos para conseguir su integración. Después de estudiar cuál de ellos es el más ventajoso para los entornos C-ITS, se ha optado por la protección extremo a extremo del servicio de movilidad, desde el MR hasta el HA, mediante IPsec e IKEv2. También se ha detectado la necesidad de definir una estrategia de comunicación entre las implementaciones de NEMO e IKEv2 para conseguir una mejor cooperación entre ellas. Una de las causas es que las asociaciones de seguridad tienen una alta dependencia de las direcciones IP que nos permiten acceder a las distintas redes (CoAs en términos de movilidad), teniendo que acudir a soluciones que permitan conservar dichas asociaciones a pesar de los cambios de IP. *Migrate* [111] y *Mobike* [112] son dos acercamientos válidos para realizar estos cambios, aunque se valora la posibilidad de juntar ambos servicios en un mismo proceso, lo que resolvería el problema. Sin embargo, se ha optado por un acercamiento intermedio, donde el servicio de movilidad, implementado por Mip6d del proyecto UMIP [32], usa *Migrate*, pero se comunica con OpenIKEv2 mediante un esquema inter-proceso, que evita a IKEv2 tener que entrar en las interioridades del núcleo del sistema.

A lo largo de la Tesis también hemos constatado que el servicio de movilidad, por sí solo, tiene como punto débil el comportamiento de las comunicaciones durante los traspasos, produciéndose cortes y reduciendo en consecuencia la calidad percibida por el usuario. Se ha comprobado que la extensión MCoA de NEMO ha sido un gran aporte a la hora de mejorar dichos traspasos, ya que nos permite disponer simultáneamente de varias interfaces de red operativas y, por tanto, traspasos con una interrupción del servicio nula. Además, disponer de esta tecnología hace que no tenga tanta influencia el tiempo de establecimiento de un nuevo canal, pues se puede simultanear este proceso con la utilización del canal que ya teníamos. La incorporación de 802.21 también ha sido decisiva a la hora de mejorar este punto, ya que se trata de una entidad capaz de tomar decisiones dependiendo de las condiciones del entorno. En un primer paso hemos tenido en cuenta la potencia de la señal 802.11p a la hora de tomar las decisiones de cuándo realizar los traspasos. Sin embargo, cuanta más información utilicemos, mejores decisiones se pueden tomar.

Después de probar nuestra propuesta poniendo énfasis en la metodología empleada (mediante un entorno de pruebas bien definido, procedimiento, métricas, procesado de datos, etc.), hemos podido validar, de forma global, la propuesta de red vehicular presentada en la Tesis, así como analizar el rendimiento de las tecnologías utilizadas, apreciando que tanto 3G como 802.11p son bastante diferentes. En el caso de 3G, puede transmitir en las bandas de 900 Mhz y 2.1 Ghz. Sin embargo, 802.11p se restringe a la banda de los 5.9 Ghz. Esta diferencia influye notablemente en las comunicaciones, pues las señales a frecuencias altas son más direccionales y con menor grado de penetración. A frecuencias tan altas como es el caso de 802.11p (5.9 Ghz), se requiere línea de visión directa entre emisor y receptor. Cualquier obstáculo afecta en gran medida a la transmisión, lo que limita mucho el rango de cobertura. Con 3G, con frecuencias mucho más bajas, las señales pueden atravesar grandes obstáculos como edificios y pequeñas colinas. Esto hace que su rango de cobertura sea mayor. En contrapartida, 802.11p no requiere de ningún tipo de negociación a nivel de enlace, con lo que las comunicaciones son más rápidas y fluidas, especialmente pensado para entornos vehiculares donde los enlaces se crean y se destruyen constantemente.

Como resumen y haciendo referencia a los objetivos que nos habíamos marcado en esta Tesis, hemos conseguido incorporar IPv6 y gran parte de sus tecnologías asociadas a las redes vehiculares, aprovechando implementaciones maduras de Internet como son la seguridad y movilidad. Nuestra propuesta ha seguido la arquitectura de referencia establecida por ISO/ETSI, explotando la sinergia existente entre las redes vehiculares y los estándares del IETF. Como servicio fundamental en nuestra propuesta se implementa un nuevo enfoque para proteger el tráfico orientado a flujos de datos y no a mensajes individuales como hasta ahora era habitual en este tipo de redes, siendo IPsec e IKEv2 los máximos protagonistas en este punto. Otro de los servicios fundamentales es el de movilidad, que hemos complementado con otras tecnologías que ayudan de forma decisiva a eliminar los cortes que se producen en los traspasos, como el estándar IEEE 802.21 o la extensión MCoA. Seguridad y movilidad, servicios que hasta ahora habían funcionado por separado, no son capaces de hacerlo juntos de forma directa, teniendo que realizar modificaciones que afortunadamente ya están contempladas por los estándares, pero que a la hora de llevarlo a cabo no evitan que surjan problemas inesperados, que hemos conseguido resolver usando para ello esquemas de comunicación inter-proceso. Todas las pruebas realizadas de nuestra propuesta han sido efectuadas en un escenario real, desplegado en torno al Campus de la Universidad de Murcia, que ha supuesto para este trabajo un salto cualitativo, ya que no abundan los trabajos que ofrezcan resultados en entornos reales, sino en simuladores. Además, esto nos ha permitido experimentar con tecnologías relevantes en entornos vehiculares ITS, como son 3G y IEEE 802.11p, centrando la atención en comprobar el rendimiento de la red y los procesos de traspaso. Los resultados obtenidos revelan que añadir seguridad al servicio de movilidad no presenta una sobrecarga significativa en la red. La velocidad del vehículo también ha resultado ser de importancia, ya que a velocidades más bajas, el vehículo permanece más tiempo en zonas de traspaso donde las señales son más débiles. Sin embargo, la aplicación de 802.21 también mitiga gran parte de este problema. La obtención de estos resultados no habría sido posible sin

un cuidado especial a la hora de definir una buena metodología en dichas pruebas, teniendo en cuenta métricas, procedimientos, etc. Todos estos objetivos conseguidos vienen a satisfacer otros de mayor importancia, como son salvar vidas y conservar el medio ambiente, aportando una contribución al estado del arte con un objetivo a medio plazo de aumentar la seguridad de los pasajeros y reducir el impacto medio-ambiental de los vehículos.

8.2. Trabajos Futuros

Al ser varias las líneas de investigación seguidas en este trabajo, no son pocas las vías futuras que hemos identificado a lo largo del desarrollo de esta Tesis Doctoral. Por ello vamos a enumerarlas en distintos apartados por temáticas.

8.2.1. Evolución de los Traspasos

Una de las líneas que más vías futuras ha abierto ha sido la que persigue mejorar el comportamiento de los traspasos en redes heterogéneas, pues esta mejora puede ser encaminada por varios frentes. En el presente trabajo hemos visto que extensiones como MCoA o la incorporación del estándar IEEE 802.21 han ayudado mucho en dicha mejora, incluso hemos hablado de reducir el tiempo en los métodos de autenticación, pero aún podemos ir un poco más allá, que son las vías futuras que identificamos aquí. Una de las vías futuras la identificamos en cuanto a los métodos de autenticación. Debido a que se puede predecir en muchos casos el trayecto que va a seguir un vehículo, puede realizarse un proceso de pre-autenticación con las distintas redes que vayan a ser visitadas, de tal forma que el proceso de autenticación sea lo más breve posible, por haber adelantado ya parte del proceso.

Otra vía futura importante dentro de esta línea es la de seguir con la implementación de IEEE 802.21, con la ayuda de ODTONE [33], para ser capaces de recoger mayor cantidad de información del entorno y efectuar los traspasos en el momento más adecuado posible, teniendo en cuenta tanto los requisitos del usuario como las condiciones del medio, así como el coste, para estar siempre lo mejor conectados posible en cada momento. Después de la aplicación de 802.21, la velocidad del vehículo no debería ser tan influyente en los resultados. Se propone, en continuación con esta línea, seguir realizando pruebas a distintas velocidades para constatar este hecho.

Y por último, dentro de este contexto, también se ha detectado que la fase de detección de direcciones duplicadas (DAD) de IPv6 consume demasiado tiempo dentro del proceso de traspaso. Como vía futura se plantea eliminar dicho mecanismo, ya que a priori, en IPv6 (direcciones de 128 bits, 64 de los cuales destinados a designar equipos en una red) es muy difícil que dos direcciones coincidan, debido a que normalmente se generan basándose en la dirección física de la interfaz de red (MAC) o es generada pseudo-aleatoriamente, como es el caso del mecanismo *Cryptographically Generated Address* [117] (CGA).

8.2.2. Mejoras en la Integración de Servicios de Movilidad y Seguridad

En otra de las líneas, en concreto la referente a la integración de los servicios de seguridad y movilidad, en el presente trabajo ha quedado constancia de la necesidad de que ambos servicios colaboren activamente. También se ha visto que en cada cambio de direccionamiento hay asociaciones de seguridad que no sobreviven a dicho cambio de IP, con lo que para evitar renegociarlas, se han ideado mecanismos de comunicación entre ambos servicios para actualizar dicha IP en las asociaciones. Se han presentado varias propuestas para interoperar, como *Migrate* o *Mobike*. Creemos que aquí se puede seguir trabajando para encontrar la mejor solución para resolver este problema, que podría estar en unir ambas implementaciones en una sola y, por tanto, sólo un proceso, con lo que esta comunicación entre servicios quedaría dentro de un único módulo software, sin necesidad de establecer ninguna normativa ni estándar que lo regule. Como trabajo futuro, se plantea este desarrollo que sea capaz de implementar ambos servicios. OpenIKEv2 está especialmente diseñado para poder ser incorporado en otras aplicaciones, por lo que lo más razonable parece incorporar la funcionalidad de IKEv2 en la implementación de NEMO de UMIP.

8.2.3. Explorar las Comunicaciones V2V

La Tesis ha demostrado la eficacia de nuestra propuesta en comunicaciones V2I basadas en IPv6, pero hemos dejado de lado la posibilidad de comunicar a los vehículos entre sí de forma directa, pues con el esquema V2I ya pueden hacerlo pero de forma indirecta. El inconveniente de esta vía indirecta es que la protección se realiza hasta el *Home Agent* situado en la central de la infraestructura de red, y por tanto el tráfico puede ser analizado y examinado por la operadora de carretera. Establecer conexiones entre coches siguiendo un esquema V2V nos aleja de este problema, pero este tipo de comunicaciones no suele basarse en IPv6 y los esquemas de seguridad se basan en protección individual de mensajes. Esto nos plantea importantes desafíos, como los de incorporar IPv6 a este tipo de conexiones y poder por tanto usar servicios de seguridad orientados a sesión como los que ofrecen IPsec e IKEv2 para proteger los datos intercambiados. Aparte habría que analizar la idoneidad de estos protocolos a este tipo de comunicaciones V2V. Sin embargo, aunque este tipo de enlaces V2V se crean y se destruyen muy rápidamente, podemos imaginar escenarios donde el vehículo necesite enviar o recibir una gran cantidad de datos con algún vehículo que vaya a estar cerca durante un periodo de tiempo suficiente para que merezca la pena usar este tipo de protección en V2V.

8.2.4. Interfaces Físicas Vs. Interfaz Virtual

Un problema evidente del uso de múltiples interfaces de red es que las aplicaciones tienen que estar preparadas para usar una u otra según múltiples condicionantes. El servicio de movilidad ha conseguido de manera aceptable ocultar esos detalles a

las aplicaciones mediante el uso de una única dirección IP, siendo la movilidad la que selecciona la interfaz adecuada en cada momento. Como vía futura se plantea el desarrollo de una capa de abstracción software que permita tener en el sistema una interfaz de red virtual cuya IP sea la comentada antes, y que internamente se vincule dicha interfaz a las que el sistema crea necesario en cada momento de forma transparente. De esta forma se abstrae absolutamente la gestión y selección de interfaces físicas, quedando enmascaradas por dicha interfaz virtual permanente.

Todos estos trabajos persiguen mejorar las comunicaciones en C-ITS y, por tanto, conseguir alcanzar sus objetivos de aumentar la seguridad de todos los medios de transporte actuales, mejorar la eficiencia optimizando trayectos y evitando congestiones, incentivar la intermodalidad mediante el uso de diferentes medios de transporte, integrar éstos dentro de las políticas de desarrollo sostenible reduciendo su impacto ecológico, y por último pero no menos importante, aumentar el confort de los pasajeros.

Bibliografía

[1] Asim Rasheed, Haleemah Zia, Farhan Hashmi, Umair Hadi, Warda Naim, and Sana Ajmal. Fleet & convoy management using vanet. *Journal of Computer Networks*, 1(1):1–9, 2013.

[2] W. Barfield and T.A. Dingus, editors. *Human Factors in Intelligent Transportation Systems*. Lawrence Erlbaum Associates, United Kingdom, 1998.

[3] J. Ehrlich, editor. *Intelligent Transportation Systems*, volume 60 of *Annals of Telecommunications*. GET - Lavoisier, France, March/April 2005.

[4] Department of Violence, Injury Prevention, and Disability (VIP), editors. *Global Status Report on Road Safety 2013*. World Health Organization, Switzerland, 2013.

[5] M.A. Chowdhury and A. Sadek. *Fundamentals of Intelligent Transportation Systems Planning*. Artech House, USA, 2003.

[6] Yacine Khaled, Manabu Tsukada, José Santa, JinHyeock Choi, and Thierry Ernst. A usage oriented analysis of vehicular networks: from technologies to applications. *Journal of Communications*, 4(5), 2009.

[7] ETSI TC ITS. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions. ETSI TR 102 638, June 2009.

[8] ISO TC 204. Intelligent transport systems - Communications Access for Land Mobiles (CALM) - Architecture. ISO 21217, April 2013.

[9] ETSI TC ITS. Intelligent Transport Systems (ITS); Communications Architecture. ETSI EN 302 665, September 2010.

[10] IEEE 1609 WG. IEEE Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services, 2010.

[11] Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 3: Network architectures, March 2010. ETSI TS 102 636-3 V1.1.1 (2010-03).

[12] S. Deering and R. Hinden. Internet Protocol version 6 (IPv6) Specification. RFC 2460 (Proposed Standard), December 1998.

[13] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Jul 2007. IEEE Std 802.11-2007.

[14] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Jun 2010. IEEE Std 802.11p-2010.

[15] ETSI TC ITS. European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band. ETSI ES 202 663, Nov. 2009.

[16] IEEE 1609 WG. IEEE Standard for Wireless Access in Vehicular Environments (WAVE), 2010.

[17] SAE. Dedicated Short Range Communications (DSRC) Message Set Dictionary, Sep 2015.

[18] COMmunications for eSafety (MeSafety) project. http://www.comesafety.org.

[19] ITSSv6 - IPv6 ITS Station Stack. https://project.inria.fr/itssv6/.

[20] FOTsis - European Field Operational Test on Safe, Intelligent and Sustainable Road Operation. http://www.fotsis.com/.

[21] Drive C2X. http://www.drive-c2x.eu/project.

[22] OVERSEE - Open Vehicular Secure Platform. https://www.oversee-project.com.

[23] PRESERVE - Preparing Secure Vehicle-to-X Communication Systems, note =.

[24] SEcure VEhicular COMmunications (SEVECOM) project. http://www.sevecom.org.

[25] EVITA - E-safety vehicle intrusion protected applications. http://evita-project.org/.

[26] Horizon 2020 Work Programme 2014 - 2015, April 2015.

[27] European Commission eSafety Initiative. http://ec.europa.eu/information_society/activities/esafety/.

[28] ENABLE - Enabling Efficient and Operational Mobility in Large Heterogeneous IP Networks. http://www.ist-enable.eu/.

[29] P. Eronen and P. Hoffman. IKEv2 Clarifications and Implementation Guidelines. RFC 4718 (Proposed Standard), October 2006.

[30] C. Kaufman, P. Hoffman, Y. Nir, and P. Eronen. Internet Key Exchange Protocol Version 2 (IKEv2). RFC 5996 (Proposed Standard), September 2010.

[31] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, and T. Kivinen. Internet Key Exchange Protocol Version 2 (IKEv2). RFC 7296 (Proposed Standard), October 2014.

[32] Jun Murai from USAGI/WIDE Project. Mobile IPv6 and NEMO for Linux.

[33] Carlos Guimarães, Daniel Corujo, and Rui Aguiar. ODTONE: Open 802.21.

[34] R. Stahlmann, A. Festag, A. Tomatis, I. Radusch, and F. Fischer. Starting European Field Tests for Car-2-X Communication: The Drive C2X Framework. In *18th ITS World Congress and Exhibition 2011*, ITS World Congress '11, 2011.

[35] C. Weib. V2X communication in Europe: From research projects towards standardization and field testing of vehicle communication technology. *Computer Networks*, 55(14):3103 – 3119, 2011.

[36] O. Shagdar, M. Tsukada, M. Kakiuchi, T. Toukabri, and T. Ernst. Experimentation towards IPv6 over IEEE 802.11p with ITS Station Architecture. In *Intelligent Vehicles Symposium (IV'12), 2012 IEEE*, pages 1–6, Madrid, Spain, jun. 2012. IEEE.

[37] Jia-Chin Lin, Chi-Sheng Lin, Chih-Neng Liang, and Bo-Chiuan Chen. Wireless communication performance based on IEEE 802.11p R2V field trials. *IEEE Communications Magazine*, 50(5):184 –191, may 2012.

[38] Fernando A. Teixeira, Vinicius F. e Silva, Jesse L.Leoni, Daniel F. Macedo, and José M. S. Nogueira. Vehicular Networks using the IEEE 802.11p Standard: An Experimental Analysis. *Vehicular Communications*, 2014.

[39] Carolina Pinart, Pilar Sanz, Iván Lequerica, Daniel García, Isaac Barona, and Diego Sánchez-Aparisi. DRIVE: a reconfigurable testbed for advanced vehicular services and communications. In *Proceedings of the 4th International Conference on Testbeds and research infrastructures for the development of networks & communities*, TridentCom '08, pages 16:1–16:8, 2008.

[40] S. Cespedes, Xuemin Shen, and C. Lazo. IP mobility management for vehicular communication networks: challenges and solutions. *Communications Magazine, IEEE*, 49(5):187–194, May 2011.

[41] Thouraya Toukabi, Manabu Tsukada, Thierry Ernst, and Lamjed Bettaieb. Experimental evaluation of an open source implementation of IPv6 GeoNetworking in VANETs. In *ITST 2011: 11th International Conference on Intelligent Transport System Telecommunications)*, September 2011.

[42] GEONET - Geo-addressing and Geo-routing for Vehicular Communications. http://www.geonet-project.eu/.

[43] C. Perkins, D. Johnson, and J. Arkko. Mobility Supporty in IPv6. RFC 6275 (Proposed Standard), July 2011.

[44] M. Tsukada, J. Santa, O. Mehani, Y. Khaled, and T. Ernst. Design and Experimental Evaluation of a Vehicular Network Based on NEMO and MANETS. *EURASIP Journal on Advances in Signal Processing*, 2010(656407):1– 18, september 2010.

[45] M.S. Hossain, Mohammed Atiquzzaman, and Will Ivancic. Performance evaluation of multihomed NEMO. *Communications (ICC)*, 2012.

[46] R. Koodli. Proxy Mobile IPv6. RFC 5213 (Proposed Standard), July 2008.

[47] R. Koodli. Mobile IPV6 Fast Handovers. RFC 5568 (Proposed Standard), July 2009.

[48] R. Wakikawa, V. Devarapalli, G. Tsirtsis, T. Ernst, and K. Nagami. Multiple Care-of Addresses Registration. RFC 5648 (Proposed Standard), October 2009.

[49] Bruno Sousa, Marco Silva, Kostas Pentikousis, and Marilia Curado. A multiple care of addresses model. In *16th IEEE Symposium on Computers and Communications (ISCC 2011)*, Confu, Greece, March 2011.

[50] Bruno Sousa, Kostas Pentikousis, and Marilia Curado. A Study of Multimedia Application Performance over Multiple Care-of Addresses in Mobile IPv6. In *16th IEEE Symposium on Computers and Communications (ISCC 2011)*, Confu, Greece, March 2011.

[51] Xiaohua Chen, Hongke Zhang, Yao-Chung Chang, and Han-Chieh Chao. Experimentation and Performance Analysis of Multi-Interfaced Mobile Router Scheme. *Simulation Modelling Practice and Theory*, pages 407–415, 2009.

[52] Choongyong Shin, Kyungon Lee, and Jinsung Cho. A Framewok for Multiple Wireless Services in Heterogeneous Wireless Networks. In *6th International Conference on Mobile Technology, Applications, and Systems, Mobility*. ACM, 2009.

[53] Luis Alejandro Flétscher Bocanegra and Antonio F. Gómez Skarmeta. Proposal for Implementation of 802.21 Information Services (MIIS) as Handover Support in VANET Networks. In *Ingeniería y Desarrollo*, December 2010.

[54] Johann M. Marquez-Barja, Hamed Almadi, Sergio M. Tornell, Carlos T.Calafate, Juan Carlos Cano, Prieto Manzoni, and Luiz A. DaSilva. Breaking the vehicular wireless communications barriers: Vertical handover techniques for heterogeneous networks. *IEEE Transactions On Vehicular Technology*, 2014.

[55] M. Raya, P. Papadimitratos, and J-P Hubaux. Securing vehicular communications. *Wireless Communications, IEEE*, 13(5):8–15, October 2006.

[56] Bassem Mokhtar and Mohamed Azab. Survey on Security Issues in Vehicular Ad Hoc Networks. *Alexandria Engineering Journal*, 2015.

[57] Shinta Sugimoto, Francis Dupont, and Ryoji Kato. Interactions between Mobile IPv6 and IPsec/IKE. *IPSJ Digital Courier*, 2006.

[58] C. Metz and B. Phan. PF-KEY Key Management API, Version 2. RFC 2367 (Proposed Standard), July 1998.

[59] Ke Xu, Minpeng Qi, Haitao Li andPeng Yang, and Hui Deng. A Novel Interffacing Solution to Make IKEv2 Work in MIPv6 Environment. In *Communications Workshops (ICC)*, June 2008.

[60] Alejandro Pérez-Méndez, Pedro J. Fernández Ruiz, Rafael Marín López, Gregorio Martínez Pérez, Antonio F. Gómez-Skarmeta, and Kenichi Taniuchi. OpenIKEv2: Design and Implementation of an IKEv2 Solution. *IEICE Transactions*, pages 1319–1329, 2008.

[61] Pedro J. Fernandez Ruiz, Fernando Bernal Hidalgo, Cristian A. Nieto Guerra, and Antonio F. Gomez Skarmeta. Mobility and security in a real VANET deployed in a heterogeneous networks. *Security and Communication Networks*, 2012.

[62] J Santa, F Pereniguez-Garcia, F Bernal, Pedro J Fernandez, R Marin-Lopez, and A F Skarmeta. A Framework for Supporting Network Continuity in Vehicular IPv6 Communications. *IEEE Intelligent Transportation Systems Magazine*, 6(1):17–34, 2014.

[63] José Santa, Fernando Pereñiguez, Antonio Moragón, Pedro J. Fernández, Fernándo Bernal, and Antonio F. Skarmeta. IPv6 Communication Stack for Deploying Cooperative Vehicular Services. *International Journal of Intelligent Transportarion System Research*, 12(2):48–60, 2014.

[64] OASIS - Operación de Autopistas Seguras, Inteligentes y Sostenibles. http://www.cenitoasis.com/.

[65] Pedro J. Fernandez, Jose Santa, Fernando Bernal, and Antonio Skarmenta. Securing Vehicular IPv6 Communications. *IEEE Transactions on Dependable and Secure Computing*.

[66] José Santa, Pedro J. Fernández, Fernando Pere níguez, and Antonio F. Skarmeta. Real experience with ipv6 communications in highways. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, 6(3):36–53, September 2015.

[67] Jose Santa, Pedro J. Fernandez, Fernando Pereñiguez, and Antonio Skarmenta. Deployment of Vehicular Networks in Highways using 802.11p and IPv6 Technologies. *International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC)*, 2015.

[68] Pedro J. Fernández, Cristian A. Nieto, José Santa, Antonio F. Gómez-Skarmeta, Johann Márquez-Barja, and Pietro Manzoni. Experience Developing a Vehicular Network Based on Heterogeneous Communication Technologies, Wireless Technologies in Vehicular Ad Hoc Networks: Present and Future Challenges. In Raul Aquino Santos, Arthur Edwards, and Victor Rangel Licea, editors, *Wireless Technologies in Vehicular Ad Hoc Networks: Present and Future Challenges*. IGI Global, 2012.

[69] Pedro Javier Fernández Ruiz, Fernando Bernal Hidalgo, José Santa Lozano, and Antonio F. Skarmeta. Deploying ITS Scenarios Providing Security and Mobility Services Based on IEEE 802.11p Technology, Vehicular Technologies - Deployment and Applications. In Lorenzo Galati Giordano and Luca Reggiani, editors, *Vehicular Technologies - Deployment and Applications*. InTech, 2013.

[70] Fernando Pereñiguez, José Santa, Pedro Javier Fernández Ruiz, Fernando Bernal Hidalgo, Antonio F. Skarmeta, and Thierry Ernst. The Use of IPv6 in Cooperative ITS: Standarization Viewpoint. In C. Campolo et al., editor, *Vehicular Ad-Hoc Networks*. Springer, 2015.

[71] Pedro J. Fernández Ruiz, Cristian A. Nieto Guerra, and Antonio F. Gómez-Skarmeta. Deployment of a Secure Wireless Infrastructure Oriented to Vehicular Networks. In *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference, Perth, Australia*, pages 1108–1114. IEEE, April 2010.

[72] Pedro J. Fernández Ruiz, Cristian A. Nieto Guerra, and Antonio F. Gómez-Skarmeta. Autenticación basada en IKEv2 y EAP para Escenarios de Redes Vehiculares. In *IX Jornadas de Ingeniería Telemática (JITEL'10), Valladolid, Spain*, Sep 2010.

[73] Pedro J. Fernández Ruiz and Antonio F. Gómez-Skarmeta. Providing Security using IKEv2 in a Vehicular Network based on WiMAX Technology. In *3rd IEEE Intelligent Vehicular Communications System Workshop (IVCS'11), Las Vegas, Nevada, USA*, pages 1001–1006. IEEE, Jan 2011.

[74] Pedro J. Fernández Ruiz, Cristian A. Nieto Guerra, and Antonio F. Gómez-Skarmeta. Seguridad y movilidad en una VANET real desplegada con diferentes tecnologías inalámbricas. In *X Jornadas de Ingeniería Telemática (JITEL'11), Santander, Spain*, Sep 2011.

[75] José Santa, Manuel Mora, Antonio Moragón, Andrés S. García, Pedro J. Fernández, Fernando Bernal, Antonio F. Skarmeta, and Jaime Arrazola.

Arquitectura de comunicaciones en OASIS: desarrollo de una pila de comunicación ITS siguiendo los conceptos de CALM y Arquitectura Europea de Comunicación ITS. In *XII Congreso Español sobre ITS, Madrid, Spain*, 2012.

[76] Pedro J. Fernández José Santa, Antonio Moragón, Andrés S. García, Fernando Bernal, and Antonio F. Gómez-Skarmeta. Architecture and development of a networking stack for secure and continuous service access in vehicular environments. In *19th ITS World Congress (ITSWC 2012). Vienna, Austria*, 2012.

[77] José Santa, Fernando Bernal, Pedro J. Fernández, Antonio Moragón, Andrés S. García, and Antonio F. Gómez-Skarmeta. Continuous IPv6 Communications in a Vehicular Networking Stack for Current and Future ITS Services. In *First International Workshop on IPv6-based Vehicular Networks (Vehi6), 2012 IEEE Intelligent Vehicles Symposium (IV'2012), Alcalá de Henares, Spain*, 2012.

[78] José Santa, Pedro J. Fernández, Fernando Pereñiguez, Fernando Bernal, and Antonio F. Gómez-Skarmeta. A Vehicle Network Mobility Framework: Architecture, Deployment and Evaluation. In *IEEE INFOCOM International Workshop on Mobility Management in the Networks of the Future World (MobiWorld 2015). Hong Kong, China*, 2015.

[79] ISO TC 204. Intelligent transport systems - Communications Access for Land Mobiles (CALM) - IPv6 Networking. ISO 21210, January 2011.

[80] ISO. Intelligent transport systems – Communication Access for Land Mobiles (CALM) – Part 2: Fast networking & transport layer protocol (FNTP), June 2013. ISO 29281-1:2013.

[81] Intelligent Transport Systems (ITS); Vehicular Communications; Part 4: Geographical Addressing and Forwarding for Point-to-Point and Point-to-Multipoint Communications; Sub-part 1: Media-Independent Functionality, June 2011. ETSI TS 102 636-4-1 V1.1.1 (2011-06).

[82] ETSI TC ITS. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service. ETSI TS 102 637-2, March 2011.

[83] ETSI TC ITS. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service. ETSI TS 102 637-3, September 2010.

[84] Defense Advanced Research Agency. Internet Protocol. RFC 791 (Proposed Standard), September 1981.

[85] C. Bao, C. Huitema, and M. Bagnulo. IPv6 Addressing of IPv4/IPv6 Translators. RFC 6052, October 2010.

[86] M. Bagnulo, A. Sullivan, P. Matthews, and I. van Beijnum. DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers. RFC 6147 (Proposed Standard), 2011.

[87] G. Tsirtsis, H. Soliman, N. Montavont, G. Giaretta, and K. Kuladinithi. Flow Bindings in Mobile IPv6 and Network Mobility (NEMO) Basic Support. RFC 6089, January 2011.

[88] S. Kent and K. Seo. Security Architecture for the Internet Protocol. RFC 4301 (Proposed Standard), December 2005.

[89] R. Atkinson. Security Arquitecture for the Internet Protocol. RFC 1825 (Proposed Standard), August 1995.

[90] S. Kent and R. Atkinson. Security Arquitecture for the Internet Protocol. RFC 2401 (Proposed Standard), November 1998.

[91] D. Piper. The Internet IP Security Domain of Interpretation for ISAKMP. RFC 2407 (Proposed Standard), November 1998.

[92] D. Maughan, M. Schertler, M. Schneider, and J. Turner. Internet Security Association and Key Management Protocol (ISAKMP). RFC 2408 (Proposed Standard), November 1998.

[93] D. Harkins and D. Carrel. The Internet Key Exchange (IKE). RFC 2409 (Proposed Standard), November 1998.

[94] C. Kaufman. Internet Key Exchange (IKEv2) Protocol. RFC 4306 (Proposed Standard), December 2005.

[95] Y. Ohba, Q. Wu, and G. Zorn. Extensible Authentication Protocol (EAP) Early Authentication Problem Statement. RFC 5836, April 2010.

[96] V. Devarapalli and F. Dupont. Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture. RFC 4877 (Proposed Standard), April 2007.

[97] ISO. Intelligent transport systems – Communication Access for Land Mobiles (CALM) – Part 2: Fast networking & transport layer protocol (FNTP), June 2013. ISO 29281-1:2013.

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

[99] A. Conta, S. Deering, and M. Gupta. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. RFC 4443 (Proposed Standard), 2006.

[100] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. Neighbor Discovery for IP version 6 (IPv6). RFC 4861 (Proposed Standard), September 2007.

[101] David C. Plummer. An Ethernet Address Resolution Protocol. RFC 826 (Proposed Standard), November 1982.

[102] P. Ferguson and D. Senie. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2827 (Proposed Standard), May 2000.

[103] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. Extensible Authentication Protocol (EAP). RFC 3784 (Proposed Standard), 2004.

[104] D. Simon, B. Aboba, and R. Hurst. The EAP-TLS Authentication Protocol. RFC 5216 (Proposed Standard), 2008.

[105] OpenVPN software project. https://openvpn.net/.

[106] Pedro J. Fernández and Alejandro Pérez Méndez. OpenIKEv2.

[107] WIDE Project. Racoon2.

[108] Andreas Steffen. StrongSwan.

[109] Stjepan Gros, Ana Kukec, and Domagoj Jakobovic. IKEv2.

[110] J. Loughney, M. Nakhjiri, C. Perkins, and R. Koodli. Context Transfer Protocol (CXTP). RFC 4067, July 2005.

[111] S. Sugimoto, F. Dupont, and M. Nakamura. PF-KEY Extension as an Interface between Mobile IPv6 and IPsec/IKE. draft-sugimoto-mip6-pfkey-migrate-04, Dec 2007.

[112] P. Eronen. IKEv2 Mobility and Multihoming Protocol (MOBIKE). RFC 4555 (Proposed Standard), Jun 2006.

[113] T. Kivinen and H. Tschofenig. Design of the IKEv2 Mobility and Multihoming (MOBIKE) Protocol. RFC 4621, August 2006.

[114] S. Decugis. Key Management Mobility Capability (K) flag in Mobile IPv6 BU/BA messages. RFC draft, December 2007.

[115] R. Marin-Lopez, F. Pereniguez-Garcia, F. Bernal-Hidalgo, and A. Gomez-Skarmeta. Architecture for Fast EAP Re-authentication based on a new EAP method (EAP-FRM) working on standalone mode. RFC draft, March 2011.

[116] Media Independent Handover Working Group. IEEE 802.21.

[117] T. Aura. Cryptographically Generated Addresses (CGA). RFC 3972 (Proposed Standard), March 2005.

[118] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-Hashing for Message Authentication. RFC 2104 (Proposed Standard), October 1997.

[119] OpenSSL software project. http://www.openssl.org/.

[120] Valgrind software project. http://valgrind.org/.

Apéndice A

Detalles del protocolo IKEv2

El protocolo IKEv2 está definido en el RFC 4306 y posteriores actualizaciones, en donde se especifica el funcionamiento del protocolo con detalle. En las siguientes líneas resumimos numerosos detalles de importancia del protocolo, que nos ayudará a entenderlo mejor sin necesidad de indagar en los documentos RFC el IETF.

A.1. Transporte de IKEv2

IKEv2 normalmente escucha y envía paquetes UDP por el puerto 500, pero también por el puerto 4500 con un ligero cambio en el formato de los paquetes para el soporte de NAT. Todas las implementaciones de IKEv2 deben ser capaces de enviar, recibir y procesar mensajes de hasta 1280 bytes, pero se sugiere que puedan hacerlo hasta con mensajes de 3000 bytes.

Dada la naturaleza no confiable de los paquetes UDP, donde no hay nada que certifique que un paquete ha llegado a su destino, IKEv2 incluye procedimientos de recuperación de errores de transmisión, pérdida de paquetes o falsificación de los mismos. Por el contrario no dispone de ningún mecanismo de fragmentación de mensajes grandes, por lo que se debe prestar especial atención a su tamaño, ya que si supera el máximo soportado de un paquete UDP se debe intentar dejar fuera algunas propuestas de paquetes criptográficos, o bien pasar los certificados en formato *Hash & URL* para reducir el tamaño del mismo y recuperarlo posteriormente.

A.2. Retransmisión de Mensajes

Para cada par de mensajes, el iniciador del intercambio tiene que inicializar un temporizador cuando realiza el envío de la petición y se compromete a retransmitirlo en el caso de que el temporizador expire. El respondedor nunca debe retransmitir una respuesta a no ser que reciba una retransmisión de la petición anterior. El iniciador debe recordar cada petición hasta que reciba su correspondiente respuesta. De igual forma, el receptor de la petición debe conservar cada respuesta hasta que reciba una nueva petición con el siguiente número de secuencia.

A.3. Identificadores de Mensaje

Todo mensaje en IKEv2 contiene un identificador de 32 bits como parte de la cabecera. Es usado para identificar intercambios, es decir, tanto la petición como su correspondiente respuesta tendrán asignado el mismo identificador de mensaje. Esta asignación la realiza el iniciador del intercambio, con lo que cada extremo debe disponer de su propio número de secuencia para los intercambios iniciados por cada uno. Además cada extremo deberá llevar cuenta del número de secuencia de la otra parte, para así poder detectar retransmisiones y ataques por reenvíos. Se asignará el valor 0 para el intercambio inicial IKE_SA_INIT y el valor 1 para IKE_AUTH.

A.4. Ventana Deslizante de Peticiones

Para aumentar la productividad de IKEv2, un extremo puede realizar múltiples peticiones antes de recibir ninguna respuesta. Esta habilidad tiene que ser indicada por el otro extremo mediante un payload de notificación de tipo SET_WINDOW_SIZE indicando el tamaño de la ventana. Este tamaño es igual al número de peticiones que el extremo puede llegar a procesar en paralelo. Elegir un tamaño de ventana 1 supone que cada petición debe esperar a que llegue la respuesta de la petición anterior.

A.5. Sincronización de Estado entre los Extremos

Los extremos en IKEv2 pueden desechar todo el estado asociado a una IKE_SA y todas sus CHILD_SAs en cualquier momento. Esto suele ser ocasionado por un bloqueo o reinicio del proceso. Lo realmente importante es que el otro extremo detecte esta situación pronto y no siga desperdiciando recursos. IKEv2 no se basa en la información de encaminamiento de la red de tipo ICMP o mensajes IKE sin protección criptográfica para detectar dicha situación. IKEv2 concluirá que el otro extremo ya no está operativo si después de muchos intentos no se recibe respuesta. IKE podrá usar intercambios INFORMATIONAL con el cuerpo del mensaje vacío para comprobar si el extremo sigue vivo. El número de reintentos o la duración de los tiempos de espera no están cubiertos en las especificaciones de IKEv2, ya que no afecta a la interacción entre distintas implementaciones.

A.6. Ataque de DoS en el Iniciador de una IKE_SA

Como el primer intercambio inicial de IKE no esta protegido criptográficamente, un atacante podría responder al mensaje del iniciador antes que el verdadero mensaje de respuesta sea recibido, impidiendo así la creación de la IKE_SA. Para contrarrestar este ataque el iniciador podría aceptar más de una respuesta en el intercambio IKE_SA_INIT, realizar para cada uno de ellos el intercambio IKE_AUTH quedándose con el que haya resultado exitoso y descartando el resto.

A.7. Número de Versión y Compatibilidad con Futuras Versiones

La versión viene compuesta por un número de versión, seguido de un punto y un numero de subversión (Ej: 2.0). El cambio de una versión a otra implica que la nueva versión es incompatible con la anterior. Sin embargo, el cambio en el número de subversión sólo implica la inclusión de nuevas capacidades y servirá por tanto para saber qué capacidades soporta cada extremo. Ambas partes en una negociación IKE deben utilizar la mayor versión posible que soporten en común. Para ello si uno de los extremos recibe un mensaje con un número de versión mayor al que soporta, debe enviarle un mensaje de notificación no cifrado indicando la mayor versión con la que puede trabajar. Aquí puede darse un ataque en implementaciones que soporten tanto la versión 1 como la 2, forzando a ambas a trabajar con la versión 1, y así aprovecharse de las vulnerabilidades que ello conlleva. Para evitar estas situaciones en donde por éste u otro motivo se trabaje en una versión inferior a la máxima que en común se puede usar, los mensajes llevan un bit en la cabecera que indica si puede trabajar con una versión superior a la que actualmente se está usando. De este modo se puede corregir la situación y negociar de nuevo en la versión adecuada. Los campos reservados para las siguientes versiones, en aquellas donde no tengan uso definido, deben establecerse a cero en los mensajes que se envíen e ignorar su contenido en los que se reciban.

A.8. Cookies

Las cookies son un mecanismo pensado para evitar que un atacante deje sin recursos de memoria y CPU al receptor, haciéndole llegar una cantidad ingente de peticiones de intercambio IKE_SA_INIT desde direcciones IP inventadas o aleatorias. Para mitigar los efectos de este ataque es recomendable hacer el mínimo uso de CPU posible y no almacenar ningún estado hasta que se sepa si el iniciador puede recibir mensajes en la dirección IP que ha indicado. Para ello, a partir de un cierto umbral de número de IKE_SAs en proceso de creación, se puede considerar la posibilidad de estar sufriendo un ataque DoS. A partir de este momento, se deben rechazar los intercambios IKE_SA_INIT que no lleven la cookie adecuada. Dicha cookie se proporciona en la respuesta de aquellos intercambios que no la contengan por medio de un payload NOTIFY. El iniciador legítimo debe de encargarse de repetir el intercambio IKE_SA_INIT incluyendo la cookie que se le ha entregado. Esta cookie, para que sea efectiva y no ocupe memoria, debe calcularse a partir de datos que vengan dados por el mensaje. Un ejemplo podría ser este:

```
Cookie = IndiceDelSecreto | Hash( Ni | IPi | SPIi | Secreto)

IndiceDelSecreto: identificador que indica qué secreto se ha
                  usado para cada cookie.
```

```
Ni: valor del nonce de la petición del intercambio.
IPi: dirección IP del iniciador de la petición del intercambio.
SPIi: SPI del iniciador de la petición del intercambio.
Secreto: secreto sólo conocido por el receptor.
         Debe de ser cambiado regularmente.
```

A.9. Negociación de los Algoritmos Criptográficos

La negociación de algoritmos criptográficos se realiza mediante el payload SA, que contiene una o más propuestas sobre un conjunto de posibles protocolos (IKE, ESP y/o AH) así como las transformaciones o algoritmos criptográficos asociados a ellos. El payload SA puede tener una o más propuestas, cada propuesta uno o más protocolos, cada protocolo una o más transformaciones y cada transformación los atributos que necesite, o bien ninguno. En la Figura A.1 se muestra un diagrama en el que se refleja esta estructura.

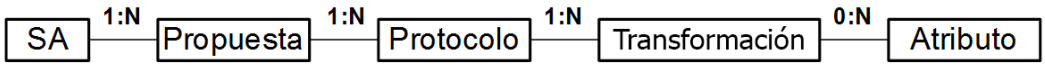


Figura A.1: Estructura lógica de una propuesta IPsec

En la petición del iniciador deberá incluirse una o varias propuestas, mientras que en la respuesta sólo se incluirá la propuesta aceptada por el receptor (si es que acepta alguna). Todos los protocolos incluidos en la propuesta seleccionada son aceptados, mientras que únicamente se podrá aceptar una transformación de cada tipo. Cada protocolo sólo podrá tener transformaciones del tipo adecuado, es decir, si se trata del protocolo ESP, entonces las transformaciones serán las de tipo ENCR para cifrar. Sin embargo el protocolo AH solo podrá tener asociado transformaciones del tipo INTEG para autenticar. El grupo Diffie-Hellman para la IKE_SA también se negocia aquí. El iniciador debe intentar adivinar un grupo de los soportados por el receptor. Si elige mal el receptor le enviará una notificación indicándole el grupo adecuado. El iniciador en este caso reintentará la negociación con dicho grupo.

A.10. Generación de Material Criptográfico

La generación de material criptográfico se obtiene fundamentalmente mediante una función pseudo-aleatoria prf basada en una función HMAC [118] que se negocia en el intercambio IKE_SA_INIT. Dicha función toma como argumentos una clave y una cantidad variable de datos, generando una cantidad fija de datos pseudo-aleatorios. Se define además la función prf+ que usa a la función prf en sucesivas iteraciones para generar una gran cantidad de datos pseudo-aleatorios, de donde se utilizarán

únicamente los bytes necesarios (no será necesario realizar las 255 iteraciones máximas sino tantas como sean necesarias para generar el suficiente material criptográfico).

```
prf+ (Clave, Datos) = T1 | T2 | T3 | T4 | ... | T255

donde:

T1 = prf (Clave, Datos | 0x01)
T2 = prf (Clave, T1 | Datos | 0x02)
T3 = prf (Clave, T2 | Datos | 0x03)
...
T255 = prf (Clave, T254 | Datos | 0xFF)

Nota: Se denota con \|" a la concatenación de datos.
```

A.10.1. IKE_SA

Para generar el material criptográfico de una IKE_SA, se parte de una clave llamada SKEYSEED generada en el intercambio IKE_SA_INIT. Dicha clave se obtiene de la siguiente manera:

```
SKEYSEED = prf ( Ni | Nr , g^ir )

donde:

Ni y Nr son los nonces intercambiados.
g^ir es el secreto compartido generado por el intercambio Diffie-Hellman.
```

Una vez obtenida SKEYSEED se procede a usar la función prf+ para generar la cantidad suficiente de datos pseudo-aleatorios para extraer las claves necesarias para una IKE_SA. Dichas claves se obtienen de la siguiente manera:

```
Claves = prf+ ( SKEYSEED, Ni | Nr | SPIi | SPIr )

donde:

Claves es una cadena de datos pseudo-aleatorios que contiene las siguientes claves:
SK_d es la clave usada para derivar futuro material criptográfico.
SK_ai y SK_ar son las claves usadas para autenticar en cada una de las direcciones.
SK_ei y SK_er son las claves usadas para cifrar en cada una de las direcciones.
SK_pi y SK_pr son las claves usadas para en el computo del payload AUTH.
Es decir, Claves = { SK_d | SK_ai | SK_ar | SK_ei | SK_er | SK_pi | SK_pr }
SPIi y SPIr son los SPIs de la IKE_SA.
Ni y Nr son los nonces intercambiados.
```

A.10.2. CHILD_SA

Para generar el material criptográfico necesario para una nueva CHILD_SA, se hará a partir del material criptográfico de la IKE_SA progenitora, más concretamente la clave SK_d. Este material se denomina KEYMAT y se generará de la siguiente manera:

```
KEYMAT = prf+ (SK_d, Ni | Nr)
```

En el caso de que el intercambio CREATE_CHILD_SA incluya un payload KE con los valores de un nuevo intercambio D-H (PFS), entonces se generará de la siguiente manera:

```
KEYMAT = prf+ (SK_d, g^ir | Ni | Nr)
```

KEYMAT contendrá todas las claves necesarias para el par de CHILD_SAs recién creadas.

A.11. Regeneración de Claves

Las asociaciones de seguridad usan claves secretas que deberían usarse durante un período limitado de tiempo y para proteger una cantidad limitada de datos. Esto hace que dichas claves tengan una caducidad. Cuando las claves de una SA caducan no debería seguir usándose dicha SA, sino que debería cerrarse y crear otra con nuevas claves. A esta acción se le llama regeneración de claves o *rekeying*.

La regeneración de claves de una SA puede hacerse de dos formas:

- Cerrando por completo la IKE_SA y sus CHILD_SA asociadas, sustituyéndolas por nuevas IKE_SA y CHILD_SA equivalentes con nuevo material criptográfico. Este tipo de regeneración se puede usar tanto para las IKE_SA como para las CHILD_SA. Tiene la ventaja de que requiere de un mínimo esfuerzo de implementación. Por contra la regeneración de claves es más lenta e interrumpe momentáneamente la comunicación, lo que podría dar lugar a pérdida de paquetes.

- Creando una SA equivalente mediante un intercambio CREATE_CHILD_SA usando la IKE_SA actual, cerrando la antigua SA cuando quede establecida la nueva. En el caso de que se use este método para la regeneración de IKE_SAs, la recién creada hereda todas las CHILD_SA de la antigua. A partir de ese momento se usará la nueva IKE_SA para mantener las CHILD_SA. Este método tiene la ventaja de hacer el cambio mucho más rápido, evitando la pérdida de paquetes ya que una SA no se cierra hasta que la nueva no esté establecida.

Hay que renovar las SA antes de que caduquen las claves y se conviertan en inútiles, es decir, una renovación proactiva. Los tiempos de vida de las claves no se negocian en IKEv2 a diferencia de como se hacía en la versión anterior. Ahora cada extremo es

responsable de forzar sus propias renovaciones de claves cuando sean necesarias. Existe la posibilidad de que ambos extremos decidan iniciar este proceso en el mismo instante, lo que podría producir eliminaciones simultáneas de SAs. Para reducir esa probabilidad se variará el tiempo de vida en una cantidad aleatoria.

A.12. Selectores de Tráfico

Un selector de tráfico consiste en un rango de direcciones IP, un rango de puertos y un tipo de protocolo IP. Con ellos se puede especificar el tipo de tráfico que se quiere proteger bajo una determinada CHILD_SA. Dichos selectores se obtienen a partir de las políticas de seguridad establecidas en la base de datos de políticas (SPD). Para establecer una CHILD_SA previamente hay que realizar una negociación de selectores de tráfico entre los extremos, de forma que el iniciador propone una serie de selectores de tráfico que el extremo debe aceptar, reducir o rechazar según sean sus políticas.

A.13. Mecanismo de Asignación de Direcciones

Un escenario muy común es aquel en el que un extremo se encuentra en otra red y quiere comunicarse con su red de procedencia a través de la pasarela de seguridad (ver sección 2.3.3). Dicho extremo dispone de una dirección IP de la red visitada, pero además necesitaría que se le asigne dinámicamente una dirección IP de su red de procedencia. De esta forma podrá establecer un túnel con la pasarela usando la dirección IP de la red visitada y en adelante hacer pasar todo el tráfico por el túnel con la IP de la red de procedencia. Esta petición se realiza mediante un payload de configuración CP en la petición de los intercambios IKE_AUTH y/o CREATE_CHILD_SA. La pasarela podrá proporcionarle una dirección mediante otro payload CP, ya sea una dirección prefijada, o preguntando a un servidor DHCP/BOOTP.

A.14. Autenticación de la IKE_SA

Sobre autenticación en IKEv2 se desarrolla en profundidad en la Sección 6.2 el Capítulo 6, donde se proponen mejoras para hacer los traspasos de una red a otra lo más imperceptibles posible.

A.15. Manejo de Errores

Durante el funcionamiento de IKEv2 pueden ocurrir errores de diverso tipo. Si se recibe una petición mal formada o que sea inaceptable por motivos de políticas, la respuesta debe contener un payload NOTIFY indicando el error. Si el error ocurre fuera del contexto de IKE (por ejemplo se recibe un error proveniente de una CHILD_SA) se debe iniciar un intercambio de tipo INFORMATIONAL que notifique el problema. Se

debe llevar especial cuidado con los errores que ocurran antes de que una IKE_SA se establezca, ya que pueden ser fruto de un intento de ataque por denegación de servicio. En estos casos es mejor no dar demasiada información ni usar demasiados recursos. En el caso en que se reciba un mensaje en los puertos 500 o 4500 perteneciente a una IKE_SA desconocida, es probable que sea debido a una caída reciente del nodo. Si el mensaje recibido es una respuesta, esta se debe ignorar y apuntar la situación como "sospechosa". Si lo que se recibe es una solicitud, entonces se puede contestar si se desea con un mensaje sin protección criptográfica que contenga un payload NOTIFY de tipo INVALID_IKE_SPI. Si se recibe un payload de notificación sin proteger no se debe responder ni cambiar el estado de ninguna SA existente, ya que el mensaje puede ser fruto de un ataque o haber sido modificado durante el trayecto. Se puede tomar este tipo de mensajes como una señal para iniciar un test de liveness1 sobre las IKE_SA, siempre sin gastar muchos recursos y evitando ataques de DoS.

A.16. NAT Trasversal

NAT existe principalmente por la escasez de direcciones IPv4, permitiendo a un conjunto de máquinas con direcciones IP privadas salir al exterior usando la dirección IP pública del gateway NAT. Esto se consigue haciendo una traducción de direcciones de forma transparente. Sin embargo, los protocolos que envían algún tipo de información sobre la dirección IP de los extremos dentro del cuerpo de los mensajes no funcionan bien con NAT, ya que el gateway no suele ser capaz de traducir esta información (a menos que esté especialmente programado para ello). Una comunicación IPsec plantea también una serie de problemas a la hora de ser tratada por un gateway NAT. Si se funciona en modo transporte, cambiar las direcciones IP de un paquete causa un error en la suma de comprobación dado que el gateway no puede regenerarlos porque están protegidos criptográficamente. IKEv2 permite, de forma opcional, negociar encapsulación UDP para los paquetes IP de IKE y ESP como solución a estos problemas, usando los números de puerto para decidir qué dirección local se usará en la traducción. Esto es menos eficiente pero simplifica el procesamiento que tiene que hacer el gateway NAT sobre los paquetes. Para usar este mecanismo se deben enviar los paquetes IP de IKE y ESP encapsulados hacia el puerto 4500 del destino. El gateway NAT se ocupará de cambiar el puerto origen del mismo, para recordar de quién proviene el paquete. IKE debe responder a la dirección IP y puerto indicados en el paquete recibido, de forma que el gateway NAT pueda reenviarlo a su receptor correcto. Los participantes disponen de un mecanismo para averiguar si alguno de ellos está tras un gateway NAT y, en caso de que así sea, activar este mecanismo de funcionamiento.

Apéndice B

Diseño e Implementación de OpenIKEv2

OpenIKEv2 es nuestra implementación de código abierto del protocolo IKEv2, implementada en la Universidad de Murcia. En dicha implementación hemos aplicamos tanto cuestiones de diseño de software (patrones de diseño) como de implementación (herramientas de desarrollo y depuración). En este Apéndice hemos incluido toda la información de utilidad para comprender cómo se diseñó e implementó OpenIKEv2.

B.1. Diseño de OpenIKEv2

En el diseño de OpenIKEv2 se ha considerado el paradigma orientado a objetos para aprovechar los mecanismos de herencia y polimorfismo, con el objetivo de realizar una implementación fácil y extensible. Más especificamente, se ha usado el lenguaje C++ para implementar dos librerías, *libopenikev2* y *libopenikev2_impl*, y una aplicación que las usa llamada *openikev2*.

Toda la lógica que hace funcionar el protocolo IKEv2 reside en la librería *libopenikev2*. Esta librería controla la máquina de estados, la generación y porterior lectura de mensajes, control de las asociaciones de seguridad y todo el procesamiento necesario para IKEv2. Dicha librería ha sido diseñada con la idea de ser independiente de la plataforma y del sistema operativo, donde los más específicos aspectos de implementación como son los relacionados con la red, la criptografía, hilos y control de IPsec han sido delegados a través de varias interfaces software que en este caso son implementadas por *libopenikev2_impl*. La Figura B.1 ilustra cómo interactuan ambas librerías. Gracias a esta separación se puede exportar más fácilmente la funcionalidad IKEv2 a cualquier aplicación, que lo único que tendría que hacer es usar las librerias y escuchar por el bus de eventos para conocer el comportamiento del protocolo. La librería *libopenikev2_impl* propone una implementación por defecto para cada interfaz disponible, siendo esta dependiente de la plataforma y del sistema operativo. En este caso se implementa para entornos Linux, ejecutándose sobre la arquitectura x86. La aplicación *openikev2* usa ambas librerías para utilizar toda la

funcionalidad que ofrece IKEv2. Esta aplicación se ejecuta como un servicio en segundo plano y pondrá a disposición del sistema el servicio ofrecido por IKEv2.

En las secciones siguientes se mostrará cómo las librerías *libopenikev2*, *libopenikev2_impl* y la aplicación *openikev2* interactúan entre ellas para proveer un servicio IKEv2 completo.

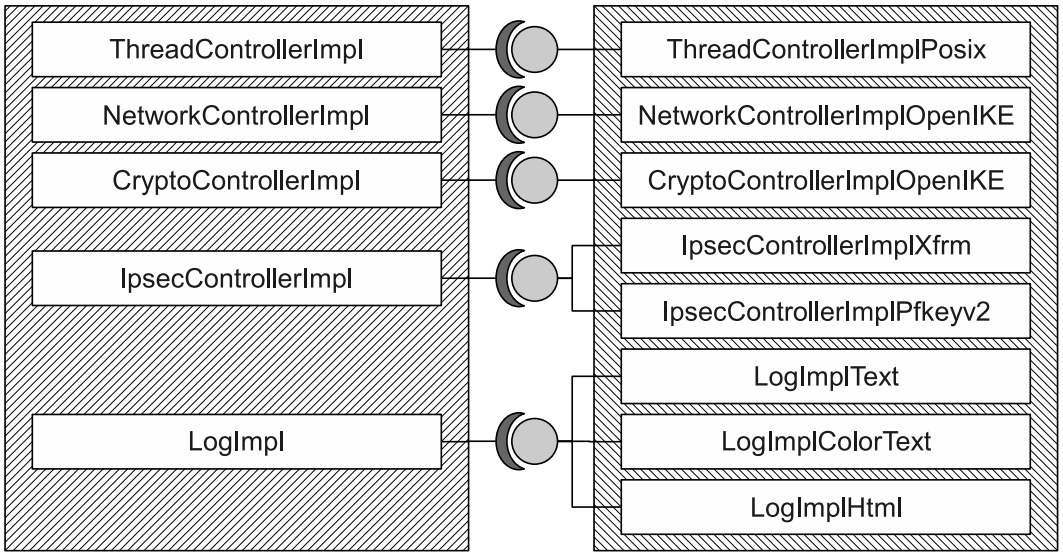


Figura B.1: Interacción entre *libopenikev2* y *libopenikev2_impl*.

B.1.1. Componentes de la Librería *libopenikev2*

La librería *libopenikev2* está compuesta de siete subsistemas, cada uno encargado de parte de la funcionalidad de IKEv2. En concreto son: *sistema de hilos*, *sistema de red*, *sistema criptográfico*, *sistema de interfaz con IPsec*, *sistema de registro de eventos*, *sistema de configuración* y *sistema de bus de eventos*. Cada uno de ellos está accesible desde cualquier parte del código por medio de una clase estática, gracias al patrón de diseño "singleton". Además, los primeros cinco subsistemas siguen también el pratrón de diseño llamado "estrategia" (mostrado en la Figura B.2). Este patrón permite cambiar dinámicamente la implementación de uno de los subsistemas mencionados y es la clave de la separación de las dos librerías.

A continuación se describe cada subsistema en detalle. Esta descripción puede ayudar a otros desarolladores como guía de cómo diseñar y desarrollar sus propias implementaciones IKEv2 a partir de la nuestra.

B.1.1.1. Subsistema de Hilos

Cada IKE SA establecida estará representada por un objeto donde se guardará el estado de la misma así como toda la información necesaria. Todas las acciones que

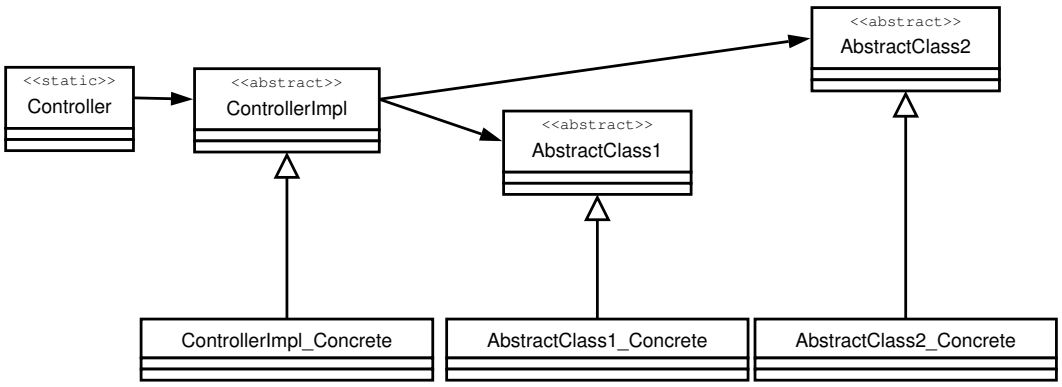


Figura B.2: Patrón de diseño "estrategia".

queramos llevar a cabo sobre una determinada IKE SA será a través de comandos. Por ello, necesita de una cola para almacenar dichos comandos y establecer así un orden de ejecución entre ellos. Existe un conjunto limitado y configurable de hilos, cada uno con capacidad de ejecutar comandos. El subsistema de hilos recorre circularmente la lista de IKE SAs disponibles y por cada una de ellas mira a ver si hay comandos pendientes de ejecutar. Si los hay, y la máquina de estados de dicha IKE SA está en el estado de reposo, se le asigna uno de estos hilos para llevar a cabo dicha ejecución. La ejecución de los hilos es concurrente, lo que nos permite tener un número indeterminado de IKE SAs ejecutándose a la vez. Cuantos más hilos configuremos, más posibilidades hay de ejecutar tareas de forma concurrente.

Este subsistema es accedido a través de la clase estática *ThreadController* (patrón de diseño "singleton"). Esta clase delega toda su funcionalidad haciendo clientela a una implementación particular a través de la clase abstracta *ThreadControllerImpl*. De esta clase abstracta heredará la clase que terminará implementando sus métodos, usando para ello las librerías necesarias. Como en nuestro caso vamos a usar los hilos del sistema Linux que siguen el estandar POSIX (*Portable Operating System Interface for Unix based systems*), decidimos llamar a esta clase *ThreadControllerImplPosix*, incluida en la librería *libopenikev2_impl*. Más aún, se requieren una serie de clases abstractas que también deben ser implementadas con el sistema elegido. Estas clases son *Mutex*, *Condition*, *Semaphore* y *Thread*. Además, existe una clase abstracta que se utiliza como interfaz llamada *Runnable* que debe ser implementada por aquellas clases que deseamos puedan ser ejecutadas por un hilo. En particular, en la librería *libopenikev2_impl* se encuentran las implementaciones de dichas clases usando de nuevo los hilos que ofrece el estandar POSIX en Linux (librería de C libpthread): *MutexPosix*, *ConditionPosix*, *SemaphorePosix* y *ThreadPosix*.

B.1.1.2. Subsistema de Red

Este subsistema es accedido a través de la clase estática *NetworkController* (patrón de diseño "singleton"). Al igual que en el subsistema anterior, esta clase delega haciendo

clientela a la clase abstracta *NetworkControllerImpl*, que implementa una de las nuevas características de IKEv2: la configuración remota de direcciones. Además hará las funciones de factoría de objetos, de cuyas clases hablamos a continuación.

Este subsistema está compuesto también por la clase abstracta *IpAddress*, que representa tanto una dirección IPv4 como IPv6. También tenemos la clase abstracta *UdpSocket*, que representa un conjunto de conexiones UDP, una para cada interfaz disponible en el sistema. Mención aparte merecen aquí las clases abstractas *EapClientController* y *EapServerController* que controlan la autenticación mediante el protocolo EAP (Extensible Authentication Protocol), otra de las nuevas características añadidas en IKEv2.

Todas las clases anteriores están localizadas en la librería *libopenikev2*. Dichas clases abstractas deben ser implementadas para poder dotar de funcionalidad al subsistema. Para ello en la librería *libopenikev2_impl* se añaden las subclases *NetworkControllerOpenIKE*, *IpAddressOpenIKE* y *UdpSocketOpenIKE*, que de nuevo usan el estandar POSIX para realizar operaciones de red en sistemas Linux.

Dependiendo del número de métodos EAP soportados por esta implementación, tendremos un par de clases por cada uno de los métodos. En principio hay implementados tres: EAP-MD5 (método sencillo), EAP-TLS(uso de certificados) y EAP-FRM (reautenticación más rápida). Por tanto sus clases asociadas serán *EapClientControllerMD5*, *EapServerControllerMD5*, *EapClientControllerTLS*, *EapServerControllerTLS*, *EapClientControllerFRM* y *EapServerControllerFRM* respectivamente. Finalmente, *NetworkControllerImplOpenIKE* implementa la asignación remota de direcciones IP usando para ello dos implementaciones distintas: una mediante asignación estática, y otra mediante el protocolo DHCP.

B.1.1.3. Subsistema Criptográfico

Este subsistema permite el acceso a funciones criptográficas tan necesarias en el protocolo IKEv2. De nuevo, se usa el patrón "singleton" para hacer este sistema accesible desde todo el código mediante la clase estática *CryptoController*. Este subsistema está compuesto además por las siguientes clases abstractas:

- *Cipher*, que se usa para encriptar y autenticar los mensajes IKEv2;
- *Random*, usada para la generación criptográficamente segura de números aleatorios;
- *DiffieHellman*, usada para generar el secreto compartido fruto de un intercambio Diffie-Hellman;
- *KeyRing*, utilizado para generar y almacenar el material criptográfico necesario para las IKE SAs y las IPsec SAs;
- *PseudoRandomFunction*, que es usada para implementar las funciones pseudo aleatorias (PRF) definidas en IKEv2; y

- *CryptoControllerImpl* que principalmente actúa como factoría de todas las clases anteriores, y es accedida a través de *CryptoController*.

Todas las clases abstractas anteriores están incluidas en la librería *libopenikev2*. Deben ser implementadas por otras subclases para poder dotar de funcionalidad criptográfica al sistema. Para ello se van a definir las siguientes clases, y en todas ellas se usará la librería criptográfica "libssl" (OpenSSL [119]) para su implementación: *CipherOpenSSL*, *RandomOpenSSL*, *DiffieHellmanOpenSSL*, *PseudoRandomFunctionOpenSSL* y *KeyRingOpenSSL*. Además también tendremos la subclase *CryptoControllerImplOpenIKE* que implementa la factoría y otras funcionalidades menores. Al igual que en otros subsistemas, todas estas clases están incluidas en la librería *libopenikev2_impl*, que como podemos deducir, ahora será también dependiente de la librería OpenSSL.

B.1.1.4. Subsistema de Interfaz con IPsec

Este subsistema se utiliza como interfaz que permite por un lado manejar la pila IPsec del sistema (crear, modificar y eliminar políticas y asociaciones de seguridad IPsec) y por otro lado recibir eventos generados por IPsec y que IKEv2 debe atender. Esta interfaz es necesaria ya que IPsec reside en el núcleo del sistema, mientras que IKEv2 reside en la capa de aplicación.

Al igual que en subsistemas anteriores, es globalmente accesible mediante la clase estática *IpsecController*, la cual delega su funcionalidad haciendo clientela a la clase abstracta *IpsecControllerImpl*, ambas clases incluidas en la librería *libopenikev2*.

En nuestra implementación OpenIKEv2 se incluyen dos formas distintas de implementar este subsistema, representadas por las siguientes subclases: *IpsecControllerImplXfrm* y *IpsecControllerImplPfkeyv2*. La primera usa la librería ***netlink*** para dialogar con el núcleo del sistema operativo a través de la interfaz ***XFRM*** y manejar la pila IPsec. Esta forma es propia de sistemas operativos basados en Linux, que tiene lo bueno de poder explotar al máximo las funcionalidades de la pila IPsec, pero no es portable a otros sistemas. La segunda usa la librería ***PFKEY v2*** [58], que es una capa intermedia que permite manejar IPsec de una forma más estandar y portable a otros sistemas. Lo malo es que, al ser más genérico, no permite explotar al máximo las funcionalidades de la pila IPsec, sino de una forma más bien básica. Esto es debido a ciertas limitaciones en las especificaciones de PFKEYv2. Ambas clases están incluidas en la librería *libopenikev2_impl*.

B.1.1.5. Subsistema de Registro de Eventos

Este subsistema premite registrar mensajes de depuración, para poder mostrar qué es lo que está ocurriendo dentro de OpenIKEv2. Existen varios tipos de mensajes de depuración, uno por cada tipo de operación (red, criptografía, generación y lectura de mensajes, transiciones de máquinas de estado, etc). Para determinar qué tipo de mensajes me interesa mostrar, se puede configurar una máscara que filtrará los tipos deseados. Este sistema, al igual que otros comentados anteriormente, son accedidos

globalmente a través de una clase estática, llamada *Log* (patrón "singleton"). Esta hace clientela a la clase abstracta *LogImpl*, delegando casi toda la funcionalidad en ella. Esta clase abstracta, incluida en la librería *libopenikev2*, será implementada por tantas subclases como formas de respresentar dichos mensajes queramos.

En la librería *libopenikev2_impl* se han implementado tres formas diferentes de mostrar los mensajes: texto plano, usando la subclase *LogImplText*; texto a color, usando la subclase *LogImplColorText* y en formato HTML usando la subclase *LogImplHtml*.

B.1.1.6. Sistema de Configuración

Este subsistema contiene toda la información necesaria para configurar un determinado escenario (establecer participantes autorizados, algoritmos criptográficos a usar, métodos de autenticación, etc.). Esta configuración puede ser establecida por medio de diferentes fuentes, como pueden ser ficheros de configuración, configuración remota, configuración estática, etc. También aquí este subsistema es accedido mediante la clase estática *Configuration*. La información de configuración se distribuye en una instancia de la clase *GeneralConfiguration* y varias instancias de la clase *PeerConfiguration*. En la primera podemos encontrar atributos cuyos valores afectan a todo el sistema IKEv2. En la segunda sólo encontraremos atributos que configuran la comunicación con cada conjunto de participantes. Además, esta clase subdivide sus atributos en dos subclases: en *IkeConfiguration* estarán todos aquellos atributos que tengan que ver con la configuración de la IKE SA, y en *IpsecConfiguration* aquellos que configuren las asociaciones IPsec SA. En la Figura B.3 podemos ver la relación entre las distintas clases.

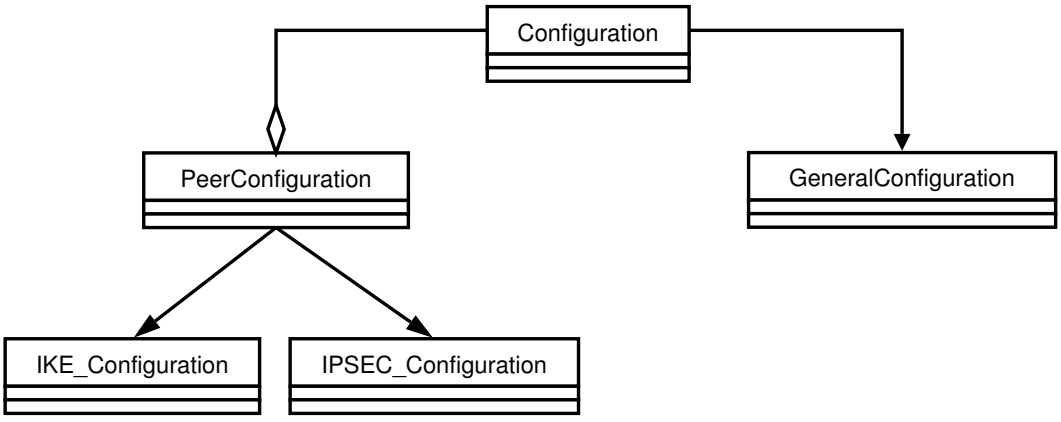


Figura B.3: Subsistema de configuración.

Existe una instancia especial de *PeerConfiguration* para participantes anónimos, que puede utilizarse en caso de no encontrarse otra configuración aplicable a un determinado participante. Si esta configuración anónima no estuviera establecida, no se premitiría la negociación IKEv2.

Para el caso de OpenIKEv2 hemos decidido obtener la configuración a través de ficheros de configuración gracias a la librería *libconfuse*, que facilita la lectura de atributos de configuración desde archivos. Esta funcionalidad de obtención de configuración esta concentrada en la aplicación *openikev2*, no en las librerías. Por tanto, la dependencia con dicha librería recae sobre la aplicación openikev2, no sobre la librería *libopenikev2_impl*.

B.1.1.7. Sistema de Bus de Eventos

Este susbsistema permite a otras aplicaciones usar la librería *libopenikev2* y recibir eventos que informen de lo que esté pasando durante las negociaciones IKEv2, como pueden ser el éxito o el fracaso de una negociación de creación, renovación o eliminado de SAs, entre otros eventos. Las aplicaciones, por tanto, usarán este subsistema para ver el resultado de sus comandos.

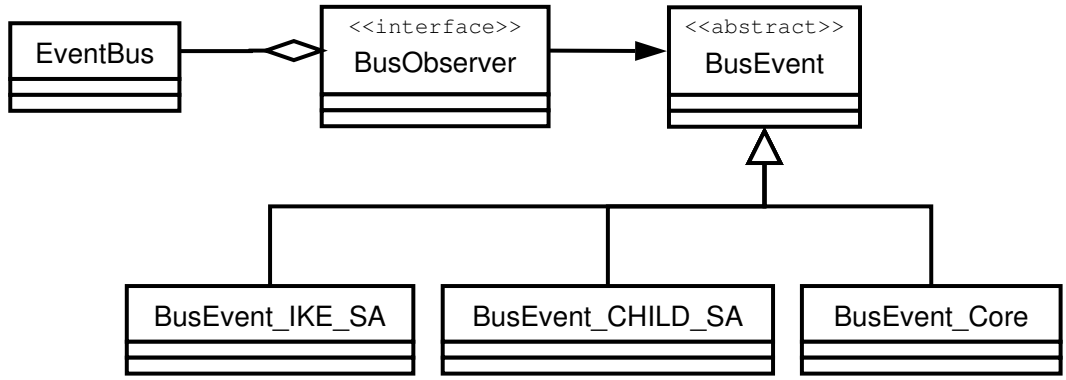


Figura B.4: Subsistema de eventos.

Este subsistema lo componen la clase *EventBus*, que se encarga de distribuir los eventos, instancias de la clase *BusEvent*. Este sistema usa el patrón de diseño "observer". Para ello, todos los objetos que quieran ser notificados por el bus de eventos, deberá implementar la clase abstracta *BusObverver* y registrarse en la instancia de *EventBus*, que mantendrá una lista de objetos interesados en sus eventos. En este registro se puede además indicar qué tipo de eventos me interesan. Hay tres tipos de eventos, cada uno implementado por cada una de estas clases: *BusEventIkeSa*, que notifica eventos relacionados con la IKE SA; *BusEventChildSa*, eventos relacionados con las IPsec SAs y *BusEventCore*, que informa sobre el resto de posibles eventos. En la Figura B.4 se muestra la estructura de este subsistema.

B.2. Implementación de OpenIKEv2

En esta sección se muestran los detalles de implementación de OpenIKEv2 más relevantes.

B.2.1. Herramientas de Trabajo

Ya que hemos seguido un diseño orientado a objetos, hemos seleccionado un lenguaje de programación orientado a objetos para poder sacar partido a los mecanismos de herecia y polimorfismo. El lenguaje seleccionado ha sido C++. Este es un lenguaje muy extendido que nos permite usar tanto el paradigma orientado a objetos como el imperativo, que ofrece la ventaja a su vez de usar librerías desarrolladas en C, como puede ser la librería criptográfica OpenSSL o todas las librerías del núcleo de Linux. Por otro lado, es un lenguaje compilado generando binarios que se ejecutan a gran velocidad, tan necesaria en entornos de producción. También posee mecanismos para tener un control más cuidadoso de la memoria dinámica usada. Uno de estos mecanismo son los llamados *smart pointers*, que son objetos que almacenan en su interior un puntero. Estos objetos cuidan del acceso a ese puntero, aportando funcionalidades adicionales como la verificación de acceso a NULL y la liberación automática de la memoria asignada a dicho puntero. Esto ayuda enormemente a reducir problemas con la memoria utilizada, como es el caso del goteo de memoria, que hace que en una aplicación que se está ejecutando durante largos periodos de tiempo termine, tarde o temprano, consumiendo toda la memoria del sistema. En OpenIKEv2, adicionalmente se ha luchado contra este problema de goteo usando las herramientas que ofrece *valgrind* [120].

Para desarrollar OpenIKEv2 se ha elegido Linux como sistema operativo, ya que numerosos proyectos de investigación europeos suelen requerir sistemas operativos de software libre. Además, siguiendo con la filosofía de software libre, se han utilizado herramientas de desarrollo que también la siguen, como:

- Desarrollo software: *gcc, make, autotools.*
- Entorno de programación: *kdevelop3.*
- Depuración: *gdb, valgrind, dmalloc, electric-fence.*
- Documentación: *doxygen.*
- Repositorio y Control de versiones: *subversion.*

Apéndice C

Herramientas para la Toma y Procesamiento de Datos

Para realizar las pruebas donde valoraremos el rendimiento de las comunicaciones no sólo vale con tener la propia infraestructura necesaria para desplegar una red vehicular, sino que también tenemos que ser capaces de probarla, y para ello necesitamos de herramientas para generar y capturar tráfico, obtener estadísticas, generar gráficas, etc. En este apéndice veremos las herramientas, tanto propias como de terceros, que hemos utilizado para tal fin.

C.1. Entorno para las Herramientas de Pruebas

Para realizar todas las fases que las pruebas requieren, hemos creado un entorno bajo sistemas Linux compatibles que parte de un directorio "raíz" que incluye una serie de subdirectorios como vemos en la Figura C.1, cada uno con un nombre y un fin en particular dentro de este proceso de generación, captura y procesamiento de los datos. A continuación vamos a enumerarlos agrupándolos por las propias fases de las que se compone una prueba.

C.1.1. Generación y Captura de Datos

La generación del tráfico que se usará para analizarlo posteriormente y deducir resultados necesitamos una serie de programas y *scripts* que deben ser ejecutados en cada entidad involucrada en la prueba. Por tanto, por cada una de estas entidades existe un directorio donde irán dichos programas y *scripts*. Esto son los directorios "MR" para el *Mobile Router*, "AR" para el *Home Agent* o un *Router* de acceso situado en la infraestructura, y "HOST" para un equipo conectado a la red móvil del *Mobile Router*, que en nuestro caso es un portátil con Linux. Empezamos con lo que nos encontramos en el directorio "MR".

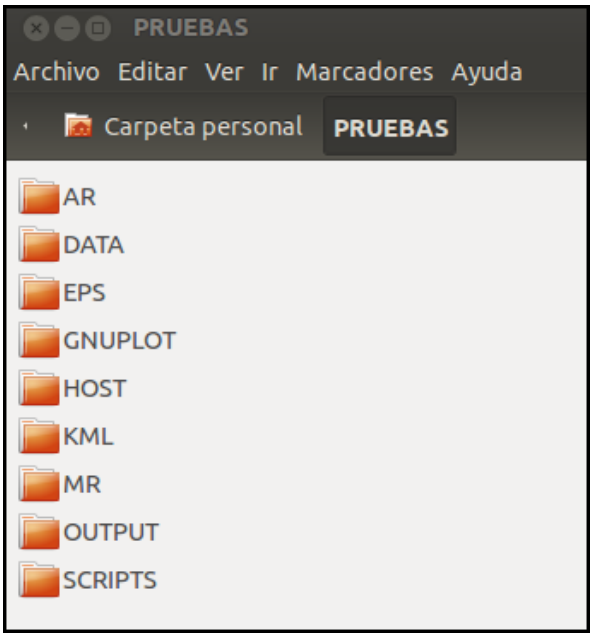


Figura C.1: Estructura de directorios del entorno de pruebas

C.1.1.1. Subdirectorio MR

Antes de empezar cualquier prueba, hay que sincronizar los relojes de todas las entidades que intervienen en la prueba, pues de ello depende que los resultados sean fiables. Para ello, en este directorio tenemos dos *scripts*:

- **SYNCRO_DATE.sh**: Nos permite sincronizar el reloj del sistema al marcado por el dispositivo GPS que el *Mobile Router* lleva incorporado. El reloj del GPS se supone exacto.

```
gpsd /dev/ttyS2
sleep 5
( echo '?WATCH={"enable":true, "json":true}'; sleep 5 ) |
        telnet ::1 2947 | grep TPV  | cut -d ',' -f4 |
        cut -d ':' -f2 | cut -d '.' -f1 > temp

DATE=$(tail -1 temp)
rm temp

echo System clock:
date -u +%s
date -u

echo GPS clock:
echo $DATE
date -d @${DATE}
```

```
echo
echo Setting system clock to GPS clock...
date -u -s @${DATE}

DATE_OK=$(echo $?)
if [ $DATE_OK -ne 0 ]; then
    echo Date is still missing in GPS. Please, try again later.
    exit 1
fi

hwclock -w
```

- **SHOW_DATE.sh**: Muestra en pantalla la hora, una vez por segundo, para verificar que todos los dispositivos están sincronizados.

```
while [ 1 ]; do date +%s; sleep 1; done
```

Después tenemos otra serie de *scripts* para la obtención de datos, como son:

- **GPS.sh**: Obtiene los datos relativos a la posición del vehículo en cada momento. Toda la información obtenida del GPS es volcada en un fichero, donde ya se incluyen marcas de tiempo.

```
FILENAME="GPS.txt"
DIRECTORY="/opt"
TEST_ID=$1

gpsd /dev/ttyS2

sleep 5

( echo '?WATCH={"enable":true, "json":true}'; sleep 10000 ) |
       telnet ::1 2947 >> ${DIRECTORY}/DATA/${TEST_ID}/${FILENAME} |
       tail -f ${DIRECTORY}/DATA/${TEST_ID}/${FILENAME}
```

- **SIGNAL.sh**: Obtiene la fuerza de la señal 802.11p en cada instante, guardándola en un fichero junto a marcas de tiempo.

```
FILENAME="SIGNAL.txt"
DIRECTORY="/opt"
TEST_ID=$1
MAC=$2
OUTPUT=${DIRECTORY}/DATA/${TEST_ID}/${MAC}_${FILENAME}
COUNT=1

while [[ 1 ]]; do
echo -n "$COUNT " | tee -a ${OUTPUT} ;
             date -u +%s | head -c 10 | tee -a ${OUTPUT};
             echo -n " " | tee -a ${OUTPUT} ;
```

```
            cmg rssi g5d0 2> /dev/null | grep ${MAC} | tr -s " " |
                  cut -d " " -f6 | tee -a ${OUTPUT}
sleep 1
    COUNT=$(expr $COUNT \+ 1)
done
```

- **DUMP.sh**: Ejecuta capturadores de tráfico (*tcpdump*) por las interfaces 3G y 802.11p. Todo el tráfico es capturado y volcado en sus respectivos ficheros, uno por interfaz, incluyendo también marcas de tiempo.

```
FILENAME="DUMP_$2.pcap"
DIRECTORY="/opt"
TEST_ID=$1
IFACE=$2
CONTADOR=1

DUMP_FILE=${DIRECTORY}/DATA/${TEST_ID}/${CONTADOR}_${FILENAME}

/usr/sbin/tcpdump -s0 -n -i $IFACE -tt -w $DUMP_FILE -f ip6
```

Para utilizar todos estos *scripts* de una forma más automatizada en cada una de las pruebas, también disponemos de otros que nos ayudan a controlar este proceso, iniciar la captura, ver el estado y parar cuando haga falta dicha captura de datos:

- **START.sh <id_prueba>**: Ejecuta los *scripts* GPS.sh, SIGNAL.sh y un DUMP.sh por cada interfaz disponible. Recibe como argumento un identificador de la prueba, que se usará para guardar los datos obtenidos en el directorio "DATA", en un subdirectorio llamado como dicho identificador de la prueba.

```
control_c()
{
   echo "Exiting..."
   killall telnet
   killall gpsd
   killall tail
}

trap control_c SIGINT

./SIGNAL.sh $1 8b &
./DUMP.sh $1 g5d0 &
./DUMP.sh $1 tap0 &
./GPS.sh $1
```

- **STATUS.sh**: Muestra el estado actual de la prueba y la salida que están produciendo los *scripts* anteriores. Es una forma de verificar que todo está funcionando correctamente.

```
while [ 1 ]; do
   clear
   ifconfig | grep inet | grep -v fe80 | grep -v "192.168" |
          grep -v "127.0" | grep -v "::1/128"
   ping -c 1 155.54.95.98
   ping -c 1 10.8.0.1
   sleep 5
done
```

- **STOP.sh**: Para las capturas en marcha de forma controlada, eliminando de memoria cualquier proceso involucrado en la prueba, dejando el sistema listo para una nueva prueba.

```
echo "Exiting..."
killall telnet
killall gpsd
killall tail
```

Con todos estos *scripts* tenemos preparado el *Mobile Router* para registrar cualquier actividad que se produzca a través de sus interfaces de comunicaciones, siendo posible correlacionar todos estos hechos con el lugar y el tiempo, pudiendo saber así, dónde y cuándo se han producido.

C.1.1.2. Subdirectorio AR

Este directorio incluye los *scripts* necesarios para realizar las pruebas que hagan uso del tráfico UDP y TCP, ya que necesitan de un cliente y un servidor para funcionar. Estos archivos deberán ser copiados a un equipo dentro de la Estación Central ITS, que podrá ser cualquier *router* de acceso o el propio *Home Agent*. En nuestro caso, en este lado se va a actuar como servidor en el caso de TCP y como cliente en el caso de UDP. Esto será al contrario en el HOST, el otro extremo de esta comunicación. Por tanto los ficheros que tenemos son:

- **SYNCRO_DATE.sh**: Nos permite sincronizar el reloj del sistema con el del MR. En este caso se deberá introducir a mano la marca de tiempo que esté mostrando **SHOW_DATE.sh** en el MR.

- **SHOW_DATE.sh**: Muestra en pantalla la hora, una vez por segundo, para verificar que todos los dispositivos están sincronizados.

- **TCP.sh <id_prueba>**: Inicia la prueba TCP, quedando a la espera la recepción del tráfico procedente del HOST, conectado a la red móvil del MR. Los datos son almacenados dentro de un directorio llamado "DATA", en un subdirectorio llamado como el identificador de la prueba. Esta prueba usa el programa *iperf* para recibir el tráfico TCP.

- **UDP.sh <id_prueba> <tasa_de_transferencia>**: Inicia la prueba UDP, enviando tráfico UDP a una determinada tasa de transferencia pasada por argumento. Los valores que hemos utilizado para las pruebas han sido de 500.000, 1.000.000 y 2.000.000 bits/segundo. Al igual que en TCP, la herramienta *iperf* es la encargada de generar el tráfico UDP. Los resultados son guardados en el directorio "DATA", en un subdirectorio con nombre igual al identificador de la prueba.

Todas estas pruebas se detienen con un simple **CTRL+C**.

C.1.1.3. Subdirectorio HOST

En este subdirectorio se encuentran los *scripts* necesarios para la ejecución de las pruebas en la parte del HOST, conectado a la red móvil del MR. Necesitaremos sincronizar igualmente el reloj del sistema con la hora obtenida en el MR, y también generar o recibir tráfico dependiendo del caso. Para el tráfico UDP se actúa aquí de servidor y, por tanto, receptor del tráfico UDP, al contrario que TCP, que en este caso actúa de cliente y generador de tráfico TCP. Aquí también habrá un *script* específico para la prueba de RTT con tráfico ICMPv6, haciendo uso de la herramienta *ping*. Enumeramos pues los *scripts* presentes en este subdirectorio:

- **SYNCRO_DATE.sh**: Nos permite sincronizar el reloj del sistema con el del MR. En este caso se deberá introducir a mano la marca de tiempo que esté mostrando **SHOW_DATE.sh** en el MR.
- **SHOW_DATE.sh**: Muestra en pantalla la hora, una vez por segundo, para verificar que todos los dispositivos están sincronizados.
- **RTT.sh <id_prueba>**: Inicia la prueba ICMPv6, gracias a la herramienta *ping*, para estudiar el retardo del envío de un paquete entre el HOST y el *Home Agent*, o también llamado RTT (*Round-Trip Time*). Los resultados son guardados en el directorio "DATA", en un subdirectorio con nombre igual al identificador de la prueba.
- **TCP.sh <id_prueba>**: Inicia la prueba TCP, generando tráfico TCP dirigido al *router* de acceso elegido o al propio *Home Agent*. Esta prueba usa el programa *iperf* para generar el tráfico TCP. Los resultados son guardados en el directorio "DATA", en un subdirectorio con nombre igual al identificador de la prueba.
- **UDP.sh <id_prueba>**: Inicia la prueba UDP, quedando a la espera de recibir tráfico UDP a una determinada tasa de transferencia establecida en el otro extremo. Al igual que en TCP, la herramienta *iperf* es la encargada de recibir el tráfico UDP. Los resultados son guardados en el directorio "DATA", en un subdirectorio con nombre igual al identificador de la prueba.

Todas estas pruebas se detienen con un simple **CTRL+C**.

C.1.2. Procesamiento de los Datos Obtenidos

Después de realizadas las pruebas y obtenidos los datos, es el momento de procesarlos. Para ello disponemos de otros subdirectorios destinados a este proceso.

C.1.2.1. Subdirectorio DATA

El directorio "DATA" es fundamental ya que acumula todos los datos obtenidos por todas las entidades. Como se muestra en la Figura C.2, todos los datos van a parar a dicha carpeta pero en equipos diferentes, lo que implica que antes de iniciar el procesado de los datos hay que unificar todo el contenido en sólo uno de ellos, en nuestro caso en el HOST, pues dispone de librerías y herramientas necesarias para el procesado de los datos y la generación de representaciones gráficas, como *perl* o *gnuplot*.

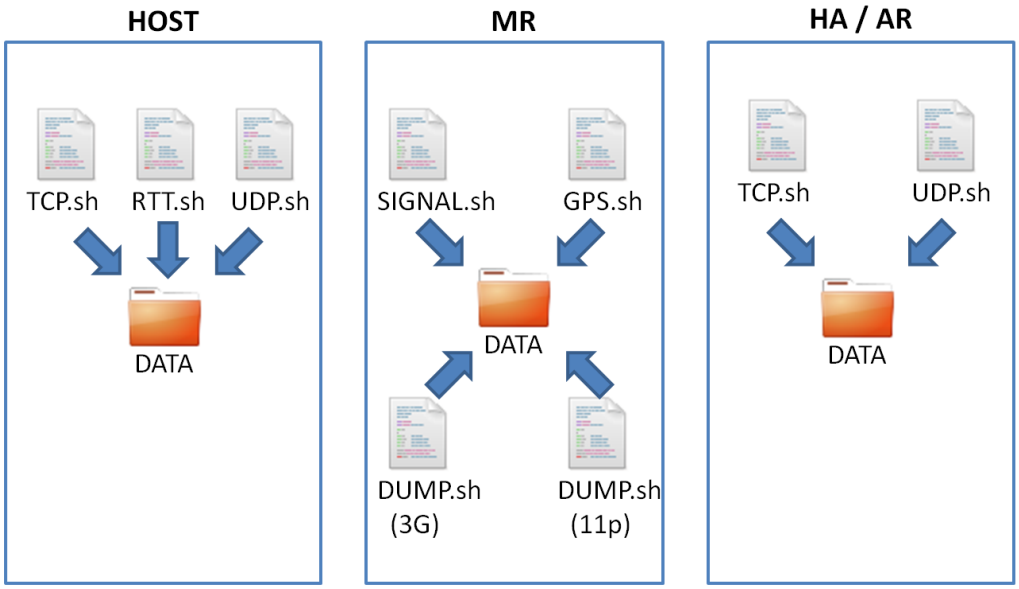


Figura C.2: Origen de los datos obtenidos en las pruebas

Así que antes de empezar a probar, dentro del subdirectorio "DATA", y dentro de ella en otro subdirectorio con nombre el identificador de la prueba, encontraremos los siguientes ficheros:

- **1_DUMP_g5d0.pcap**: Este es el archivo de captura de paquetes de la interfaz 11p creada por *tcpdump* en el MR. Si la captura se interrumpiera y se reanudara de nuevo, se crearía otro archivo pero empezando por 2, y así sucesivamente, evitando sobrescribir capturas y poderlas parar y reanudar sin problemas.
- **1_DUMP_tap0.pcap**: Este es el archivo de captura de paquetes de la interfaz OpenVPN, que a su vez va por la interfaz 3G. Es creada también con *tcpdump*

en el MR. Para evitar sobrescribir capturas, también soporta el poderlas parar y reanudar sin problemas.

- **XX_SIGNAL.txt**: Este archivo contiene los niveles de señal de una estación base 802.11p, y para dar soporte a poder registrar el nivel de señal de más de una estación base, se le añade al principio del nombre del archivo los dos últimos dígitos hexadecimales de la dirección MAC de cada estación base. Así habrá tantos archivos de señal como estaciones bases haya. En nuestro caso sólo tenemos una estación base 802.11p.

```
21 1380724042 -93
22 1380724043 -86
23 1380724045 -83
24 1380724046 -87
25 1380724048 -88
26 1380724049 -91
27 1380724050 -90
28 1380724052 -88
29 1380724053 -91
30 1380724055 -87
```

- **GPS.txt**: En este archivo están registradas las posiciones del vehículo, su velocidad, altitud, etc. durante el transcurso de la prueba. Está en formato *json* para poder ser procesado con más facilidad. Es generado por el servicio *gpsd* que ejecuta el MR.

```
{"class":"VERSION","release":"2.94","rev":"2013-06-26T14:53:43",
        "proto_major":3,"proto_minor":2}
{"class":"DEVICES","devices":[{"class":"DEVICE","path":"/dev/ttyS2",
        "activated":1380724020.26,"native":0,"bps":9600,"parity":"N",
        "stopbits":1,"cycle":1.00}]}
{"class":"WATCH","enable":true,"json":true,"nmea":false,"raw":0,
         "scaled":false,"timing":false}
{"class":"TPV","tag":"GGA","device":"/dev/ttyS2",
         "lat":38.023176667,"lon":-1.174960000,"alt":127.000,"mode":3}
{"class":"TPV","tag":"GSA","device":"/dev/ttyS2",
         "lat":38.023176667,"lon":-1.174960000,"alt":127.000,"epv":6.785,"mode":3}
...
```

- **RTT.txt**: Valores de RTT obtenidos en la prueba, tal y como los registra la aplicación *ping*.

```
1380724779 64 bytes from 2001:720:1710:3::1000: icmp_seq=1 ttl=63 time=374 ms
1380724780 64 bytes from 2001:720:1710:3::1000: icmp_seq=1 ttl=63 time=196 ms
1380724781 64 bytes from 2001:720:1710:3::1000: icmp_seq=1 ttl=63 time=94.8 ms
1380724782 64 bytes from 2001:720:1710:3::1000: icmp_seq=1 ttl=63 time=159 ms
1380724783 64 bytes from 2001:720:1710:3::1000: icmp_seq=1 ttl=63 time=7.77 ms
1380724785 64 bytes from 2001:720:1710:3::1000: icmp_seq=1 ttl=63 time=6.15 ms
1380724786 64 bytes from 2001:720:1710:3::1000: icmp_seq=1 ttl=63 time=7.59 ms
1380724787 64 bytes from 2001:720:1710:3::1000: icmp_seq=1 ttl=63 time=9.24 ms
...
```

- **UDP_XXXXXXX.txt**: Valores ofrecidos por *iperf* en la prueba con tráfico UDP, donde XXXXXXX es la tasa de transferencia usada en bits/segundo. Para nuestro caso tendremos tres archivos: a 500.000, 1.000.000 y 2.000.000 bits/s. De estas pruebas se extrae el valor porcentual de PDR.

```
------------------------------------------------------------
Server listening on UDP port 5001
Binding to local address 2001:720:1710:6:7a92:9cff:fe0a:3f4a
Receiving 1470 byte datagrams
UDP buffer size: 180224 Byte (default)
------------------------------------------------------------
[  3] local 2001:720:1710:6:7a92:9cff:fe0a:3f4a port 5001 connected with 2001:720:1710:95::
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0- 1.0 sec  63960 Bytes  511680 bits/sec  10.090 ms  919/  971 (95%)
[  3]  1.0- 2.0 sec  60270 Bytes  482160 bits/sec  19.240 ms    0/   49 (0%)
[  3]  2.0- 3.0 sec  57810 Bytes  462480 bits/sec  12.645 ms    0/   47 (0%)
[  3]  3.0- 4.0 sec  67650 Bytes  541200 bits/sec  17.924 ms    0/   55 (0%)
...
```

- **TCP.txt**: Valores obtenidos en la prueba con tráfico TCP, tal y como los registra la aplicación *iperf*. De esta prueba se obtiene el valor de ancho de banda máximo en cada momento usando TCP.

```
------------------------------------------------------------
Client connecting to 2001:720:1710:95:f66d:4ff:fe99:d334, TCP port 5001
TCP window size: 19820 Byte (default)
------------------------------------------------------------
[  3] local 2001:720:1710:6:2532:607d:aa37:7d2 port 53787 connected with 2001:720:1710:95::
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  131072 Bytes  1048576 bits/sec
[  3]  1.0- 2.0 sec  131072 Bytes  1048576 bits/sec
[  3]  2.0- 3.0 sec  131072 Bytes  1048576 bits/sec
[  3]  3.0- 4.0 sec  131072 Bytes  1048576 bits/sec
[  3]  4.0- 5.0 sec  131072 Bytes  1048576 bits/sec
...
```

Todos estos archivos contienen los datos que nos permitirán elaborar estadísticas y obtener resultados que nos servirán para llegar a conclusiones que nos ayuden en nuestro proceso investigador. Para ello entra en juego el siguiente subdirectorio "SCRIPTS".

C.1.2.2. Subdirectorio SCRIPTS

En este subdirectorio se encuentran, como su nombre indica, todos los *scrips* necesarios para procesar toda la información de una determinada prueba. Entre estos destaca uno que es el que tenemos que ejecutar para lanzar el proceso sobre una determinada prueba que le debemos indicar por argumento mediante su identificador. Empezando por este, se enumeran a continuación los *scripts* que contiene este subdirectorio:

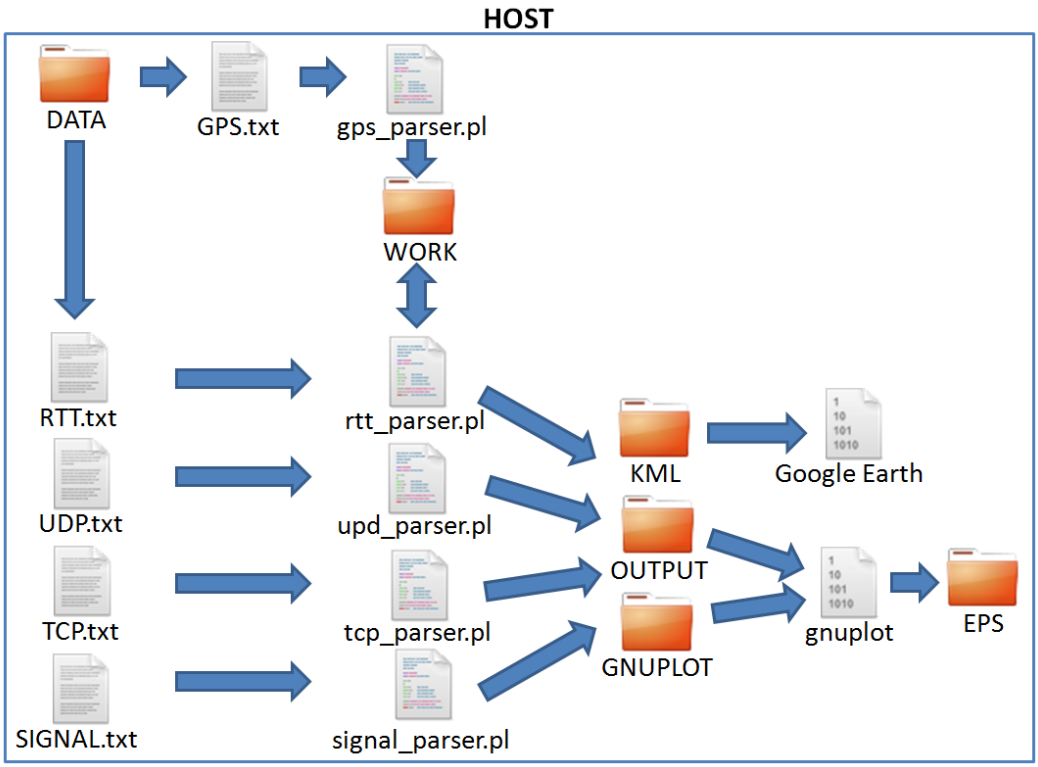


Figura C.3: Flujo de trabajo en el procesado de los datos

- **make_test.sh <id_prueba>**: Este *script* es el encargado de llamar en el orden adecuado a los demás *scripts* que se irán encargando de cada parte del procesado de los datos y de la elaboración de estadísticas y representaciones gráficas como mostramos en la Figura C.3. Este mismo orden voy a utilizar a continuación para seguir enumerando *scripts*.

- **directory_maker.sh <id_prueba>**: Este *script* es el encargado de crear la estructura de directorios necesaria para albergar los ficheros resultantes del procesado de datos. Dentro de cada directorio involucrado, se creará un subdirectorio con el identificador de la prueba donde quedarán recogidos los ficheros resultantes de dicha prueba.

- **gps_parser.pl <id_prueba>**: Este *script* escrito en *perl* se encarga de obtener las posiciones geográficas del recorrido del vehículo en la prueba indicada y se asocia cada posición a una marca de tiempo. El fichero resultante va a parar a la carpeta "WORK" ya que será utilizado por otros *scripts*.

- **rtt_parser.pl <id_prueba>**: Se encarga de procesar los datos de las pruebas con tráfico ICMPv6 para encontrar los valores de RTT, generando un fichero con los valores de RTT obtenidos asociándolos a una marca de tiempo y a una posición

geográfica, y otro con estadísticas, ambos en el subdirectorio "OUTPUT". También se generan *scripts* de *gnuplot* en el subdirectorio "GNUPLOT", que harán uso de los ficheros guardados en "OUTPUT" para generar gráficas en formato *post-script encapsulado* (EPS) en el directorio "EPS". Además, para poder ver estos datos en un entorno de tres dimensiones gracias a la aplicación Google Earth, generamos ficheros en formato KML en el subdirectorio "KML" para representar la misma información sobre el plano terrestre, centrándonos así en la posición geográfica de los datos.

- **udp_parser.pl <id_prueba>**: Se encarga de procesar los datos de las pruebas con tráfico UDP para encontrar los valores porcentuales de PDR y la tasa de transferencia alcanzada en cada momento, generando en primer lugar un fichero con dichos valores obtenidos asociándolos a una marca de tiempo y a una posición geográfica, y en segundo lugar otro con estadísticas, ambos en el subdirectorio "OUTPUT". También se generan *scripts* de *gnuplot* en el subdirectorio "GNUPLOT", que harán uso de los ficheros guardados en "OUTPUT" para generar gráficas en formato *post-script encapsulado* (EPS) en el directorio "EPS". Además, para poder ver estos datos en un entorno de tres dimensiones gracias a la aplicación Google Earth, generamos ficheros en formato KML en el subdirectorio "KML" para representar la misma información sobre el plano terrestre, centrándonos así en la posición geográfica de los datos, como vemos en un ejemplo en la Figura C.4.

Figura C.4: Representación gráfica del PDR en el plano terrestre

- **tcp_parser.pl <id_prueba>**: Al igual que en los dos anteriores, también se encarga de procesar los datos de las pruebas, pero esta vez con tráfico TCP, para encontrar los valores de ancho de banda máximo alcanzados en cada momento, generando en primer lugar un fichero con dichos valores obtenidos asociándolos a una marca de tiempo y a una posición geográfica, y en segundo lugar otro con estadísticas, ambos en el subdirectorio "OUTPUT". También se generan *scripts* de *gnuplot* en el subdirectorio "GNUPLOT", que harán uso de los ficheros guardados en "OUTPUT" para generar gráficas en formato *post-script encapsulado* (EPS) en el directorio "EPS". Además, para poder ver estos datos en un entorno de tres dimensiones gracias a la aplicación Google Earth, generamos ficheros en formato KML en el subdirectorio "KML" para representar la misma información sobre el plano terrestre, centrándonos así en la posición geográfica de los datos.

- **signal_parser.pl <id_prueba>**: Se encarga de obtener los valores de fuerza de la señal 802.11p y asociarlos a una marca de tiempo y una posición geográfica, resultando un fichero que se guarda en "OUTPUT". Como la captura de la fuerza de señal se hace a lo largo de las diferentes pruebas, debemos obtener el periodo de tiempo de estas muestras para cada prueba. Por eso, para cada una de ellas se ha guardado una marca de tiempo inicial y final. Teniendo en cuenta estas marcas, se genera una representación gráfica para cada prueba mediante *gnuplot* que se guardará en "GNUPLOT" y posteriormente resultando en un fichero en formato EPS en el subdirectorio "EPS", además de un fichero KML para poderlo representar sobre un plano en 3D en el directorio "KML".

Algunas de estas gráficas las hemos mostrado a lo largo de la Tesis, sobre todo en el capítulo de resultados.