UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INFORMÁTICOS

**Sistema de comunicaciones V2X para vehículos
inteligentes como soporte para sistemas
cooperativos**

TESIS DOCTORAL

JOSÉ JAVIER ANAYA CATALÁN
INGENIERO INDUSTRIAL

Madrid, 27 de mayo de 2016

DEPARTAMENTO DE INTELIGENCIA ARTIFICIAL

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INFORMÁTICOS

Sistema de comunicaciones V2X para vehículos inteligentes como soporte para sistemas cooperativos

AUTOR: JOSÉ JAVIER ANAYA CATALÁN
INGENIERO INDUSTRIAL

DIRECTOR: JOSÉ EUGENIO NARANJO
DOCTOR INGENIERO INFORMÁTICO

Memoria para optar al título de:
Doctor Ingeniero Informático
por la
Universidad Politécnica de Madrid

Madrid, 27 de mayo de 2016

Tribunal nombrado por el Magnifico y Excelentísimo Sr. Rector de la Universidad Politécnica de Madrid, el día ____ de _____ de 20 ____.

Presidente: _____

Secretario: _____

Vocal: _____

Vocal: _____

Vocal: _____

Realizado el acto de defensa y lectura de la Tesis: "Sistema de comunicaciones V2X para vehículos inteligentes como soporte para sistemas cooperativos". En _____, el día ____ de _____ de 20 ____ en la Escuela Técnica Superior de Ingenieros Informáticos, este Tribunal decide otorgar la calificación de _____.

EL PRESIDENTE:

EL SECRETARIO:

_____

LOS VOCALES:

_____

_____

_____

Declaración de Autoridad

Yo, don José Javier ANAYA CATALÁN, declaro que esta tesis titulada: 'Sistema de comunicaciones V2X para vehículos inteligentes como soporte para sistemas cooperativos' y el trabajo que se presenta en ella son míos. También confirmo que:

- Este trabajo fue realizado total o parcialmente, en el transcurso de una beca de investigación en esta universidad.
- Cualquier parte de esta tesis no ha sido previamente sometida a un título o cualquier otra calificación en esta Universidad o cualquier otra institución.
- Todas las reseñas utilizadas en este texto están claramente referenciadas al autor o autores. Con la excepción de estas citas, esta tesis es enteramente trabajo propio.
- He reconocido todas las fuentes de ayuda.
- Cuando los resultados de la presente tesis se basan en el trabajo realizado por la colaboración con otros miembros de la comunidad investigadora he dejado explícitamente indicadas las tareas desarrolladas en el transcurso de la investigación.

En _____, a día ____ de _____ de 20 ____.

Firma:

Agradecimientos

Quisiera expresar mi más sincero agradecimiento a todas las personas que me han permitido la realización de esta tesis doctoral.

A José Eugenio Naranjo, por su apoyo y valiosos consejos durante toda la tutorización de esta tesis.

A mis compañeros que me han acompañado durante la realización de esta tesis, especialmente a Edgar, Óscar y Alberto por aguantar mis sugerencias, ideas locas, frustraciones y ayudarme a desarrollarla con sus consejos e indicaciones.

A mis padres, familia y amigos por su apoyo constante e incondicional.

Al Ministerio de Economía y Competitividad a través de las ayudas para contratos predoctorales para la formación de doctores.

A los proyectos CICYT iVANET - Comunicaciones en malla para vehículos e infraestructuras inteligentes (CICYT TRA2010-15645) y ADAS-ROAD sistema avanzado de asistencia a la conducción para entornos interurbanos y sistemas de comunicación, modelado y actuación (CICYT Retos. TRA2013-48314-C3-2-R) que han contribuido a la financiación para la realización de esta tesis.

Abstract

In the socioeconomic development of the European Union, the cooperative intelligent transport systems (C-ITS) are targeted towards reducing accidents, improving environmental quality (energy conservation and pollution reduction), increasing efficiency of transportation, increasing the capacity of the current infrastructure and business productivity. These C-ITS are part of the 2020 plan of the European Union for Smart Cities. A previous step to getting the cooperative applications is to create VANETs (Vehicular Area Networks) and for that the inter-vehicle communications (V2V) and vehicle to infrastructure (V2I) are necessary, generically known as V2X. However, there is currently no commercially available technology or any hardware or software to support these communications in the vehicular environment, leaving the equipment available to the restricted use of automakers and electronics companies, exclusively. Therefore, the implementation of this technology is far from widespread use, especially oriented towards the field of research, in development projects of cooperative systems and even autonomous vehicles.

Consequently, this thesis proposes the development of hardware and control software, that is capable of providing services to communications V2X, following European and international standards in the field.

Once implemented the core system, the proper functioning of this communications technology has been validated through implementation and commissioning of four cooperative systems involving the three fundamental elements of the road: the vehicle, route and users.

This thesis has been developed in collaboration with the research group *Intelligent Systems for Mobility and Communication Accessible (SIMCA)* belonging to *Department of Artificial Intelligence* of the *School of Computer Engineers* and the *University Automobile Research Institute (INSIA)* both belonging to *Technical University of Madrid*.

In order to achieve this, the most advanced technology available at that time has been used: Two completely different communication technologies have been applied, WSA (Wireless Sensor and Actuator Network) and DSRC (Dedicated Short Range Communication), which follow different communication standards.

At first a novel routing algorithm was developed with the WSA technology, based on geolocation and creating a VANET type collection data tree. Once demonstrated the ability to create a sufficiently robust and reliable VANET a cooperative system has been developed for the avoidance of collisions with pedestrians.

Later, with the release of the European standards for inter-vehicle communications, through the European Telecommunications Standards Institute (ETSI), the European

DSRC technology (ITS-G5) has undergone an important step forward to create a VANET with this technology, the DSRC module UPM-INSIA has been designed as part of this thesis, with the ability of perform V2X communications following European Standards to guarantee the interoperability. In order to do so, intervehicular communication standards have been analyzed and developed a software that meets the specifications to provide geocommunication. During the implementation of the standards different lacks and errors of the current delivered version have been detected which have been solved.

Together with the DSRC-UPM-INSIA modules, a set of cooperative ADAS applications have also been developed to demonstrate their functionalities and capacities. In this thesis three cooperative systems have been developed with the DSRC technology that tackle different problems:

- Pedestrian warning system for urban and interurban roads, using V2V and V2P communications.
- Bike rider presence warning system using V2V and V2P communications.
- Motorcycle warning system for road with poor visibility, including blind spot warning, using V2V communications

The cooperative system based on WSA and three cooperative systems based DSRC presented in this thesis have been tested in controlled conditions and real traffic situations. The systems have obtained optimal results.

Resumen

En el desarrollo socioeconómico de la Unión Europea se contemplan los sistemas inteligentes de transporte cooperativos (C-ITS) como un medio para reducir la siniestralidad, mejorar la calidad medioambiental (ahorro de energía y reducción de la contaminación), aumento de la eficiencia del transporte, de la capacidad de la infraestructura existente y de la productividad de las empresas, especialmente las de transporte. Estos C-ITS están englobados en el plan 2020 de la Unión Europea sobre las Smart Cities. Un paso previo a conseguir las aplicaciones cooperativas es la puesta en servicio de VANETs (Vehicular Area NETWORKs) y para ello son necesarias las comunicaciones intervehiculares (V2V) y con la infraestructura (V2I), denominadas genéricamente como V2X. Sin embargo, en la actualidad no existe disponible comercialmente ningún tipo de tecnología ni hardware ni software para dar soporte a estas comunicaciones en el entorno vehicular, quedando los equipos disponibles al uso restringido de los fabricantes de automóviles o empresas de electrónica, por lo que la implantación de esta tecnología queda muy lejos de un uso masivo orientándose sobre todo hacia el ámbito de la investigación, en proyectos de desarrollo de sistemas cooperativos e, incluso, de vehículos autónomos.

Por tanto, en esta tesis doctoral se ha planteado el desarrollo de un hardware de comunicaciones y su software de control. Este debe ser capaz de dar servicios a comunicaciones V2X, siguiendo los estándares Europeos e internacionales en el ámbito y todos los elementos que caracterizan esta tecnología.

Asimismo, el correcto funcionamiento de esta tecnología de comunicaciones ha sido validado mediante la implementación y puesta en servicio de cuatro sistemas cooperativos que involucran a los tres elementos fundamentales de la carretera: el vehículo, la vía y los usuarios.

Para ello se ha colaborado con el grupo de investigación *Sistemas Inteligentes para la Movilidad y Comunicación Accesible (SIMCA)* perteneciente al *Departamento de Inteligencia Artificial* de la *Escuela Técnica Superior de Ingenieros Informáticos* y con el *Instituto Universitario de investigación del Automóvil (INSIA)* ambos de la *Universidad Politécnica de Madrid*.

Durante el desarrollo de esta tesis se ha utilizado la tecnología disponible más puntera en ese momento. En concreto se han utilizado dos tecnologías totalmente diferentes de comunicación, WSA (Redes inalámbricas de sensores y actuadores) y DSRC (Comunicaciones dedicadas de corto alcance), las cuales tienen diferentes estándares de comunicación.

Inicialmente y gracias a la tecnología WSA se desarrolló un novedoso algoritmo de enrutamiento basado en geoposicionamiento creando una VANET de recolección de datos.

Una vez demostrada la capacidad de crear una VANET lo suficientemente robusta y fiable se desarrolló un sistema cooperativo para prevenir accidentes con peatones.

Posteriormente con la liberación de la normativa europea de comunicaciones intervehiculares, a través del Instituto Europeo de Estandarización de las Telecomunicaciones (ETSI), se pudo avanzar hacia la tecnología DSRC. Para ello se ha desarrollado un módulo de comunicaciones ITS de diseño propio, denominado ITS DSRC-UPM-INSIA. Los estándares de comunicación intervehicular han sido analizados y se ha desarrollado un software que cumple las especificaciones para poder realizar una geocomunicación. Durante la implementación de éstas se han detectado diferentes problemas y errores los cuales se han ido solucionando.

Finalmente gracias a los módulos DSRC-UPM-INSIA se han desarrollado aplicaciones cooperativas de asistencia al conductor ADAS (*Advanced Driver Assistance System*) para demostrar las funcionalidades y capacidades de los módulos. En esta tesis se muestran tres sistemas cooperativos que se han desarrollado con la tecnología DSRC que abordan diferentes problemáticas:

- Un sistema de aviso a peatones en entornos urbanos e interurbanos, usando comunicaciones V2V y V2P.
- Sistema de aviso de la presencia de ciclistas en las inmediaciones del vehículo, usando comunicaciones V2V y V2P.
- Aviso de motocicletas en carreteras con baja visibilidad incluyendo avisos de ángulo muerto, usando comunicaciones V2V.

El sistema cooperativo basado en WSA y los tres sistemas cooperativos basados DSRC expuestos en la presente tesis han sido probados en condiciones controladas y de tráfico real obteniéndose unos resultados óptimos.

Tabla de Contenidos

Tribunal de la Tesis	V
Declaración de Autoridad	VII
Agradecimientos	IX
Abstract	XI
Resumen	XIII
Tabla de Contenidos	XV
Índice de Ilustraciones	XXI
Índice de tablas	XXV
Glosario	XXVII
Acrónimos	XXIX
1. INTRODUCCIÓN	1
1.1. Motivación de la tesis e identificación del problema	1
1.2. Contribución de la tesis	3
1.2.1. Divulgación científica	4
1.3. Estructura de la memoria	6
2. ESTADO DEL ARTE	9
2.1. Tecnología de sistemas cooperativos y comunicaciones	11
2.1.1. Sistemas ADAS	11
2.1.1.1. Día Uno: aplicaciones en el despliegue de los C-ITS en Europa	12
2.1.1.2. Sistema de detección de ángulo muerto	13
2.1.1.3. Control de cruce adaptativo (ACC)	14
2.1.1.4. Sistema anticolidión	14
2.1.1.5. Detección de peatones y ciclistas	14
2.1.2. Sistemas de posicionamiento global por satélite	15
2.1.2.1. NAVSTAR-GPS	16
2.1.2.2. GLONASS	17
2.1.2.3. Galileo	17

2.1.2.4. Beidou	18
2.1.3. Comunicación entre vehículos	19
2.1.3.1. Honda	19
2.1.3.2. OKI	19
2.1.3.3. Nissan	20
2.1.3.4. Toyota	20
2.1.3.5. eCall	21
2.1.3.5.1. Beneficios del <i>eCall</i>	21
2.1.3.5.2. Implantación del <i>eCall</i>	22
2.1.3.6. Desarrollo de plataformas	22
2.2. Revisión Bibliográfica	23
2.2.1. Sistemas cooperativos (C-ITS)	23
2.2.2. Modelo OSI	25
2.2.3. VANETs	27
2.2.4. Algoritmos de enrutamiento	30
2.2.4.1. Algoritmos GeoUnicast	33
2.2.4.2. Algoritmos Broadcast	35
2.2.4.3. Algoritmos GeoBroadcast	37
3. COMUNICACIONES V2X: METODOLOGÍA Y APORTACIONES	41
3.1. Tecnología WSAAN	43
3.1.1. Descripción de la tecnología	43
3.1.1.1. Estándar de comunicaciones: WSAAN	44
3.1.1.2. <i>Hardware</i>	46
3.1.1.3. Software	47
3.1.2. Algoritmos de enrutamiento	48
3.1.2.1. Estimador de calidad de la conexión	49
3.1.2.2. Ruta de validación	49
3.1.2.3. Balizado adaptativo	49
3.1.3. Trabajo desarrollado: Adaptación de la tecnología WSAAN para comunicaciones V2X	50
3.1.3.1. Arquitectura	50
3.1.3.2. Algoritmo GCTP: Geonetworking Control Tree Protocol	51
3.1.3.2.1. Control de Datos (DC)	52
3.1.3.2.2. Motor de Reenvío (FE)	53
3.1.3.2.3. Motor de enrutamiento (RE)	53
3.1.3.2.4. Estimador de Enlace (LE)	53
3.1.3.2.5. Estructura de los paquetes del algoritmo GCTP	54
3.1.3.2.6. Implementación	56
3.1.4. Resultados	57
3.1.4.1. Pruebas GCTP	57
3.1.4.1.1. Estudio de saturación	57
3.1.4.1.2. Prueba de rendimiento	60
3.1.4.1.3. Pruebas en condiciones reales	62
3.2. Tecnología DSRC	69
3.2.1. Descripción de los protocolos de comunicación normalizados	69
3.2.1.1. Escenarios	71

3.2.1.2.	Arquitectura de los protocolos de geoenrutamiento	72
3.2.1.3.	Estándar de GeoNetworking	72
3.2.1.4.	Protocolos Transporte	73
3.2.2.	Algoritmos de enrutamiento	73
3.2.2.1.	GeoUnicast: <i>Greedy</i>	73
3.2.2.2.	GeoBroadcast: Simple GeoBroadcast	75
3.2.3.	Utilización de la tecnología DSRC para comunicaciones V2X	76
3.2.3.1.	Módulo de comunicaciones DSRC-UPM-INSIA	77
3.2.3.2.	Sistema Operativo	79
3.2.3.3.	Software de soporte para los estándares	79
3.2.4.	Problemas y modificaciones en los estándares	80
3.2.4.1.	Módulos	80
3.2.4.1.1.	Conocimiento de sí mismo	80
3.2.4.1.2.	Vecindad	80
3.2.4.2.	Adquisición de datos	81
3.2.4.2.1.	Posicionamiento, fallo de GPS	81
3.2.4.2.2.	Sincronización temporal y Roadsites	82
3.2.4.3.	Uso de las áreas geográficas en coordenadas cartesianas	82
3.2.4.3.1.	Circunferencia	84
3.2.4.3.2.	Rectángulo	84
3.2.4.3.3.	Elipse	85
3.2.4.4.	Acceso a la información del módulo	86
3.2.5.	Arquitectura para sistemas cooperativos	86
3.2.5.1.	Segunda tarjeta de comunicaciones (IEEE 802.11ac)	87
3.2.5.2.	Pila OSI IEEE 802.11ac	87
3.2.5.3.	Pila OSI para geoposicionamiento	87
3.2.5.3.1.	Nivel físico	87
3.2.5.3.2.	Nivel de enlace	88
3.2.5.3.3.	Nivel de red	89
3.2.5.3.4.	Nivel de transporte	89
3.2.5.3.5.	Nivel de aplicación	89
3.2.5.3.6.	Protocolos de comunicación entre niveles	90
3.2.5.3.6.1.	Servicio de acceso al protocolo <i>Link Layer Control</i> (LLC-SAP)	90
3.2.5.3.6.2.	Servicio de acceso al protocolo GeoNetworking (GN-SAP)	90
3.2.5.3.6.3.	Servicio de acceso al protocolo BTP (BTP-SAP)	91
3.2.5.4.	Arquitectura de geoposicionamiento implementada en los módulos DSRC-UPM-INSIA	92
3.2.6.	Algoritmos	93
3.2.6.1.	Algoritmo LLC	94
3.2.6.2.	Algoritmo GeoNetworking	96
3.2.6.3.	Algoritmo <i>Basic Transport Protocol</i>	98
3.2.6.4.	Aplicación de soporte para elementos de la red sin capacidad de comunicaciones IEEE 802.11p (GNES)	100
3.2.6.4.1.	Características GNES	100

3.2.6.4.2. Cabeceras GNES	103
3.2.6.4.3. Protocolo de acceso al servicio GNES (GNES-SAP)	104
3.2.6.4.3.1. <i>GNES-SAP request</i>	104
3.2.6.4.3.2. <i>GNES-SAP indication</i>	105
3.2.6.4.4. Elementos de las BDs	105
3.2.7. Resultados	106
3.2.7.1. Comprobación del algoritmo de GeoBroadcast	106
3.2.7.1.1. Prueba estática	106
3.2.7.1.2. Dinámica	108
3.2.7.2. GeoUnicast, GeoBroadcast, rendimiento y latencia	110
3.2.7.3. Comprobación de funcionamiento de mensajes	114
4. APLICACIONES DE SISTEMAS COOPERATIVOS	117
4.1. Motivación de las aplicaciones desarrolladas	119
4.2. Sistema distribuido de detección de obstáculos en entornos urbanos . . .	121
4.2.1. Detección de obstáculos	122
4.2.1.1. Detección mediante sensor láser	122
4.2.1.2. Detección mediante visión	123
4.2.2. Alerta de obstáculos	123
4.2.2.1. Vehículo emisor	123
4.2.3. Operación de Transmisión de Datos	126
4.2.4. Vehículo receptor	126
4.2.4.1. Mantenimiento GCTP	127
4.2.4.2. Receptor de datos del sistema de comunicación	127
4.2.4.3. Análisis de datos	127
4.2.4.4. Interfaz Hombre-máquina (HMI)	128
4.2.4.5. Simulador	129
4.2.4.6. Resultados	130
4.3. Aviso a peatones en vías urbanas e interurbanas	133
4.3.1. Primera fase: Solución del problema	133
4.3.1.1. Comunicación inalámbrica	134
4.3.1.2. Trayectoria del vehículo	134
4.3.1.2.1. Modelo de bicicleta	134
4.3.1.2.2. Área de Destino Geográfica (ADG)	136
4.3.1.3. Aplicación móvil de demostración	137
4.3.2. Segunda fase: Desarrollo de la aplicación móvil	138
4.3.2.1. Adquisición de los datos	138
4.3.2.2. Interfaz gráfica	139
4.3.3. Pruebas	139
4.4. <i>BikeWarn</i> : Detección de bicicletas mediante <i>iBeacons</i>	143
4.4.1. Utilidad: detección de <i>ibeacons</i>	144
4.4.2. Aplicación móvil	145
4.4.3. Pruebas y resultados	146
4.5. <i>MotoWarn</i> : Aviso de motocicletas en ángulos muertos	149
4.5.1. Interfaz desarrollada	150
4.5.2. Posicionamiento en la interfaz	151
4.5.3. Pruebas de la aplicación	154

4.5.3.1. Pruebas controladas	154
4.5.3.1.1. Seguimiento	155
4.5.3.1.2. Adelantamiento	156
4.5.3.1.3. Cruce en recta	158
4.5.3.1.4. Cruce en curva	159
4.5.3.2. Pruebas en carreteras convencionales	160
4.5.3.2.1. Seguimiento y adelantamiento	161
4.5.3.2.2. Cruce con baja visibilidad	163
5. CONCLUSIONES Y TRABAJOS FUTUROS	165
5.1. Conclusiones	167
5.2. Aportaciones fundamentales	171
5.3. Trabajos futuros	173
Bibliografía	175
 A. Estándares de comunicación	 A- I
A.1. Estándar de GeoNetworking	A- I
A.1.1. Elementos del estándar	A- I
A.1.1.1. “GeoNetworking address format”	A- I
A.1.1.2. Elementos de posicionamiento	A- II
A.1.1.3. Almacenamiento de datos	A- III
A.1.1.3.1. “Location Table”	A- III
A.1.1.3.2. “Local Position Vector”	A- III
A.1.1.3.3. “Buffers”	A- IV
A.1.2. Cabeceras del estándar	A- IV
A.1.2.1. “Basic Header”	A- VI
A.1.2.2. “Common Header”	A- VII
A.1.2.3. “GeoNetworking Extender Header”	A- VIII
A.1.2.3.1. GeoUnicast	A- VIII
A.1.2.3.2. Topological Scoped Broadcast	A- IX
A.1.2.3.3. Single-Hop Broadcast packet header	A- IX
A.1.2.3.4. Geographical Scoped AnyCast/BroadCast	A- X
A.1.2.3.5. Beacon	A- XI
A.1.2.3.6. Location Service	A- XI
A.1.3. Operaciones de comunicación	A- XII
A.1.3.1. Operaciones de envío	A- XIII
A.1.3.1.1. GUC sender	A- XV
A.1.3.1.2. GAC y GBC sender	A- XVI
A.1.3.1.3. TSB y SHB sender	A- XVII
A.1.3.2. Operaciones de recepción	A- XVIII
A.1.3.2.1. Detector de Paquetes Duplicados	A- XIX
A.1.3.2.2. Beacon y GUC receive	A- XX
A.1.3.2.3. GAC y GBC receive	A- XXI
A.1.3.2.4. TSB y SHB receive	A- XXII
A.1.3.2.5. LS Request y LS Reply receive	A- XXIII

A.1.4. Protocolo de acceso al servicio de geoposicionamiento (GN-SAP)	A- XXIV
A.1.4.1. Request	A- XXIV
A.1.4.2. Indication	A- XXV
A.1.5. Definición de las áreas geográficas para GeoBroadcast y GeoAnycast	A- XXVI
A.2. Estándar Protocolo básico de transporte (<i>Basic Transport Protocol</i> , BTP)	A- XXIX
A.2.1. Servicio de acceso al protocolo BTP (BTP-SAP)	A- XXX
A.2.1.1. Request	A- XXX
A.2.1.2. Indication	A- XXXI

Índice de Ilustraciones

2.1.	Ejemplo de sensores montados en un vehículo	11
2.2.	Fases en el desarrollo de los sistemas cooperativos (ERTICO, 2015)	12
2.3.	Método de trilateración para el cálculo de la posición en el espacio	16
2.4.	Torre OSI	26
2.5.	Arquitectura CALM	29
2.6.	GeoUnicast	34
2.7.	Broadcast	35
2.8.	GeoBroadcast	37
3.1.	Pila de protocolo IEEE 802.15.4	44
3.2.	Mota Maxfor MTM-CM3100	46
3.3.	Esquema del algoritmo GCTP	48
3.4.	Arquitectura de la VANET basada en WSAN	50
3.5.	Esquema del algoritmo GCTP	52
3.6.	Calculo de Distancia Esperada (DE)	54
3.7.	Esquema de los paquetes GCTP	55
3.8.	Flujograma GCTP	56
3.9.	Número máximo de vehículos que la red puede manejar	59
3.10.	Rendimiento de la red WSAN	60
3.11.	Tramo de la autopista A3 donde se han realizado las pruebas	62
3.12.	Datos del nodo 13 recogidos durante la prueba donde se muestra el tipo de conectividad	63
3.13.	Detalle de una reconfiguración de la VANET durante la prueba	64
3.14.	Primera reconfiguración de la VANET	66
3.15.	Segunda reconfiguración de la VANET	67
3.16.	Topologically-Scoped Broadcast	72
3.17.	Detalles de la arquitectura OSI de GeoNetworking	73
3.18.	Componentes y DSRC-UPM-INSIA	78
3.19.	Elementos de la elipse	85
3.20.	Pila OSI para TCP-IP	88
3.21.	Pila OSI para geoposicionamiento	88
3.22.	Formato de trama transmitida por la capa de enlace	89
3.23.	<i>Request GN packet</i>	90
3.24.	<i>Indication GN packet</i>	91
3.25.	<i>Request BTP packet header</i>	91
3.26.	<i>Indication BTP packet header</i>	92
3.27.	Arquitectura de geoposicionamiento implementada	93
3.28.	Esquema del algoritmo LLC	94

3.29. Algoritmo de LLC.	95
3.30. Algoritmo de GeoNetworking.	96
3.31. Algoritmo <i>Basic Transport Protocol</i>	99
3.32. Algoritmo de <i>GeoNetworking Extension and Support</i> (GNES).	101
3.33. Cabecera GNES	103
3.34. Elemento con GPS	105
3.35. Elemento <i>iBeacon</i>	106
3.36. Resultados prueba estática del círculo	107
3.37. Resultados prueba estática del rectángulo	107
3.38. Resultados prueba estática del elipse	108
3.39. Trayecto de los vehículos en la prueba dinámica	109
3.40. Resultados prueba dinámica	110
3.41. Distribución de los módulos DSRC-UPM-INSIA para las pruebas	111
3.42. Distribución de los módulos DSRC-UPM-INSIA para las pruebas del algoritmo GeoBroadcast y GeoUnicast	112
3.43. Esquema de tiempos de retardo	113
3.44. Resultados de latencia	114
3.45. Distribución de los módulos DSRC-UPM-INSIA para las pruebas	115
4.1. Resumen del anuario de accidentes 2013 de la DGT	120
4.2. Sistema distribuido de detección de obstáculos	121
4.3. Patrones de peatones	122
4.4. Detección de escáner láser y HGO	123
4.5. Campos identificativos de cada objeto contenido en el <i>payload</i> del paquete de datos	124
4.6. Transformación de coordenadas	125
4.7. Primera reconfiguración de la VANET	126
4.8. Latencia en función de la distancia	130
4.9. Simplificación modelo de coche	135
4.10. Simplificación modelo de coche	135
4.11. Posible área de destino	137
4.12. Primera interfaz de AAV	138
4.13. Diferentes pantallas de la aplicación	139
4.14. Resultados de prueba de la aplicación de aviso de peatones	140
4.15. Resultados de prueba de la aplicación de aviso de peatones	142
4.16. <i>iBeacons</i> utilizados	145
4.17. Funcionamiento de <i>BikeWarn</i>	146
4.18. Acotamiento de la bicicleta	147
4.19. Trayectoria de la bicicleta y del vehículo durante la prueba	147
4.20. Diagrama de la pantalla de <i>MotoWarn</i>	152
4.21. Seguimiento del vehículo a la motocicleta	155
4.22. Adelantamiento de la motocicleta al vehículo	156
4.23. Cruce del vehículo y la motocicleta en recta	158
4.24. Seguimiento del vehículo a la motocicleta	159
4.25. Muestra de las características de visibilidad de la M-315	160
4.26. Seguimiento y adelantamiento de la motocicleta al vehículo	161
4.27. Secuencia de imágenes de seguimiento y adelantamiento	162

4.28. Cruce de la motocicleta y el vehículo en curva sin visibilidad	163
4.29. Secuencia de imágenes de cruce	164
A.1. <i>GeoNetworking address format</i>	A- II
A.2. <i>Short Position Vector</i>	A- II
A.3. <i>Long Position Vector</i>	A- III
A.4. Estructura de los paquetes GeoNetworking	A- IV
A.5. Estructura de los paquetes GeoNetworking con detalle de <i>GN Header</i> . .	A- IV
A.6. Estructura de los paquetes GeoNetworking con seguridad	A- IV
A.7. Basic Header	A- VI
A.8. Composición del campo LT	A- VI
A.9. <i>Common Header</i>	A- VII
A.10. <i>GeoUnicast packet header</i>	A- IX
A.11. Topological Scoped Broadcast packet header	A- IX
A.12. <i>Single-Hop Broadcast packet header</i>	A- X
A.13. <i>Geographical Scoped AnyCast/BroadCast packet header</i>	A- X
A.14. Beacon packet header	A- XI
A.15. <i>Location Service Request packet header</i>	A- XI
A.16. <i>Location Service Reply packet header</i>	A- XII
A.17. Interacciones de GeoNetworking.	A- XIII
A.18. Esquema de generación de paquetes.	A- XIV
A.19. Esquema de <i>GeoUnicast Sender</i>	A- XV
A.20. Esquemas de GAC y GBC Sender.	A- XVI
A.21. Esquemas de TSB y SHB Sender.	A- XVII
A.22. Esquema de recepción de paquetes.	A- XVIII
A.23. Esquema del detector de Paquetes duplicados	A- XIX
A.24. Esquemas de recepción de paquetes del tipo Beacon y GUC.	A- XX
A.25. Esquemas de recepción de paquetes del tipo GAC y GBC.	A- XXI
A.26. Esquemas de recepción de paquetes del tipo TSB y SHB.	A- XXII
A.27. Esquema de recepción de paquetes del tipo <i>Location Service</i>	A- XXIII
A.28. Circunferencia	A- XXVII
A.29. Rectángulo	A- XXVIII
A.30. Elipse	A- XXVIII
A.32. Estructura de los paquetes BTP	A- XXIX
A.33. BTP header	A- XXX

Índice de tablas

2.1.	Enumeración de protocolos de enrutamiento en VANETs (Li and Wang, 2007)	32
2.2.	Comparación de protocolos de geocast extraída de (Maihofer, 2004) . . .	38
3.1.	Características MTM-CM3100.	47
3.2.	Detalle de los campos del paquete de enrutamiento de la figura 3.7. . . .	55
3.3.	Composición de GCTP Trama de datos de la figura 3.7.	56
3.4.	Casos de uso teóricos.	58
3.5.	Composición de los datos transmitidos en el <i>payload</i>	60
3.6.	Resultados de la prueba estática a distancias de 1hop.	61
3.7.	Resultados de la prueba estática a distancias de 2 hop (14 → 13 → Raíz)..	61
3.8.	Resultados de una prueba de GCTP en autopista	64
3.9.	Velocidades de circulación durante el adelantamiento	65
3.10.	Características placa base módulos de comunicaciones	77
3.11.	Valores de GNET type	104
3.12.	Resultados experimento GeoBroadcast dinámico	110
3.13.	Mensajes GBC y GAC enviados y recibidos por los nodos	112
3.14.	Resultados de latencia de los módulos DSRC-UPM-INSIA	113
3.15.	Paquetes SHB, TSB y GBC enviados y recibidos	115
4.1.	Resultado experimento de comunicaciones	131
A.1.	Campo NH en <i>GeoNetworking Basic Header</i>	A- VI
A.2.	Codificación del campo LT	A- VII
A.3.	Campo NH en <i>GeoNetworking Common Header</i>	A- VIII
A.4.	GeoNetworking Header Types and Header Sub-Types	A- VIII
A.5.	Elementos necesarios para definir el área	A- XXVI

Glosario

ADAS Son sistemas para ayudar al conductor en la conducción.

BROADCAST El direccionamiento de un paquete a todos los destinos utilizando un código especial en el campo de dirección. Cuando se transmite un paquete con este código, todas las máquinas de la red lo reciben y procesan. Este modo de operación se conoce como difusión (broadcasting).

DSRC Productos específicos unidireccional o bidireccional de corto alcance que utilizan comunicación inalámbrica de alcance medio diseñado específicamente para el uso en automóviles que poseen un conjunto protocolos y estándares específicos.

ENRUTAMIENTO Es la función de buscar un camino entre todos los posibles en una red de nodos.

FRAMEWORK Framework o infraestructura digital en el desarrollo de software es una estructura conceptual y tecnológica de soporte, normalmente con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

GEOCAST Acción de difundir un mensaje en un área geográfica determinada

INTEROPERABILIDAD Habilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

ONDA VERDE Una ola verde u onda verde es un fenómeno inducido intencionalmente, en el cual una serie de semáforos se coordinan para permitir el flujo continuo del tráfico sobre varias intersecciones en una misma dirección.

PAYLOAD Son los datos transmitidos en una comunicación. Concretamente, es la parte de la transmisión la cuál era el propósito fundamental de la comunicación, el *payload* no incluye información enviada como cabeceras o metadatos.

RED AD-HOC Red que no depende de una infraestructura fija, cada nodo participa en el encaminamiento mediante el reenvío de datos hacia otros nodos.

RUTEADOR En una red de computadoras existe un nodo llamado ruteador, que se encarga de trazar una ruta entre un nodo origen a un nodo destino

SMART CITY Desarrollo urbano basado en la sostenibilidad capaz de responder adecuadamente a las necesidades básicas de instituciones, empresas, y habitantes, tanto en el plano económico, como en los aspectos operativos, sociales y ambientales.

UNICAST La transmisión de punto a punto con un emisor y un receptor se conoce como unidifusión (unicasting).

VANET Es una red de comunicaciones punto a punto entre vehículos

VISIÓN DIRECTA Es la comunicación entre dos nodos de la red inalámbrica sin necesidad de que otros nodos actúen como relés.

WIDGET Un *widget* o “artilugio” es una pequeña aplicación o programa. Entre sus objetivos está el de dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.

Acrónimos

ADAS	Advanced Driver Assistance System (Sistemas avanzados de asistencia a la conducción)
ADG	Área de Destino Geográfico
API	Application Programming Interface (interfaz de programación de aplicaciones)
AU	Application Unit
BSMD	Bounded Secured Managed Domain
BLE	Bluetooth Low Energy
BTP	Basic Transport Protocol
CALM	Communications Access for Land Mobiles (Acceso a comunicaciones móviles terrestres)
CAM	Cooperative Awareness Message
CCA	Control de Crucero Adaptativo
C-ITS	Cooperative Intelligent Transport Systems (Sistemas Inteligentes de Transporte Cooperativos)
CTP	Collection Tree Protocol
DC	Data Control (Control de datos)
DE	Distancia Esperada
DGT	Dirección General de Tráfico
DL	Data Layer (Capa de datos)
DENM	Decentralized Environmental Notification Message
DnGN	Dispositivo sin GeoNetworking
DSRC	Dedicated Short Range Communications (Comunicaciones dedicadas de corto alcance)
DSSS	Direct Sequence Spread Spectrum

	Espectro ensanchado por secuencia directa
ERTICO	E uropean R oad transport T elematics I mplementation C oordination O rganisation (ITS Europe) (Organización para la Coordinación e Implementación de Sistemas Telemáticos en el Transporte por Carretera)
ETSI	E uropean T elecommunications S tandards I nstitute
FE	F orwarding E ngine (Motor de reenvío)
FIFO	F irst I nput F irst O uput
GAC	G eographically- S coped A nycast
GBC	G eographically- S coped B roadcast
GCTP	G eoNetworking C ollection T ree P rotocol
GN	G eoNetworking
GNEP	G eoNetworking E xtension P rotocol
GNES	G eoNetworking E xtension and S upport
GN_Addr	G eoNetworking A ddress (Dirección de GeoNetworking)
GNSS	G lobal N avigation S atellite S ystems (Sistemas de Navegación Global por Satélite)
GPS	G lobal P ositioning S ystem (American GNSS) (Sistema de posicionamiento Global)
GUC	G eographically- S coped U nicast
HMI	H uman- M achine I nterface (Interfaz Humano-Máquina)
I2V	I nfrastructure t o V ehicle (Comunicación de infraestructura a vehículo)
IEEE	I nstitute of E lectrical and E lectronics E ngineers
INSIA	I nstituto U niversitario de I nvestigación del A utomóvil
IoT	I nternet o f T hings (Internet de las cosas)
ISO	I nternational O rganization for S tandardization (Organización Internacional de Normalización)
ITS	I ntelligent T ransport S ystems (Sistemas Inteligentes de Transporte)
JVM	V irtual J ava M achine (Máquina Virtual de Java)
LE	L ink E stimator (Estimador de enlace)
LL_Addr	L ink L ayer A ddress (Dirección de la capa de enlace)
LLC	L ink L ayer C ontrol (Control de la capa de enlace)

LLP	Link Layer Protocol
LocT	Location Table (Tabla de localizaciones)
LocPV	Local Position Vector (Vector de posición local)
LPV	Long Position Vector
LS	Location Service
MAC	Media Access Control (Control de acceso al medio)
MANETs	Movile Ad-Hoc Networks
MAP	Road Topology
NH	Next Hop (Siguiente salto)
OBU	On Board Unit
OSI	Open System Interconnection
PD	Paquete de Datos
PE	Paquete de Enrutamiento
PHY	PHYsical layer (Capa física)
PT	Protocolo de Transporte
PV	Position Vector (Vector de posición)
R2R	Road side to Road side (Comunicación de estación de carretera a estación de carretera)
R2V	Road side to Vehicle (Comunicación de estación de carretera a vehículo)
RE	Routing Engine (Motor de enrutamiento)
RET	Routing Table (Tabla de enrutamiento)
RL	Routing Layer (Capa de enrutamiento)
RSU	Road Side Unit
SAM	Service Announcement Message
SAP	Service Access Protocol (Servicio de acceso al protocolo)
SHB	Single-hop Broadcast
SPAT	Signal Phase And Timing
SPV	Short Postion Vector
SIMCA	Sistemas Inteligentes para la Movilidad y Comunicación Accesible
TC	Traffic Class
TCP	Transmission Control Protocol

TSB	T opologically- s coped B roadcast
UDP	U ser D atagram P rotocol
UE	U nión E uropea
uit	U nión I nternacional de T elecomunicaciones
UPM	U niversidad P olitécnica de M adrid
UTM	U niversal T ransverse M ercator (Sistema de coordenadas universal transversal de Mercator)
V2I	V ehicle to I nfrastructure Communication (Comunicación de vehículo a infraestructura)
V2P	V ehicle to P edestrian Communication (Comunicación de vehículo a peatón)
V2R	V ehicle to R oad side Communication (Comunicación de vehículo a estación de carretera)
V2V	V ehicle to V ehicle Communication (Comunicación de vehículo a vehículo)
V2X	V ehicle to X Communication (Comunicación de vehículo a x)
VANETs	V ehicular A d hoc N ETworks (Redes vehiculares punto a punto)
WAVE	W ireless A ccess in V ehicular E nvironments
WORA	W rite O nce, R un A newhere (Programa una vez y ejecútalo en cualquier dispositivo)
WPAN	W ireless P ersonal A rea N etwork
WSAN	W ireless S ensors and A ctuators N etworks

A mi familia presente y ausente...

Capítulo 1

INTRODUCCIÓN

1.1. Motivación de la tesis e identificación del problema

En menos de 40 años, se calcula que las ciudades contendrán al 70% de la población mundial, lo que generará grandes retos a hacer frente por la gran densidad de usuarios de las vías.

Para asegurar la buena calidad de vida de la ciudadanía y la sostenibilidad de las urbes se ha iniciado ya el camino hacia un modelo de ciudades inteligentes (smart cities), un concepto surgido hace dos décadas, pero que ha tomado fuerza en los últimos tiempos para hacer frente a la necesidad de transformación de las ciudades.

La sociedad actual se basa en la movilidad, tanto de las personas como de los bienes producidos, lo que conlleva unos costes asociados relacionados principalmente con cuatro parámetros: número de víctimas en accidentes, volumen de emisiones contaminantes, demoras por congestiones y nivel de confort percibido. Por ello en el concepto de smart cities ideado se han marcado diferentes metas relacionadas con la movilidad:

- Tener a los usuarios informados aumenta la satisfacción de los usuarios.
- No tener retrasos aumenta la eficiencia del sistema de transporte, la capacidad de la infraestructura existente y la productividad de las empresas especialmente de las empresas de transporte.
- No tener accidentes o intentar reducirlos al máximo.
- Reducir el impacto medioambiental, la contaminación y el ahorro de energía.

La capacidad de mejora de las soluciones clásicas está casi agotada y puede resultar poco efectiva en comparación al esfuerzo a invertir. Por ejemplo la imposibilidad de ampliación

de la infraestructura en algunas zonas o la limitación en cuanto a beneficios alcanzables con las medidas convencionales de seguridad. Estas razones han motivado la introducción de nuevas alternativas bajo la denominación de Sistemas Inteligentes de Transporte (ITS según las siglas en inglés).

Dentro de los ITS se encuentran los sistemas cooperativos, que se espera que produzcan avances en los objetivos anteriormente enumerados mediante “sistemas de asistencia avanzada al conductor” (ADAS, Advanced Driver Assistance System) que pueden ayudar en la regulación del tráfico rodado, un uso más eficiente de los medios de transporte, reducción de tiempos empleados en los desplazamientos, posibilitar una conducción adecuada a las condiciones de la vía, una reducción de la contaminación y descenso de la siniestralidad vial como consecuencia directa o indirecta de los ADAS.

La base de los sistemas cooperativos es el intercambio de información entre los diversos actores para que puedan interactuar entre ellos. Se podrá implementar un amplio espectro de aplicaciones y servicios que ayudarán a la seguridad, la eficiencia, el confort y a reducir el impacto medioambiental del transporte, mediante el intercambio de información entre vehículos, con la infraestructura y centros de control. Esta información permite ampliar el horizonte que es posible “ver” para adoptar acciones. Este intercambio de información es de vital importancia dado que estamos hablando de un sistema dinámico donde un retraso de unos pocos milisegundos puede marcar la diferencia entre un uso correcto de los datos y un accidente.

La dificultad de los sistemas cooperativos reside en crear una red de comunicaciones estable con la que poder intercambiarse información en tiempo real que dé soporte a todas las aplicaciones que irán embebidas en los vehículos. Estas redes de comunicación tienen que optimizar el enrutamiento de paquetes dado que las posiciones de los nodos de la red varían su posición geográfica causando potenciales pérdidas de información, por ello se tiene que realizar una reestructuración de los parámetros de enrutamiento constante para asegurar la robustez del sistema así como dar una seguridad en las comunicaciones, tanto en la entrega de paquetes como en la privacidad de los mismos. A estas redes de comunicaciones se las denomina VANETs (Vehicle ad-hoc Networks).

Durante los años precedentes diversos grupos de investigación han trabajado en el problema del enrutamiento en redes vehiculares donde se han ido desarrollando diferentes algoritmos de enrutamiento, con diferentes grados de eficiencia según el escenario al que se enfrentan.

Actualmente las pocas VANETs que se pueden encontrar en el mercado han sido desarrolladas sin seguir un estándar bien definido por empresas privadas haciendo sistemas propietarios, cerrados y con altos costes de mantenimiento y uso. Sin embargo, desde las

diferentes instituciones europeas e internacionales, se está realizando un gran esfuerzo para que este tipo de redes vehiculares sean abiertas a la competencia, con lo que se está promoviendo la creación de estándares que regulen esta transferencia de datos entre vehículos y con la infraestructura, de manera que todos los usuarios de la vía, independientemente de su fabricante, puedan acceder a todos los servicios derivados de las comunicaciones y, por extensión, de los sistemas cooperativos.

La presente tesis se enfoca en proponer una solución a la problemática que hay en el uso de VANETs para el desarrollo de sistemas cooperativos, tanto a nivel hardware como software, mediante el desarrollo de una tecnología propia de comunicaciones que cumpla los requisitos de estandarización pertinentes en el sector, además de ser robusta, fiable, interoperable y de bajo coste, de manera que pueda ser utilizada para el desarrollo de todo tipo de proyectos que involucren Sistemas Inteligentes de Transporte Cooperativos (C-ITS). Además, esta tecnología debe ser flexible, por lo que se debe tener el control de todo el software que la compone a fin de que se pueda adaptar a todo tipo de aplicaciones, así como del hardware, para añadir todas las nuevas funcionalidades y conectividad que se considere oportuno.

1.2. Contribución de la tesis

La presente tesis se ha desarrollado en colaboración con el grupo de investigación “*Sistemas Inteligentes para la Movilidad y Comunicación Accesible (SIMCA)*” perteneciente al “*Departamento de Inteligencia Artificial*” de la “Escuela Técnica Superior de Ingenieros Informáticos” de la “Universidad Politécnica de Madrid” y en el “*Instituto Universitario de Investigación del Automóvil (INSIA)*” de la “Universidad Politécnica de Madrid”. Tanto el grupo SIMCA como el INSIA llevan numerosos años investigando los sistemas inteligentes de transporte cooperativos (C-ITS), llevando a cabo trabajos y colaboraciones reconocidas nacional e internacionalmente en diferentes congresos y revistas.

En el presente documento se detallan los pasos seguidos hasta conseguir tener un dispositivo capaz de crear una VANET operativa que dé soporte a sistemas cooperativos, donde se han seguido los estándares internacionales, garantizando la interoperabilidad en cualquier situación. También se detalla la creación de varios ADAS de seguridad vial usando sistemas cooperativos basados en estos dispositivos.

Todos los algoritmos y la programación realizada han sido creados en lenguajes que soportan la multiplataforma haciendo posible que se exporten a otra plataforma de comunicaciones aunque no hayan sido específicamente desarrollados para esta.

La presente tesis tiene dos fases diferenciadas:

- En un principio del desarrollo de la presente tesis se centraron los esfuerzos en el desarrollo de algoritmos de enrutamiento mediante la creación de un algoritmo de enrutamiento lo suficientemente robusto como para dar soporte a una VANET mediante dispositivos WSA.
- Posteriormente con la salida comercial de tarjetas de comunicación IEEE 802.11p y la publicación de los estándares europeos de redes de geocomunicación por el Instituto Europeo de Estándares de Telecomunicación (ETSI - European Telecommunications Standards Institute) se centraron los esfuerzos en la creación de un módulo dedicado de comunicaciones de corto alcance (DSRC-UPM-INSIA) con todas las funcionalidades necesarias para la creación y mantenimiento de una VANET que sirviera como base para el desarrollo de aplicaciones C-ITS.

1.2.1. Divulgación científica

Parte del trabajo desarrollado en esta Tesis se ha expuesto en las siguientes publicaciones y contribuciones a congresos:

- Revistas:
  - Felipe Jiménez Alonso; José Eugenio Naranjo Hernández; Francisco Serradilla García; Elisa Pérez; María José Hernández; Trinidad Ruiz; José Javier Anaya Catalán; Alberto Díaz Álvarez. "Intravehicular, Short- and Long-Range Communication Information Fusion for Providing Safe Speed Warnings". Sensors-Basel. 16 - 1, pp. 131 - 146. 01/2016. ISSN 1424-8220
  - Alberto Díaz Álvarez; Francisco Serradilla García; José Javier Anaya Catalán; Felipe Jiménez Alonso; José Eugenio Naranjo Hernández. "Estimación de la autonomía de un vehículo eléctrico según el estilo de conducción. Dyna Ingeniería e Industria". 90 - 3, pp. 294 - 299. 05/2015. ISSN 0012-7361
  - José Javier Anaya Catalán; Edgar Talavera Muñoz; Felipe Jiménez Alonso; Francisco Serradilla García; José Eugenio Naranjo Hernández. "Vehicle to Vehicle GeoNetworking using Wireless Sensor Networks". Ad Hoc Networks. 27, pp. 133 - 146. 04/2015. ISSN 1570-8705
  - Felipe Jiménez Alonso; José Eugenio Naranjo Hernández; Oscar Gómez Casado; José Javier Anaya Catalán. "Vehicle Tracking for an Evasive Manoeuvres Assistant Using LowCost Ultrasonic Sensors". Sensors. pp. 22689 - 22705. 11/2014. ISSN 1424-8220
  - José Javier Anaya Catalán; Edgar Talavera Muñoz; Nuria Gómez Blas; Felipe Jiménez Alonso; José Eugenio Naranjo Hernández. "A novel GeoBroadcast

algorithm for V2V Communications over WSN“. Electronics. pp. 521 - 533. 09/2014. ISSN 2079-9292

- José Javier Anaya Catalán; Edgar Talavera Muñoz; Felipe Jiménez Alonso; José Eugenio Naranjo Hernández; José Gabriel Zato Recellado. "Vehicular GeoNetworking based Communications to support Cooperative Safety Systems“. Securitas Vialis. 18, pp. 54 - 69. 06/2014. ISSN 1888-9697
- Alberto Díaz Álvarez; Francisco Serradilla García; José Eugenio Naranjo Hernández; José Javier Anaya Catalán; Felipe Jiménez Alonso. "Modeling the Driving Behavior of Electric Vehicles Using Smartphones and Neural Networks“. IEEE Intelligent Transportation Systems Magazine. 6 - 3, pp. 44 - 53. 05/2014. ISSN 1939-1390
- Fernando García; Felipe Jiménez Alonso; José Javier Anaya Catalán; José Maria Armingol; José Eugenio Naranjo Hernández; Arturo de la Escalera. "Distributed Pedestrian Detection Alerts Based on Data Fusion with Accurate Localization“. Sensors. pp. 11687 - 11708. 09/2013. ISSN 1424-8220

■ Congresos:

- José J. Anaya, Edgar Talavera, David Giménez, Nuria Gómez, Felipe Jiménez, José E. Naranjo, "Vulnerable Road Users Detection using V2X Communications“ 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Las Palmas, Spain, September 2015.
- José J. Anaya, Edgar Talavera, David Giménez, Felipe Jiménez, José E. Naranjo, "Detección de usuarios vulnerables utilizando comunicaciones V2X“ XV Congreso Español sobre Sistemas Inteligentes de Transporte. Madrid, Spain, April 2015
- Felipe Jiménez, José E. Naranjo, Oscar Gómez y José J. Anaya, "Device for steering system autonomous control to be implemented in any kind of road vehicle“, FISITA World Automotive Congress 2014, Maastricht, The Netherlands, June 2014.
- José J. Anaya, Pierre Merdrignac, Oyunchimeg Shagdar, Fawzi Nashashibi y José E. Naranjo, "Vehicle to Pedestrian Communications for Protection of Vulnerable Road Users“, IEEE IV Symposium 2014, Dearborn, Michigan, USA, June 2014.
- José J. Anaya, Edgar Talavera, Felipe Jiménez, José G. Zato, Nuria Gómez y José E. Naranjo, "GeoNetworking based V2V Mesh Communications over WSN“, IEEE ITSC Conference, The Hague, Netherlands, October, 2013.
- José J. Anaya, Edgar Talavera, Felipe Jiménez, José G. Zato y José E. Naranjo, "Comunicaciones V2V basadas en GeoPosicionamiento“, Jornada sobre Vehículos y Transportes: I+D+I de la UPM, INSIA, Madrid, 19 September, 2013.

- José E. Naranjo, Edgar Talavera, José J. Anaya, Felipe Jiménez, José G. Zato y Nuria Gómez, "Highway Test of V2V Mesh Communications over WSN", IEEE ITS Conference, Anchorage, USA, September, 2012

- Demostraciones:

- Demostración de un sistema de comunicaciones intervehiculares basado en WSN el 7 de junio de 2012 en el Congreso Internacional de Vehículos Inteligentes 'IEEE IV'12' organizado por la Sociedad IEEE Sistemas Inteligentes de Transporte (IEEE Intelligent Transportation Systems Society, ITSS).

Durante la realización de la tesis se han realizado un par de estancias en centros extranjeros:

- Julio-Septiembre de 2013: Colaboración con el grupo Imara en Institut National de Recherche en Informatique et en Automatique (INRIA) Paris Rocquencourt. Se ha colaborado en el proyecto SCORE@F (Système COopératif Routier Expérimental Français) y se han desarrollado aplicaciones V2P.
- Abril-Julio de 2014: Colaboración con Partners for Advanced Transportation Technology (PATH) de la University of California, Berkeley. Se ha colaborado en la determinación de la posición de los vehículos en tiempo real de un sistema CACC perteneciente al proyecto "Development of a Connected Vehicle Platform to Support Cooperative Adaptive Cruise Control for a Fleet of Instrumented Research Vehicles".

Adicionalmente gracias al trabajo desarrollado y a la divulgación de este se ha conseguido el "Premio a la investigación en el campo de la automoción" de la "Fundación Eduardo Barreiros" del año 2014, por la creación de un "Sistema Integrado de evitación automática de colisiones".

1.3. Estructura de la memoria

La memoria de esta tesis doctoral se desarrolla en una estructura como sigue:

- En el capítulo 2 se presenta el "ESTADO DEL ARTE" donde se realiza un análisis del estado de la técnica en las soluciones propuestas en el campo de los sistemas cooperativos así como en los módulos DSRC, se hace un estudio de la bibliografía existente de los diferentes algoritmos de geoposicionamiento y georutamiento existentes.

- En el capítulo 3 se presenta “COMUNICACIONES V2X: METODOLOGÍA Y APORTACIONES” donde se explican las diversas fases en las que se ha desarrollado la presente tesis, dividiéndose ésta en dos grandes bloques:
  - En la sección 3.1: “Tecnología WSA” se explica el proceso seguido para el desarrollo de un algoritmo de geoposicionamiento así como los pasos seguidos para la verificación del algoritmo.
  - En la sección 3.2: “Tecnología DSRC” se explica los pasos seguidos para el desarrollo de los módulos DSRC-UPM-INSIA, análisis de los estándares ETSI, selección de hardware y desarrollo de los algoritmos necesarios para su finalización. También se detallan las pruebas realizadas para la verificación del correcto funcionamiento del módulo y los resultados de éstas.
- En el capítulo 4 se presentan las “APLICACIONES DE SISTEMAS COOPERATIVOS” donde se describe el desarrollo de diferentes aplicaciones “ADAS” para cada plataforma de comunicación. Se realiza una descripción de las características de cada una y se presentan los resultados obtenidos de éstas.
- En el capítulo 5 se presentan las “CONCLUSIONES Y TRABAJOS FUTUROS” obtenidos tras la realización del presente trabajo doctoral, así como futuras líneas de investigación por las que se debería continuar en futuros desarrollos.
- Finalmente se expone la bibliografía utilizada durante la realización del presente documento.
- En el anexo A se expone de manera pormenorizada los estándares utilizados durante el desarrollo de la presente tesis.

Capítulo 2

ESTADO DEL ARTE

En este capítulo se expone un análisis de la tecnología actual en el ámbito de la seguridad en vehículos, ADAS, sistemas cooperativos, VANETs y algoritmos de GeoNetworking tanto en el apartado industrial como en el académico, a fin de poner en contexto el trabajo realizado durante la presente tesis.

2.1. Tecnología de sistemas cooperativos y comunicaciones

En este punto se hace una pequeña introducción a los sistemas industriales, tanto a modelos comercializados, como prototipos e ideas propuestos en el sector industrial en el campo de los sistemas ADAS, VANETs (comunicaciones) y sistemas de posicionamiento global.

2.1.1. Sistemas ADAS

Los sistemas avanzados de asistencia a la conducción (ADAS) son sistemas desarrollados para automatizar, adaptar y/o mejorar la seguridad de los vehículos.

FIGURA 2.1: Ejemplo de sensores montados en un vehículo

Las características de seguridad están diseñadas para evitar colisiones y accidentes ofreciendo tecnologías que alertan al conductor de posibles problemas para que actúe en consecuencia o evitando colisiones mediante la aplicación de dispositivos de seguridad que toman el control del vehículo.

El sector dedicado a los ADAS es uno de los que tienen un mayor crecimiento en la electrónica del automóvil (Figura 2.1). Las tecnologías ADAS pueden basarse en todo tipo de sensores instalados en el propio vehículo o en la comunicación de datos entre vehículos (V2V, *Vehicle-to-Vehicle*) y con la infraestructura (V2I, *Vehicle-to-Infrastructure*). Todas estas tecnologías son el soporte de los denominados sistemas cooperativos, los cuales permiten el desarrollo de aplicaciones de seguridad, eficiencia y confort que no se basan en la información proporcionada por el propio vehículo como un elemento individual, sino que son capaces de utilizar la información proveniente de otros vehículos y de la infraestructura para desarrollar su funcionalidad, extendiendo su alcance más allá del horizonte visual del conductor y los sensores, cuya culminación es la conducción automática

cooperativa. La Unión Europea a través del organismo ERTICO (*European Road Transport Telematics Implementation Coordination Organisation*) ha fijado diferentes fases para la implementación de los sistemas cooperativos, en la Figura 2.2 se pueden ver estas fases. Se puede observar que ya estamos en el periodo de la “fase 1” la implementación de las comunicaciones V2V básicas.

FIGURA 2.2: Fases en el desarrollo de los sistemas cooperativos (ERTICO, 2015)

En el documento *Guide about technologies for future C-ITS service scenarios* (ERTICO, 2015) el ERTICO describe las tecnologías que se usaran en el futuro en los ITS, las normas e iniciativas que se plantean implementar en los sistemas europeos de C-ITS. Así como las diferentes fases de implementación de los sistemas C-ITS y la definición de las aplicaciones a implementar en cada fase.

Las próximas generaciones de sistemas ADAS aprovecharán cada vez más la conectividad inalámbrica para ofrecer un mejor uso de las comunicaciones entre vehículos y compartir la información de los diferentes vehículos. A continuación se detallan algunos sistemas ADAS cooperativos.

2.1.1.1. Día Uno: aplicaciones en el despliegue de los C-ITS en Europa

Los miembros del consorcio encargado de la implementación de los sistemas cooperativos en Europa, han decidido en general seguir un enfoque de implementación por fases (Figura 2.2) con un despliegue inicial de mínimos, casos de uso no complejos donde se consiguen beneficios para el usuario incluso con una penetración limitada de C-ITS

en vehículos equipados y *RoadSide Unit (RSU)* (Infraestructura) instaladas en la carretera en zonas de limitadas, *Día Uno* (ERTICO, 2015). En las siguientes fases la complejidad de los casos de uso se incrementará incluidas otras aplicaciones de seguridad con mayor penetración de equipos C-ITS en los vehículos y una mayor cobertura de la infraestructura de prevención de accidentes.

Las aplicaciones tienen diferentes requisitos operacionales y funcionales, por ejemplo diferentes requisitos de latencia que conducirían a tecnologías de comunicación específicas. En particular, los servicios relacionados con la seguridad, pero también servicios de eficiencia de tráfico y casos de uso que se basan en el intercambio de información entre los vehículos y la infraestructura vial (V2X), actualmente se centran en la tecnología *WiFi* vehicular (ETSI ITS G5 dentro de la banda de 5,9 GHz).

Las aplicaciones destinadas al *Día Uno* por el consorcio son:

- Advertencia de lugares peligrosos.
- Advertencia de vehículo lento.
- Advertencia de retenciones más adelante.
- Advertencia de obras en la carretera.
- Advertencia vehículo parado.
- Recomendación de velocidad adecuada según la señalización vial.
- Datos de los vehículos colindantes.
- Fase de los semáforos y tiempo hasta el cambio de fase en las intersecciones.
- Advertencia de vehículo de emergencias.
- Luz de freno de emergencia.
- Indicación de motocicleta acercándose.

2.1.1.2. Sistema de detección de ángulo muerto

Tiene como objetivo alertar al conductor si hay otro vehículo en el ángulo muerto del vehículo o cerca de él. Los avisos pueden ser visuales, acústicos o por vibración. Existen dos filosofías de detección; la primera se basa en los sensores del propio vehículo, para ello los sensores vigilan constantemente la zona lateral próxima al vehículo. Generalmente son sensores radar de medio o corto alcance a 24 GHz o bien sistemas de procesamiento de

imágenes. La otra filosofía es mediante la comunicación entre vehículos (V2V) donde se comunican las posiciones respectivamente. Existen sistemas que pueden alertar de forma continua de la existencia de vehículos en el ángulo muerto independientemente de las intenciones del conductor, mientras que otros únicamente actúan cuando el conductor expresa la intención de cambiar de carril (al activar el intermitente o girar el volante).

2.1.1.3. Control de crucero adaptativo (ACC)

Permite fijar una velocidad de conducción y mantenerla de forma automática sin necesidad de pisar el acelerador ni el freno, manteniendo la distancia de seguridad con el coche que circula delante. Si el carril de circulación está libre, el ACC mantiene la velocidad del vehículo fijada por el conductor, mientras que si el conductor se aproxima a un vehículo que circula por delante, o bien otro vehículo entra en su carril, el ACC ajustará la velocidad para mantener una distancia de seguridad. Para calcular la distancia con el coche precedente se puede utilizar un telémetro láser, bien mediante un radar o comunicaciones V2V donde los vehículos se transmiten la posición y la velocidad actual a la que circulan para que el resto de vehículos transmitan esta información al ACC y puedan adaptar la velocidad para evitar un accidente.

2.1.1.4. Sistema anticolidión

Existen dos opciones para estos sistemas, la individual y la cooperativa. En la individual los sensores instalados en el vehículo monitorizan constantemente su entorno, de forma que cuando un objeto (vehículo, peatón, obstáculo, etc) es detectado, el sistema determina si el vehículo está en una situación en la que puede producirse una colisión inminente. En la cooperativa todos los actores de la circulación están continuamente intercambiándose información de la posición, velocidad y dirección, de esta manera el sistema puede determinar con tiempo si el vehículo está en una trayectoria de colisión con otro vehículo. En caso afirmativo en ambos casos puede enviar un aviso al conductor para que pueda evitar la colisión o directamente toma el control del vehículo accionando los frenos o cambiando la trayectoria del vehículo si el conductor no responde al aviso o no hay tiempo suficiente como para que el conductor evite la colisión.

2.1.1.5. Detección de peatones y ciclistas

Existen diferentes sistemas que realizan la detección usando diferentes técnicas de los que algunos se basan en los sensores del vehículo exclusivamente y otros se han implementado mediante sistemas cooperativos.

Los sistemas basados en sensores se parecen a los sistemas anticolidión. Se trata de una tecnología de ayuda a la conducción que no solo avisa al conductor de la presencia de peatones o de ciclistas, sino que también llega a aplicar la frenada autónoma del vehículo. Para ello el sistema se sirve de una cámara instalada en el parabrisas y de un radar en el parachoques. El sistema cruza la imagen que recibe con una base de datos capaz de ayudarlo a detectar formas de peatones y ciclistas y diferenciarlos de otros objetos de la carretera. En el caso de ser un peatón, el vehículo puede pasar por dos fases: una primera en la que avisa al conductor de la presencia de la persona, y si éste no responde, disminuye el tiempo requerido para aplicar la frenada, reduciendo el espacio entre las pastillas y los discos de freno. En la segunda fase, si el conductor sigue sin reaccionar, el sistema activa los frenos de forma automática.

Los sistemas cooperativos se basan en comunicar la posición del ciclista al vehículo mediante comunicaciones. Los sistemas de Volvo (for life, 2014) y de la DGT (DGT, 2015) utilizan comunicaciones móviles para establecer la comunicación entre ambos. Estos sistemas implican que tanto el ciclista como el conductor del vehículo deben de tener un dispositivo móvil con conexión a la red móvil, que tengan la aplicación instalada y en funcionamiento. Así como unos servidores donde se guarden las posiciones de los ciclistas y vehículos para poder ser consultadas en todo momento.

2.1.2. Sistemas de posicionamiento global por satélite

Dadas las características de los sistemas cooperativos en vehículos, todo mensaje emitido debe estar firmado con un identificador del vehículo. Una simple identificación numérica no es suficiente dado que los vehículos tienen la capacidad de variar de posición, por lo cual es necesario que tengan una identificación numérica del vehículo y la posición geográfica desde la que fueron emitidos. Para el posicionamiento geográfico hay que utilizar algún sistema de posicionamiento global, lo más común y asequible es utilizar satélites para ello.

Todos los sistemas de posicionamiento global se basan en el mismo principio para conseguir conocer la posición de un objeto situado en cualquier punto de la superficie de la tierra.

Un sistema global de navegación por satélite, o *GNSS* por su acrónimo en inglés, está formado por una constelación de satélites los cuales transmiten una serie de señales. Estas señales son utilizadas para el posicionamiento en cualquier parte del globo terráqueo. Con la recepción y procesamiento de las señales de varios satélites se puede determinar las coordenadas geográficas y la altitud de un punto dado, las 24 horas del día y en todas las condiciones climatológicas.

Cuando se desea determinar la posición, el receptor debe localizar como mínimo cuatro satélites de la red, de los que recibe las señales indicando la identificación y la hora del reloj de cada uno de ellos. Con base a estas señales, el receptor sincroniza su reloj interno y calcula el tiempo que tardan las señales de los diferentes satélites en llegar al equipo obteniendo la distancia a cada uno. Conocidas las distancias se determina fácilmente la propia posición relativa respecto a los satélites. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene la posición absoluta o coordenada real del punto de medición, mediante trilateración (Figura 2.3).

FIGURA 2.3: Método de trilateración para el cálculo de la posición en el espacio

Actualmente existen dos sistemas de navegación por satélite que cumplen con las condiciones para ser considerados GNSS. El sistema de Estados Unidos de América (NAVSTAR-GPS) y el de la Federación Rusa (GLONASS). Adicionalmente existen varios programas para desarrollar GNSS por parte de diferentes países o instituciones, como es el caso de China, Japón, India o la Unión Europea.

2.1.2.1. NAVSTAR-GPS

El NAVSTAR-GPS (*NAVigation System with Timing And Ranging – Global Position System*), conocido simplemente como GPS, es el sistema GNSS de Estados Unidos de América. Éste sistema fue desarrollado, instalado y empleado por el Departamento de Defensa de los Estados Unidos. Inicialmente el objetivo del sistema era ofrecer a las fuerzas

armadas de los Estados Unidos la posibilidad disponer de la posición geográfica de forma autónoma o individual, de vehículos o de armamento, con un coste relativamente bajo, con disponibilidad global y sin restricciones temporales.

El GPS funciona mediante una red de 24 satélites situados en 6 planos orbitales a 20.200 km de altura con trayectorias sincronizadas para cubrir toda la superficie de la Tierra. Utiliza el sistema WGS84 (*World Geodetic System 84* (Sistema Geodésico Mundial 1984 (Agency, 2014)) que consiste en un patrón matemático de tres dimensiones que representa la tierra por medio de un elipsoide, al que se le estima un error medio de cálculo menor a 2 cm.

En 1984 un avión de pasajeros coreano fue derribado por la Unión Soviética al invadir por error su espacio aéreo. Ello llevó al gobierno de los Estados Unidos a ofrecer a los usuarios civiles cierto nivel de uso del GPS, llegando finalmente a ceder el uso global y sin restricciones temporales.

2.1.2.2. GLONASS

GLONASS (*GLObal'naya NAVigatsionnaya Sputnikovaya Sistema o GLObal Orbiting Navigation Satellite System*) fue desarrollado por la antigua Unión Soviética. Hoy en día es administrado por la Federación Rusa y constituye el homólogo del GPS estadounidense.

Este sistema consta de una constelación de 31 satélites (24 activos, 3 de repuesto, 2 en mantenimiento, uno en servicio y otro en pruebas) situados en tres planos orbitales a altura de 19.100 km. GLONASS utilizó inicialmente el sistema geodésico ruso PZ-90 que era sensiblemente diferente del sistema WGS-84. En septiembre de 2007 fue adaptado y actualizado, llamándose ahora PZ-90.02

La aparición en el mercado de receptores que permiten recibir señales pertenecientes a los dos sistemas GLONASS y GPS hace interesantes las posibilidades de GLONASS en la medición como apoyo al GPS norteamericano. El uso del sistema GLONASS con otros GNSS permite aumentar considerablemente las prestaciones de los Sistemas de Navegación por Satélite: precisión, acceso, integridad y continuidad de los servicios de navegación para los usuarios de la aviación.

2.1.2.3. Galileo

Galileo es el sistema en desarrollo por la Unión Europea con el objetivo de evitar la dependencia de los sistemas GPS y GLONASS. Al contrario que los dos anteriores, será de uso civil.

Este sistema se compondrá de una constelación de 30 satélites divididos en tres órbitas, a una altura aproximada de 24.000 km, que cubrirán toda la superficie del planeta. Se espera que esté completamente operativo a partir de 2019 (nueve años más tarde de lo previsto).

Galileo será compatible con NAVSTAR por lo que los receptores podrán combinar las señales de los satélites de ambos para aumentar la precisión de las medidas. El sistema está concebido para usuarios multimodales. Ofrecerá cinco servicios:

- *Open Service (OS)*. Orientado a aplicaciones para el público en general. Proveerá señales para proporcionar información precisa de tiempo y posicionamiento de forma gratuita.
- *Safety-of-Life (SoL)*. Se utilizará para la mayoría de las actividades de transporte en las cuales la vida humana podría estar en peligro.
- *Commercial Service (CS)*. Estará orientado a aplicaciones de mercado que requieren un nivel de prestaciones superior que las que ofrece el servicio abierto. Brindará servicios de valor añadido a cambio del pago de un canon.
- *Public Regulated Service (PRS)*. Servicio “robusto” y de acceso controlado para las aplicaciones gubernamentales. El servicio PRS será utilizado principalmente por los cuerpos policiales y los servicios aduaneros.
- *Search and Rescue Service (SAR)*. Este servicio brindará importantes mejoras al sistema de Búsqueda y Salvamento (SAR) existente (localización precisa de alertas, recepción casi en tiempo real de mensajes de socorro transmitidos desde cualquier punto de la Tierra).

2.1.2.4. Beidou

Beidou es el proyecto de la República Popular China para obtener un sistema de navegación por satélite. La primera generación, *BeiDou-1*, está operativa desde el 2000 y es un sistema de posicionamiento por satélite local dando servicio a China y a sus países vecinos. La segunda generación, también llamada *Compass* o *BeiDou-2*, será un sistema de posicionamiento global con un funcionamiento similar al GPS.

A diferencia de los sistemas GPS, GLONASS y Galileo, que utilizan satélites en órbitas bajas y ofrecen un servicio global, la primera generación, *BeiDou-1* utiliza satélites en órbitas geoestacionarias a una altura de 35.786 kilómetros. Esto implica que el sistema no requiera de una gran constelación de satélites, pero su cobertura queda limitada a las zonas sobre la tierra desde las cuales éstos sean visibles. En este caso el territorio de China.

2.1.3. Comunicación entre vehículos

Los vehículos conectados son aquellos que hacen uso de tecnologías de comunicaciones para el intercambio de información con otros vehículos (V2V), con la infraestructura viaria (V2I) y con las personas (V2P), de forma que se pueden llevar a cabo acciones orientadas a mejorar la fluidez en el tráfico, aumentar la seguridad mediante la correcta anticipación de eventos, y comunicarse de forma más eficiente con las fuerzas de seguridad y los servicios de emergencias entre otras funciones, habilitando el funcionamiento de los sistemas cooperativos (C-ITS).

Estas tecnologías dan lugar a redes de comunicación ad-hoc (red que no depende de una infraestructura fija, cada nodo participa en el encaminamiento mediante el reenvío de datos hacia otros nodos) en las que los nodos son los propios vehículos o puntos de acceso en la infraestructura. Así, un vehículo (nodo) puede recoger información del tráfico en su entorno inmediato y difundirlo a través de la red de comunicación hacia otros vehículos. Estos vehículos podrán interpretar esos datos y ofrecérselos al conductor.

Varias compañías están desarrollando sistemas de comunicación entre vehículos para la prevención de accidentes. A continuación se detallan las principales compañías que están desarrollando estos sistemas.

2.1.3.1. Honda

Honda ha desarrollado un sistema para prevenir los atropellos a peatones al alertar al conductor del vehículo y al peatón en caso de que se prevea una posible colisión (Honda, 2013). Este sistema incluye un módulo integrado en el vehículo y un *smartphone* especial con la capacidad de comunicaciones en la banda de los 5.9GHz donde hay instalada una aplicación. Tanto el módulo del vehículo como el *smartphone* del peatón se envían datos a través de la aplicación: su posición, velocidad, ángulo de movimiento, etc. A partir de estos datos ambas partes calculan la posibilidad de que se produzca un atropello, mostrando una alerta en caso de que la probabilidad sea lo suficientemente alta.

De una manera similar pero orientado a la comunicación entre vehículos y motocicletas, mediante comunicaciones V2V, el sistema alerta de las posibles colisiones entre vehículos y motocicletas.

2.1.3.2. OKI

OKI ha desarrollado el *Simple and Compact IVC Unit* (Patrascu, 2010). El *IVC Unit* es un dispositivo compacto para la comunicación entre vehículos que integra una antena para la

comunicación, un módulo GPS para la geolocalización, un sensor giroscópico y una serie de LEDs y *buzzers* que actúan como interfaz de usuario. Este dispositivo mide 50mm x 57mm x 26mm por lo que podrá ser colocado fácilmente en el salpicadero de los coches.

Este dispositivo compartiría sus datos con otros vehículos a través de internet. Cuando detecte que alguno de esos vehículos puede suponer un peligro para el conductor emitiría un aviso para alertar al usuario. Si el conductor no tiene visión directa del vehículo que se aproxima, gracias a la comunicación entre vehículos, el módulo de OKI conoce la posición, velocidad y ángulo de movimiento del resto de vehículos cercanos. Con esos datos puede predecir la trayectoria de esos vehículos y avisar al conductor en caso de que se pueda producir una situación de riesgo con alguno de ellos.

2.1.3.3. Nissan

Nissan ha desarrollado un sistema que mediante comunicaciones V2I (Nissan Motor CO., 2015) es capaz de advertir a un vehículo que se aproxima a una intersección si la velocidad con la que se aproxima es la adecuada o es excesiva para detenerse con seguridad porque el semáforo está en rojo o va a cambiar de fase antes de que pueda pasarlo. El sistema también puede advertir al conductor de un posible accidente porque se aproxima un vehículo con la prioridad a la intersección. El sistema también puede indicar la velocidad adecuada para que el conductor vaya en la onda verde de la calle por la que circula. Para el correcto funcionamiento del sistema debe haber un módulo DSRC de tipo infraestructura en cada intersección y los vehículos deben poseer un localizador GNSS y un módulo DSRC con el que comunicarse con la infraestructura.

2.1.3.4. Toyota

Toyota ha desarrollado un sistema de ayuda para intersecciones (connected Vehicles, 2015), advierte al conductor de una velocidad excesiva al aproximarse a la intersección si el semáforo está en fase roja o se va a poner en rojo. El sistema también tiene un avisador de cambio de fase a verde para intentar minimizar los atascos en las ciudades al advertir al conductor de que el semáforo va a cambiar de fase roja a verde, de esta forma se consigue reducir el tiempo de reacción del conductor.

Toyota también tiene un control de crucero adaptativo (CCA) mediante comunicaciones V2V, este sistema mejora al CCA mediante radar en que el sistema es más ágil y preciso para detectar aceleraciones y deceleraciones siendo capaz de mantener la distancia entre vehículos mejor y reaccionar con correcciones más suaves y precisas.

2.1.3.5. eCall

El “*eCall*” (european Commission, 2015) es un sistema de seguridad electrónico que en caso de accidente grave llamará automáticamente a los servicios de emergencia. Incluso si el conductor queda inconsciente, el sistema informará al personal de rescate del lugar exacto del choque, y el rescate estará en camino en pocos minutos. Se quiere que este sistema esté en funcionamiento en toda la Unión Europea además de Islandia, Noruega y Suiza.

En cuanto el dispositivo *eCall* del vehículo detecte un impacto producido por un accidente, automáticamente enviará una llamada de emergencia al centro de emergencia más cercano y le transmitirá la ubicación geográfica exacta del lugar del accidente junto con otros datos. El sistema *eCall* también se puede realizar una llamada manualmente pulsando un botón, esto es útil por ejemplo, si se presencia un accidente. En el momento de la llamada además de la conexión automática de datos se realiza una conexión de voz entre el vehículo y el centro de llamadas de emergencia. De esta manera, cualquier ocupante del vehículo capaz de responder preguntas puede proporcionar detalles adicionales del accidente al centro de llamadas.

2.1.3.5.1. Beneficios del *eCall*

En 2009, murieron unas 35000 personas y 1,5 millones resultaron heridas en unos 1,15 millones de accidentes de tráfico en la red de carreteras de la UE (para la Seguridad del Automóvil, 2007). Al obtenerse una alerta inmediata cuando haya un accidente y conocer la ubicación exacta del lugar de este podría acortar los tiempos de respuesta de los servicios de emergencia en un 50% en zonas rurales y en un 40% en zonas urbanas. Gracias a esta ganancia de tiempo, se espera que el *eCall* evite varios cientos de muertes en la Unión Europea cada año, se estima que las muertes y heridos se reducirán entre un 5% y un 15%. El *eCall* también actuará en el rápido tratamiento de las personas heridas, dándoles a las víctimas de accidentes una mejor perspectiva de recuperación. Llegando antes al lugar del accidente también se podrán despejar antes los restos del choque, reduciendo así el riesgo de accidentes secundarios, disminuyendo los tiempos de congestión, consumo de combustible y emisiones de CO₂.

En términos de financiación, las pérdidas económicas de la UE como consecuencia de los accidentes ascienden a más de 160.000 millones de euros por año. Si todos los coches estuvieran equipados con el sistema *eCall*, se podrían ahorrar 20.000 millones de euros como mínimo, en esta cantidad están incluidos el ahorro en servicios sanitarios al tener que tratar dolencias menos graves, ahorro de combustible al disminuirse los atascos producidos por los accidentes y la mejora de la productividad (para la Seguridad del Automóvil, 2007).

2.1.3.5.2. Implantación del *eCall*

Para la implantación del sistema *eCall* se creó el consorcio HeERO (Rooke, 2015) el cual está formado por nueve países: Croacia, República Checa, Finlandia, Alemania, Grecia, Italia, Los países bajos, Rumanía y Suecia. Durante tres años (desde Enero de 2011 a Diciembre de 2013), los socios del consorcio han llevado a cabo el lanzamiento de un servicio de llamadas de emergencia en el vehículo basado en un servicio telefónico de gestión unificada de emergencia, 112 interoperable y armonizado.

La segunda fase del proyecto HeERO - HeERO2 - comenzó el 1 de enero de 2013 con una duración 2 años. Al consorcio se han unido seis nuevos países: Bélgica, Bulgaria, Dinamarca, Luxemburgo, España y Turquía. Además, otros países que deseaban asociarse al HeERO, pero que por diversos motivos no pudieron, son ahora *partners* asociados, un estatus que les permite beneficiarse de las experiencias de los pilotos HeERO 1 y 2 pero sin acceso a financiación.

2.1.3.6. Desarrollo de plataformas

Para el desarrollo de todos estos sistemas y plataformas cooperativas son necesarios dispositivos capaces de establecer la posición del vehículo o de la infraestructura y comunicarse entre todos los agentes implicados. Por ello hay grandes compañías de semiconductores (NXP (nxp, 2015), Qualcomm (Qualcomm, 2015)) y de desarrollo de equipos (TNO (tno, 2015), Hitachi (Lagana et al., 2012), Unex (Unex, 2015)) que han centrado sus esfuerzos en la creación de estos dispositivos. De momento al ser un nicho de mercado y un sector incipiente, la información disponible de las características internas de los dispositivos es limitada e incluso inexistente.

2.2. Revisión Bibliográfica

Una vez presentados el estado de la tecnología implementada en los vehículos actuales se pasa a hacer un repaso en esta sección del estado de investigación en el campo de los sistemas cooperativos así como en los campos de investigación relacionados, VANETs, redes de comunicación y algoritmos de enrutamiento.

2.2.1. Sistemas cooperativos (C-ITS)

Los sistemas cooperativos se definen como aquellos sistemas de seguridad, eficiencia y confort en el entorno vial basados en el intercambio de información entre vehículos y/o infraestructura, mediante comunicaciones inalámbricas (Jiménez, 2012). Incluso se puede extender el intercambio de información a usuarios que no están en el interior de un vehículo en el instante de recibir el servicio. Es decir, el vehículo se encuentra conectado en un entorno cooperativo, lo que posibilita que además de poseer datos propios y de percibir su entorno por medio de sensores embarcados, puede recibir información de otros vehículos (sensores e información del vehículo), de la infraestructura o de centrales de tráfico, así como disponer de posicionamiento por satélite o por medio de balizas en la vía.

Las aplicaciones en los diferentes campos es casi infinita por lo que nos vamos a enfocar en sistemas cooperativos dedicados a la seguridad vial dado que la esta tesis está centrada en este campo.

Los sistemas de seguridad en el automóvil tradicionalmente se clasificaban en dos grandes grupos como eran seguridad activa o primaria y seguridad pasiva o secundaria. Sin embargo, la proliferación de unidades electrónicas de procesamiento en el vehículo y la compartición de información entre las mismas han dado lugar a que la frontera pase a ser mucho más difusa teniéndose que actualizar estos términos de la siguiente manera:

- Sistemas de asistencia al conductor: sistemas desarrollados con el objetivo fundamental de evitar errores en el conductor durante la circulación, por ejemplo en (Handmann et al., 2000) explican un avisador al conductor de objetos en la vía.
- Sistemas de seguridad primaria: dispositivos que contribuyen a evitar o minimizar los actos y comportamientos inseguros del conductor y del propio vehículo, susceptibles de causar accidentes como por ejemplo en (McCall and Trivedi, 2006) con un avisador de cambio de carril.
- Sistemas precolisión: elementos que permiten, utilizando información proporcionada por sensores embarcados, que se desarrollen acciones coordinadas por los sistemas

de control del vehículo y protección de ocupantes, durante las fases de precolisión y colisión, con el objeto de reducir o eliminar daños. En (García et al., 2013) se describe un sistema de precolisión que actúa sobre los frenos y la dirección en caso de detección de un accidente inminente.

- Sistemas de seguridad secundaria: dispositivos que tienen como finalidad evitar o minimizar los daños producidos a personas o cosas transportadas en el vehículo o con las que éste puede interactuar, cuando tiene lugar un accidente. En (Broughton, 2003) se describen los beneficios observados entre 1980 y 1998 de la implementación de la seguridad secundaria en Reino Unido.
- Sistemas de seguridad terciaria: dispositivos embarcados para el registro de información dinámica durante la fase previa y de colisión (caja negra), así como aquellos que tienen como función mejorar la fase de aviso y actuación de los equipos de rescate de víctimas del siniestro. En (Chauvel and Haviotte, 2010) explican los accidentes típicos en Francia y el tiempo de reacción de los servicios médicos así como las mejoras potenciales de un sistema de aviso de accidente.

Los principales sistemas cooperativos en la mejora de la seguridad vial según (ERTICO, 2016) se pueden clasificar en:

- Aviso al conductor de situaciones especiales presentes en la carretera con el fin de que el conductor esté alertado de las mismas y pueda actuar en consecuencia: aviso de zonas de la carretera con características especiales, carretera deslizante, condiciones climatológicas o condiciones de conducción fuera de lo normal como en (Papadimitratos et al., 2009).
- Sistemas de asistencia, cuyo objetivo fundamental es reducir la carga sobre el conductor, permitiéndole estar más atento a las tareas más importantes para la seguridad y reducir su cansancio: control de la velocidad, aviso de distancia de seguridad o reducción cooperativa del deslumbramiento. En (Moriarty and Langley, 1998) proponen un sistema cooperativo para la selección del carril más adecuado en una autopista en función de las velocidades del resto de vehículos.
- Sistemas de evitación de colisiones orientados específicamente a detectar situaciones críticas de accidente inminente (a ser posible, todavía en la fase de accidente evitable) para proporcionar los avisos correspondientes o actuar sobre los sistemas del vehículo si es el caso.
  - Aviso de colisión por alcance como en (Huang et al., 2002).
  - Aviso de vehículo anormalmente lento o parado en la vía.

- Detección de vehículos u obstáculos en general.
  - Supervisión del ángulo muerto para cambio de carril, como en (Doshi and Trivedi, 2010).
  - Asistencia para adelantamiento seguro, como en (Olaverri-Monreal et al., 2010).
  - Evitación de colisiones en intersecciones, como en (Tan and Huang, 2006) y en (Doshi and Trivedi, 2010).
  - Aviso para evitar sobrepasar un semáforo en rojo, como en (Geyer et al., 2011).
  - Aviso de vehículo circulando en sentido contrario.
-
- Sistemas de interacción entre seguridad primaria y secundaria, que emplean la información propia de sistemas de seguridad primaria para el despliegue de medidas de seguridad secundaria: accionamiento de sistemas precolisión.
  - Sistemas de seguridad terciaria de actuación tras el accidente: llamada de emergencia *eCall*.
  - Automatización, entendiendo como tal el control del vehículo para la circulación y no tanto la actuación automática sobre algún sistema como en los sistemas anteriores: autopista automatizada.

Para que un sistema cooperativo pueda funcionar necesita una red de comunicaciones en la que apoyarse. En la siguiente sección se hace una pequeña introducción al estándar de comunicaciones OSI y posteriormente se estudiarán las opciones existentes y las soluciones tomadas.

2.2.2. Modelo OSI

Los estándares son fundamentales en comunicaciones ya que garantizan la interoperabilidad de los medios y dispositivos, impidiendo la existencia de tecnologías propietarias y permitiendo el desarrollo de aplicaciones universales. Por tanto en esta sección se hará una pequeña introducción a la división de los estándares de comunicaciones que afectan a los vehículos, así como a la estructura lógica que deben poseer para que puedan interactuar entre ellos.

Propiamente el modelo OSI es una recomendación o estándar de comunicaciones que deben seguir los fabricantes para que todos los dispositivos fabricados puedan comunicarse entre ellos con independencia de la tecnología utilizada, pero dada la naturaleza de la tesis es necesario hacer un inciso sobre la estructura de las redes de comunicación y sus diferentes partes.

Como se ha comentado el modelo OSI es una guía o recomendación de cómo deben estar estructurados los sistemas de comunicación. El modelo OSI (modelo de interconexión de sistemas abiertos (en inglés, *Open System Interconnection*)) o *torre OSI* es un modelo de referencia para los protocolos de la red de arquitectura en capas, creado en el año 1980 por la Organización Internacional de Normalización (ISO, *International Organization for Standardization*). Se ha publicado desde 1983 por la Unión Internacional de Telecomunicaciones (UIT) y, desde 1984, la ISO también lo publicó como estándar.

El modelo de referencia OSI, es una normativa formada por siete capas que define las diferentes fases por las que deben pasar los datos para viajar de un dispositivo a otro sobre una red de comunicaciones (figura 2.4). Siguiendo el esquema de este modelo se han creado numerosos protocolos. El advenimiento de protocolos más flexibles donde las capas no están tan desmarcadas y la correspondencia con los niveles no era tan clara puso a este esquema en un segundo plano. Pero sigue siendo un referente en las comunicaciones.

FIGURA 2.4: Torre OSI

Este modelo está dividido en siete capas o niveles:

- Nivel físico: es la primera capa del Modelo OSI. Es la que se encarga de la topología de red y de las conexiones globales de la computadora hacia la red, se refiere tanto al medio físico como a la forma en la que se transmite la información.
- Nivel de enlace de datos: esta capa se ocupa del direccionamiento físico, del acceso al medio, de la detección de errores, de la distribución ordenada de tramas y del control del flujo.

- Nivel de red: se encarga de identificar el enrutamiento existente entre una o más redes. En este nivel se realiza el direccionamiento lógico y la determinación de la ruta de los datos hasta su receptor final.
- Nivel de transporte: capa encargada de efectuar el transporte de los datos de la máquina origen a la de destino, independizándolo del tipo de red física que esté utilizando.
- Nivel de sesión: esta capa es la que se encarga de mantener y controlar el enlace establecido entre dos computadores que están transmitiendo datos de cualquier índole.
- Nivel de presentación: el objetivo de esta capa es encargarse de la representación de la información, de manera que aunque distintos equipos puedan tener diferentes representaciones internas de caracteres los datos lleguen de manera reconocible.
- Nivel de aplicación: ofrece a las aplicaciones la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos.

2.2.3. VANETs

Una red de comunicaciones VANET es una red de comunicaciones entre vehículos (Vehicular Ad hoc NETwork). Para la creación de las VANETs se utiliza tecnología de comunicaciones de corto alcance (DSRC, *Dedicated Short Range Communication*) basada en el estándar ETSI ITS-G5 (663, 2009) que es el estándar europeo equiparable al estándar IEEE 802.11p (for Information technology, 2010). Esta tecnología de comunicaciones sirve para dar soporte en tiempo real a los sistemas cooperativos para el intercambio de información. Las VANETs están incluidas en las denominadas WAVE (Wireless Access in Vehicular Environments) que permiten diferentes tipos de comunicaciones entre los interlocutores de la red.

En (Al-Sultan et al., 2014) hacen una descripción de los elementos de la arquitectura de una VANET necesarios para su correcto funcionamiento. Los principales componentes del sistema son la unidad de aplicación (*Application Unit*, AU), la unidad de abordó (*OnBoard Unit*, OBU) y las estaciones de apoyo en los laterales de la carretera (*RoadSide Unit*, RSU). La principal diferencia entre una RSU y una OBU es la finalidad para la que han sido diseñadas, la OBU esta imbuida en un vehículo y por lo cual puede cambiar de posición geográfica, mientras que la RSU es estática y está en una posición fija durante toda su vida útil. La finalidad de la RSU y la OBU es dar soporte a las comunicaciones a aplicaciones de diversa naturaleza, seguridad, control de tráfico, información viaria .... Estas aplicaciones

pueden encontrarse también en centros de control de tráfico o en los propios vehículos. Cada vehículo está equipado con una OBU y un conjunto de sensores para recopilar y procesar la información para, a continuación, enviar esta mediante un mensaje a otros vehículos o RSUs a través de la red inalámbrica. También puede llevar una única AU o múltiples que utilicen las aplicaciones proporcionadas mediante la conexión con el OBU. La RSU también puede conectarse a internet mediante conexiones adicionales. La distinción entre el AU y el OBU puede ser solamente lógica si el OBU posee una HMI (*Human-Machine Interface*).

También en (Al-Sultan et al., 2014) hacen una pequeña descripción de las tecnologías utilizadas en las VANETs para la comunicación inalámbrica. Estas se pueden crear utilizando sistemas de telefonía móvil (2/2.5/2.75/3/4/4.5/5G) que tienen diferentes anchos de banda, WLAN/Wi-Fi que tiene un rango de comunicación aproximado de 140 metros en exteriores, WiMax con el estándar IEEE 802.16e que puede llegar a coberturas de 15 km, DSRC/WAVE que pueden llegar a rangos de comunicación de 1 km o una combinación de todas las anteriores.

Las VANETs actuales se están diseñando según la arquitectura “acceso a comunicaciones terrestres móviles” conocida como CALM por sus siglas en inglés “*Communications Access for Land Mobsiles*”, que engloba un conjunto de normas y estándares de comunicación para C-ITS (Ver Figura 2.5). Esta arquitectura y normas están especificadas en la ISO/TC 204 WG16 con la cual se ha intentado normalizar y aunar las diferentes comunicaciones móviles. Las principales características de la arquitectura CALM son:

- Los estándares contenidos en ella no solo definen las comunicaciones sino que define las relaciones de los diferentes bloques.
- La idea principal de la arquitectura es la abstracción de las aplicaciones ITS de los protocolos de comunicación, lo cual posibilita la selección automática de diferentes perfiles de comunicación en función de las necesidades de las aplicaciones, la capacidad de la tecnología usada para el acceso al medio y la saturación actual del medio de transmisión.
- Define la información que transmiten las diferentes capas de la arquitectura de forma concisa y precisa.
- Sigue el concepto BSMD por sus siglas en inglés (*Bounded Secured Managed Domain*): Es un concepto introducido en la ISO 21217 la cual especifica que cada estación de ITS supone un dominio el cual está autogestionado, es seguro y está encapsulado de esta manera la seguridad y manejabilidad aumenta y hace que los módulos ITS sean más robustos que un sistema descentralizado.

FIGURA 2.5: Arquitectura CALM

En la Figura 2.5 se pueden observar los diferentes bloques definidos en la arquitectura CALM las cuales se pueden relacionar con las capas del modelo OSI (*Open System Interconnection*) que se ha simplificado y ampliado con el fin de definir la arquitectura de un módulo ITS (RSU/OBU), esta arquitectura consta de seis partes (aplicaciones (*Applications*), gestión (*Management*), acceso (*Access*), redes y transporte (*Networking & Transport*), servicios (*Facilities*) y seguridad (*Security*)). Las capas definidas en la arquitectura CALM se pueden relacionar con la pila OSI como:

- Acceso: capas OSI 1 y 2.
- Redes y Transporte: capas OSI 3 y 4.
- Servicios: capas OSI 5, 6 y 7.

Donde la funcionalidad de las diferentes partes es:

- El bloque de seguridad proporciona los medios necesarios para asegurar las comunicaciones y el propio módulo ITS.

- El bloque de gestión proporciona las herramientas necesarias para el mantenimiento y configuración del módulo ITS.
- Los bloques, capas y entidades están interconectados a través de interfaces, que por lo general son “puntos de acceso de servicio” (SAP) [ISO 24102-3].
- Todo esto es utilizado por aplicaciones de ITS. Las aplicaciones se conectan a las herramientas de comunicación a través de una “interfaz de programación de aplicaciones” (API), que es la encargada de proporcionar las funcionalidades de la MA-SAP (SAP de gestión), FA-SAP (SAP de servicios) y SA-SAP (SAP de seguridad) (Ver Figura 2.5).

Se considera que la arquitectura CALM va más allá de la definición de las comunicaciones porque identifica la funcionalidad de los seis bloques y considera los módulos ITS como un elemento BSMD. Las normas CALM no especifican ninguna aplicación C-ITS, pero sin embargo definen claramente su método de interconexión con las comunicaciones.

La parte más importante de la arquitectura CALM es la parte de comunicaciones, porque aunque los módulos ITS puedan contener aplicaciones la parte esencial que deben tener todos para posibilitar la interoperabilidad entre ellos es la parte de comunicaciones. El bloque de “redes y transporte” es el más importante para el buen funcionamiento de las comunicaciones y que los paquetes con datos alcancen su destino. Este se organiza a través de los algoritmos de enrutamiento que consisten en diferentes métodos para conseguir transmitir los mensajes por la red. Por lo cual en las siguientes secciones vamos a hacer un repaso a los diferentes algoritmos de enrutamiento propuestos por la comunidad científica.

2.2.4. Algoritmos de enrutamiento

El enrutamiento (o encaminamiento o ruteo) es la acción de buscar un camino o ruta entre dos puntos o nodos en una red entre todos los posibles. Cuando se trata de una red inalámbrica, este camino se puede realizar mediante técnicas multisalto (*multihop*), las cuales permiten que los mensajes recorran el camino desde un origen a un destino mediante una cadena de saltos a través de una serie de nodos intermedios de la red, donde cada nodo intermedio se comporta como un *router* o encaminador. La complejidad de la búsqueda de este camino óptimo se complica cuando los nodos de la red son móviles, como es el caso de los vehículos conectados. Hay que definir una medida para poder comparar el rendimiento de diferentes redes y tecnologías, denominada métrica, la cual no es exclusiva. La métrica es un valor que toman los diferentes protocolos de enrutamiento para poder determinar cuál es la mejor ruta hacia un nodo destino de la red. No es inusual que exista más de un único camino hacia un nodo y, por lo tanto, se deberá emplear algún método para determinar

cuál de esos caminos es mejor. En algunos casos será aquel cuya distancia es menor o en otros casos es aquella que tiene mejor ancho de banda. Esto dependerá del protocolo de enrutamiento que se esté utilizando, ya que cada uno usa una métrica diferente. Como se ha comentado existen diferentes métricas, por ejemplo, el número de saltos (el número de saltos o *hops* es el número de nodos involucrados en la comunicación sin contar al emisor) necesarios para ir de un nodo a otro (es sencilla y suele ofrecer buenos resultados), medición del retardo de tránsito entre nodos vecinos (se expresa en unidades de tiempo y sus valores no son constantes sino que dependen del tráfico de la red).

La mejor ruta es aquella que mantiene acotado el retardo entre pares de nodos de la red, consigue ofrecer altas cadencias efectivas independientemente del retardo medio de tránsito y ofrece el menor costo. El criterio más sencillo es elegir el camino más corto, es decir la ruta que pasa por el menor número de nodos. Una generalización de este criterio es el de “coste mínimo”. En general, el concepto de distancia o coste de un canal es una medida de la calidad del enlace basado en la métrica que se haya definido (Rao and Murthy, 2005).

La dificultad del enrutamiento en las VANETs radica en que los nodos de la red tienen la posibilidad de cambiar su posición geográfica, por lo cual se modifican de manera continua las métricas y calidad de la conexión entre los diferentes nodos de la red, lo que implica que un nodo no pueda determinar la ruta hasta otro de manera determinista, por lo que se han tenido que desarrollar nuevos algoritmos de enrutamiento específicos para las VANETs. Estos algoritmos ya no se pueden basar como en los algoritmos tradicionales, tanto redes cableadas o *Wireless* donde los nodos de destino tienen una dirección conocida e identificada, en mensajes de “*broadcast*” o difusión (a todos los nodos contenidos en la red), “*multicast*” (a los nodos de varias redes), “*anycast*” (al nodo más cercano en latencia) o “*unicast*” (a un nodo específico).

Dada la forma de operar de las VANETs puede que interesar transmitir determinados mensajes a una posición o a un área geográfica concreta. “Geocast” es la acción de difundir un mensaje en un área geográfica determinada, existen nuevos tipos de mensajes, adicionales a los tradicionales (Festag, 2014):

- GeoUnicast: consiste en transmitir un mensaje al vehículo que se encuentre en una posición concreta.
- GeoBroadcast: consiste en transmitir un mensaje a todos los vehículos que se encuentren en un área concreta.
- GeoAnycast: consiste en transmitir un mensaje al vehículo más cercano que se encuentre en un área concreta.

Estos nuevos mensajes han introducido una complejidad adicional a los algoritmos tradicionales de enrutamiento lo que ha provocado que sea necesaria la exploración de nuevos métodos de enrutamiento, que a continuación pasamos a detallar.

TABLA 2.1: Enumeración de protocolos de enrutamiento en VANETs (Li and Wang, 2007)

<i>Routing Protocol</i>	<i>Routing Type</i>	<i>Position Information? (How to use)</i>	<i>Hierarchical Structure?</i>	<i>Network Simulator</i>	<i>Simulation Scenario</i>
AODV (Perkins and Royer, 1999)	Unicast	No	No	-	-
DSR (Johnson and Maltz, 1996)	Unicast	No	No	-	-
GPSR (Karp and Kung, 2000)	Unicast	<i>Packet Forwarding</i>	No	-	-
PRAODV/ PRAODV-M (Namboodiri et al., 2004)	Unicast	<i>Route-Selection (lifetime prediction)</i>	No	NS2	<i>Simple highway model (20km segment only)</i>
AODV-bis (Ooi and Faisal, 2004)	Unicast	<i>Route-Req-Forwarding</i>	No	-	-
GSR (Lochert et al., 2003)	Unicast	<i>Packet Forwarding</i>	No	NS2	<i>Real city model (from map)</i>
GPCR (Lochert et al., 2005)	Unicast	<i>Packet Forwarding</i>	No	NS2	<i>Real city model (from map)</i>
A-STAR (Liu et al., 2004)	Unicast	<i>Packet Forwarding (also use traffic info.)</i>	No	NS2	<i>Grid city model</i>
COIN (Blum et al., 2003)	Unicast	<i>Cluster Formation</i>	Yes	Own	<i>Real highway model</i>
LORA_CBF (Santos et al., 2005)	Unicast	<i>Packet Forwarding (also location prediction)</i>	Yes	OPNET	<i>Simple circle and square road</i>
Flooding	Broadcast	No	No	-	-
UMB (Korkmaz et al., 2004)	Broadcast	<i>Packet Forwarding</i>	No	Own	<i>Simple intersection road</i>
V-TRADE/ HV-TRADE (Sun et al., 2000)	Broadcast	<i>Classify Forwarding Group</i>	No	Own	<i>Simple intersection</i>
BROADCASTMM (Durreli et al., 2005)	Broadcast	<i>Formation of Cells</i>	Yes	Own	<i>Simple highway</i>
Msg Dis Protcl (Briesemeister et al., 2000)	Geocast	<i>Packet Forwarding</i>	No	Own	<i>Simple highway model (10 km long)</i>
IVG (Bachir and Benslimane, 2003)	Geocast	<i>Packet Forwarding</i>	No	Glomosim	<i>Simple highway model (10 km long, 100/200 nodes)</i>
Cached Geocast (Maihofer and Eberhardt, 2004)	Geocast	<i>Packet Forwarding</i>	No	NS2	<i>Quadratic network (size from 1 km to 4km, 100 nodes)</i>
Abiding Geocast (Maihöfer et al., 2005)	Geocast	<i>Packet Forwarding</i>	No	-	-

En (Li and Wang, 2007) hacen un análisis de las características de arquitectura y protocolos de enrutamiento desarrollados para VANETs. Estos los dividen en cinco categorías:

- *Ad Hoc Routing*: dada la naturaleza de las VANETs y MANETs (*Mobile Ad-Hoc Networks*) que tienen que autorganizarse, autogestionarse, tienen poco ancho de banda y corto alcance. En esta categoría engloban los algoritmos diseñados a tal propósito.
- *Position-Based Routing*: por lo general las VANETs están restringidas a movimientos bidireccionales, vías de circulación, en esta categoría engloban los algoritmos que organizan la red basada en las posiciones geográficas y/o la dirección de circulación.
- *Cluster-Based Routing*: en esta categoría engloban los algoritmos que se segmentan el espacio creando regiones o *clusters* para organizar la red.

- *Broadcast Routing*: en esta categoría están los algoritmos de *broadcast*, un mensaje es difundido por toda la red.
- *Geocast Routing*: en esta categoría están los algoritmos que son capaces de realizar un *broadcast* limitado a un área geográfica.

Finalmente hacen un resumen en una tabla que hemos reproducido en la Tabla 2.1 donde listan algunos algoritmos de enrutamiento que serán explicados con detalle en los siguientes apartados, el tipo de enrutamiento que proporcionan, que método usan para transmitir la información de su posición, si el algoritmo posee una estructura jerarquizada, y cómo ha sido testeado, tipo de simulador y escenario simulado.

Existen varios tipos de transmisión de la posición de los diferentes de algoritmos:

- *Packet Forwarding*: difusión de la información a todos los nodos que estén en el campo de visual de comunicaciones.
- *Cluster Formation*: difusión solo a los miembros de un *cluster* formado previamente.
- *Formation of cells*: entrega de la información solo a los nodos que se encuentran en el interior de un área concreta.
- *Route selection*: se encarga de calcular la ruta un RSU.

A continuación se procede a detallar el estado del arte para los diferentes tipos de enrutamiento, analizando algoritmos de enrutamiento específicos para redes donde la movilidad es esencial y VANETs. La frontera entre definiciones no queda del todo bien definida haciendo que en función de los autores, fecha de publicación y país definan el tipo de algoritmo con el mismo nombre que en las redes cableadas cuando en realidad se refieren a algoritmos de enrutamiento geográfico, por ejemplo Unicast-GeoUnicast. Por esa razón se ha tomado la decisión de agruparlos en una categoría única en función de la naturaleza de la transmisión.

2.2.4.1. Algoritmos GeoUnicast

Los algoritmos GeoUnicast son aquellos que sirven para la comunicación entre dos nodos de la red, esta comunicación es punto a punto. Esta comunicación puede ser directa porque ambos nodos se vean mutuamente o mediante diversos saltos. El resto de nodos de la red no tiene que tratar el mensaje más que como simple *router* si fuera necesario como se puede observar en la Figura 2.6. La problemática existente con los algoritmos de GeoUnicast reside en la estrategia adoptada para hacer llegar el mensaje al destinatario,

buscando el mínimo número de saltos o más saltos pero con mejor calidad de la señal: ¿debe existir confirmación de que el mensaje ha sido recibido o no?, ¿cómo identificar donde se encuentra el destinatario del mensaje?, etc. Por esa razón existe una extensa literatura donde se presentan diversas estrategias en los algoritmos de GeoUnicast, a continuación pasaremos a detallar los algoritmos que han tenido más relevancia en el campo del georrutamiento con mensajes *Unicast* en ITS.

FIGURA 2.6: GeoUnicast: los vehículos naranjas son los que están en comunicación, los vehículos verdes son los que actúan como repetidores o conectores entre los vehículos emisor y receptor, por último los vehículos azules no intervienen en la comunicación (636-1, 2014)

Un algoritmo básico de unicast es el *Greedy* el cual se apoya en el geoposicionamiento. El mensaje se entrega al nodo más cercano al destino que se encuentre en el horizonte de visión, este puede ser el objetivo del mensaje o un nodo que actuará como repetidores o conectores entre los nodos.

En (Bhoi and Khilar, 2014) proponen un algoritmo de georrutamiento unicast para ciudades denominado IJS, donde se necesita un HV (*Helping Vehicle*) en cada intersección que actúa como relé y distribuidor de paquetes dado que este algoritmo solo permite la comunicación directa entre OBUs que se encuentren en la misma vía circulando en la misma dirección. Finalmente comparan su algoritmo con A-Star (Liu et al., 2004), GyTAR (Jerbi et al., 2009) y GSR (Lochert et al., 2003) obteniendo mejores resultados.

En (Johnson and Maltz, 1996) se presenta DSR, *Dynamic Source Routing*, un algoritmo de unicast que utiliza un enrutamiento dinámico de la fuente que se adapta rápidamente a los cambios de los nodos de la red, no como otros algoritmos que utilizan un vector de distancia o utilizando identificadores de calidad de la conexión. Este algoritmo no está pensado específicamente para VANETs sino para ser utilizado en dispositivos portátiles que su movimiento no sea continuado y existan largos periodos de estancia estática y por lo cual la necesidad de reconfiguración de la red no sea constante.

En (Karp and Kung, 2000) se presenta el algoritmo de unicast GPSR, *Greedy Perimeter Stateless Routing*, este consiste en un algoritmo de *Greedy* optimizado para reducir la necesidad de intercambio de paquetes de conocimiento de la red, son mensajes que se difunden por la red de tipo broadcast de un salto llamados *beacons*, el algoritmo está basado en geoposicionamiento. Según los autores tienen un éxito en simulación en la entrega de

mensajes del 94% para redes de 200 nodos, mejoran la flexibilidad de DSR (Johnson and Maltz, 1996) dado que el algoritmo es más escalable, dado que GPSR solo tiene en cuenta los vecinos próximos y no toda la red conocida.

En (Lochert et al., 2005) proponen una solución para el problema del enrutamiento en ciudades, dado que a veces se produce el problema que el nodo más cercano geográficamente al objetivo lleva a un camino sin salida al no tener comunicación directa por existir un objeto que la impida. Su solución es el algoritmo GPCR, *Greedy Perimeter Coordinator Routing*, que determina la existencia de un cruce mediante la densidad de nodos en un área determinada, y de esta manera difundir el mensaje por la calle que más se aproxime al objetivo.

En (Saleet et al., 2011) proponen un algoritmo de unicast, IGRP (*Intersection-Based Geographical Routing Protocol*). Es un algoritmo desarrollado para ciudades, el cual necesita tener un grafo de la ciudad con sus calles e intersecciones para determinar cuál es el camino más óptimo para entregar el paquete al nodo de destino.

2.2.4.2. Algoritmos Broadcast

Los algoritmos de broadcast son aquellos que sirven para difundir un mensaje a todos los nodos de una red (Figura 2.7), en el caso de las VANETs no hay subredes por lo cual el mensaje se difundirá por toda la red. El protocolo estándar indica que cuando un nodo recibe un mensaje de broadcast este debe difundirlo, siempre que no sea repetido, lo cual puede provocar un efecto de inundación o *flooding* haciendo que la red se sature. Otro problema asociado al broadcast y las VANETs es cuando existe una falta de conexión entre dos subgrupos de ésta. Todos estos problemas han sido tratados en los artículos que comentamos a continuación proponiendo diferentes soluciones.

FIGURA 2.7: Broadcast: los vehículos naranjas son los que están en comunicación (636-1, 2014)

En (Fang and Luo, 2013) se propone un algoritmo de difusión basado en un doble temporizador mediante el cual se evita la duplicidad de paquetes y la sobrecarga de la red. El primer temporizador se utiliza para asegurarse de que todos los nodos colindantes sean capaces de recibir el mensaje y calcular la distancia relativa a la última posición del emisor

del paquete. Cuando el primer temporizador termina, el segundo temporizador se activa para asegurarse de que sólo el receptor más alejado reenvía el paquete.

En (Ni et al., 1999) y (Tonguz et al., 2010) se afronta el problema de las tormentas de datos producidas por los algoritmos de broadcast. Para solucionarlo se define un factor de probabilidad de reemisión el cual está estrechamente relacionado con la distancia del módulo receptor y el módulo emisor, y con el tiempo transcurrido desde la emisión del paquete.

En (Mehar et al., 2012) presentan el algoritmo “*Dissemination protocol for Heterogeneous Cooperative Vehicular Networks*”, DHVN, para solucionar los problemas de las VANETs por conectividad, los canales no fiables y por las características de las ciudades. DHVN tiene en cuenta la topología de las carreteras, la capacidad de comunicación heterogénea de los vehículos, la conectividad de red y la posible partición en caso de baja densidad de tráfico.

En (Viriyasitavat et al., 2010) se presenta UV-CAST, “*Urban Vehicular Broadcast Protocol*”, es un protocolo de broadcast especializado en la complejidad asociada al callejero de las ciudades. Sin un control adecuado un sólo mensaje de broadcast puede provocar una tormenta de datos que inutilizara toda la VANET. Para solucionarlo UV-CAST limita los mensajes por intersecciones. Este protocolo ha sido testeado en el simulador SUMO (Krajzewicz et al., 2012) en un escenario de Manhattan (Nueva York, EEUU) con excelentes resultados de pérdida de paquetes (99% de recepción de paquetes), por lo cual se probó en el centro de Pittsburg (Pensilvania, EEUU) con buenos resultados, los vehículos a menos de 300 metros del lugar de emisión reciben los mensajes.

En (Korkmaz et al., 2004) se presenta el algoritmo de broadcast para zonas urbanas UMB, “*Urban Multi-Hop Broadcast*”. Este protocolo necesita un módulo de repetición en cada intersección del entorno urbano el cual es el responsable de la difusión en todas direcciones y determina que vehículo entre dos intersecciones tiene que actuar como relé para llegar a la siguiente intersección. Se demuestra a través de simulaciones que el protocolo tiene una alta tasa de éxito.

En (Durrezi et al., 2005) se presenta el protocolo BROADCAST, el cual es un broadcast especializado en carreteras el cual divide esta en celdas o segmentos en el cual existe un RSU en el centro que es capaz de comunicarse con los RSU de las celdas adyacentes. Cuando se produce un mensaje de difusión solo los RSU reenvían el mensaje.

2.2.4.3. Algoritmos GeoBroadcast

Llamamos algoritmos de GeoBroadcast a los algoritmos de broadcast focalizados en un área geográfica limitada (Figura 2.8). Este tipo de algoritmos han sido ideados y desarrollados específicamente para difundir mensajes en VANETs.

FIGURA 2.8: GeoBroadcast: los vehículos naranjas son los que están en comunicación, los vehículos verdes son los que actúan como repetidores o conectores entre los vehículos emisor y receptor, por último los vehículos azules no intervienen en la comunicación (636-1, 2014)

Debido a las características de la red y de sus posibles aplicaciones hay veces que interesa acotar el rango de un mensaje de broadcast, de esta manera se consiguen varios beneficios: Se evita los problemas de *flooding*, se reduce la carga de la red, se pueden difundir mensajes específicos en un área concreta (accidente, obras, información de tráfico, etc). El desarrollo de este tipo de mensaje ha tenido no pocas dificultades pues hay que tener en cuenta muchas variables como por ejemplo cómo actuar si el emisor del mensaje no se encuentra dentro del área de difusión, cómo evitar el *flooding* en el interior del área de destino, etc. por esa razón en este apartado se comentan los artículos que desarrollan diferentes estrategias y algoritmos de GeoBroadcast.

En (Bachir and Benslimane, 2003) se presenta el algoritmo *Inter-Vehicles Geocast* (IVG) que difunde un mensaje de alerta por la red VANET, este se difunde por el área en función del tiempo transcurrido desde la emisión del aviso y la dirección de los vehículos en relación al objeto emisor.

En (Maihofer, 2004) se realiza un análisis de diferentes técnicas de geocast. En este artículo se explican las diferentes técnicas utilizadas para hacer llegar el mensaje al área objetivo y por último se simulan los algoritmos más representativos de cada técnica de enrutamiento en el simulador NS-2 con diferente número de nodos [100-1000] y diferente densidad de estos. Los resultados obtenidos se pueden observar en la Tabla 2.2, extraída de dicho artículo, donde las seis técnicas están ordenadas en columnas y en las filas están los parámetros analizados:

- Estrategia: la estrategia de difusión utilizada por cada algoritmo.

- Escalabilidad: el tamaño de red que es capaz de soportar el algoritmo si los mensajes se pueden enviar una vez o múltiples veces.
- Complejidad en la entrega del mensaje: cuantos reenvíos produce la emisión de un mensaje en la red.
- Memoria: cantidad de memoria utilizada para el uso del algoritmo.
- Robustez: capacidad de evitar fallos si en un nodo se produce un problema.
- Tiempo estable: indica si se pueden producir fluctuaciones en el tiempo de difusión de mensajes.
- Huecos: capacidad de los algoritmos de solucionar espacios vacíos sin nodos.

TABLA 2.2: Comparación de protocolos de geocast extraída de (Maihofer, 2004)

<i>Criterion</i>	Flooding	LBM (Ko and Vaidya, 1999)	Voronoi (Stojmenovic et al., 2003)	Mesh (Boleng et al., 2001) (Camp and Liu, 2003)	GeoGRID (Liao et al., 2000) (Liao et al., 2001)	GeoNode (Navas, 2001) (Navas and Imielinski, 1997)
<i>Fix network/ad hoc</i>	Ad hoc	Ad hoc	Ad hoc	Ad hoc	Ad hoc	Fix
<i>Estrategia</i>	Flooding	Directed flooding	Directed flooding	Multipath routing	Directed flooding	Multicast ^{8,10} , unicast ^{9,10}
<i>Escalabilidad/envío una vez</i>	Bajo	Medio	Medio	Bajo	Bajo-alto ³	Medio ^{8,10} , alto ⁹
<i>Escalabilidad/envío varias veces</i>	Bajo	Medio	Medio	Medio-alto	Bajo-alto ³	Alto
<i>Complejidad mensaje (primera vez)</i>	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)^6$, $O(\sqrt{n})^7$ +gw election	$O(\sqrt{n})$ routing protocol
<i>Complejidad mensaje (segunda vez)</i>	$O(n)$	$O(n)$	$O(n)$	$O(\sqrt{n}) - O(n)^4$	$O(n)^6$, $O(\sqrt{n})^7$ +gw election	$O(\sqrt{n})$ routing protocol
<i>Requerimientos de memoria</i>	No $O(n)^1$	Bajo $O(n)^1$	Bajo $O(n)^1$	Medio $O(n)^6$	Bajo $O(n)^{6,1}$, $O(n)^{7,2}$	Bajo-medio $O(n)^8$, $O(n)^9$, $O(g)^{10}$
<i>Robustez</i>	Medio-alto ¹¹	Medio-alto ¹¹	Medio-alto ¹¹	Medio-alto ¹¹	Medio	Medio
<i>Solución a huecos parciales</i>	Si	Limitado	Si	Limitado	Limitado	Si
<i>Entrega garantizada</i>	No	No	No	No	No	No
<i>Tiempo estable</i>	No	No	No	No	No	Si
<i>Ajuste de grupos de multicast</i>	No	No	No	No	No	Si
<i>Depende de otro protocolo</i>	No	No	No	Si	No	Si

Donde:

n = número de nodos en la red.

g = número de grupos de geocast.

j = número de grupos geocast unidos.

1: Guardado de los últimos paquetes para detectar duplicados.

2: Guardado de la información del vecino.

3: Depende de la movilidad del nodo.

4: $O(\sqrt{n})$ suponiendo una distribución regular de los nodos y no hay cambios en la topología; en el peor de los casos $O(n)$ .5: $O(n^2)$ si el estado de información se mantiene en los nodos intermedios, $O(n\sqrt{n})$ si la fuente es usada.

6: Flooding-based.

7: Ticket-based.

8: GPS-Multicast.

9: GeoRouter.

10: DNS.

11: Depende de la congestión de la red y otros parámetros (ver (Ni et al., 1999)).

En (Ko and Vaidya, 1999) se plantean cómo hacer llegar el mensaje al área objetivo para su posterior difusión en un área cuadrada. Para ello se presenta el algoritmo LBM (*Location Based Multicast*) el cual utiliza varias técnicas de difusión mediante inundación en áreas que abarcan el área destino. Estas áreas de tránsito pueden ser rectangulares, esta área de destino está definida por dos vértices el emisor y el vértice más alejado del área de difusión, lo cual

implica que cuanto más lejos este el área de destino más extensa será el área de tránsito, o triangulares donde los vértices son el emisor y dos vértices no contiguos del área de difusión.

En (Stojmenovic et al., 2003) proponen un algoritmo para determinar la ruta más óptima para alcanzar el área objetivo del mensaje mediante la utilización de diagramas de Voronoi.

En (Boleng et al., 2001) evolucionan el algoritmo LBM (Ko and Vaidya, 1999) para ello sustituyen la técnica de *flooding* en las áreas de tránsito del paquete al área de difusión de este por algoritmos de maya unicast, de esta manera siempre se elige el camino más óptimo para alcanzar el área objetivo reduciéndose los mensajes necesarios.

En (Camp and Liu, 2003) intentan solucionar el algoritmo LBM (Ko and Vaidya, 1999) mandando mensajes de comprobación del camino, si el mensaje de comprobación llega correctamente al área de destino manda los datos, si falla modifica el área de tránsito y lo vuelve a intentar, si tras algunos intentos no consigue llegar al área de destino realiza el envío del mensaje mediante *flooding*.

En (Celes et al., 2013) se presenta el algoritmo GeoSPIN, *Geocast routing based on SPatial Information*, El cual se basa en el conocimiento de la ruta del vehículo para poder predecir rutas y redes de encaminamiento de paquetes.

En (Maihofer and Eberhardt, 2004) proponen la existencia de *buffers* de almacenamiento de paquetes que no se hayan podido difundir por la inexistencia de un nodo adecuado que actúe como relé para que el paquete llegue al área de destino objetivo. Por ejemplo esta falta relé se puede dar por las características intrínsecas de las VANETs en una autopista.

Capítulo 3

COMUNICACIONES V2X: METODOLOGÍA Y APORTACIONES

En esta sección se ha decidido unificar la metodología con el trabajo desarrollado, ya que de esta manera se puede describir más fielmente los diferentes elementos desarrollados, así como su análisis e implementación, quedando todo más hilado y comprensible. El objetivo principal de esta tesis es el desarrollo de un sistema de comunicaciones V2X para vehículos inteligentes que sea eficiente, robusto, interoperable, de código abierto y de bajo coste.

Durante el transcurso de la tesis se ha utilizado la tecnología actual más adecuada disponible en ese momento para conseguir crear una VANET. Las tecnologías han ido cambiando dado que es un entorno que se encuentra en el límite entre desarrollo e investigación. No existiendo en muchos casos sistemas físicos y hardware disponible para desarrollos concretos como el que se persigue, con lo que nos hemos visto obligados adaptar tecnología, tanto hardware como software para lograr los propósitos perseguidos.

Al principio de la misma se optó por una tecnología de comunicación basada en redes de sensores, WSN (*Wireless Sensor and Actuators Networks*), Dado que la tecnología DSRC (*Dedicated short-range communications*) solo existía en papel, no habiendo disponibilidad de dispositivos físicos, protocolos y los estándares disponibles eran simples borradores sin utilidad para la implementación. Además, la hipótesis de partida es que todos los protocolos, aplicaciones y desarrollos que se realicen sobre WSN son directamente extrapolables a cualquier otra tecnología de comunicaciones, aunque esta no haya sido específicamente desarrollada para automoción ni movilidad. En el apartado 3.1 se explicará más detalladamente las características de la tecnología así como los algoritmos y trabajos realizados sobre ella y los resultados obtenidos. También se explican las razones de porque se abandonó esta tecnología y se pasó a otra más adecuada.

Conforme el desarrollo de la tesis avanzaba, la tecnología en la que se basa el DSRC comenzó a aparecer en el mercado, así como estándares más desarrollados y definitivos, por lo cual, se decidió apostar por una tecnología específicamente diseñada para entornos de transporte y automoción. En el apartado 3.2 se explicará más detalladamente las características de la tecnología así como los algoritmos y trabajos realizados sobre ella y los resultados obtenidos.

3.1. Tecnología WSA

En este apartado se explican todas las características de la tecnología de comunicación WSA, hardware como software así como su utilización como soporte para las comunicaciones vehiculares, los algoritmos desarrollados y los resultados obtenidos.

3.1.1. Descripción de la tecnología

Una red WSA es una red de nodos, equipados con sensores, que colaboran en una tarea común. Las redes de sensores están formadas por un grupo de nodos con capacidades sensitivas y de comunicación inalámbrica los cuales permiten formar redes ad-hoc sin infraestructura física preestablecida ni administración central. Esta clase de redes se caracterizan por su facilidad de despliegue, por ser autoconfigurables y por ofrecer servicios de encaminamiento entre nodos sin visión directa, así como registrar datos referentes a los sensores locales de cada nodo. Otra de sus características es su gestión eficiente de la energía, que les permite obtener una alta tasa de autonomía que las hacen plenamente operativas.

La idea de estas redes es repartir aleatoriamente estos nodos en un territorio grande, en el cual los nodos observan hasta que sus recursos energéticos se agoten. Los atributos «pequeño», «barato» y «autónomo» dieron a conocer la idea como polvo inteligente (*smart dust*) de ahí el nombre genérico de “motes” o “motas”.

Las WSA se están utilizando para el Internet de las Cosas (IoT), también suele referirse como “el Internet de todas las Cosas”. Es un concepto que se refiere a la interconexión digital de objetos cotidianos con Internet. Si los objetos de la vida cotidiana tuvieran incorporados módulos de comunicación, podrían ser identificados y gestionados por otros equipos, de la misma manera que si lo fuesen por seres humanos.

Por ejemplo, si los libros, termostatos, refrigeradores, la paquetería, lámparas, botiquines, etc. estuvieran conectados a Internet y equipados con dispositivos de identificación, no existirían, en teoría, cosas fuera de stock o carencia de medicinas o caducadas, sabríamos exactamente la ubicación, cómo se consumen y se compran productos en todo el mundo; el extravío sería cosa del pasado y sabríamos qué está encendido o apagado en todo momento. Esto es extrapolable a los vehículos los cuales estarían identificados y posicionados en todo momento, permitiendo un uso más eficiente del control de tráfico, avisos a emergencias, creación de sistemas cooperativos, etc, lo cual constituye lo que se viene a denominar “*Internet of Vehicles*”.

3.1.1.1. Estándar de comunicaciones: WSN

Como se ha comentado anteriormente los estándares son fundamental en comunicaciones ya que garantizan la interoperabilidad de los medios y dispositivos. Por lo cual en esta sección se hará una pequeña introducción del estándar específico de los WSN.

Las redes de sensores operan con el estándar de comunicaciones IEEE 802.15.4 (for Information technology, 2006) el cual se encuentra en el grupo de estándares 802.15 *Wireless Personal Area Network* (WPAN), este estándar define el nivel físico (PHY) y el control de acceso al medio (MAC) de redes inalámbricas de área personal con tasas bajas de transmisión de datos (*low-rate wireless personal area network*, LR-WPAN), la actual revisión del estándar se aprobó en 2006.

El estándar define una tasa de transferencia de 250 Kbps. en un área de comunicación de 10 metros. Se puede reducir los consumos a cambio de reducir las tasas de transferencia para ello se definen varios niveles físicos. Hay definidas tres tasas alternativas de 20, 40 y 100 Kbps. Se pueden lograr tasas aún menores con la consiguiente reducción de consumo de energía. La característica fundamental de 802.15.4 entre las WPAN's es la obtención de costes de fabricación excepcionalmente bajos por medio de la sencillez tecnológica, sin perjuicio de la generalidad o la adaptabilidad. Entre los aspectos más importantes se encuentra la adecuación de su uso para tiempo real por medio de *slots* de tiempo garantizados, evasión de colisiones por CSMA/CA y soporte integrado a las comunicaciones seguras. También se incluyen funciones de control del consumo de energía como calidad del enlace y detección de energía.

FIGURA 3.1: Pila de protocolo IEEE 802.15.4

Los dispositivos que operan con IEEE 802.15.4 se relacionan entre sí a través de una red inalámbrica sencilla. La definición de los niveles se basa en el modelo OSI (Figura 2.4), los niveles inferiores de la torre OSI se definen en el estándar, se prevé la interacción con el resto de niveles, posiblemente por medio de un subnivel de control de enlace lógico (LLC) basado en IEEE 802.2, que acceda a MAC. En la Figura 3.1 se puede ver la sección de la pila OSI descrita anteriormente.

El nivel físico (PHY) provee el servicio de transmisión de datos sobre el medio físico propiamente dicho, así como la interfaz con la entidad de gestión del nivel físico, por medio de la cual se puede acceder a todos los servicios de gestión del nivel y que mantiene una base de datos con información de redes de área personal relacionadas. De esta forma, PHY controla el transceptor de radiofrecuencia y realiza la selección de canales junto con el control de consumo y de la señal. Opera en una de tres posibles bandas de frecuencia de uso no regulado.

- 868-868,8 MHz: Europa, permite tres canales de comunicación.
- 902-928 MHz: Norte América, hasta treinta canales.
- 2400-2483,5 MHz: uso en todo el mundo, hasta dieciséis canales.

La versión original del estándar, de 2003, especifica dos niveles físicos basados en espectro ensanchado por secuencia directa (*direct sequence spread spectrum*, DSSS): uno en las bandas de 868/915 MHz con tasas de 20 y 40 Kbps; y otra en la banda de 2450 MHz con hasta 250 Kbps. La revisión de 2006 incrementa las tasas de datos máximas de las bandas de 868/915 MHz, que permiten hasta 100 y 250 Kbps. Adicionalmente define cuatro niveles físicos en base al método de modulación usado. Tres de ellas preservan el mecanismo por DSSS: las bandas de 868/915 MHz, que usan modulación en fase binaria o por cuadratura en offset (*offset quadrature phase shift keying*, ésta segunda opcional). En la banda de 2450 MHz se usa esta segunda técnica. También, se define una combinación opcional de modulación binaria y en amplitud para las bandas de menor frecuencia, basadas por lo tanto en una difusión de espectro paralela, no secuencial (PSSS). Si se usan éstas bandas de menor frecuencia, se puede cambiar dinámicamente el nivel físico usado de entre los soportados.

El control de acceso al medio (MAC) transmite tramas MAC usando para ello el canal físico. Además del servicio de datos, ofrece una interfaz de control y regula el acceso al canal físico y al balizado de la red. También controla la validación de las tramas y las asociaciones entre nodos, y garantiza *slots* de tiempo. Por último, ofrece puntos de enganche para servicios seguros.

El estándar no define niveles superiores ni subcapas de interoperabilidad. Existen extensiones, como la especificación de ZigBee, que complementan al estándar en la propuesta de soluciones completas.

3.1.1.2. *Hardware*

Como se ha comentado en el apartado anterior una de las características del estándar IEEE 802.15.4 es la posibilidad de incluirlo en dispositivos de bajo coste disponibles en el mercado en el momento del desarrollo de esta parte de la tesis.

FIGURA 3.2: Mota Maxfor MTM-CM3100

Existen diferentes familias de motas en función de los microprocesadores y de los chips de radiofrecuencia que porten. Las principales familias son: Imote2, IRIS, Kmote, MICA2, MICAz, TelosB, TinyNode y UCMote Mini.

Concretamente para la implementación de los protocolos y algoritmos para dar soporte a las comunicaciones vehiculares, se utilizaron los dispositivos MTM-CM3100 de Maxfor Technology Inc., basados en la plataforma TelosB. En la figura 3.2 se puede observar uno de estos dispositivos utilizados y en la tabla 3.1 se enumeran las principales características de este dispositivo utilizado.

Las razones de elegir este tipo de motas para dar soporte a las comunicaciones vehiculares son que operaban en una banda de frecuencia de 2.4GHz, el ancho de banda eran las más elevadas con rangos de transmisión de hasta 150 metros, suficientes como para soportar la implementación de aplicaciones ITS y sistemas cooperativos básicos, así como que tenían la posibilidad de extraer la información en tiempo real mediante el puerto serie incorporado. Se han seleccionado porque esta tecnología diseñada para internet de las cosas ha sido originalmente diseñada para funcionar en modo broadcast, incorporando diversos algoritmos de enrutamiento que son utilizados para gestionar en modo ad-hoc el funcionamiento de la red.

TABLA 3.1: Características MTM-CM3100.

ITEM	Specification
<i>Processor</i>	TI MSP430F1611
<i>Memory</i>	Program flash(48KB), Data RAM(10K Byte), External Flash ROM(1MB)
<i>Current Draw</i>	Active mode(1.8mA), Sleep mode(5.1uA)
<i>ADC</i>	12bit resolution(8 channels)
<i>Interface</i>	Serial I/F : USB, RS232
<i>Connector</i>	SMC Q type 12P male R/A External system interface (ERNI) 51 pin connector External sensor interface
<i>RF Chip</i>	TI CC2420
<i>Freq. band</i>	2.4GHz 2.485GHz (IEEE 802.15.4 compliant)
<i>Sensitivity</i>	-95 dBm typ.
<i>Transfer</i>	250Kbps
<i>RF power</i>	10dBm
<i>Range</i>	150m(outdoor), 20 30(Indoor)
<i>Current Draw</i>	Active mode RX:18.8mA, Active mode TX:17.4mA, Sleep mode : 1uA
<i>Power supply</i>	2.1V 3.6V(CC2420 input power)
<i>Antenna</i>	Dipole Antenna
<i>Electro-mechanical Power</i>	3V (AA or AAA battery)
<i>Dimension</i>	60×32 mm
<i>Weight</i>	14g

3.1.1.3. Software

Las motas al tener unos componentes muy limitados en computación y memoria, así como unos requerimientos limitados de energía, hacen que sea necesario un sistema operativo propio que atienda todas estas necesidades y que de soporte a los sensores embebidos. Para ello la Universidad de California, Berkeley, en cooperación con *Intel Research* y *Crossbow Technology*, han estado desarrollando desde el año 1999 el sistema operativo TinyOS. Desde entonces ha ido evolucionando y creciendo hasta ser un consorcio internacional, *TinyOS Alliance*.

TinyOS es un sistema operativo de código abierto especializado en redes de sensores inalámbricos (WSNs). TinyOS es un sistema operativo integrado escrito en el lenguaje de programación nesC como un conjunto de tareas y procesos de cooperación.

La separación lógica entre TinyOs y una aplicación desarrollada para este sistema operativo es difusa y se puede considerar que es un único ente. Como se ha dicho anteriormente TinyOs está escrito en nesC que es un lenguaje de programación derivado de C optimizado para la memoria limitada de las motas.

Los programas de TinyOs se construyen a partir de unidades básicas de software, las cuales algunas contienen abstracciones de hardware. Estas unidades básicas o componentes se interrelacionan mediante interfaces. TinyOS proporciona interfaces y componentes para abstracciones comunes como la comunicación de paquetes, el enrutamiento, la detección, actuación y almacenamiento.

TinyOs es un sistema no-bloqueante pero a pesar de esto permite mantener una alta concurrencia con una pila, lo que obliga a los programadores a escribir lógica compleja uniendo muchos controladores de eventos pequeños. Para apoyar a los cálculos más grandes, TinyOS ofrece tareas, que son similares a una llamada a procedimientos en diferido y son atendidas cuando el manejador de tareas lo permite. Un componente programa una tarea y el sistema operativo la atenderá y ejecutará cuando pueda ser atendida. Las tareas son no preferentes y se ejecutan en orden de llegada (FIFO).

3.1.2. Algoritmos de enrutamiento

El algoritmo *Collection Tree Protocol* (CTP) (Tinyos, 2015) es un módulo de TinyOS, está pensado para para recopilar información de redes de sensores o de los dispositivos conectados del internet de las cosas. Algunos ejemplos pueden ser: redes de sensores desplegados para monitorear microclimas en las explotaciones agrícolas o despliegues que miden el consumo de energía en edificios de oficinas o residenciales. Los nodos de estas redes recopilan información sobre el mundo físico usando sus sensores y posteriormente retransmiten las lecturas a una estación base central o servidor utilizando la comunicación inalámbrica *multi-hop*.

Debido a la posibilidad de que los nodos de la red puedan cambiar la ubicación física y como consecuencia variar los parámetros de conectividad, produce que la recopilación de información de los nodos en una red de sensores sea complicada. El enrutamiento multisalto en un entorno inalámbrico dinámico requiere que un protocolo deba adaptarse rápidamente a los cambios en la red, mientras que la energía-limitada de las motas dicta que tales mecanismos no requieran demasiada comunicación entre los nodos.

CTP calcula las rutas desde cada nodo en la red al nodo raíz en la red. CTP (ver Figura 3.3) usa tres mecanismos distintos para crear las rutas:

FIGURA 3.3: Esquema del algoritmo GCTP

3.1.2.1. Estimador de calidad de la conexión

El estimador de calidad de la conexión o “*Link Quality Estimation*” sirve para determinar la calidad de conexión. Los enlaces en una red inalámbrica son muy dinámicos y muestran un comportamiento a ráfagas en escalas de tiempo cortas. Este establecimiento de enlaces inalámbricos sugiere que una calidad de enlace debe ser ágil para que sea exacta. El estimador de enlace utilizado en CTP utiliza la información de la capa física y de información provista por otros nodos para determinar la calidad de conexión.

3.1.2.2. Ruta de validación

En un entorno inalámbrico no centralizado y cambiante, una ruta que es confiable en un momento dado puede llegar a ser poco fiable o incluso no estar disponible a los pocos segundos. Debido a los cambios en las condiciones de enrutamiento, se pueden llegar a formar bucles y causar una congestión de la red y un aumento en el consumo energético. Por lo cual, es importante detectar estos problemas en la ruta hasta la raíz rápidamente cuando ocurran. CTP hace una validación del camino de datos para detectar estos problemas de transmisión del paquete de datos. Lo hace mediante el uso de paquetes específicos donde se difunden los datos de la topología de la red y así poder detectar rápidamente el problema y solucionarlo.

3.1.2.3. Balizado adaptativo

En los protocolos de enrutamiento normalmente se retransmiten *beacons* o paquetes de control, estas tramas son usadas para conocer la topología de la red, estas son enviadas a intervalos fijos (por ejemplo, cada 30 segundos). Este intervalo representa una solución de compromiso básico. Un pequeño intervalo de envío, es decir frecuentes *beacons*, hace al protocolo más sensible a los cambios en la red, pero utiliza más ancho de banda y de energía. Un intervalo grande entre los envíos de los *beacons* usa menos energía, pero puede hacer que los problemas topológicos persistan durante mucho tiempo. CTP utiliza un balizamiento adaptativo para romper este equilibrio. Cuando la topología es incompatible y tiene problemas, envía *beacons* más rápido. De lo contrario, disminuye la tasa de balizamiento de manera exponencial. Por lo tanto, CTP puede responder rápidamente a las situaciones adversas mientras que existe bajo costo operativo de control a largo plazo si la red está estable.

3.1.3. Trabajo desarrollado: Adaptación de la tecnología WSN para comunicaciones V2X

A continuación se detalla el trabajo desarrollado sobre esta tecnología primero se describirá la arquitectura implementada y por último los algoritmos desarrollados para la creación de una VANET utilizando tecnología WSN.

3.1.3.1. Arquitectura

Se pretende realizar comunicaciones V2X en modo GeoBroadcast donde cada mota es una OBU o RSU. A partir de las características técnicas de las motas, se ha diseñado un sistema de comunicaciones entre vehículos basado en IEEE 802.15.4 para crear una VANET que dé soporte a sistemas cooperativos, siguiendo la arquitectura que aparece en la figura 3.4, vehículo tendrá un módulo compuesto por un ordenador, un GPS y una mota.

FIGURA 3.4: Arquitectura de la VANET basada en WSN

El procedimiento básico implementado es: el ordenador recoge la información del GPS, la encapsula en un paquete de datos y se la entrega a la mota para que proceda a su difusión por la red, el resto de motas recogen esta información y se la retransmiten al ordenador de abordó para que este conozca la posición del resto de vehículos.

Para comprobar si las motas eran capaces de formar una VANET que diera soporte a un sistema cooperativo se programaron con el algoritmo de enrutamiento *Collection Tree Protocol* (CTP). Se comprobó que era posible la comunicación entre módulos siempre que se comportaran como RSU, es decir, no tuvieran movilidad. En el caso de que los módulos se

comportaran como OBU el algoritmo no valía por la baja tasa de actualización de las rutas establecidas.

Se estudió el algoritmo con detenimiento, comprobado sus características y analizado sus fortalezas y debilidades para la creación de VANETs. El CTP es muy bueno a la hora de la elección de rutas eficientes, el problema radica que estas rutas no garantizan que sean óptimas geográficamente y su tiempo de actualización es muy elevado. Estas características lo hacen inviable para el uso en VANETs de vehículos dada la elevada capacidad de movimiento de estos. Por eso se decidió crear un nuevo algoritmo de enrutamiento para las motas para que cumpliera un enrutamiento por posición geográfica o geoenrutamiento capaz de dar soporte a VANETs. El nuevo algoritmo se le denominó *Geonetworking Control Tree Protocol* (GCTP).

3.1.3.2. Algoritmo GCTP: Geonetworking Control Tree Protocol

Como se ha explicado anteriormente este algoritmo se desarrolló por la necesidad de solucionar los problemas derivados en el geoenrutamiento dinámico que el CTP no es capaz de solucionar, el cual se produce cuando los nodos están en movimiento en lugar de estar estáticos, como es el caso de los vehículos. Habiendo observado que las características del CTP en composición estática eran excelentes se decidió crear el GCTP tomando como base el CTP, incluyendo el geoposicionamiento en el direccionamiento de los paquetes de datos y estructura de la red. Sin embargo, sólo la estructura modular funcional de CTP ha sido replicada, ya que las funcionalidades de cada módulo han sido totalmente redefinidas y rediseñadas, en consecuencia, se puede considerar que este algoritmo de GeoNetworking como un algoritmo totalmente novedoso y una aportación relevante de esta tesis.

A este nuevo algoritmo se le denomina *GeoNetworking Collection Tree Protocol* (GCTP) y se fundamenta en la utilización de una mensajería específica (*beacons o balizas*) para crear la red, estos mensajes son enviados por todos los nodos de la red en modo broadcast para la construcción y el mantenimiento de la tabla de enrutamiento, adicionalmente, existen mensajes de datos para transportar los datos del nivel de aplicación entre los nodos de la red. La arquitectura de la red en malla sigue una estructura de árbol, donde se encaminan los mensajes de todos los nodos hacia un nodo raíz (root), en caso de no existir comunicación directa utiliza otros nodos como relés para que los mensajes lleguen al nodo raíz.

El GCTP consta de cuatro componentes de software, divididos en dos grupos: transmisión de datos (*Data Layer -DL*) y enrutamiento (*Routing Layer -RL*).

- El grupo de transmisión de datos está formado por dos componentes:
  - El control de datos (*Data Control -DC*).

- El motor de reenvío (*Forwarding Engine* – FE).
- El grupo de enrutamiento está formado por:
  - El motor de enrutamiento (*Routing Engine* – RE).
  - El Estimador de Enlace (*Link Estimator* - LE).

FIGURA 3.5: Esquema del algoritmo GCTP

Cada nodo de la red posee una instancia con todos ellos, podemos ver en la Figura 3.5 un esquema de las iteraciones que tienen entre todos los módulos del algoritmo GCTP. En los siguientes subapartados se explica detalladamente el funcionamiento de los diferentes módulos y sus funciones.

3.1.3.2.1. Control de Datos (DC)

La misión de este módulo es la de recibir los datos del GPS procesar estos datos y transmitirlos a FE y LE. Los datos proporcionados por el sistema de geolocalización son: posición UTM (Norte y Este), la velocidad y una marca de tiempo. También recibe los datos

de la capa de aplicación, una vez recibida esta información, el DC la empaqueta junto a un identificador de nodo y la envía a FE, que se encarga de difundirla entre los nodos de la red. De forma similar, cuando el FE recibe un paquete de datos de otros nodos, éste lo envía al DC que a su vez lo transmite al dispositivo conectado al nodo de comunicación.

3.1.3.2.2. Motor de Reenvío (FE)

Este módulo es el encargado de analizar los mensajes de datos que circulan por la red y decidir si son sensibles de analizar o se descartan. En el caso de que se tenga que analizar se desempaqueta la información y se analizan las cabeceras, en función de estas se actúa de una forma u otra. Si el paquete es para ese nodo se pasa al módulo superior DC (ver Figura 3.5), si la información es para otro nodo y se está actuando como relee se reempaqueta la información con nuevas cabeceras y se envía al vecino que le indica la Tabla de Enrutamiento (RET) (en los siguientes apartados explicaremos que es esta tabla, como se crea y como se actualiza). El FE es también responsable de la detección y reparación de los bucles de enrutamiento, así como la supresión de paquetes duplicados.

3.1.3.2.3. Motor de enrutamiento (RE)

La misión del motor de enrutamiento es examinar los paquetes entrantes y salientes de enrutamiento (*beacons*) de la red, así como la creación y actualización de la Tabla de enrutamiento (RET). Esta RET contiene una lista de vecinos cercanos (visión directa), de la que el nodo puede seleccionar a su nodo con mejor conectividad (padre) dentro del árbol de enrutamiento. La RET se completa utilizando la información extraída de los *beacons* recibidos de los diferentes nodos cercanos.

Para rellenar la RET el RE utiliza el identificador del nodo que ha enviado el *beacon* junto con el identificador de los nodos vecinos a este, con esta información RE es capaz de calcular cuál de todos los nodos vecinos es el más adecuado para ser el padre potencial. El padre potencial será el que menor métrica tenga, esta métrica se denomina “Distancia Esperada” (DE), la DE es calculada por otro módulo el LE, finalmente la información calculada se comunica a cada nodo vecino a través del intercambio de mensajes de enrutamiento (*beacons*).

3.1.3.2.4. Estimador de Enlace (LE)

Como hemos comentado en el apartado anterior la función de LE es determinar la calidad de la comunicación de los nodos vecinos. Esto se hace mediante la DE. Se puede definir DE_{ab} como la distancia Manhattan (se optó por esta medida por la limitación

computacional de los dispositivos de comunicación), en metros, entre el nodo a y el nodo b .

FIGURA 3.6: Calculo de Distancia Esperada (DE)

Si hay visión directa entre el nodo a y el nodo b , sólo hay un salto entre ambos nodos y en este caso se denomina $DE1hop_{ab}$. Si no hay una línea de visión directa entre el nodo a y el nodo b , el DE_{ab} se calcula como la suma de los $DE1hop_{xy}$ de los nodos requeridos de la red para llegar al nodo b desde el nodo a . Se puede ver un pequeño esquema de esta lógica en la Figura 3.6. El LE de un nodo calcula el $DE1hop_{ax}$ gracias a la recepción de los *beacons* de los otros nodos x , donde se encuentra la posición GPS en coordenadas UTM.

3.1.3.2.5. Estructura de los paquetes del algoritmo GCTP

Como se ha dejado entrever en apartados anteriores existen dos tipos diferentes de paquetes en el algoritmo para dar soporte a las comunicaciones vehiculares. Los paquetes de enrutamiento (*beacons*) y los paquetes de datos de la aplicación. La Figura 3.7 muestra la estructura de los paquetes utilizados para la creación de la VANET y el intercambio de información de las aplicaciones.

Los paquetes de enrutamiento son utilizados para establecer las tablas de rutas necesarias y por consiguiente para garantizar una transmisión eficaz de la información entre los nodos de la VANET. En este paquete el campo “Cabecera PHY”, “Cabecera MAC” y “Pie MAC” representan la información estándar de una red IEEE 802.15.4. Los datos de “Cabecera LE”, “GCTP Trama de enrutamiento (RE)” y “Pie LE” representa la información Geo-referenciada a proporcionar al algoritmo GCTP la información necesaria, en la Tabla 3.2

FIGURA 3.7: Esquema de los paquetes GCTP

se puede ver con detalle dicha información, para definir las tablas de rutas que son la base de la VANET.

TABLA 3.2: Detalle de los campos del paquete de enrutamiento de la figura 3.7.

Trama	Campo	tamaño	Descripción
<i>LE header</i>	NE	4 bit	Identificador de red
	Reservado	4 bit	Reservado
	SeqNo	8 bit	Numero de secuencia
<i>GCTP routing frame</i>	P	1 bit	Indicador de enrutamiento
	C	1 bit	Indicador de congestión
	Reservado	6 bit	Reservado
	ID	16 bit	Identificación del nodo emisor
	DEroot	16 bit	DE a la raíz desde el nodo emisor
	UTM_N	32 bit	Coordenada norte del nodo en UTM
	UTM_E	32 bit	Coordenada este del nodo en UTM
<i>LE footer</i>	DE1Hop. n1	8 bit	DE1hop entre el nodo emisor y el nodo n1
	IDn1	16 bit	Identificador del nodo n1
	...	...	...
	DE1Hop. nn	8 bit	DE1hop entre el nodo emisor y el nodo nn
	IDnn	16 bit	Identificador del nodo nn

La estructura del paquete de datos contiene la información a ser transmitida desde un nodo de comunicación a otro, para dar soporte a las aplicaciones que utilizan el algoritmo GCTP y los datos necesarios para sustentar el funcionamiento del algoritmo. Del mismo modo los paquetes de enrutamiento, los paquetes de datos están formados por los 3 campos estándar del IEEE 802.15.4 (“Cabecera PHY”, “Cabecera MAC” y “Pie MAC”). Adicionalmente existe una sección del paquete “GCTP Trama de datos” donde se especifica la ruta que debe llevar el paquete para alcanzar el nodo destino así como información útil para detectar posibles problemas de bucles y reenvíos, en la Tabla 3.3 se puede ver con más detalle la composición de esta cabecera, por ultimo la sección “Payload” contiene la carga útil con la información de la aplicación.

TABLA 3.3: Composición de GCTP Trama de datos de la figura 3.7.

Trama	Campo	tamaño	Descripción
GCTP data frame	P	1 bit	Indicador de enrutamiento
	C	1 bit	Indicador de congestión
	Reservado	6 bit	Reservado
	THL	16 bit	Numero de saltos del paquete de datos
	DEorigen	16 bit	DE desde el origen a la raíz
	Origen	32 bit	ID del emisor del paquete
	SeqNo	32 bit	Numero de paquete
	CollectID	8 bit	DE1hop entre el nodo emisor y el nodo n1

3.1.3.2.6. Implementación

El algoritmo GCTP se ha implementado junto con otros módulos de TinyOs para poder interactuar con el ordenador de abordo como se especificó en la figura 3.5. El algoritmo de GCTP se ha implementado en un módulo de TinyOS y junto a otros módulos se creó un programa de geoenrutamiento que da soporte a VANETs. En la figura 3.8 se puede ver el flujograma con el que opera el algoritmo de GCTP en las motas.

FIGURA 3.8: Flujograma GCTP

Cuando a la mota le llega un mensaje por el puerto serie lo primero que se realiza es una comprobación de que el mensaje es correcto, si no es así se descarta, entonces se extrae la información necesaria para actualizar la RET. En base a la RET se calcula la ruta más óptima hasta el *root*, se encapsula la información y se envía a la VANET a través del módulo de radio. Al mismo tiempo se recalcula las DE con todos los nodos de la RET y se envía un mensaje de balizado (*beacon*).

Cuando una mota recibe un mensaje a través del módulo de radio se valida que los datos contenidos sean correctos y que el paquete sea para esa mota; si no lo son se descarta el mensaje. A continuación se analiza el tipo de datos contenidos en el paquete, si son de tipo *beacon* se extrae la información y se actualiza la RET. Si son del tipo de datos la mota analiza si la mota destino es ella, si es así transmite esa información contenida en el *payload* al PC a través del puerto serie. Si la mota no es el destino del paquete de datos y el paquete no es repetido prepara los datos con nuevas cabeceras y lo reenvía por el módulo de radio a la VANET. Si el paquete recibido es repetido lo descarta y avisa de un posible bucle en la VANET.

3.1.4. Resultados

Una vez desarrollados los algoritmos se ha procedido a comprobar su funcionamiento mediante pruebas en entornos interurbanos.

Primero se ha testado el correcto funcionamiento del algoritmo con diversas pruebas en entornos semiurbanos y autopistas. Cuando se ha confirmado el correcto funcionamiento de redes VANETs con GCTP se ha procedido al desarrollo de una aplicación ADAS distribuida para la detección y aviso de obstáculos en las vías urbanas, sección 4.2

3.1.4.1. Pruebas GCTP

Para comprobar el correcto funcionamiento del algoritmo se realizaron varias pruebas para verificar el rendimiento de la VANET en diferentes situaciones para chequear si la red sería capaz garantizar los requisitos mínimos de eficiencia, latencia y robustez para poder actuar como proveedor de información para aplicaciones ADAS.

3.1.4.1.1. Estudio de saturación

Con el fin de aplicar un algoritmo de GeoNetworking es importante demostrar la viabilidad del sistema para servir a las comunicaciones en escenarios reales con un elevado número de vehículos, dada la limitación de medios la mayoría del tiempo las pruebas de campo se han efectuado con un número limitado de nodos. Por lo cual se estudió si el algoritmo de GCTP sería capaz de dar servicio incluso en el peor de los escenarios posibles.

Dadas las cualidades del algoritmo GCTP, cada nodo de comunicación envía dos tipos de paquetes, de datos y de enrutamiento, con diferentes longitudes y frecuencia de transmisión. El paquete de datos son 39 bytes y tiene una frecuencia de transmisión de 2 Hz, Las características del paquete de enrutamiento son variables, dependiendo de su tamaño, el

número de vecinos visibles por nodo y su frecuencia de transmisión en la saturación de la red y la calidad de la conexión con sus vecinos (distancia, interferencias y movilidad).

Teniendo en cuenta estas características, se han diferenciado 4 casos de uso necesarios de comprobar la funcionalidad de la red en diferentes situaciones de tráfico. Estos casos se presentan en la Tabla 3.4. Dado que el ancho de banda máximo de la red es de 256 Kbps, también se ha incluido es el número máximo de vehículos que puede ser soportada por la red sin saturación.

TABLA 3.4: Casos de uso teóricos.

Caso de uso	Longitud PD	Longitud PE	Frecuencia PD (Hz)	Frecuencia PE (Hz)	Máximo número de vehículos
1	39	62	2	4	165
2	39	32	2	0.2	388
3	39	47	2	1	262
4	39	62	2	1	234

Donde el primer caso representa el peor escenario, con un ambiente de alta movilidad y la necesidad de una reconfiguración de la red constante. El segundo es el mejor de los casos, con una reconfiguración de la red mínimo. El tercero y cuarto son casos intermedios, pero el cuarto con la longitud máxima en los paquetes de enrutamiento.

Por otra parte, el rango de operación de cada nodo se limita a aproximadamente 100 m. por la potencia del dispositivo. Por lo tanto, podemos analizar cuántos vehículos circulan en 200 m. de la carretera en condiciones de saturación de acuerdo con la velocidad. Este número depende de la densidad de tráfico que sigue la Eq. 3.1.

$$I = V \cdot D \quad (3.1)$$

Donde I es la intensidad del tráfico en veh/h, V es la velocidad en km/h y D es la densidad en veh/h. También se puede definir la densidad de tráfico como se muestra en la Eq. 3.2 (Aparicio et al., 2008).

$$D = ((1 - W_T - 2W_B)(L_C + H_{CC}V + W_T(L_C + V(h_{CT} + h_{TV}) + L_T) + W_B(L_C + V(h_{CT} + h_{BC}) + L_B))^{-1} \quad (3.2)$$

Donde C se refiere a coches, B se refiere a autobuses y T se refiere a camiones. L_k es la longitud del vehículo, con k siendo C , B o T . h_{kj} es el tiempo mínimo entre el vehículo k y el vehículo j , con k y j siendo C , B o T . W_B y W_T representan la proporción de autobuses y camiones que existen en el tráfico.

El valor h viene definido por la Eq. 3.3

$$h_{kj} = \frac{\frac{V_k^2}{2a_k} + t_r \cdot V_k - \frac{V_j^2}{2a_j}}{V_k} \quad (3.3)$$

Donde k es el vehículo que circula delante y j es el vehículo posterior. V_i es la velocidad del vehículo, a_i es la aceleración del vehículo y t_r representa el tiempo de reacción del conductor.

Si suponemos que la velocidad es la misma para cada vehículo de ese tramo de carretera se puede simplificar la Eq. 3.3 como se muestra en la Eq. 3.4.

$$h_{kj} = \frac{V}{2a_k} + t_r - \frac{V}{2a_j} \quad (3.4)$$

Cuando ambos vehículos son del mismo tipo, por motivos de seguridad se asume que el segundo vehículo frena un 10% más que el primero. La deceleración elegida para los tres tipos de vehículos son: $a_C = -0,8g$, $a_B = -0,4g$ y $a_T = -0,3g$. Las longitudes 5m, 12m y 18m y las proporciones de ese tipo de vehículo en esa sección de carretera son 0.85, 0.075 y 0.075 para coches, autobuses y camiones respectivamente. Por último t_r es alrededor de 2 s (Aparicio et al., 2008).

FIGURA 3.9: Representación del número máximo de vehículos que la red puede manejar para cada caso de uso y el número máximo de vehículos de acuerdo con la velocidad para una carretera con tres carriles por sentido, teniendo en cuenta una rango de comunicación de 100 m. para cada nodo

Con las ecuaciones [3.1-3.4] se puede calcular la densidad de los vehículos en función de la velocidad, como se muestra en la Figura 3.9. Esta figura representa la densidad del tráfico en función de la velocidad para los cuatro casos de uso analizados en una autopista con tres carriles por sentido. El algoritmo es capaz de gestionar la estructura de la red en todos

los casos sin saturación, lo que garantiza la correcta entrega del paquete de datos mediante el algoritmo GCTP en las diferentes condiciones de la carretera, manteniendo siempre el máximo número de nodos conectados en cada caso de uso.

3.1.4.1.2. Prueba de rendimiento

La prueba de rendimiento consistió en una transmisión constante de la información a partir de los nodos de la red, intercambiando paquetes de datos a una velocidad de 2 Hz y con los datos que se describen en la Tabla 3.5, que corresponden a información útil para aplicaciones ADAS, con una longitud de 13 bytes.

TABLA 3.5: Composición de los datos transmitidos en el *payload*.

Trama	Nombre del campo	tamaño	Descripción
Payload	Tipo	4 bit	Indicador de red
	ID	4 bit	Identificador de nodo emisor de la información
	UTM_N	24 bit	Coordenada norte del nodo en UTM
	UTM_E	24 bit	Coordenada este del nodo en UTM
	UTM_Zone	8 bit	Región a la que esta referenciado UTM
	Velocidad	8 bit	Velocidad del nodo
	TST	16 bit	Tiempo de emisión del mensaje
	Ctrl	8 bit	Validación de datos

Al mismo tiempo los nodos están transmitiendo mensajes de enrutamiento con un periodo de [0.25-5]s en función de las variaciones en la red. En situaciones donde existe una constante reconfiguración de la red los nodos emiten mensajes de enrutamiento a una frecuencia de 4Hz mientras que si la red se mantiene estable (una retención, por ejemplo) los nodos emiten a 0.2HZ.

FIGURA 3.10: Rendimiento de la red. Promedio de latencia frente a la distancia al transmisor, y los paquetes de datos perdidos vs. distancia al transmisor

Primero se probó el algoritmo de forma estática para poder comprobar la latencia del algoritmo, con dos o tres nodos situados a diferentes distancias en configuraciones adecuadas para asegurarse de que la red estuviera en uno y dos saltos. En el segundo paso, siempre con tres nodos, se probaron las características de la red en una situación de

circulación a baja velocidad, para poder realizar un seguimiento del rendimiento de la red en condiciones controladas. Por último, estas pruebas se completaron con experimentos en situaciones de tráfico reales. Los parámetros de la red analizados son la latencia del proceso de comunicación (tiempo desde cuando que se genera un mensaje en un nodo hasta que el mensaje es entregado a un nodo receptor), el número de paquetes de datos entregados y la tasa de paquetes de datos perdidos.

TABLA 3.6: Resultados de la prueba estática a distancias de 1hop.

Tiempo del experimento (seg.)	Distancia entre nodos (m.)	Promedio de la latencia (seg.)	Desviación típica de la latencia (seg.)	Paquetes recibidos	Paquetes perdidos	Total paquetes
56	10	0.01	0.0037	107	6 (5.31 %)	113
62.5	20	0	0.011	118	6 (4.84 %)	124
68	30	0.04	0.05	130	7 (5.11 %)	137
59	50	0.04	0.049	119	6 (4.80 %)	125
52.5	100	0.08	0.043	104	5 (4.59 %)	109

Los resultados de la prueba estática se muestran en la Fig. 3.10 y en la Tabla 3.6, donde 3 nodos están transmitiendo y recibiendo al mismo tiempo, a una distancia 1 salto (1hop). La prueba se realizó a una tasa de transmisión de 4HZ; el emisor y receptor se sincronizan usando los relojes GPS, cuya precisión, en este caso, fue de 100 ms. Este ratio de retransmisiones es el peor de los casos que se puede adoptar sin llegar a la saturación de la red.

TABLA 3.7: Resultados de la prueba estática a distancias de 2 hop (14 → 13 → Raíz).

Tiempo del experimento (seg.)	ID Nodo	Distancia al nodo raíz (m)	Promedio de la latencia (seg.)	Suma de la latencia (seg.)	Paquetes recibidos	Paquetes perdidos	Total paquetes
37	13	20	0.01	0.02	67	5 (6.9 %)	72
	14	38	0.01	0.05	68	4 (5.5 %)	72
63	13	26	0.01	0.05	117	8 (6.4 %)	125
	14	57	0.01	0.03	118	7 (5.6 %)	125
60	13	116	0.01	0.03	112	8 (6.6 %)	120
	14	234	0.01	0.05	116	4 (3.3 %)	120

Esta primera prueba demuestra que el rendimiento de la red está garantizada gracias a la eficacia de GCTP, el mantenimiento de la latencia media entre nodos es muy inferior a la tasa de transmisión de 250 ms., lo que asegura un servicio adecuado a sistemas ADAS debido a el hecho de que los mensajes siempre se actualizan. Del mismo modo, el número de paquetes perdidos es aproximadamente un 5 % (Coates et al., 2000), el cual se considera aceptable para asegurar el rendimiento de sistemas ADAS. La Tabla 3.6 presenta los resultados detallados de las pruebas estáticas a una distancia 1 hop y en la Tabla 3.7 se presentan los resultados a una distancia de 2 hop con tres nodos, nodo 13, nodo 14 y nodo raíz (0). En la primera tabla, el rendimiento de la red es estable a todas las distancias, al mantener una baja tasa de mensajes perdidos y la latencia es inferior a 100 ms. En el caso de la tabla que representa la prueba con 2 hops, se mantienen los valores de latencia inferior a 100 ms. Se puede observar un ligero aumento de las tasas de pérdida de paquetes con respecto a la prueba de 1 hop, causados por posibles interferencias durante la prueba, pero se considera que es en el rango valido para

garantizar la fiabilidad de las comunicaciones. La prueba demostró que el rendimiento del algoritmo GCTP es muy eficiente y eficaz para aplicaciones de seguridad, lo que demuestra la alta fiabilidad del sistema, y está en cumplimiento con los requisitos de los sistemas de comunicación vehicular.

3.1.4.1.3. Pruebas en condiciones reales

Una vez demostrado que el algoritmo GCTP tiene capacidad suficiente para atender las condiciones más adversas de comunicaciones y que la latencia y efectividad en la entrega de mensajes es suficiente se ha pasado a probar su efectividad en el mundo real haciendo pruebas de campo en un entorno normal de circulación. Para ello se ha elegido un tramo de 10 km de autopista, concretamente un tramo de la autopista A3 que está cerca y es fácilmente accesible desde el Campus Sur de la UPM, este tramo consta en su mayoría de 3 carriles por sentido. Se han utilizado 3 vehículos especialmente instrumentados para la ocasión, los conductores se han comportado como usuarios normales de la vía, siguiendo pautas de conducción segura y las normas de regulación, es decir, respetando la señalización en todo momento, con estas condiciones se ha emulado el máximo número de situaciones reales que se producen en una VANET. En la figura 3.11 se puede ver el recorrido de uno de los vehículos para dicha prueba.

FIGURA 3.11: Tramo de la autopista A3 donde se han realizado las pruebas

Para probar el rendimiento de la red, se ha utilizado la misma carga útil que en la prueba estática para no desvirtuar los resultados y que sean coherentes unos con otros. El

payload incluido en los paquetes de datos se puede ver en la Tabla 3.5, estos datos han sido enviados por cada nodo al nodo raíz de GCTP con una frecuencia de 2 Hz. El algoritmo de enrutamiento de cada nodo tiene que poder encaminar cada paquete de datos al destino, pero siempre tratando de mantener una estructura de red geográfica óptima y gestionar automáticamente los saltos necesarios para llegar al nodo raíz. Este comportamiento asegura que la información de cada nodo siempre se actualiza con los datos de sus vecinos cercanos geográficamente. Por lo tanto, la VANET conoce la información relevante de los otros nodos, y de esta forma es posible monitorizar el rendimiento de la red desde cada nodo destino.

Gracias a que la información incluida en la carga útil, incluye la posición GPS y la velocidad de cada vehículo, es posible tener una imagen en tiempo real del estado de la red, así como el cálculo de las métricas para estimar el correcto rendimiento de la red.

FIGURA 3.12: Datos del nodo 13 recogidos durante la prueba donde se muestra el tipo de conectividad

Como se ha comentado anteriormente estas pruebas se han realizado con tres vehículos instrumentados: nodo 0, 13 y 14, donde el nodo 0 es el nodo raíz. Los tres vehículos han recorrido los 10 km de la autopista A3 durante la cual la velocidad media de circulación ha sido de unos 100 km/h y la velocidad máxima de 110 km/h con unas características de congestión media/densa ente 2750 y 4500 vehículos/hora según los datos de la dirección General de Tráfico, DGT, para la hora y día de la realización de las pruebas.

En la prueba los tres vehículos se han comportado como otros vehículos, haciendo adelantamientos, cambios de carril y distintas distancias entre los vehículos, en

determinadas ocasiones se han intercalado vehículos no instrumentados para comprobar la conectividad con obstáculos que podían ocluir la línea de visión directa de los nodos. Las posiciones relativas de los tres nodos también han cambiado de acuerdo con las condiciones del tráfico: situaciones de conducción en pelotón en el mismo carril, cambios de carril, circulación en carriles adyacentes, etc.

TABLA 3.8: Resultados de una prueba de GCTP en autopista

Tiempo de muestreo	ID Nodo	Saltos	Latencia Media (s)	Paquetes recibidos	Paquetes perdidos	Total paquetes
1320	13	1	0.08	1698	-	-
1320	13	2	0.11	673	-	-
1320	13	1/2	0.09	2361	279 (10 %)	2640
1320	14	1	0.01	2027	-	-
1320	14	2	0.02	487	-	-
1320	14	1/2	0.01	2514	126 (4.7 %)	2640

FIGURA 3.13: Detalle de una reconfiguración de la VANET durante la prueba

Los datos del experimento han sido recogidos por el nodo 0. El análisis de estos datos demuestra que la red mantiene su configuración hasta el momento que un nodo se aleja más de 90 metros del nodo raíz, en ese momento se reconfigura la red. Cuando ocurre esto el GCTP recalcula la tabla de rutas para adaptarse a la nueva configuración de red.

La prueba ha sido realizada durante 22 minutos aproximadamente, obteniéndose los resultados que se pueden observar en la Tabla 3.8. Estos resultados son similares a la prueba de rendimiento y validan los resultados. El rendimiento de la red se mantiene en situaciones

de comunicación tanto con 1 salto como con 2. El algoritmo de enrutamiento cumple con su objetivo de que la red sólo se reconfigura cuando las condiciones de los nodos le exigen, reduciendo así el número de cambios en la estructuras de la red. Durante esta prueba se produjeron 14 cambios en el enrutamiento de la VANET. Un parámetro se desvió de los resultados esperados, el nodo 13 tiene una tasa de paquetes perdidos superior a lo esperado, el 10%. Analizado los datos para averiguar el origen de esta anomalía y solucionarla se ha llegado a la conclusión que este resultado es debido a las circunstancias de tráfico al final de la prueba, donde la distancia entre el nodo 13 y los otros dos nodos superó el máximo de 90 metros. En este caso, los paquetes de datos emitidos durante este período se consideran paquetes perdidos, a pesar de que cumplen con las especificaciones de algoritmo de enrutamiento (Figura 3.12). Mediante la eliminación de esta última parte del experimento, la tasa de paquetes perdidos se reduce al 6,6 % (Coates et al., 2000).

TABLA 3.9: Velocidades de circulación durante el adelantamiento

	Nodo 13	Nodo 14	Nodo 0
Reconfiguración 1	98 km/h	80 km/h	100 km/h
Reconfiguración 2	83 km/h	83 km/h	95 km/h

Un momento delicado en toda VANET y que puede causar problemas en esta se produce cuando la topología de la red se reconfigura para adecuarse a un cambio de las posiciones de los nodos. Por eso se ha analizado más detalladamente estos casos. La Figura 3.13 representa un detalle de la ruta de prueba en el que se producen dos cambios de estructura de la red. La situación inicial los tres vehículos circulan en una sección con una separación de 90 m; el primer nodo es el 14, seguido por el 13 y por último el nodo 0. En esta situación inicial, la estructura de red es $14 \rightarrow 13 \rightarrow 0$. A continuación, el vehículo con el nodo 0 aumenta su velocidad y comienza un adelantamiento de los otros dos vehículos. Las diferentes velocidades de circulación de los tres vehículos se muestran en la Tabla 3.9. La Figura 3.13 muestra las posiciones GPS del nodo 13 y 14 durante estas dos reconfiguraciones de la estructura de la red; primero un cambio desde una estructura inicial ($14 \rightarrow 13 \rightarrow 0$) a un enlace directo entre los nodos 13 y 14 al nodo raíz ($14 \rightarrow 0$; $13 \rightarrow 0$). En este caso, la duración del cambio de estructura de red fue de 0,4 s, lo que significa que ningún paquete con información se pierde durante el proceso y el cambio es transparente a nivel de datos.

En la Figura 3.14 se muestra un detalle de la ruta de los tres coches y la estructura de la red durante la maniobra. La figura está dividida en dos secciones, en la parte superior, una vista aérea del segmento de la carretera en donde se llevó a cabo la maniobra y la estructura de la red; en la parte inferior se muestra con la representación gráfica de estos datos en un solo eje, para que los resultados pueden ser apreciadas más claramente. El algoritmo GeoNetworking decide cambiar la estructura de red, el nodo 0 ha superado al nodo 13 y está circulando detrás de nodo 14. En este momento, la distancia entre los nodos 13 y 14 es mayor de 90 metros y es

FIGURA 3.14: Primera reconfiguración de la VANET

mayor que la distancia entre los nodos 13 y 0 y los nodos 14 y 0, por eso el algoritmo decide cambiar la configuración de enrutamiento. Una vez que este cambio es efectivo, se mantiene la estructura de la red, mientras el vehículo con el nodo 0 sigue la ruta, superando el vehículo con el nodo 14 (Fig. 3.15). Después de esta segunda maniobra de adelantamiento, la distancia entre los nodos 0 y 13 es mayor que 90 metros, y el algoritmo GeoNetworking decide hacer un segundo cambio en la estructura de la red, como se muestra en la Fig. 3.15. En este caso la estructura de enrutamiento de red reconfigura de enlace directo de los dos nodos con el nodo raíz a $13 \rightarrow 14 \rightarrow 0$, para optimizar la estructura de enrutamiento de acuerdo con las posiciones geográficas de los nodos. En este caso, la duración de la estructura de la red reconfiguración fue de 0,6s, lo que podría causar un retardo de 0,1s en la transmisión de un mensaje desde los nodos 13 y 14. Sin embargo, este retraso no se aprecia en los conjuntos de datos registrados durante la prueba.

FIGURA 3.15: Segunda reconfiguración de la VANET

3.2. Tecnología DSRC

En este apartado se explican todas las características de la tecnología de comunicación *Dedicated Short Range Communications* (DSRC) como soporte a las comunicaciones vehiculares, hardware, software, estándares utilizados, la arquitectura y algoritmos desarrollados, así como los resultados obtenidos.

Con la liberación de las normas de comunicación ETSI ITS-G5 (663, 2009) y IEEE 802.11p (for Information technology, 2010), la aparición del hardware adecuado para soportar dicha norma y la liberación de las normas de ITS para comunicaciones vehiculares de la ETSI (*European Telecommunications Standards Institute*), ha posibilitado dar un paso más hacia el uso de redes vehiculares con unas tasas de transferencia de datos muy superiores y empezar a pensar que en un futuro cercano el uso de VANETs para funciones de seguridad y gestión de tráfico será posible.

Como parte del trabajo de esta tesis, se ha desarrollado la tecnología hardware y software necesaria para implementar un módulo de comunicaciones DSRC robusto, interoperable y flexible, que pueda dar soporte a aplicaciones basadas en sistemas cooperativos. Para ello se han seguido los siguientes pasos: Un estudio pormenorizado de los estándares de la ETSI de comunicaciones vehiculares y del protocolo IEEE 802.11p para conocer las características de estos y poder implementarlos, dado que no existen aún en *open source*, creación de un módulo ITS capaz de ejecutar ambas normas; desarrollo de un *software* multiplataforma, que pueda ejecutarse en módulos desarrollados por otros organismos o empresas, capaz de administrar las VANETs; prueba de los protocolos implementados y por ultimo creación de aplicaciones de sistemas cooperativos.

3.2.1. Descripción de los protocolos de comunicación normalizados

En este apartado se hace una introducción a los principales protocolos de comunicación de la ETSI respecto a las VANETs, debido a que el estudio pormenorizado de estos estándares es parte esencial de la presente tesis dado que aún no han sido implementados en un módulo DSRC. Este estudio ha descubierto algunas inconsistencias o fallos en los estándares que han sido necesarios subsanar para su implementación práctica, por ello se procederá a explicar las partes más importantes y esenciales de éstos. Estos estándares son los llamados estándares de geoposicionamiento o GeoNetworking. Como se ha explicado anteriormente los elementos de una VANET están en continuo movimiento haciendo que los caminos de enrutamiento de paquetes no sean constantes, por lo cual los algoritmos contenidos en los estándares tienen que ser robustos y capaces de modificar las rutas dinámicamente para adaptarse al entorno de cada momento.

La ETSI define los módulos de comunicación DSRC como módulos ITS que significa por sus siglas en inglés sistemas de transporte inteligente (*Intelligent Transport Systems*). En los estándares de GeoNetworking a los actores de la red se les divide en dos grandes grupos:

- Vehículos (**V**): Todo elemento que tenga la posibilidad de cambiar su posición con relativa facilidad móvil que se encuentre en la vía urbana o interurbana y que posea un módulo de comunicación ITS.
- Puntos de carretera (**R**): Los “Puntos de carretera” o “*Roadside*” son módulos de comunicación ITS fijos en la vía que sirven como puntos de apoyo a la red y/o recolección de datos. Estos pueden estar conectados a otras redes de comunicaciones, por ejemplo una intranet, Internet o incluso redes 3G, 4G o 5G, aunque no es un requisito del estándar, es una posibilidad.

ETSI tiene una familia de normativas relacionadas con los algoritmos de GeoNetworking en las comunicaciones vehiculares en sistemas inteligentes de transporte:

- ETSI EN 302 636: “*Intelligent Transportation System (ITS); Vehicular communications; GeoNetworking*”.
- ETSI EN 302 637 (*all parts*): “*Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; multiple parts*”.
- ETSI TS 102 687: “*Intelligent Transport Systems (ITS); Transmitter Power Control Mechanism for Intelligent Transport Systems operating in the 5 GHz range*”.
- ETSI TS 102 760-2: “*Intelligent Transport Systems (ITS); Test specifications for Intelligent Transport Systems; Communications Access for Land Mobiles (CALM); Medium Service Access Points (ISO 21218); Part 2: TSS&TP (Test Suite Structure and Test Purposes)*”.
- ETSI TS 102 797-2: “*Intelligent Transport Systems (ITS); Road Transport and Traffic Telematics (RTTT); Test specifications for Intelligent Transport Systems, Communications Access for Land Mobiles (CALM), Interface Manager (ISO 24102); Part 2: Test Suite Structure and Test Purposes (TSS & TP)*”.
- ETSI TS 102 859-2: “*Intelligent Transport Systems (ITS); Testing; Conformance test specifications for Transmission of IP packets over GeoNetworking; Part 2: Test Suite Structure and Test Purposes (TSS&TP)*”.
- ETSI TS 102 868-2: “*Intelligent Transport System (ITS); Testing; Conformance test specification for Co-operative Awareness Messages (CAM); Part 2: Test Suite Structure and Test Purposes (TSS&TP)*”.

- ETSI TS 102 869-2: “*Intelligent Transport System (ITS); Testing; Conformance test specification for Decentralized environmental Notification Messages (DNM); Part 2: Test Suite Structure and Test Purposes (TSS&TP)*”.
- ETSI TS 102 870-2: “*Intelligent Transport Systems (ITS); Testing; Conformance test specifications for GeoNetworking Basic Transport Protocol (BTP); Part 2: Test Suite Structure and Test Purposes (TSS&TP)*”.
- ETSI TS 102 871-2: “*Intelligent Transport Systems (ITS); Testing; Conformance test specifications for GeoNetworking ITS-G5; Part 2: Test Suite Structure and Test Purposes (TSS&TP)*”.

El desarrollo de estos estándares está, además, realimentado a través de proyectos Europeos tales como COMeSafety2, EasyWay, ITS SV6, Intellidrive, euroFOT, FOT-NET2, FOTsis, PRESERVE, TeleFOT o Drive C2X.

En las siguientes secciones se explicarán los estándares más relevantes desde los aspectos más generales a los más complejos. Se entrará en detalle solo en las partes que se consideren más importantes para su implementación.

3.2.1.1. Escenarios

En el estándar ETSI EN 302 636-1 (636-1, 2014) se exponen las principales características que debe cumplir un sistema de comunicaciones inalámbrico ITS. En este documento se explican a grandes rasgos sin entrar en el detalle una serie de protocolos de red basados en las posiciones geográficas de los elementos de la red. En él se explican algunos de los métodos ya descritos en el capítulo 2 y otros métodos de transmisión de paquetes por la red vehicular.

- **GeoUnicast:** Comunicación punto a punto entre dos actores de la red utilizando si fuera necesario otros nodos como soporte para mantener la comunicación.
- **GeoBroadcast:** Consiste en difundir un mensaje en un área específica.
- **Topologically-Scoped Broadcast:** La comunicación “*Topologically-scoped broadcast*” (TSB) o “GeoDifusión por vecindad”, consiste en realizar una difusión de un paquete a los “*n*-vecinos” más próximos, es decir el paquete se difundirá por la red “*n*-saltos”. Para el caso de $n = 1$ se denomina como Single-Hop Broadcast (SHB). En la figura 3.16 se observa un ejemplo de este tipo de comunicación, en el este caso $n = 2$.

En el estándar ETSI EN 302 636-2 (636-2, 2007) se definen los posibles escenarios de comunicación entre los diferentes actores de la red “V” y “R”

FIGURA 3.16: *Topologically-Scoped Broadcast*: los vehículos naranjas son los que están en comunicación, y los vehículos azules no intervienen en la comunicación

- Punto a punto: Comunicación entre un módulo ITS y otro módulo ITS.
- Punto a multipunto: Comunicación entre un módulo ITS y múltiples módulos ITS.
- GeoAnycast: Comunicación entre un módulo ITS y un módulo ITS arbitrario contenido en un área geográfica.
- GeoBroadcast: Comunicación entre un módulo ITS y todos los módulos ITS contenidos en un área geográfica.

Por lo cual se pueden distinguir cada uno de los escenarios descritos anteriormente para cada tipo de comunicaciones, “V2V”, “V2R” y “R2V”. Dado que utilizar las comunicaciones inalámbricas con los protocolos de GeoNetworking no tendría mucho sentido en el escenario “R2R”, pues lo más lógico será que utilicen otro tipos de redes de comunicación, más sencillas, para comunicarse entre ellos.

3.2.1.2. Arquitectura de los protocolos de geoenrutamiento

En el estándar ETSI EN 302 636-3 (636-3, 2014) se detalla cómo es la arquitectura de los protocolos de geoenrutamiento. Este protocolo de comunicación sustituye la parte de IP de la torre OSI de comunicaciones, dando soporte a TCP, UDP, BTP (*Basic Transport Protocol*) (anexo A.2) y otros protocolos basados en IP y GeoNetworking como se puede observar en la Figura 3.17, realizando las funciones que hace el protocolo IP, tales como: enrutamiento de paquetes, eliminación de bucles si se llegaran a producir y detección de paquetes repetidos, así como la organización de la red.

3.2.1.3. Estándar de GeoNetworking

En el estándar ETSI EN 302 636-4-1 (636-4-1, 2014) es donde se describe el funcionamiento de GeoNetworking por lo cual es el estándar más importante de todos. El resto de estándares que se explican en el presente documento sirven de soporte o aclaración de éste. El estándar (636-4-1, 2014) es la piedra angular de todas las comunicaciones basadas

FIGURA 3.17: Detalles de la arquitectura OSI de GeoNetworking

en geoposicionamiento pues en él se explican los algoritmos y pasos que siguen los diferentes paquetes de datos en función del tipo que sean.

En el Anexo A.1 se ha detallado con minuciosidad todos los elementos destacables de este estándar así como una explicación detallada de los diferentes elementos y procesos recogidos en estos.

3.2.1.4. Protocolos Transporte

Como se ha comentado en la sección 3.2.1.3 en los estándares de geoposicionamiento existe el GN que ocupa el nivel de red y los protocolos de transporte. Se pueden observar algunos de ellos en la Figura 3.17

En el Anexo A.2 se explica con detalle el estándar ETSI EN 302 636-5-1 (636-5-1, 2014) que es el protocolo de transporte implementado de la capa de transporte.

3.2.2. Algoritmos de enrutamiento

A continuación se explican detalladamente los algoritmos específicos que vienen recogidos en el estándar ETSI EN 636-4-1 (636-4-1, 2014) para la comunicación GeoUnicast y GeoBroadcast.

3.2.2.1. GeoUnicast: *Greedy*

Cuando se desea mandar un mensaje de un nodo a otro nodo de la red sin que las aplicaciones del resto de nodos tengan conocimiento de este mensaje se envía un mensaje del tipo GeoUniCast (GUC). En los algoritmos de GeoNetworking se especifica que este tipo de mensajes son enviados mediante el algoritmo *Greedy*, este algoritmo ya se ha comentado en el estado del arte pero ahora lo vamos a explicar más detalladamente.

Cuando *Greedy* recibe el encargo de enviar un mensaje se le indica la posición geográfica donde se encuentra el nodo al que se le quiere mandar el mensaje y la dirección identificativa de este (GN_Addr). *Greedy* calcula la distancia existente entre la posición actual del nodo y la de destino, para tener una referencia, a continuación busca en su base de datos todos los nodos que están en su campo de visión (nodos vecinos) y determina cual es el que está más cerca para enviarle el mensaje.

Greedy entonces comunica a la instancia que le ha llamado cual es el vecino al que tiene que enviar el mensaje mediante su GN_Addr. En el caso de que el nodo sea el más cercano al destino bien, porque no hay ningún vecino en esa dirección, o porque no hay vecinos, si está habilitada la posibilidad de almacenar el paquete a la espera de que se presenten opciones óptimas para su envío, este se almacena en el buffer correspondiente al tipo de paquete y se devuelve un GN_Addr con valor 0 para indicar que el mensaje ha sido almacenado. En cualquier otro caso se le indica a la instancia que le ha llamado que difunda el mensaje a todos los nodos de la red a ver ellos conocen un camino más óptimo para alcanzar dicho nodo. Se puede observar el pseudocódigo del algoritmo en Alg. 1.

Si resulta que el más cercano es el nodo objetivo del paquete, el nodo receptor tratará el mensaje y lo subirá al nivel de transporte, si por el contrario, el mensaje es entregado a otro nodo este nodo ejecutara *greedy* para encontrar el nodo destino actuando como relé.

Algoritmo 1 Pseudo código del algoritmo *Greedy*.

Entrada: Posición de destino.

Salida: LL address del siguiente salto: dirección de envío o 0.

```

 $MFR = Dist(PV_P, LocPV)$ 
1: for  $i \in LocT$  do
2:   if  $i$  es Vecino then
3:     if  $Dist(PV_P, PV_I) < MFR$  then
4:        $NH \leftarrow i$ 
5:        $MFR \leftarrow DIST(PV_P, PV_I)$ 
6:     end if
7:   end if
8: end for
9: if  $MFR < DIST(PV_P, LPV)$  then
10:  SET NH_LL_ADDR  $\leftarrow$  NH_LL_ADDR
11: else
12:  if LocT no tiene vecinos y el flag TC_SCF esta habilitado then
13:     $B \leftarrow P$ 
14:    SET NH_LL_ADDR  $\leftarrow$  0
15:  else
16:    SET NH_LL_ADDR  $\leftarrow$  Broadcast
17:  end if
18: end if
19: return NH_LL_ADDR

```

Dónde:

- P es el paquete que se quiere enviar (paquete de tipo GUC o GBC en algunos casos específicos que se explican en el algoritmo de GeoBroadcast).
- i es el i -ésimo elemento de la LocT.
- NH (next hop) es el identificador del siguiente salto en la LocT. NH.LL_ADDR es la dirección MAC de NH (*link layer address*).
- NH_LL_ADDR es la dirección MAC del siguiente salto.
- LocPV es donde se guarda la posición actual del módulo (*local position vector*).
- PV_P es la posición donde se tiene que enviar el paquete, esta información viene contenida en el propio paquete a enviar (*Position vector Packet*).
- PV_I es la última posición conocida del i -ésimo elemento de la Location Table (LocT).
- MFR es el *most forward within radius*, variable donde se almacena la mínima distancia entre la posición a enviar el paquete y un vecino.
- B es el buffer de envío de paquetes. Dependiendo del tipo de paquete será el buffer de Unicast o el de Broadcast.
- LocT es la *location table*.
- TC es el tipo de tráfico de datos (*traffic class*) del GN SAP Request o el de la Common header si se está reenviando el paquete.

3.2.2.2. GeoBroadcast: Simple GeoBroadcast

Cuando se desea mandar un mensaje de un nodo a una región limitada por un área de interés se utiliza un mensaje del tipo GeoUniCast (GUC). En los algoritmos de GeoNetworking se especifica que este tipo de mensajes son enviados mediante el algoritmo *Simple GeoBroadcast*.

Cuando el algoritmo recibe el encargo de calcular el siguiente salto para un tipo de paquete GeoBroadcast, lo primero que hace es calcular su posición respecto al área de difusión, mediante los cálculos especificados en la ETSI EN 302 931 (931, 2014) (sección A.1.5), si se encuentra dentro del área comunica a la instancia superior que el siguiente salto es de tipo *broadcast* y termina el algoritmo. Si el nodo emisor del mensaje está en la LocT y su posición es certera, se calcula si el emisor está dentro del área de difusión o no, si el emisor está dentro descarta el paquete y se lo indica a la estancia superior, si el emisor no está dentro del área llama al algoritmo de GeoUnicast *Greedy* y devuelve a la instancia superior

lo que conteste *Greedy*. En el caso de que el nodo emisor del mensaje no está en la LocT o su posición no es certera se le indica a la instancia superior que el siguiente salto es de tipo *broadcast*. En el Alg. 2 se puede observar el pseudocódigo de *Simple GeoBroadcast*.

Algoritmo 2 Pseudo código del algoritmo *Simple GeoBroadcast*

Entrada: Posición de destino.

Salida: LL address del siguiente salto: dirección de envío, 0 o -1

```

1: Calcular  $F(Lat, Long)$ 
2: if  $F \geq 0$  then
3:   return NH_LL_ADDR  $\leftarrow$  BCAST
4: else
5:   if  $\exists PV_{SE} \text{ Y } PAI = True$  then
6:     Calcular  $F(Lat_{SE}, Long_{SE})$ 
7:     if  $F < 0$  then
8:       return NH_LL_ADDR  $\leftarrow$  Greedy(A)
9:     else
10:      Descartar P
11:      return -1
12:   end if
13: else
14:   return NH_LL_ADDR  $\leftarrow$  BCAST
15: end if
16: end if

```

Dónde:

- P es el paquete que se quiere enviar.
- Lat y Long son la latitud y longitud del LocPV, respectivamente.
- PV_{SE} es la posición del módulo que envía el paquete contenida en la LocT con latitud Lat_{SE} , longitud $Long_{SE}$, y el indicador de calidad de la posición PAI_{SE}
- A es el punto central del área de destino del paquete.
- NH_LL_ADDR es la dirección MAC del siguiente salto.
- BCAST es la dirección genérica de *broadcast*.
- Greedy() es el algoritmo de GeoUnicast.
- B es el buffer de envío de paquetes específico de paquetes de tipo de GeoUnicast.

3.2.3. Utilización de la tecnología DSRC para comunicaciones V2X

A continuación se detalla el trabajo desarrollado sobre esta tecnología primero se describirá el módulo desarrollado, el sistema operativo implementado y por último el software utilizado en los módulos.

3.2.3.1. Módulo de comunicaciones DSRC-UPM-INSIA

Antes de decantarnos por el desarrollo de un módulo propio de DSRC se indagó en los módulos comerciales. Debido a que es una tecnología reciente y que hay poca oferta los precios de dichos módulos es elevado, con el añadido que en el precio solamente viene incluido el hardware sin una implementación de los estándares. Por lo cual se decidió crear unos módulos propios DSRC-UPM-INSIA (Figura 3.18(e)).

Para la creación de los módulos primeramente se definieron unos requisitos básicos: los módulos tenían que ser asequibles, para que su integración en los vehículos del actual parque automovilístico no suponga una barrera, también tiene que ser de reducido tamaño, para que su integración en los vehículos sea lo más sencilla posible al poderse colocar en algún lugar recóndito del vehículo y de esta manera sería viable su montaje en vehículos de dimensiones limitadas como motocicletas.

Lo primero que se realizó es una búsqueda de las tarjetas de comunicación que fueran capaces de soportar el estándar IEEE 802.11p. Todas las tarjetas encontradas tienen una particularidad tienen una conexión miniPCI lo cual limita mucho los dispositivos donde se pueden integrar fácilmente siguiendo los requisitos básicos, Figura 3.18(b).

Al final y después de mucho buscar se encontró una placa base de reducidas dimensiones la cual tiene un bajo consumo energético (15W a 18V) y la potencia de cálculo es suficiente para procesar el estándar de comunicaciones y aplicaciones adicionales, las características de esta placa base se pueden ver en la Tabla 3.10 al tener dos bahías miniPCI se le puede añadir un módulo que le de nuevas cualidades por ejemplo una tarjeta de comunicaciones adicional para darle otras utilidades al módulo, un GPS integrado o una tarjeta de telefonía móvil. Al no tener partes móviles es más robusto a las vibraciones producidas por la rodadura de los vehículos.

TABLA 3.10: Características placa base módulos de comunicaciones

CPU	500 MHz	
Memoria	DRAM	256 MB
	Almacenamiento	CompactFlash
Comunicaciones	2 bahías miniPCI	
	1 Canal Ethernet (Via VT6105M 10/100)	
	2 puertos USB 2.0	
	Puerto serie DB9	

Dado que es un módulo DSRC diseñado para la creación de VANETs necesita poseer un módulo de geolocalización. Inicialmente se ha elegido un GPS bluetooth (Figura 3.18(c)) que opera a 10 Hz para poder separar físicamente este del módulo DSRC-UPM-INSIA, para lo cual se le ha proporcionado al módulo de comunicación Bluetooth mediante un adaptador

USB. Finalmente se ha desarrollado un módulo de geoposicionamiento (Figura 3.18(d)) el cual tiene mejores cualidades de precisión al poder recibir la señal de los satélites de GPS, GLONAS y Galileo así como tener corrección diferencial, también opera a 10Hz, y se conecta al módulo DSRC-UPM-INSIA por medio de USB.

FIGURA 3.18: Componentes y DSRC-UPM-INSIA

Dado que la mayoría de módulos estarán en los vehículos es importante que toda la información posible de este esté disponible para las aplicaciones ADAS por lo cual se ha añadido un lector de *Bus Can* al DSRC-UPM-INSIA que recoge toda la información proporcionada por el vehículo: velocidad, ángulo de volante, la posición de los pedales, si están o no accionadas las luces, limpiaparabrisas, etc.

Dado que el módulo DSRC-UPM-INSIA está concebido únicamente como un módulo de comunicaciones, es necesario que los sistemas cooperativos se ejecuten en otra máquina y se comuniquen con él. Existen varias alternativas para comunicarse con este, mediante el USB, Rs232 o *Ethernet*, lo que implica un cable que una a ambos. Actualmente dentro de un vehículo el computador más potente que hay es el smartphone del conductor, por lo cual se ha decidido utilizar la segunda bahía miniPCI colocando una tarjeta de comunicaciones inalámbricas que soporte el estándar IEEE 802.11ac de esta manera se dota al módulo de la posibilidad de comunicarse inalámbricamente con dispositivos que no soporten el estándar IEEE 802.11p. Utilizando esta tarjeta de comunicaciones se han configurado por software

dos redes inalámbricas con IEEE 802.11ac diferentes. Una específica por cada módulo con seguridad de acceso *WPA* de 128 bits, la otra es una red abierta la cual tiene muy limitadas las posibilidades de interactuar con ella.

La configuración final del módulo DSRC-UPM-INSIA es: una CompactFlash de 8GB, suficiente para ejecutar el OS y los programas desarrollados para soportar el estándar, dos tarjetas de comunicaciones inalámbricas, una que soporta el estándar IEEE 802.11p y otra con el estándar IEEE 802.11ac que crea dos redes inalámbricas, un módulo de geoposicionamiento y un adaptador Bluetooth-USB (Figura 3.18(a)).

Una vez elegidos todos los componentes y ensamblados se ha procedido a elegir el sistema operativo más adecuado para que el módulo pueda cumplir con todas las exigencias software que los estándares requieren.

3.2.3.2. Sistema Operativo

Para permitir las comunicaciones en VANETs las tarjetas de red deben operar en modo Ad-Hoc para que no sea necesario un punto de acceso. Tras probar diferentes sistemas operativos nos hemos decantado por utilizar uno basado en Linux pues es fácil de poner las tarjetas de red en modo Ad-Hoc y al ser de código abierto la modificación de los drivers de los componentes son más sencillas en caso de ser necesario.

3.2.3.3. Software de soporte para los estándares

Para la implementación de los estándares se ha decidido utilizar en la medida de lo posible lenguajes de programación multiplataforma para que de esta manera todo el software desarrollado pueda ejecutarse en cualquier dispositivo que cumpla los requisitos mínimos. Estos requisitos son: poseer una tarjeta que soporte el estándar IEEE 802.11p y que tenga el driver adecuado.

Por ello se ha utilizado principalmente *Java SDK* el cual gracias a su Máquina Virtual (JVM) es capaz de adaptarse a todos los sistemas sin importar las diferentes configuraciones hardware. Cuando no se ha podido utilizar *Java* por las limitaciones intrínsecas a esta plataforma se han utilizado lenguajes de programación que siguen esta filosofía y que solucionaban los problemas plantados como son *C++* y *Python* los cuales son lenguajes de programación multiparadigma y multiplataforma.

Gracias a estos lenguajes de programación se ha conseguido realizar la programación de los estándares antes descritos de forma que se puedan ejecutar en cualquier dispositivo.

3.2.4. Problemas y modificaciones en los estándares

Al implementar los diferentes estándares, se han ido descubriendo diferentes casos que o bien no han sido contemplados a la hora de desarrollar los estándares o no se han especificado en estos. Por lo que se han ido tomando diferentes decisiones para implementarlos y que se detallan a continuación.

3.2.4.1. Módulos

En el Anexo A.1.1 se detalla que en el estándar de GeoNetworking (GN) existe una Base de datos donde se guarda la información de todos los nodos conocidos de la red, esta base de datos se la conoce como *Location Table* (LocT) y un *Local Position Vector* (LocPV) el cual contiene toda la información del nodo. Según el estándar (636-4-1, 2014) estas bases de datos se renuevan cada vez que hay nuevos datos, que en el caso de la LocT ocurre cuando llega un paquete nuevo. Estas bases de datos son accesibles solo por dicho estándar.

Todos los submódulos de GN acceden a esta base de datos para tomar sus decisiones por lo cual el manejo de estas bases de datos es crucial y crítico.

3.2.4.1.1. Conocimiento de sí mismo

En el estándar se añaden nuevos módulos a la base de datos cuando llega un mensaje, el problema reside en que si llega un mensaje previamente retransmitido por el propio módulo comprueba si está contenido en la LocT y al no estar lo añade, poniéndose a sí mismo en la LocT, produciendo un problema en potencia pues puede darse el caso que un submódulo intente enviar un mensaje a si mismo pensando que es un módulo externo. Este comportamiento puede llevar a la pérdida de paquetes en el mejor de los casos.

La solución que se ha adoptado es una comprobación previa al incluir un nuevo módulo, se comprueba que la GN_Addr del módulo del mensaje no coincide con la GN_Addr del módulo receptor contenido en LocPV, de esta manera se evita este potencial error.

3.2.4.1.2. Vecindad

Uno de los campos de la LocT es la vecindad, este parámetro indica si el módulo contenido en la LocT es accesible mediante visión directa o no. Este dato es crucial para enviar los mensajes de un módulo a otro, solo se pueden utilizar como relés los módulos que tengan este campo activo en la tabla.

El problema que se ha detectado ocurre cuando un módulo deja de estar en visión directa y no hay comunicación entre ambos, bien porque no envíen mensajes de difusión (TSB, GBC o GAC) o de uno a otro (GUC) o porque no haya ningún módulo que haga de repetidor, en la LocT seguirá figurando como vecino cuando en realidad no lo es hasta que se elimine de esta porque los datos se hayan desfasado, se ha comprobado empíricamente que este tiempo es demasiado elevado para que la red funcione adecuadamente.

Hay dos posibilidades de afrontar este problema:

- Reduciendo el tiempo de estancia en la LocT antes de que se elimine el módulo de esta.
- Dada la naturaleza del estándar donde hay un paquete periódico (*Beacon*), que se explica en el Anexo A.1.3.1, utilizar un múltiplo reducido de la constante del *Beacon* para poner la vecindad como falsa.

Se ha implementado la segunda opción dado que ya hay implementado en la LocT un algoritmo periódico que elimina los módulos desfasados y no es necesario hacer muchos cambios. La primera opción se descartó porque si se reduce el periodo de estancia en la LocT el conocimiento de la red que no sea en visión directa se reduce produciendo una atomización de la red.

3.2.4.2. Adquisición de datos

En los estándares se establece que la adquisición de datos para la creación de la red de comunicación viene determinado por módulos externos que se comunican con GN a través del servicio de acceso al protocolo (SAP) entre el módulo de mantenimiento y GN (GN_Mgmt_SAP). Esto puede crear problemas cuando falla la adquisición de datos. En los estándares no se indica cómo se debe actuar en caso de algún fallo. A continuación se especifican los problemas detectados y las soluciones adoptadas para solucionarlos.

3.2.4.2.1. Posicionamiento, fallo de GPS

Un punto capital en el funcionamiento de las redes basadas en geoposicionamiento es poseer un conocimiento exacto de la posición. Aunque el estándar incorpora la calidad del establecimiento de la posición con el parámetro PAI, no tiene en cuenta otras circunstancias importantes como las que se exponen a continuación.

Si se produce un fallo en el GPS de recepción o corrupción de los datos y son transmitidos a GN los dará por buenos y producirá problemas en las comunicaciones y el mantenimiento de la red. Dada la naturaleza de los GPS utilizados durante el desarrollo de esta tesis, si un

GPS pierde la señal sigue transmitiendo la última posición conocida valida y esto produce problemas en la red creada por GN.

La solución adoptada es que cuando se detecta un problema en el GPS los valores de latitud y longitud se ponen a 0. Detectar un fallo en el GPS no es una tarea sencilla dado que puede deberse a muchos posibles casos. Falta de señal, fallo generalizado del GPS y para cada uno hay que detectarlo de una forma diferente.

3.2.4.2.2. Sincronización temporal y Roadsites

Si un módulo está definido como “Roadsite” se presupone que este módulo nunca cambia la posición por lo cual incluirle un GPS es un desperdicio de recursos. Se puede decir a GN que siempre retransmita la misma posición pues al colocarlo en su posición esta es conocida, el problema reside que sin un GPS en el módulo no hay una fuente de tiempo para que el resto de módulos puedan incluirlo en la red.

Por esa razón se ha tomado la decisión de utilizar la hora del sistema como referencia de los tiempos TST. Dado que puede haber cierta deriva en los relojes internos de los módulos se han sincronizado con la hora proporcionada por algún servidor externo que se accede a él a través de Internet.

Esta solución plantea un problema adicional, que pasa si un módulo nunca se puede conectar al servidor porque no tiene acceso a Internet. Por ejemplo un módulo instalado en un vehículo, suponiendo que el módulo estaba sincronizado al principio con el tiempo se producirá inevitablemente una deriva que acabara haciendo inservible ese módulo. Se ha encontrado una solución intermedia entre ambas proposiciones, si hay GPS integrado en el módulo, se sincroniza cada cierto periodo el reloj interno del módulo con la hora proporcionada por el GPS, si no hay GPS pero hay acceso a Internet el reloj interno se sincroniza con los servidores globales. En el caso de que no haya ninguna de las dos posibilidades cada cierto periodo de tiempo habrá que comprobar la deriva y poner a la hora correcta manualmente.

3.2.4.3. Uso de las áreas geográficas en coordenadas cartesianas

El estándar ETSI EN 363-4-1 (636-4-1, 2014) (Anexo A.1.5) define las normas que siguen los módulos de comunicación que reciben un paquetes de los tipo “GeoBroadcast” o “GeoAnycast” lo primero que el módulo realiza es el análisis del área de interés para conocer si se encuentra en el área de interés definido en el paquete o no. Para ello hay que utilizar las ecuaciones reflejadas en dicho estándar (ecuaciones: A.3, A.4 o A.5) en función del tipo de área. Estas ecuaciones son muy fáciles de aplicar en un sistema de coordenadas cartesianas,

pero el problema viene cuando se intentan aplicar en el sistema de posicionamiento global, empiezan a surgir problemas de aplicación, especialmente en los casos del área rectangular y del área elipsoidal.

Por lo cual se optó por hacer una transformación a coordenadas polares que simplifica las operaciones. Lo más importante es calcular la distancia entre puntos en coordenadas geográficas ec. 3.5, pues esta distancia se comparará con la distancia entre el centro y el borde de las áreas. También se optó por simplificar la ecuación A.2 resultando la ecuación 3.7

$$D_{0M} = \text{distHaversine}(P_0(lat, lon), P_{\text{Módulo}}(lat, lon)) \quad (3.5)$$

dónde:

- D_{0M} es la distancia entre el punto central del área de difusión y el módulo
- distHaversine es la distancia entre dos puntos de la tierra calculada con la Fórmula del Haversine 3.6
- $P_0(lat, lon)$ es el punto central del área de difusión.
- $P_{\text{Módulo}}(lat, lon)$ es el punto donde se encuentra el módulo.

$$\begin{aligned} R &= 6371,137 \\ \Delta lat &= lat2 - lat1 \\ \Delta long &= long2 - long1 \\ a &= \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat1) \cdot \cos(lat2) \cdot \sin^2\left(\frac{\Delta long}{2}\right) \\ c &= 2 \cdot \text{atan}^2(\sqrt{a}, \sqrt{1-a}) \\ d &= R \cdot c \end{aligned} \quad (3.6)$$

dónde:

- R es el radio de la tierra en metros
- $lat1$ y $lat2$ son la latitud en grados del punto 1 y 2 respectivamente.
- $long1$ y $long2$ es la longitud en grados del punto 1 y 2 respectivamente
- d es la distancia entre puntos en metros.

$$F(lat, lon) = \begin{cases} 1 & \text{si } D_{0M} < D_{0A} \rightarrow P_{\text{Módulo}} \text{ está dentro área} \\ 0 & \text{si } D_{0M} = D_{0A} \rightarrow P_{\text{Módulo}} \text{ esta en el borde del área} \\ -1 & \text{si } D_{0M} > D_{0A} \rightarrow P_{\text{Módulo}} \text{ esta fuera área} \end{cases} \quad (3.7)$$

dónde:

- D_{0M} es la distancia entre el centro del área y el módulo.
- D_{0A} es la distancia entre el centro del área y el borde del área.

Otro punto importante para determinar si P_M está dentro del área es la orientación de esta, en el caso de la circunferencia no importa, pero en el caso del rectángulo y de la elipse es de una importancia capital. El estándar (931, 2014) define el ángulo azimutal (θ) medido desde el Norte y creciente en sentido horario, en coordenadas polares el ángulo (α) está definido en sentido antihorario siendo 0° el eje X . Por ello hubo que hacer una transformación de grados N-E a grados en polares. Para posteriormente poder calcular el ángulo relativo que tiene el módulo respecto del área de interés.

3.2.4.3.1. Circunferencia

Para este caso no es necesario hacer una transformación específica pues la distancia D_{0A} es constante, r , así pues solo es necesario comparar D_{0M} y D_{0A}

3.2.4.3.2. Rectángulo

Así como para la circunferencia y la elipse los cálculos necesarios en polares se simplifican ostensiblemente para el caso del rectángulo no es tan claro.

Lo primero que hay que calcular son los cuatro ángulos que tienen los vértices del rectángulo, que dependen de la longitud de los lados (ec. 3.8) y la orientación de este en el plano.

$$\beta = \arctg\left(\frac{b}{a}\right) \quad (3.8)$$

Una vez que conocemos los cuatro ángulos de los vértices β y el ángulo que forma el módulo con el centro del área calculamos el ángulo que tiene el módulo respecto del área α . Entonces podemos calcular ρ con la ecuación 3.9 en función del ángulo que forma la recta imaginaria entre el módulo ($P_{\text{Módulo}}$) y el centro del área (P_0), respecto al rectángulo.

$$\rho(\alpha) = \begin{cases} \left| \frac{a}{\cos(\alpha)} \right| & \text{si } \alpha = [0, \beta) \\ \left| \frac{b}{\sin(\alpha)} \right| & \text{si } \alpha = [\beta, 180 - \beta) \\ \left| \frac{a}{\cos(\alpha)} \right| & \text{si } \alpha = [180 - \beta, 180 + \beta) \\ \left| \frac{b}{\sin(\alpha)} \right| & \text{si } \alpha = [180 + \beta, 360 - \beta) \\ \left| \frac{a}{\cos(\alpha)} \right| & \text{si } \alpha = [360 - \beta, 360) \end{cases} \quad (3.9)$$

dónde:

- ρ es la distancia del centro del rectángulo (O) a un punto de esta en función del ángulo.
- α es el ángulo entre el centro del rectángulo (O) y un punto M .

Una vez hallado $\rho(\alpha) = D_{0A}$ se puede aplicar la ecuación 3.7 para determinar si el módulo está dentro o fuera del área de interés.

3.2.4.3.3. Elipse

Las características de la elipse (ver figura 3.19) permiten que el paso a coordenadas polares sea más sencillo por la propiedad de la elipse que hace que la suma de las distancias de un punto de esta a sus focos sea constante 3.10.

FIGURA 3.19: Características de la elipse

$$MF_1 + MF_2 = 2a \quad (3.10)$$

dónde:

- a es la medida del semieje mayor de la elipse.
- MF_1 es la distancia del punto M al foco 1 (MF_1).
- MF_2 es la distancia del punto M al foco 2 (MF_2).

Gracias a esta característica podemos deducir la ecuación 3.11 que especifica la distancia del centro de la elipse a un punto de esta en función del ángulo (ver figura 3.19).

$$\rho(\alpha) = \frac{1}{\sqrt{\frac{\cos^2(\alpha)}{a^2} + \frac{\cos^2(\alpha)}{b^2}}} \quad (3.11)$$

dónde:

- ρ es la distancia del centro de la elipse (O) a un punto de esta en función del ángulo.
- α es el ángulo entre el centro de la elipse (O) y un punto M .

Una vez hallado $\rho(\alpha) = D_{0A}$ se puede aplicar la ecuación 3.7 para determinar si el módulo está dentro o fuera del área de interés.

3.2.4.4. Acceso a la información del módulo

El conocimiento de los nodos de la red es esencial para que las aplicaciones a las que se les va a dar soporte, esto es debido a que la utilidad principal del geoposicionamiento es la seguridad vial, la regulación del tráfico así como otros servicios. Si para el conocimiento de la red cada aplicación que opera en el módulo o aplicación externa debe enviar un “Beacon” propio se está utilizando ancho de banda innecesariamente cuando el propio GN en el módulo tiene una base de datos con el conocimiento de todos los nodos de la red. Por tal motivo se han implementado una serie de peticiones al GN para que devuelva esos datos.

Ya existen puertos asignados en el protocolo BTP para distintos servicios de aplicaciones por lo cual se ha optado en crear un nuevo puerto de comunicación en el protocolo BTP que será el encargado de tratar estas nuevas peticiones a través de unos nuevos subtipos de paquetes de GN. Se pueden observar en la Tabla ?? que complementarían la Tabla A.4.

Header Type (HT)	Header Sub-type (HST)	Encoding	Description
DB		7	Acceso a datos del módulo
	All	0	Obtiene LocPV + LocT
	Local	1	Obtiene LocPV
	Database	2	Obtiene LocT
	Module	3	Obtiene LPV de un módulo específico
	Neighbour	4	Obtiene los GNADDR de los módulos vecinos

3.2.5. Arquitectura para sistemas cooperativos

Para la implementación se ha tenido que ir complementando el modelo OSI (Figura 2.4), dado que los protocolos de geoposicionamiento influyen en varias capas del modelo OSI.

Por las características de los módulos DSRC-UPM-INSIA que poseen dos tarjetas inalámbricas de comunicaciones una que opera en IEEE 802.11p y otra en 802.11ac, existen dos pilas ISO diferentes que se comunican internamente entre ellas y externamente entre pilas semejantes.

3.2.5.1. Segunda tarjeta de comunicaciones (IEEE 802.11ac)

Esta tarjeta se ha incluido para que la interacción con los módulos sea más cómoda y para crear un punto de acceso donde dispositivos que no tengan la posibilidad de tener comunicaciones 802.11p se conecten a este módulo y puedan mandar paquetes a través de la red de GN.

Esta configuración puede causar problemas de seguridad en el módulo pues para que los dispositivos sin posibilidad de conectarse a redes IEEE 802.11p se conecten a la red el punto de acceso tiene que estar sin clave, por ello sobre esta tarjeta se han creado dos redes wi-fi: Una específica del módulo con una SSID personalizada y con seguridad “WPA” para comunicarse con este y poder acceder a todos los parámetros de GN y administrar el módulo en general. La segunda red tiene el mismo SSID en todos los módulos y sin seguridad para que todo dispositivo se pueda conectar, esta red está muy limitada en sus atributos, únicamente tiene un puerto disponible para comunicarse con la aplicación desarrollada para fines de comunicación que se explica más detalladamente en la sección 3.2.6.4.

3.2.5.2. Pila OSI IEEE 802.11ac

Esta pila corresponde al protocolo de comunicaciones TCP-IP por lo cual no se explica con detenimiento pues no es el objetivo de este trabajo. En la Figura 3.20 se pueden observar las distintas capas que componen dicho protocolo. En el protocolo TCP/IP el nivel de transporte abarca la función de varias capas.

3.2.5.3. Pila OSI para geoposicionamiento

A continuación se explican los protocolos que están involucrados en la geocomunicación. En la Figura 3.21 se pueden observar la estructura y a que capas corresponde de la torre OSI.

3.2.5.3.1. Nivel físico

En el nivel físico está situada la tarjeta de comunicaciones IEEE 802.11p para la cual se está realizando un driver específico para que pueda operar a las frecuencias requeridas.

FIGURA 3.20: Pila OSI para TCP-IP

FIGURA 3.21: Pila OSI para geoposicionamiento

3.2.5.3.2. Nivel de enlace

Dado que el protocolo de GeoNetworking tiene la cualidad de poder emitir mensajes en Broadcast o Unicast hay que filtrar los mensajes por MAC que es la identificación a más bajo nivel que existe, para ello se ha desarrollado un driver específico LLC (*Link Layer Control*) el cual es el encargado de filtrar los mensajes adecuados para el módulo. De esta manera se aligera la carga computacional de los niveles superiores.

FIGURA 3.22: Formato de trama transmitida por la capa de enlace

Este driver es el encargado de generar los paquetes y de interpretarlos. Este driver utiliza “*Raw socket*” para enviar y recibir los mensajes generando la cabecera MAC e interpretarla. En la Figura 3.22 se pueden observar los campos de la cabecera MAC. Hay que prestar especial atención al campo “Tipo de datos” el cual indica el tipo de datos que están contenidos en el campo de datos. En este caso el identificador para GeoNetworking es el 0x0707.

3.2.5.3.3. Nivel de red

El encargado de este nivel es el protocolo de GeoNetworking que ha sido explicado en la sección 3.2.1.3.

3.2.5.3.4. Nivel de transporte

En el nivel de transporte hay dos protocolos que se complementan en función de las necesidades de la aplicación, el “*Basic Transport Protocol*” (BTP) que es explicado en el anexo A.2 y el “*GeoNetworking Extension and Support*” (GNES) que se explica con más detalle en la sección 3.2.6.4. Estos protocolos son los encargados de asegurarse de que la información es correcta y dar servicio a las diferentes aplicaciones de ITS.

3.2.5.3.5. Nivel de aplicación

En este nivel se encuentran las aplicaciones de ITS, son protocolos encargados de diferentes campos de la comunicación ITS y dan soporte a las aplicaciones ITS. Los más importantes son CAM (*Cooperative Awareness Message*), DENM (*Decentralized Environmental Notification Message*), MAP (*Road Topology*), SPAT (*Signal Phase And Timing*), SAM (*Service Announcement Message*) que son estándares de ITS e ISO para las comunicaciones de tráfico. Adicionalmente se ha creado el *GeoNetworking Extension Protocol* (GNEP) que es el protocolo para que las aplicaciones ITS se puedan comunicar con GNES.

3.2.5.3.6. Protocolos de comunicación entre niveles

Dado que los diferentes estándares indican que campos tienen que intercambiarse entre las diferentes capas, pero ninguna información de cómo se deberían transmitir los datos ni el formato por lo que se ha decidido hacerlo como si fueran una cabecera formada por bytes.

3.2.5.3.6.1. Servicio de acceso al protocolo *Link Layer Control* (LLC-SAP)

Para que el protocolo de GeoNetworking mande y reciba mensajes se ha creado un protocolo de acceso al LLC: el LLC-SAP.

LLC-SAP *Request*

GeoNetworking cuando quiere mandar un mensaje tiene que ponerse en contacto con el nivel de enlace mediante un “Request”, este mensaje consta de la MAC de la tarjeta de comunicaciones del nodo de destino, y la información a mandar.

LLC-SAP *Indication*

Cuando el nivel de enlace recibe un mensaje destinado para ella, lo procesa y se lo comunica al GeoNetworking mediante un “Indication”. Este mensaje está compuesto de la información recibida y de la MAC de la tarjeta que lo ha enviado.

3.2.5.3.6.2. Servicio de acceso al protocolo GeoNetworking (GN-SAP)

En el Anexo A.1.4 se explican los campos de intercambio entre GeoNetworking y el nivel de transporte pero no se especifica el formato que deben tener estos mensajes. A continuación se muestra como es el intercambio entre esos dos campos.

GN-SAP *Request*

En la Figura 3.23 se puede observar los campos necesarios a rellenar por los protocolos situados en el nivel de transporte para poder enviar los datos a través del GeoNetworking.

FIGURA 3.23: *Request GN packet*

Los campos contenidos en la Figura 3.23 son explicados con detalle en el Anexo A.1.4

GN-SAP Indication

En la Figura 3.24 se puede observar los campos que devuelve GN.

FIGURA 3.24: Indication GN packet

Los campos contenidos en la Figura 3.24 son explicados con detalle en el Anexo A.1.4

3.2.5.3.6.3. Servicio de acceso al protocolo BTP (BTP-SAP)

En el Anexo A.2.1 se explican los campos que se intercambian entre el “Basic Transport Protocol” y el nivel de aplicación pero no se especifica el formato que deben tener estos mensajes, a continuación se muestra como es el intercambio de información entre las dos capas.

BTP-SAP Request

En la Figura 3.25 se puede observar la configuración de datos que envía el nivel de aplicación para que BTP retransmita el mensaje por la red.

FIGURA 3.25: Request BTP packet header

Los campos son explicados con detalle en el Anexo A.2.1.1

BTP-SAP Indication

En la Figura 3.26 se puede observar la configuración de datos que envía BTP al nivel de aplicación.

FIGURA 3.26: *Indication BTP packet header*

Los campos son explicados con detalle en el Anexo A.2.1.2

3.2.5.4. Arquitectura de geoposicionamiento implementada en los módulos DSRC-UPM-INSIA

Finalmente en la Figura 3.27 se puede observar la arquitectura implementada en los módulos DSRC-UPM-INSIA, los módulos de la arquitectura están coloreados con los mismos colores que en la figura de la pila OSI para que sea fácilmente identificable a que capa pertenecen.

Dónde:

- I: es el módulo encargado del “*Link Layer Control*” (LLC) (Sección 3.2.5.3.2).
- II: es servicio de acceso al protocolo LLC (Sección 3.2.5.3.6.1).
- III: es el módulo de GeoNetworking encargado de la comunicación por geoposicionamiento (Sección 3.2.1.3).
- VI: es servicio de acceso al protocolo GN (GN-SAP Anexo A.1.4).
- V: es el módulo “*Basic Transport Protocol*” (BTP) (Anexo A.2).
- VI: es servicio de acceso al protocolo BTP (BTP-SAP Anexo A.2.1).
- VII: es un módulo especialmente desarrollado para dar más funcionalidades a los protocolos de geoposicionamiento (GNES) (Sección 3.2.6.4).
- VIII: es servicio de acceso al protocolo GNES (GNES-SAP Sección 3.2.6.4.3).

- IX: son las aplicaciones que se sirven de la geocomunicación (En la sección 3.2.7 hay algunos ejemplos de aplicaciones desarrolladas).

FIGURA 3.27: Arquitectura de geoposicionamiento implementada

3.2.6. Algoritmos

En este apartado se describe el diseño del *software* desarrollado para que los algoritmos de geoposicionamiento puedan ser implementados y funcionen en tiempo real en los módulos DSRC-UPM-INSIA. A la hora de implementar los algoritmos se han intentado hacer mediante un lenguaje de alto nivel para que su implementación en diferentes máquinas sea lo más sencillo posible al abstraerse las funciones. Por ello siempre que ha sido posible se ha utilizado *Java* porque es un lenguaje de programación de propósito general, concurrente y orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "*write once, run anywhere*"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para

ejecutarse en otra plataforma deferente. De esta manera, toda la lógica de control del sistema de comunicaciones V2X puede ser migrada a otro dispositivo físico compatible con *Java* sin necesidad de realizar modificaciones en el código fuente. Esto es una gran ventaja, sobre todo a la hora de integrar las comunicaciones en dispositivos de terceros, tales como la consola de un vehículo o su electrónica integrada.

3.2.6.1. Algoritmo LLC

Se han analizado las características de las VANETs, en la capa de enlace, al operar de forma inalámbrica no hay medio físico que evite que todos los nodos se escuchen entre sí por lo cual se puede considerar una comunicación *broadcast*. Todos los nodos de las VANETs escuchan y luego filtran las tramas en función de si la trama va para ellos o no. En un principio se pensó en utilizar *sockets UDP* pero esto conllevaba que la arquitectura del algoritmo no se ajustaría al estándar. Por lo cual se buscaron alternativas encontrando los *Raw sockets*.

El *Raw socket* se utiliza comúnmente para enviar y recibir paquetes sin ningún formato específico en la capa de transporte. Es responsabilidad del usuario indicar en el *Raw socket* las características de los datos que van en la carga útil del paquete. En los *sockets* estándares, la carga útil a transmitir es encapsulada según el protocolo de la capa de transporte elegido (por ejemplo, TCP, UDP). Por el contrario, el *Raw socket* suele recibir paquetes crudos, incluyendo la cabecera. Cuando la transmisión de paquetes, la adición automática de una cabecera puede ser una opción configurable del socket.

FIGURA 3.28: Esquema del algoritmo LLC

De esta manera se cumple el estándar y se pueden identificar los paquetes pertenecientes a GeoNetworking, así como filtrar los paquetes que van destinados al nodo de los que son para otros nodos, liberando recursos de los dispositivos.

Dado que Java es un lenguaje interpretado que funciona en base a una máquina virtual que separa la lógica de la aplicación del hardware, implica que hay funciones que no son posibles de ejecutar, en este caso con *Java* no es posible utilizar *Raw socket*, la solución se haya en el uso de JNI. JNI son las iniciales de “*Java Native Interface*” el cual es un *framework* de programación que permite que un programa escrito en Java ejecutado en la Máquina Virtual de Java (JVM) pueda interactuar con programas escritos en otros lenguajes como C, C++ y ensamblador. El problema de esta solución es que se crea una dependencia funcional con el OS dado que hay que compilar el algoritmo LLC en la primera instalación del sistema de comunicaciones y cada vez que se produzca una modificación en el algoritmo LLC, por eso se ha desarrollado esté de la manera más simple posible, delegando toda la complejidad a Java.

FIGURA 3.29: Algoritmo de LLC.

Se ha configurado el “*Raw socket*” en C++ y mediante JNI se ha enlazado con las clases de Java encargadas de gestionar el LLC. Por esto se puede decir que el LLC es un híbrido de Java y C++. En realidad la parte de C++ es la encargada de gestionar toda la comunicación de bajo nivel así como filtrar las tramas, y la parte de Java hace de interfaz para que el algoritmo de GeoNetworking pueda mandar y recibir mensajes.

La Figura 3.28 se puede ver un esquema del algoritmo de LLC la parte Java para comunicarse con GeoNetworking y la parte de C++ que se comunica con la red de comunicaciones.

El *LLC SAP Indication* es la interfaz con la que se comunica el protocolo de red con el algoritmo LLC cuando este quiere difundir un mensaje a través de la VANET y el *LLC SAP Request* es la interfaz con la que se comunica el algoritmo LLC con el protocolo de red cuando llega una trama desde la VANET con destino a ese módulo.

El algoritmo LLC se encarga de crear la trama de GeoNetworking (Figura 3.22), encapsula los datos con la cabecera MAC que se encarga de rellenar, con la MAC de destino proporcionada por la capa de red, la MAC de la tarjeta de comunicaciones 802.11p y con el número de identificación de tipo de trama (0x0707) correspondiente a GeoNetworking cuando hay que enviar una trama y desencapsular, analizar la cabecera MAC y decidir que hace con la trama recibida cuando llega una desde la VANET (Ver figura 3.29), si hay que transmitirla a la capa de red utiliza el *LLC SAP Request* donde le transmite el mensaje y la MAC del módulo emisor de dicho paquete.

3.2.6.2. Algoritmo GeoNetworking

La implementación del algoritmo de GeoNetworking ha sido llevada a cabo en Java siguiendo el estándar ETSI EN 302 636-4 (sección 3.2.1.3), donde se han detectado varios problemas y se han adoptado las soluciones necesarias para el correcto funcionamiento del estándar, estos problemas y soluciones se detallaron en la sección 3.2.4.

FIGURA 3.30: Algoritmo de GeoNetworking.

En la Figura 3.30 se muestra el esquema del algoritmo de GeoNetworking desarrollado. GeoNetworking pertenece al nivel de red del modelo OSI por lo cual se tiene que comunicar con la capa inferior, nivel de enlace de datos (Algoritmo LLC), y con la capa superior, nivel de transporte (Algoritmo BTP), para ello tiene una serie de interfaces (SAP):

- GN SAP *Request*: es utilizada por el algoritmo BTP para comunicarse con el GeoNetworking cuando quiere mandar un mensaje por la VANET.
- GN SAP *Indication*: es utilizada por el algoritmo de GeoNetworking para comunicarse con el BTP cuando se ha recibido un mensaje de la VANET dirigido al módulo, ha sido procesado por el GeoNetworking y este ha decidido que se tiene que comunicar al nivel superior BTP.
- LLC SAP *Request*: es utilizada por el GeoNetworking para enviar un mensaje a la VANET a través del LLC. Bien porque es un mensaje procedente del BTP o porque es un mensaje procedente de la VANET que debe ser retransmitido.
- LLC SAP *Indication*: es utilizado por el algoritmo LLC para comunicar al GeoNetworking de la llegada de un mensaje a través de la VANET.

En el algoritmo de GeoNetworking se pueden diferenciar dos secciones en función de su utilidad, envío y recepción de mensajes:

- Envío: En la figura 3.30 los bloques pertenecientes a la parte de envío de paquetes del algoritmo de GeoNetworking está delimitado por el borde discontinuo y marcada como “I”. Esta sección del algoritmo es la encargada de enviar los mensajes recibidos del algoritmo BTP a través del *GN SAP Request*. GeoNetworking analiza los datos recibidos para comprobar que todos los parámetros son correctos y contesta al BTP a través del *GN SAP Indication* si el paquete ha sido aceptado o no, en caso de ser rechazado devolverá la razón del rechazo. Las razones de rechazo pueden ser por ejemplo: el tamaño del de la carga útil (*payload*) excede el máximo aceptable, el tiempo de vida del paquete es excesivo, el intervalo de repetición es demasiado corto, etc. Si el mensaje ha sido aceptado creará las cabeceras necesarias para su transmisión y se enviará a la capa de nivel de enlace, algoritmo LLC, a través del *LLC SAP Request*.
- Recepción: En la figura 3.30 los bloques pertenecientes a la parte de recepción de paquetes del algoritmo de GeoNetworking está delimitado por el borde punteado y marcada como “II”. Esta sección del algoritmo es la encargada de recibir los paquetes tratados por el nivel de enlace, lo primero que se realiza es un análisis del paquete para determinar las operaciones futuras, comunicar al nivel de transporte,

a través del *GN SAP Indication*, reenviar el mensaje, a través del *LLC SAP Request*, o descartar el paquete. Las operaciones de comunicar y reenviar no son excluyentes.

- Comunicar: Se produce cuando un mensaje recepcionado tiene como destino ese módulo y no es repetido.
- Reenviar: Hay varias posibilidades por las que se puede reenviar un mensaje: El módulo receptor hace de relé, el mensaje es de tipo GBC, GAC o TSB.
- Descartar: El mensaje es repetido, este puede proceder de otro módulo pero el contenido es el mismo.

En el caso de que se quiera conocer con más detalle los pasos seguidos en cada una de las tareas por el algoritmo de GeoNetworking en la figura A.22 se pueden ver.

3.2.6.3. Algoritmo *Basic Transport Protocol*

El algoritmo BTP ha sido implementado usando Java siguiendo el estándar ETSI EN 363-5 (636-5-1, 2014) descrito en la anexo A.2. Dado que el algoritmo BTP está situado en el nivel de transporte se tiene que comunicar con la capa inferior, nivel de red (Algoritmo GeoNetworking), y con la capa superior, nivel de aplicación (Utilidades ITS), por las características del servicio que tiene que desempeñar, múltiples aplicaciones ITS dependen de él. En los anexos del estándar se especifican 5 utilidades o aplicaciones a las que debe de atender y adicionalmente se ha creado una aplicación de soporte para elementos de la red sin capacidad de comunicaciones IEEE 802.11p, cada aplicación tiene asignado un puerto específico por lo cual el BTP tiene que gestionar las conexiones a esos puertos. Para ello tiene una serie de interfaces (SAP):

- *BTP SAP Request*: es utilizada por las utilidades ITS para comunicarse con BTP cuando alguna quiere mandar un mensaje.
- *BTP SAP Indication*: es utilizada por el BTP para comunicarse con las utilidades ITS cuando se ha recibido un mensaje de la VANET dirigido al módulo.
- *GN SAP Request*: es utilizada por BTP para enviar un mensaje a la VANET a través del GeoNetworking.
- *GN SAP Indication*: es utilizado por GeoNetworking para comunicar al BTP de la llegada de un mensaje a través de la VANET.

Para la gestión de los puertos se ha creado un servidor independiente por cada utilidad o aplicación que debe dar soporte, para manejar la interconexión de las aplicaciones y

utilidades ITS. Se ha diseñado de esta manera porque es necesario que BTP pueda atender al mismo tiempo todas las utilidades y aplicaciones, además existe la posibilidad que las aplicaciones y utilidades tengan la posibilidad de comunicarse entre ellas, por lo cual el tener un BTP por cada aplicación o utilidad es inviable. Por esa razón ha habido que desarrollar un algoritmo capaz de manejar los diferentes servidores que dan servicio al nivel de aplicación y comunicarse con GeoNetworking de forma eficiente.

FIGURA 3.31: Algoritmo *Basic Transport Protocol*.

Cuando al BTP le llega los datos de una aplicación o utilidad a través de *BTP SAP Request* identifica el servidor por donde han entrado los datos por el número de puerto, lo cataloga por si fuera necesario utilizar esa información posteriormente, encapsula los datos con la cabecera del BTP (ver figura A.33) proporcionados por la aplicación o utilidad y se lo retransmite a GeoNetworking a través del *GN SAP Request*.

Cuando el GeoNetworking recibe un mensaje y se lo pasa al BTP a través de *GN SAP Indication* identifica a que servidor se lo tiene que entregar por el número de puerto destino que se encuentra en la cabecera del paquete, extrae los datos y se lo transmite a las aplicaciones que se encuentren conectadas en ese momento al servidor correspondiente

identificado por el puerto usando el *BTP SAP Indication*. En la figura 3.31 se puede ver un esquema simplificado del algoritmo.

3.2.6.4. Aplicación de soporte para elementos de la red sin capacidad de comunicaciones IEEE 802.11p (GNES)

Anteriormente se ha comentado que los módulos DSRC-UPM-INSIA poseen adicionalmente una tarjeta de comunicaciones IEEE 802.11ac la cual crea dos redes *Wi-Fi* donde está protegida con seguridad *WPA de 128 bits* y la otra abierta. Esto es para que las aplicaciones o utilidades internas puedan interactuar con el módulo con condiciones de seguridad a través de la red segura específica de cada módulo y dar soporte a otros usuarios de la vía sin capacidad de comunicación mediante tarjetas de comunicaciones que soporten el estándar IEEE 802.11p a través de la red inalámbrica abierta.

Esta red abierta se ha habilitado pensando usar para aumentar la cobertura y seguridad en las “Smart Cities”. Los agentes que no tienen la posibilidad de utilizar una tarjeta de comunicaciones IEEE 802.11p no pueden interactuar con los estándares de geocomunicación, gracias a esta red adicional pueden retransmitir su posición y características a la red de geoposicionamiento, si esto se hace sin control pueden crear un grave problema en la estabilidad de la red, pues un módulo con 802.11p puede intentar usar un dispositivo sin tarjeta de comunicaciones 802.11p como relé de comunicaciones, lo cual sería catastrófico, al no poder actuar como tal pues no tienen la posibilidad tecnológica de operar en esa red.

Para solucionar este problema se ha desarrollado un algoritmo/aplicación para dar soporte y ampliar el estándar de geoposicionamiento, se la ha bautizado como “*GeoNetworking Extension and Support*” (GNES). Esta aplicación se encarga de hacer de puente entre las aplicaciones de seguridad y Geonetworking, así como a dispositivos sin el módulo de geoposicionamiento (DnGN).

Como se ha explicado en la sección 3.2.4.4 es importante que las aplicaciones tengan conocimiento de todos los elementos de su alrededor, este caso también se aplica a los dispositivos externos que se conecten a la aplicación.

3.2.6.4.1. Características GNES

Al tener que dar soporte a los dispositivos sin soporte de GeoNetworking, estos únicamente pueden conocer el estado de los nodos de la red de GeoNetworking pero no conocer la presencia de los elementos sin módulo de GeoNetworking, para solucionar este inconveniente se han creado dos nuevas bases de datos en la aplicación, que mejoran

el conocimiento de los usuarios de la vía mejorando potencialmente las aplicaciones cooperativas.

- *BD_Local*: es una base de datos donde se almacenan las características de los dispositivos conectados directamente al módulo.
- *BD_External*: es una base de datos donde se almacenan las características de los dispositivos conectados a otros módulos de la red de GN.

Estas dos bases de datos se comportan exactamente igual que la LocT de GN (ver Anexo A.1.1.3.1).

FIGURA 3.32: Algoritmo de *GeoNetworking Extension and Support* (GNES).

Para que cada módulo tenga conocimiento de los DnGN que están conectados a otros módulos es necesario que todos los módulos retransmitan periódicamente la *BD_Local*, de

esta manera todos los módulos de la red pueden crear su propia *BD_External*. Esto se realiza mediante un mensaje SHB (*Single Hop Broadcast*, difusión de un salto) con una frecuencia configurable, por defecto a 1Hz, de esta manera al ser el tiempo menor que el de los *Beacons* de GeoNetworking la posición de los módulos se actualiza más frecuentemente.

A GNES se le puede conectar varias aplicaciones simultáneamente esto se debe tratar cuidadosamente pues las diferentes peticiones y mensajes no pueden ser entregados a otra aplicación pues entonces se producirán errores en las peticiones. Una posible solución es que todas las peticiones y mensajes que lleguen a GNES se entreguen a todas las aplicaciones conectadas, es lo más sencillo pero puede producir problemas en las aplicaciones al analizar los datos y lo que es más importante producir problemas de seguridad, por lo cual se ha implementado un sistema de identificadores de paquetes y de aplicaciones para minimizar los problemas y que solo se entreguen a las aplicaciones los paquetes solicitados. En la figura 3.32 se puede ver un esquema del algoritmo desarrollado.

Este algoritmo como se puede ver en la figura 3.27 pertenece al nivel de transporte y hace de intermediario entre las aplicaciones y el BTP, por lo cual tiene una serie de interfaces (SAP) para interactuar con estos.

- *GNES SAP Request*: es utilizada por las utilidades ITS para comunicarse con GNES cuando alguna quiere mandar un mensaje o pedir información del estado de la red.
- *GNES SAP Indication*: Es utilizada por el GNES para comunicarse con las utilidades ITS cuando se ha recibido un mensaje de la VANET dirigido al módulo o para contestar a una petición de información recibida.
- *BTP SAP Request*: Es utilizada por GNES para enviar un mensaje a la VANET a través del BTP.
- *BTP SAP Indication*: Es utilizado por el BTP para comunicar al GNES de la llegada de un mensaje a través de la VANET.

En el algoritmo existen varias partes diferenciadas:

- Se reciben datos de una aplicación por el *GNES SAP Request*: Se analiza el tipo de petición, en función del tipo se actúa.
  - Consulta de una base de datos: Se almacena un identificador del paquete para poder contestar a la aplicación que hizo la petición y no a otra. Se consulta la base de datos que se requiera *BD_Local* y/o *BD_External* se extraen los datos y se crea un mensaje de contestación a la aplicación correspondiente y se envía por el *GNES SAP Indication*.

- Enviar los datos por la VANET: Se encapsulan los datos con las cabeceras del GNES y se le transmiten al BTP por el *BTP SAP Request*.

Si el peticionario es un DnGN entre los datos de la petición van sus datos de identificación por lo cual se añaden a la *BD_local* o se actualizan si ya se encontraba en ella.

- Envío periódico de la *BD_Local*: Como se ha comentado cada cierto tiempo los datos contenidos en la *BD_Local* son encapsulados y enviados al BTP a través del *BTP SAP Request* para que los envíe en un SHB a la VANET.
- Se recibe un mensaje a través del *BTP SAP Indication*: Se analiza el tipo de paquete que retransmite el BTP:
  - El mensaje contiene la *BD_Local* de otro nodo: Se desempaqueta y se incluyen o actualiza la información de los DnGN en la *BD_External*.
  - El mensaje no contiene la *BD_Local* de otro nodo: Se extraen los datos de la carga útil del mensaje, analiza la cabecera de GNES y se transmite a la aplicación que corresponda a través del *GNES SAP Indication*.

3.2.6.4.2. Cabeceras GNES

Dado que GNES da nuevas funcionalidades al estándar debe tener cabeceras adicionales se han reducido al máximo posible para dejar más espacio a la carga útil de los datos o *payload*. En la Figura 3.33 se puede observar los campos de la cabecera de GNES.

FIGURA 3.33: Cabecera GNES

Donde los campos de la Figura 3.33 son:

- *GN Type*: éste campo indica el tipo de paquete de GN.
- *GNES Type*: éste campo indica el tipo de paquete de GNES se pueden observar sus posibles valores en la Tabla 3.11.
- *Sequence number*: número de identificación del paquete.
- *Data*: Datos.

Se puede observar que esta cabecera contiene el campo “*GN Type*” perteneciente a GN, esto es debido a que cuando BTP devuelve un “*Indication*” (ver Anexo A.2.1.2) el campo “*GN Type*” no viene referenciado por lo cual los datos de la carga útil pueden ser interpretados de diferentes formas dependiendo del “*GN Type*” y del “*GNES Type*”.

TABLA 3.11: Valores de GNES type

<i>GNES Type</i>	Codificación	Descripción
No GNES	0x00	Sin uso de las funciones de GNES
Save option	0x01	Empezar o parar de salvar los datos en ficheros. Hay que incluir un byte adicional en el <i>Payload</i> para indicar si se quieren guardar los datos o cerrar los ficheros (Grabar = 0x00, otro valor dejar de grabar)
<i>Time SHB</i>	0x02	Pone un nuevo valor al temporizador del envío de la <i>Local_BD</i> de GNES mediante un SHB. Hay que incluir dos bytes adicionales para el nuevo tiempo en ms.
<i>SHB with BD_Local</i>	0x03	Indica que en el <i>Payload</i> está contenida toda la base de datos <i>DB_Local</i> de GNES.
<i>BD_Local</i>	0x04	Devuelve la <i>BD_Local</i> en el <i>Payload</i> , el mensaje no se retransmite por GN.
<i>BD_External</i>	0x05	Devuelve la <i>BD_External</i> en el <i>Payload</i> , el mensaje no se retransmite por GN.
<i>iBeacon Req</i>	0x0B	<i>Request</i> enviado con la información correspondiente a un iBeacon.
<i>iPedestrian Request</i>	0x0C	<i>Request</i> enviado con la información de un peatón.
<i>String</i>	0x99	Función para enviar y recibir <i>Strings</i> en el <i>Payload</i> .

En la Tabla 3.11 existen valores que tendrían repercusión en el comportamiento de GNES estas funciones se han implementado principalmente para el *debug* de la aplicación, en la versión final de GNES probablemente no existirán tales valores.

3.2.6.4.3. Protocolo de acceso al servicio GNES (GNES-SAP)

Como todos los protocolos, GNES posee una serie de servicios para dar soporte a las aplicaciones que lo utilizan, cuando GNES transmite información a las aplicaciones (*indication*) y para cuando GNES recibe información de las aplicaciones (*request*).

3.2.6.4.3.1. GNES-SAP request

Para enviar un mensaje a través de GNES es necesario hacer un “Request” éste tiene los mismos campos que en el caso del “*BTP request*” y en el *Payload* hay que incluir la cabecera de GNES junto con los datos, en el Anexo A.2.1.1 se pueden observar los campos del “*BTP request*”.

3.2.6.4.3.2. GNES-SAP indication

Como en los casos del GN y BTP cuando se le hace una petición o llega un mensaje destinado a tal aplicación se genera un “Indication”, éste tiene los mismos campos que en el caso del “BTP indication” y en la carga útil esta la cabecera de GNES junto con los datos, en el Anexo A.2.1.2 se pueden observar los campos del “BTP request”.

3.2.6.4.4. Elementos de las BDs

Se ha explicado en los apartados anteriores que existen dos bases de datos de las aplicaciones que se conectan a GNES, si estas quieren estar en las bases de datos tienen que mandar periódicamente, igual que los módulos, su posición. En la Figura 3.34 podemos observar los datos que deben mandar.

FIGURA 3.34: Elemento con GPS

Donde los campos de la Figura 3.34 son:

- ID: Es un identificador del tipo de datos que están contenidos en el la sección Datos. En función del tipo la longitud varía:
  - 0x00: en el campo de datos hay la información de un DnGN que posee un método de geoposicionamiento, estos datos están codificados en forma de LPV (*Long Position Vector* (Figura A.3)).
  - 0x01: en el campo de datos hay la información de un DnGN, detectado mediante un *iBeacon*, los datos ocupan 24 bytes se puede ver su composición en la figura 3.35.
- Datos: es la información de posicionamiento del DnGN.

Las características de los *iBeacons* no permiten conocer la posición exacta por lo cual se ha creado un tipo específico de elemento que se puede observar en la Figura 3.35.

Donde los campos de la Figura 3.34 son:

- GN_ADDR: es la *GeoNetworking Address*.

FIGURA 3.35: Elemento *iBeacon*

- TST: tiempo de recepción de la información.
- Lat: latitud medida en $\frac{1}{10} \cdot \mu^\circ$ en el momento de la recepción de la información.
- Lon: longitud medida en $\frac{1}{10} \cdot \mu^\circ$ en el momento de la recepción de la información.
- TX Power: potencia de emisión de la señal del *iBeacon*.
- RSSI: “Received Signal Strength Indicator”, potencia de recepción de la señal del *iBeacon*.

3.2.7. Resultados

Una vez desarrollados los módulos DSRC-UPM-INSIA y los algoritmos se ha procedido a testear el rendimiento de estos. Estos test han servido para probar, mejorar y optimizar las VANETs. Para ello se ha comprobado el funcionamiento de los algoritmos de los módulos DSRC-UPM-INSIA así como la latencia del sistema.

3.2.7.1. Comprobación del algoritmo de GeoBroadcast

La complejidad del algoritmo de Geobroadcast hace que su testeo no sea sencillo, dado que existe la posibilidad de que éste llame al algoritmo de GeoUnicast. Por ello se han diseñado diferentes pruebas para ir probando todos los aspectos de éste. Una prueba estática, una prueba dinámica y una prueba donde interviene el algoritmo de GeoUnicast, de esta forma se hace una doble comprobación.

3.2.7.1.1. Prueba estática

Para la realización de esta prueba se situó un vehículo en el centro de la pista de pruebas del INSIA que actuó como vehículo emisor, se marcaron las áreas objetivo donde difundir un mensaje de tipo GeoBroadcast y se procedió con otro vehículo a hacer varias pasadas por el interior de las áreas de difusión para comprobar el correcto funcionamiento del desarrollo

del algoritmo y de la limitación de las áreas de difusión. En las Figuras 3.36-3.38 se pueden observar los resultados de estas pruebas.

FIGURA 3.36: Resultados prueba estática del círculo

FIGURA 3.37: Resultados prueba estática del rectángulo

El vehículo emisor emitía mensajes de GeoBroadcast a una frecuencia de 1 Hz y el vehículo receptor realizó pasadas a su alrededor a diferentes velocidades y distancias para comprobar el comportamiento de los algoritmos.

FIGURA 3.38: Resultados prueba estática del elipse

Observando la figura puede parecer que hay mensajes recibidos que están fuera del área y secciones que debería haber alguna recepción, esto es debido a las variaciones de posición producidas por los GPS, por las que el área cambia de localización.

Una vez comprobado el correcto funcionamiento de las áreas de difusión y del algoritmo de GeoBroadcast se procedió a hacer una prueba dinámica.

3.2.7.1.2. Dinámica

Para la prueba dinámica se utilizaron dos vehículos (un vehículo emisor y otro receptor) equipados con módulos DSRC-UPM-INSIA, esta consistió en un recorrido entre la localidad de Rivas Vaciamadrid y el campus Sur de la Universidad Politécnica de Madrid de una longitud aproximada de 10 kilómetros, por la autopista A3 con tráfico real respetando la señalización y comportándose como un vehículo más, durante varias fases del experimento se interpusieron vehículos entre medias de los dos vehículos de las pruebas y se comprobó que la comunicación era correcta y que no dificultaban las comunicaciones. El vehículo emisor fue emitiendo un mensaje de tipo GeoBroadcast cada segundo con área de difusión de un círculo de 100 metros de radio centrado en la posición actual del vehículo emisor.

Las gráficas 3.39 y 3.40 han sido creadas con la información registrada por el vehículo receptor, de esta forma se puede apreciar el funcionamiento del sistema desde el punto de vista de un vehículo. En la figura 3.39 se puede ver el recorrido de los dos vehículos realizado, para que se pueda apreciar con más claridad el trayecto de ambos vehículos el recorrido

FIGURA 3.39: Trayecto de los vehículos en la prueba dinámica

del vehículo emisor se ha desplazado 50 metros hacia el norte y así evitar que ambos se superpongan, se puede observar que hay una sección en donde no se refleja la trayectoria del vehículo emisor, esto es debido a que la distancia entre ambos vehículos era excesiva y no se estableció comunicación entre ambos nodos durante ese periodo.

La Figura 3.40(a) representa la velocidad a la que iban los vehículos en cada instante, aquí se aprecia mejor el tramo sin información del vehículo emisor. En la Figura 3.40(b) se representan las distancias existentes entre los vehículos y la distancia entre el centro del área de difusión y el vehículo receptor, gracias a esta figura se aprecia que la pérdida de información es por la excesiva distancia entre vehículos. La diferencia entre la distancia al centro del área y la distancia entre vehículos se debe a la forma que se ha tenido de tomar los datos, para tomar los datos de los vehículos la aplicación desarrollada para tal fin, hacia una consulta cada segundo al módulo requiriéndole los datos de la base de datos, mientras que los datos del área eran transmitidos a la aplicación por el módulo según eran recepcionados. Este comportamiento produce que los datos entre ambos estén desfasados en el tiempo y debido a las altas velocidades de circulación existe este desfase.

Se puede observar claramente que el sistema funciona correctamente dado que no hay ninguna recepción de mensajes a mayor distancia de 100 metros. En la Tabla 3.12 se pueden observar los resultados del experimento donde hay una tasa de recepción del 94 %. Durante el experimento han existido 9 transiciones entre más de 100 metros y menos. Estas transiciones pueden influir en que los paquetes teóricos enviados sean más dado que la frecuencia de envío y de actualización de los datos es de 1 Hz se puede haber descartado algún paquete por estar fuera de los 100 metros y posteriormente al analizar los datos del posicionamiento

(a) Velocidades

(b) Distancias

FIGURA 3.40: Resultados prueba dinámica

de los vehículos estar ambos situados a una distancia menor de 100 metros, pudiéndose dar como paquete perdido cuando en realidad no ha sido así. Por ello se puede decir que existe una tasa de recepción de entre el 94 % y el 97 %.

TABLA 3.12: Resultados experimento GeoBroadcast dinámico

	Tiempo (segundos)	Número de paquetes enviados	Número de paquetes recibidos	% de paquetes recepcionados
Tiempo del experimento	500.8	500	-	-
Tiempo a menos de 100 metros	361.962	362	341	94.2

3.2.7.2. GeoUnicast, GeoBroadcast, rendimiento y latencia

Dado que el algoritmo de GeoBroadcast puede invocar al algoritmo de GeoUnicast se ha utilizado para probar el correcto funcionamiento de éste y ya de paso verificar el correcto funcionamiento de todas las funciones del algoritmo de GeoBroadcast. Para comprobar el algoritmo de GeoUnicast (*Greedy*) se ha realizado un experimento en el cual se han

distribuido diferentes módulos como se puede observar en la figura 3.41 la potencia de emisión se ha reducido para que el rango efectivo no supere los 200 metros de esta manera se asegura que el “nodo 0” no tenga comunicación directa con el “nodo 2” ni el “nodo 3”, el “nodo 1” con el “nodo 3”, el “nodo 2” con el “nodo 0” y el “nodo 3” con los “nodo 0” y “nodo 1”.

(a) Distribución de los nodos

FIGURA 3.41: Distribución de los módulos DSRC-UPM-INSIA para las pruebas

Desde el nodo 0 se han emitido mensajes que utilizan el algoritmo de GeoBroadcast (GBC y GAC) para el testeo de la red y la comprobación del correcto funcionamiento de los algoritmos de GeoBroadcast y GeoUnicast.

Para poder analizar el correcto funcionamiento de los módulos se han puesto éstos en modo “*debug*” en el cual se guardan en archivos de texto la información de cada mensaje que es enviado y recibido por el módulo para de esta forma poder analizar los datos exhaustivamente.

Una vez comprobado que los diferentes tipos de paquetes de datos tienen un comportamiento correcto se ha procedido a analizar la latencia de la red con todos los sistemas en funcionamiento. Para ello se han estado emitiendo distintos mensajes del tipo GBC/GAC (figuras 3.42(a), 3.42(c) y 3.42(b)) desde el “nodo 0” con destino el “nodo 3” con el fin de ver la latencia de la red y los tiempos de retardo que se producen con los diferentes saltos o “*hops*”.

Se han mantenido los módulos en funcionamiento durante aproximadamente unos 50 minutos donde se han enviado cerca de 3000 paquetes de datos del “módulo 0” al “módulo 3”, al mismo tiempo cada módulo estaba emitiendo cada segundo un paquete del tipo SHB

FIGURA 3.42: Distribución de los módulos DSRC-UPM-INSIA para las pruebas del algoritmo GeoBroadcast y GeoUnicast

con la información contenida en las bases de datos de GNES, por tanto se puede considerar que la red estaba funcionando en condiciones reales de funcionamiento.

En la tabla 3.13 se pueden observar los paquetes que se han enviado desde el “nodo 0” y cuantos paquetes han recibido el resto de nodos así como los paquetes perdidos en cada interacción. Se puede observar que las tasas de pérdida de paquetes son muy reducidas. Estos datos difieren de los datos presentados en la Tabla 3.12 y puede llevar al equívoco de que son erróneos o no reales, esto es debido a la forma de realizar la prueba, al haber un mayor número de nodos y estar estos en una posición fija, sin interferencias de vehículos y con unas condiciones radioeléctricas constantes la red es capaz de comportarse de forma más robusta y eficiente que respecto a la prueba dinámica donde todas las variables son cambiantes.

TABLA 3.13: Mensajes GBC y GAC enviados y recibidos por los nodos

	Nodo 0	Nodo 1	Nodo 2	Nodo 3
Mensajes enviados/tratados	2957	2955	2954	2947
mensajes perdidos	0	2 (0.07%)	3 (0.1%)	10 (0.34%)

TABLA 3.14: Resultados de latencia de los módulos DSRC-UPM-INSIA

	Hop 0 (ms)	Hop 1 (ms)	Hop 2 (ms)
Media	1.419	8.802	15.546
Desviación típica	2.318	3.892	5.090
Varianza	5.375	15.144	25.905

Para calcular la latencia de la red se han tomado los diferentes tiempos que se han empleado para cada paso, en la Figura 3.43 se pueden observar en forma de esquema los diferentes tiempos donde $T_P X$ es el tiempo empleado en el procesamiento del paquete recibido por el módulo X y $T_T X$ es el tiempo empleado para la transmisión del paquete de un módulo a otro.

FIGURA 3.43: Esquema de tiempos de retardo

Para calcular el tiempo de los saltos o “hops” de la red se ha utilizado la ecuación 3.12.

$$hop_n = hop_{n-1} + T_T n + T_P n \quad (3.12)$$

Dónde:

- hop_n es el tiempo empleado en realizar n saltos.
- hop_{n-1} es el tiempo empleado en realizar $n - 1$ saltos.
- $T_T n$ es el tiempo usado en el tránsito desde el nodo $n - 1$ al nodo n .
- $T_P n$ es el tiempo usado por el algoritmo de GeoNetworking en procesar y resolver el mensaje.

Una vez analizados los datos se han obtenido los resultados mostrados en la Tabla 3.14 y la Imagen 3.44. En estos resultados hay que tener en cuenta que la resolución de los experimentos es de $\pm 1ms$ y en ocasiones no es suficiente para medir determinadas partes de los algoritmos de GeoNetworking, pues estas consiguen procesar los paquetes y decidir que hacer con ellos ($T_P X$) en menos de $1ms$.

El experimento se ha realizado por razones de eficiencia en el software con un retraso medio de $2,5ms$ desde que se recibe el mensaje hasta que se procesa, de esta manera se evitan problemas de inanición en el resto de threads, este valor es fácilmente configurable en caso de que se necesite.

FIGURA 3.44: Resultados de latencia

Con los resultados de estas pruebas se demuestra que el sistema es capaz de operar en condiciones normales de funcionamiento y con unas latencias suficientemente bajas como para servir de medio de transmisión para sistemas ADAS centrados en la seguridad vial, que son los más críticos con la latencia de los mensajes. Los últimos estudios indican que una latencia adecuada para sistemas cooperativos de seguridad, los más exigentes, deben rondar los $10 - 20ms$ por salto en la red (Biswas et al., 2006). Nuestro sistema tiene una latencia menor a $5ms$ en un salto y en el peor de los casos en solo $20ms$ un mensaje ha recorrido una distancia de “3 hops” (en función de la distribución de los vehículos y configuración de la red puede ser una distancia entre 450-900 metros), si suponemos dos vehículos aproximándose en dirección contraria a $150km/h$ cada uno, la distancia entre ambos se vería reducida aproximadamente a una velocidad de $84m/s$, dada la latencia del sistema esta distancia se habría reducido solamente 1,7 metros desde que un vehículo ha emitido un mensaje y el otro lo ha recibido, un 0,4% de la distancia en el peor de los casos con 450 metros de separación y $20ms$ de latencia. Si se eliminaran los tiempos de espera desde la recepción a cuando se procesan los mensajes la latencia de un salto se vería reducida a $< 1ms$.

3.2.7.3. Comprobación de funcionamiento de mensajes

Una vez comprobado el correcto funcionamiento de los algoritmos de difusión de mensajes se ha procedido a comprobar el resto de mensajes descritos en los estándares

Para ello se han distribuido los módulos de la misma manera que en el experimento anterior y desde el nodo 0 se han emitido los diferentes mensajes posibles contenidos en el estándares de GeoNetworking e implementados (GUC (figura 3.45(c)), GBC/GAC (figuras 3.42(a), 3.42(c) y 3.42(b)) y TSB (figura 3.45(b))) para el testeo de la red y la comprobación

del correcto funcionamiento de los diferentes algoritmos, para probar el caso de TSB (figura 3.45(b)) se han añadido nodos adicionales para poder comprobar el correcto funcionamiento de este tipo de paquetes.

FIGURA 3.45: Distribución de los módulos DSRC-UPM-INSIA para las pruebas

Se ha comprobado que todos los mensajes eran recibidos correctamente por el nodo o nodos objetivo en función del tipo de mensaje por lo cual se ha dado por terminado el desarrollo de los módulos DSRC-UPM-INSIA, los resultados se pueden observar en la Tabla 3.15.

TABLA 3.15: Paquetes SHB, TSB y GBC enviados y recibidos

	Enviados	Recibidos	Perdidos
SHB	580	580	0 (0 %)
TSB	612	610	2 (0.33 %)
GUC	658	654	4 (0.61 %)

Capítulo 4

APLICACIONES DE SISTEMAS COOPERATIVOS

Una vez comprobado que los algoritmos desarrollados e implementados en los diferentes dispositivos funcionan correctamente y con unas latencias adecuadas se ha procedido a desarrollar sistemas cooperativos ADAS que ayuden a aumentar la seguridad y la capacidad de las infraestructuras. Según (Dobbs, 2013) el aumento de las infraestructuras físicas para aumentar la capacidad es más costoso y lleva más tiempo de ejecutar y obtiene peores resultados que utilizar tecnología ITS para aumentar la capacidad de las carreteras.

4.1. Motivación de las aplicaciones desarrolladas

Según los datos estadísticos de la DGT del año 2013 (publicación más reciente a fecha de la escritura de esta tesis) hubo 126,400 víctimas en accidentes de tráfico las cuales 57,732 se produjeron en vías interurbanas y 68,668 en vías urbanas, de los cuales 12,344 fueron peatones casi un 10 % del total de víctimas producidas. En total hubo 1,680 muertos ese año donde casi un 50 % de las víctimas mortales fueron los denominados usuarios vulnerables, 802 víctimas mortales entre peatones (378), ciclistas (69) y motoristas (355). En la figura 4.1 se puede ver un resumen de las víctimas producidas durante el 2013 a 30 días del accidente.

A lo largo del tiempo la solución para intentar reducir las víctimas se ha basado en métodos pasivos y activos de seguridad. Los métodos pasivos más representativos son: cinturones de seguridad, *airbags*, mejoras en el chasis, carrocería y cristales... Entre métodos activos destacan: ABS (*antilock brake system*), ESP (*electronic stability program*), TCS (*traction control system*) o ASR (*anti-skid regulation*). Estos están instalados en el vehículo y se encargan de la seguridad de los ocupantes. En los últimos tiempos se han empezado instalar en los vehículos sensores capaces de detectar las condiciones que rodean al vehículo. Gracias a ellos se ha empezado a implementarse sistemas de protección a usuarios de la vía. Algunos de los sistemas son de detección de cambio de carril, detección de peatones, control de somnolencia, etc. El problema es que esta información adicional se queda contenida en el vehículo que posee esos sensores.

Durante el doctorado se han desarrollado varias aplicaciones basadas en sistemas cooperativos utilizando VANETs. Estos sistemas cooperativos intentan reducir las víctimas en los usuarios vulnerables al máximo en los casos de accidentes donde hay involucrados varios vehículos y, si es inevitable, que se reduzca lo máximo posible la gravedad de las lesiones a los implicados, mediante el intercambio de información de todos los agentes implicados.

Por tal razón se han desarrollado cuatro aplicaciones, que se detallan en los siguientes apartados, sobre los módulos WSA y DSRC-UPM-INSIA, para cada tipo de usuario vulnerable de la vía:

- Peditores: Sistema distribuido de detección de obstáculos en entornos urbanos
- Peditores: Aviso a peditores en vías urbanas e interurbanas.
- Bicicletas: *BikeWarn*.
- Motocicletas: *MotoWarn*.

FIGURA 4.1: Resumen del anuario de accidentes 2013 de la DGT

4.2. Sistema distribuido de detección de obstáculos en entornos urbanos

Viendo la efectividad de las redes de comunicación basadas en módulos WSN con el algoritmo GCTP, explicado en la sección 3.1.3.2, se ha decidido dar un paso más y crear un sistema ADAS en colaboración con el Laboratorio de Sistemas Inteligentes, LSI, de la universidad Carlos III de Madrid.

Gracias a que ambos grupos trabajamos en áreas distintas de conocimiento se ha desarrollado un sistema distribuido de detección de obstáculos en entornos urbanos, pensado principalmente para solucionar los accidentes por atropello producidos en los pasos de peatones de varios carriles, donde un vehículo de grandes dimensiones, autobús o camión, bloquea la visión de los vehículos que circulan por los carriles adyacentes, ver Figura 4.2(a).

El objetivo de este sistema es que evite estas situaciones y que sea asequible dado que los sistemas de detección de peatones por LIDAR (*Laser Imaging Detection and Ranging*) y cámaras de infrarrojos o estereoscópicas son caros y el montaje en vehículos personales encarecerían éste hasta precios que estarían fuera de mercado, en cambio, la inclusión de estos sistemas en autobuses o camiones no supondría un sobre coste en el vehículo sustancial.

FIGURA 4.2: Sistema distribuido de detección de obstáculos

Al colocar el sistema de detección de obstáculos en los vehículos de grandes dimensiones que circulan por ciudad, es necesario un sistema que difunda la información de los obstáculos a los vehículos colindantes que no poseen este sistema. Esto se soluciona con los

módulos desarrollados, de esta manera todo vehículo tiene información de los obstáculos en la carretera. En la Figura 4.2(b) se puede observar como el sistema interactuaría y se evitaría el accidente.

4.2.1. Detección de obstáculos

Este sistema lo ha desarrollado fundamentalmente el LSI de la UC3M dado que tienen más experiencia en este campo. Para que el sistema funcione correctamente hay que identificar correctamente a los peatones. Con tal fin se utilizan dos sensores, un sensor láser LIDAR y una cámara estereoscópica.

4.2.1.1. Detección mediante sensor láser

El sensor láser LIDAR devuelve un *array* de 401 puntos cada punto está separado 0.25 grados y posee la distancia a el objeto más próximo en esa dirección. Según las características de los puntos contenidos en los datos, se juntan los puntos cercanos formando polilíneas, de esta manera se pueden clasificar los diferentes objetos en función de la forma de estas polilíneas: grandes obstáculos, límites de carretera, vehículos (en forma de L), pequeños obstáculos y peatones (F. García; F. Jiménez; J.E. Naranjo; J.G. Zato; F. Aparicio; J.M. Armingol; de la Escalera, 2012).

Para determinar si un objeto detectado es un peatón se seleccionan los obstáculos con un tamaño cercano a un peatón, dado que basándose en estudios antropométricos, la forma de la polilínea de un peatón tiene un patrón típico. Por último, se incluye una etapa de seguimiento (Garcia et al., 2011). En la Figura 4.3 se puede ver el patrón de las polilíneas para la detección de peatones.

FIGURA 4.3: Patrones de peatones

4.2.1.2. Detección mediante visión

Con la detecciones del escáner láser se marcan áreas de interés donde la cámara busca patrones de peatones para descartar falsos positivos o confirmar los peatones detectados. Para tal fin el sistema de visión se apoya en los descriptores de Histograma de Gradientes Orientados (HGO) y en Maquinas de Soporte Vectorial (Dalal and Triggs, 2005). Ver Figura 4.4. Finalmente se confirma o desmiente la presencia de un peatón con la comparación de estas áreas objetivo con patrones almacenados en una base de datos hecha con anterioridad.

FIGURA 4.4: Detección de escáner láser y HGO

4.2.2. Alerta de obstáculos

Una vez que el vehículo que incorpora los sensores de detección de objetos tiene los peatones identificados y los está monitorizando, tiene que retransmitir los datos al resto de vehículos. Para este fin se utiliza el sistema de comunicaciones diseñado con módulos WSAW y el algoritmo GCTP para dotar a los vehículos de comunicaciones V2V.

Se pueden hacer dos diferenciaciones: el vehículo que envía la información y el que la recibe.

4.2.2.1. Vehículo emisor

Una vez que se detectan los obstáculos se tiene que asignar unas coordenadas de localización a éstos y ser retransmitidos por la red de comunicaciones. Para ello se utiliza el algoritmo GCTP (ver 3.1.3.2). La información de los objetos se transmite en el *payload* o carga útil de los paquetes del GCTP. Esta carga útil se transmite en los paquetes de datos que se ha descrito en detalle en la sección 3.1.3.2.5. En la Figura 4.5 se observa la estructura

de datos de cada objeto detectado por el sistema y que será transmitida al resto de vehículos de la red. Todos los datos de los objetos detectados en los alrededores del vehículo emisor se concatenaran para crear un paquete de datos a retransmitir en el *payload* del paquete de datos.

	0							1							2							3										
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	ID tipo								Número identificador																UTM Norte							
4	UTM Norte																UTM Este															
8	UTM Este																Velocidad Norte															
12	Velocidad Norte								Velocidad Este																TST							
16	TST																Control de Datos															

FIGURA 4.5: Campos identificativos de cada objeto contenido en el *payload* del paquete de datos

Donde los campos de la Figura 4.5 son:

- El campo “ID Tipo” identifica el tipo de obstáculo que es detectado por el sistema.
  - 1 si son los datos del vehículo emisor.
  - 2 si son los datos de un vehículo detectado.
  - 3 si son los datos de un peatón detectado.
  - 4 si son los datos de otro objeto detectado (objeto no identificado).
- El “Número Identificador” representa el identificador único asociado con cada objeto del sistema, nodo de la red u obstáculos.
- “UTM Norte” y “UTM Este” representan la posición GPS en coordenadas UTM para el objeto dado.
- “Velocidad Norte” y “Velocidad Este” representan el vector de velocidad del objeto detectado.
- El “TST” es un identificador temporal que representa el momento en que se generó el mensaje.
- “Control de datos” es un CRC para garantizar la correcta transmisión de los datos.

La obtención de estos datos no es trivial dado que los obstáculos detectados se referencian sobre el vehículo emisor como se ve en la Figura 4.6 (a), por lo cual hay que transformar los datos de posición y velocidad del objeto a coordenadas UTM para que puedan ser fácilmente entendidas por los vehículos receptores.

Para hacer la transformación de coordenadas de los objetos detectados a UTM es necesario conocer todos los datos del vehículo emisor, dado que todos los datos del objeto

FIGURA 4.6: Transformación de coordenadas

están referenciados al vehículo (Ver Figura 4.6 (b)). Para este fin necesitamos conocer posiciones UTM norte y este, velocidad norte y este y ángulo azimutal del vehículo que ha detectado los objetos.

Esta transformación de coordenadas implica una traslación y una rotación. En la ecuación 4.1 se puede observar la nueva posición del objeto en coordenadas UTM.

$$\begin{aligned} N_{Objeto} &= N_{Vehiculo} - (X_{Objeto} \cdot \cos(\alpha_V) - Y_{Objeto} \cdot \sin(\alpha_V)) \\ E_{Objeto} &= E_{Vehiculo} + (Y_{Objeto} \cdot \cos(\alpha_V) + X_{Objeto} \cdot \sin(\alpha_V)) \end{aligned} \quad (4.1)$$

Y mediante la ecuación 4.2 se realiza la transformación de las velocidades en el sistema de coordenadas "x" "y" del vehículo emisor al sistema de coordenadas UTM.

$$\begin{aligned} V_{Norte} &= V_{NVehiculo} - (V_{xObjeto} \cdot \cos(\alpha_V) - V_{yObjeto} \cdot \sin(\alpha_V)) \\ V_{Este} &= V_{EVehiculo} + (V_{xObjeto} \cdot \sin(\alpha_V) + V_{yObjeto} \cdot \cos(\alpha_V)) \end{aligned} \quad (4.2)$$

De esta manera ya están todos los objetos en unas coordenadas globales adecuadas a todos los vehículos receptores.

4.2.3. Operación de Transmisión de Datos

Es importante remarcar que para permitir la detección en tiempo real, es obligatorio la actualización periódica de las detecciones. También es importante reducir la carga de la red. Para permitir un compromiso entre ambas restricciones, el período de actualización de la detección se definió como un parámetro, lo que permite que el sistema sea adaptado para reducir la carga de trabajo sin perder rendimiento en tiempo real.

4.2.4. Vehículo receptor

Como se ha explicado anteriormente los nodos de la red mantienen el enlace de uno a otro a una distancia máxima de 90 metros. La transmisión de datos de objetos se inicia en el momento en que el sistema de detección detecta un objeto en la ruta del vehículo. En este momento, el sistema comienza a transmitir la advertencia de los objetos a través de las comunicaciones V2V. Los vehículos receptores recuperan las tramas de datos y la información, dependiendo de si están o no están en el área de influencia del vehículo receptor, y se decide si se debe advertir al conductor.

Para el vehículo receptor se ha desarrollado una aplicación que se encarga de mostrar al conductor mediante una interfaz hombre maquina (HMI) los datos recogidos de los vehículos emisores. En la Figura 4.7 se puede observar un esquema de dicho programa que se ejecuta en el ordenador de a bordo del vehículo receptor.

FIGURA 4.7: Primera reconfiguración de la VANET

4.2.4.1. Mantenimiento GCTP

La función de este módulo es proporcionar los datos necesarios al sistema de comunicaciones para que mantenga la conectividad de los diferentes módulos de la red operativa. Para tal fin debe recoger los datos del GPS empaquetarlos y transmitírselos al módulo WSA.

4.2.4.2. Receptor de datos del sistema de comunicación

Este módulo tiene dos funciones principales. La primera es extraer los datos del paquete de datos proporcionado por el sistema de comunicación clasificarlos e introducirlos en la base de datos, añadiéndolos o actualizándolos si el objeto ya existía previamente. También debe recoger los datos proporcionados por el GPS, analizarlos, extraer la información deseada, calcular todos los parámetros nuevos acerca del vehículo receptor y guardarlos en la base de datos.

4.2.4.3. Análisis de datos

Este módulo se activa cada vez que hay una modificación en la base de datos y, dependiendo del tipo de dato, actúa de una forma u otra.

Si los datos que se modifican son de un objeto, calcula las posiciones relativas de los objetos con la información de la base de datos para dibujar en el HMI donde se encuentran respecto del vehículo receptor así como una flecha que indica la dirección en que se desplaza respecto del vehículo, la longitud de la flecha es proporcional a la velocidad del objeto.

Si el dato modificado es el del vehículo receptor recalcula todas las posiciones relativas de los objetos de la base de datos. Al mismo tiempo que se realiza esta operación verifica el tiempo de captura del objeto y el tiempo actual. Si la diferencia supera cierto umbral el objeto es eliminado de la base de datos pues se considera que el dato ya no tiene validez.

Hay que tener en cuenta que el sistema receptor solo envía un punto por cada objeto detectado. Esta información es el punto más cercano al vehículo emisor en el eje “y” y centrado en las dimensiones del objeto en el eje “x”. Por lo cual hay que calcular los puntos adicionales si fuera necesario para representar el objeto en la interfaz gráfica, conocido este punto y la dirección que tiene este objeto. Todos estos datos calculados son guardados en una base de datos gráfica que es lo que utiliza la HMI cada vez que refresca la interfaz gráfica.

Para calcular estos puntos el módulo tiene que hacer una traslación y una rotación de todos los puntos pues la HMI muestra los objetos respecto al sistema de coordenadas del

vehículo receptor. La operación es a la inversa de cómo se explicó anteriormente. Los pasos a seguir por cada punto a dibujar son:

Traslación del origen de coordenadas UTM (Norte, Este) al sistema de referencia del Receptor (Norte, Este) mediante la Ecuación 4.3.

$$\begin{aligned} P_{Norte} &= N_{Objeto} - N_{receptor} \\ P_{Este} &= E_{Objeto} - E_{receptor} \end{aligned} \quad (4.3)$$

Rotación del sistema de coordenadas del receptor (Norte, Este) al sistema de coordenadas del receptor (X, Y) usando la Ecuación 4.4.

$$\begin{aligned} P_{tx} &= P_{Norte} \cdot \cos(\alpha_{receptor}) - P_{este} \cdot \sin(\alpha_{receptor}) \\ P_{ty} &= P_{Norte} \cdot \sin(\alpha_{receptor}) + P_{este} \cdot \cos(\alpha_{receptor}) \end{aligned} \quad (4.4)$$

Traslación y escalado de las coordenadas del mapa del HMI, aplicando la Ecuación 4.5.

$$\begin{aligned} P_x &= X_{CentroMapa} - P_{tx} \cdot \frac{\text{pixeles}}{\text{metro}} \\ P_y &= Y_{CentroMapa} - P_{ty} \cdot \frac{\text{pixeles}}{\text{metro}} \end{aligned} \quad (4.5)$$

4.2.4.4. Interfaz Hombre-máquina (HMI)

Es necesario comunicarle al conductor del vehículo receptor la presencia de los obstáculos, para ello se ha desarrollado una interfaz hombre-máquina para comunicar al conductor de los objetos detectados por los vehículos emisores. La HMI ha sido diseñada utilizando Java para que sea fácilmente extrapolable a cualquier equipo la interfaz diseñada, es una interfaz sencilla para testear el sistema, en ella existen 3 secciones diferenciadas:

- La sección de configuración e inicio de la comunicación de la HMI con el módulo WSAN. En ella está la configuración de los puertos de comunicación del GPS y del dispositivo de comunicaciones, para ello hay que indicar el puerto y la velocidad en que opera cada uno, los botones de apertura de los puertos e inicio de la recepción de los paquetes del sistema.
- La sección de comunicación y configuración de la interfaz gráfica. Existe un recuadro donde se muestra la información básica del sistema, resultado de la apertura de puertos, estado general del sistema, etc. También se puede elegir el zoom a mostrar en el mapa.

- Por ultimo está la sección del mapa que es la que ocupa la mayor parte de la HMI. En esta sección se representa a el vehículo receptor y todos los objetos enviados por el vehículo emisor así como el vehículo emisor, con un aspecto parecido a la figura 4.6(b).

Por ultimo hay una sección oculta donde se muestran los datos en bruto recibidos por el GPS y la mota WSA, a la que se puede acceder si fuera necesario.

4.2.4.5. Simulador

Dada la dificultad para testear todo el sistema y la imposibilidad de utilizar el *Ivvi 2.0*, se optó por crear una serie de simuladores que proporcionaran la información al sistema como se haría en el mundo real. Para comprobar que la HMI del receptor. Primero se creó un simulador en el lado del receptor que eliminaba el sistema de comunicaciones. Una vez comprobado que todo funcionaba correctamente se modificó este simulador incluyendo el sistema de comunicaciones pero con la salvedad que solo se utiliza un GPS y se modifican los datos de este para simular varias posiciones diferentes. Posteriormente este simulador se separó del sistema receptor y se creó un sistema emisor simulado para colocarlo en otro vehículo y hacer las pruebas de comunicación.

El simulador consta de en diferentes partes:

- Configuración de las comunicaciones. Es la sección de la HMI donde se configura e inician los puertos de comunicación del GPS y del dispositivo de comunicaciones WSA. Para ello hay que indicar el puerto y la velocidad en que opera cada uno, existen unos botones de apertura de los puertos e inicio de la recepción de los paquetes del sistema. Hay un recuadro donde se muestra la información básica del sistema.
- Información actual del GPS del vehículo emisor.
- Modificación de los datos del GPS del vehículo emisor para que la simulación sea más sencilla y flexible al poder mover virtualmente al vehículo emisor. Se puede modificar la posición, velocidad y orientación añadiendo unos valores adicionales a los valores que el sistema recaba del GPS.
- Dos secciones específicas para simular vehículos y peatones detectados por el sistema emisor. Con estas secciones se pueden introducir, borrar y modificar los datos de los vehículos y peatones detectados.
- Dos cuadros de dialogo donde se muestran los peatones y vehículos que el emisor mandara periódicamente al sistema receptor.

4.2.4.6. Resultados

Dado que la parte de detección y seguimiento de objetos ha sido desarrollado por el grupo LSI de la UC3M y que no pertenece al objetivo principal de esta tesis no se van a detallar los resultados de esa parte en este apartado pero si se quiere consultar estos datos se encuentran en el artículo (García et al., 2013). En este apartado se detallaran los resultados concernientes a las comunicaciones.

Para probar la aplicación se realizaron una serie de pruebas del sistema con el simulador de emisión en un vehículo y el sistema receptor en otro vehículo para comprobar la comunicación. El rendimiento de la red se puso a prueba para demostrar la viabilidad del sistema distribuido de detección de objetos. Se realizó un experimento para medir la latencia en el proceso de comunicación. Este experimento comprueba el tiempo necesario para enviar los datos necesarios para alertar a otro vehículo de una detección determinada. Este experimento se hizo en el INSIA con unas condiciones controladas.

FIGURA 4.8: Latencia en función de la distancia

Las características del experimento para comprobar la latencia y las pérdidas de información fueron: Tasa de transmisión de paquetes de datos de 4 Hz, con una duración de 384 segundos a diferentes distancias para comprobar si la distancia afectaba el rendimiento de la red. Se analizaron los datos recogidos del experimento, la latencia requerida para cada distancia se pueden observar en la figura 4.8, después de analizar los datos con se obtuvieron los resultados de la tabla 4.1.

Este experimento demostró que el rendimiento del protocolo de comunicación es muy eficiente y eficaz para esta aplicación de seguridad con una tasa de pérdida de

TABLA 4.1: Resultado experimento de comunicaciones

Número de paquetes recibidos	1395
Número de paquetes perdidos	3
Pérdida de datos	0.20 %
Latencia media	0.17 s

aproximadamente 0,2 %, lo que demuestra la alta fiabilidad del sistema. Además, la prueba realizada demuestra que el sistema es capaz de proporcionar una rápida comunicación entre vehículos por lo cual la distancia recorrida desde la detección de un obstáculo hasta la recepción de la alerta es mínima, aumentando el tiempo de respuesta del conductor y por lo cual mejorando la seguridad vial.

4.3. Aviso a peatones en vías urbanas e interurbanas

El principal problema de las ciudades en cuanto a los accidentes donde hay involucrados vehículos son los atropellos a peatones. Dado que la tendencia actual es que toda la población posea en su bolsillo un *smartphone*, las características de estos dispositivos dan la posibilidad de aprovechar las ventajas en cuanto a conectividad de los módulos DSRC-UPM-INSIA desarrollados, permitiendo a los peatones formar parte del ecosistema de aplicaciones para sistemas cooperativos. El estilo de vida adoptado en las grandes ciudades: aglomeraciones, tráfico elevado, prisas... ha provocado un aumento de los atropellos cuando se atraviesa una vía, por lo cual se ha desarrollado una aplicación móvil que avise al peatón si se encuentra en la trayectoria de un vehículo para que éste pueda reaccionar a tiempo evitando el atropello. La población con alguna minusvalía física, visual o auditiva, tiene más riesgo de sufrir un percance al cruzar una vía por lo cual este sistema es una potencial mejora en su seguridad.

El principio de funcionamiento del sistema cooperativo consiste en que el dispositivo que posee el peatón se conecta al dispositivo del vehículo el cual le transmite la posición, velocidad y dirección, el dispositivo del peatón calcula la trayectoria probable que seguirá el vehículo y si hay posibilidades de una posible colisión. El dispositivo del peatón avisará a éste de todos los posibles problemas.

La aplicación ha sido desarrollada en diversas fases. En la primera se desarrolló la idea básica, los algoritmos de determinación de trayectoria y de colisión entre vehículo y peatón y se desarrolló una pequeña aplicación móvil de testeo del sistema cooperativo. En la segunda fase ya con los módulos DSRC-UPM-INSIA se procedió a la mejora de la interfaz de la aplicación móvil aprovechando las nuevas características de los módulos.

4.3.1. Primera fase: Solución del problema

En la primera fase se tuvieron que investigar y desarrollar los algoritmos de trayectoria, área de destino geográfica (ADG) e investigar cómo crear hacer que se comunicaran los dispositivos de los peatones y de los vehículos.

Estas fase se realizó en el transcurso de una estancia abreviada en el INRIA (Institut National de Recherche en Informatique et en Automatique) y condujeron a una publicación (Anaya et al., 2014).

4.3.1.1. Comunicación inalámbrica

Lo primero que se testeó es la posibilidad de crear un punto de acceso sobre una tarjeta de red wi-fi y que fuera sencillo para un *smartphone* conectarse automáticamente sin necesidad de ninguna interacción adicional. Tras probar diferentes sistemas operativos (IOS, Windows y *GNU/Linux*) se decidió que se utilizaría *GNU/Linux* dada la simplicidad de creación de redes ad-hoc ya que el sistema es más abierto que los otros. El único requisito para poder crear una red ad-hoc es ser superusuario. De esta forma si dos dispositivos tienen la misma configuración en la red ad-hoc pueden comunicarse entre ellos, siempre que no haya un conflicto de IPs.

El problema que surgió fue que los sistemas operativos de los *smartphones* están muy limitados a la hora de la configuración de la tarjetas de red, siendo imposible poner estas en modo ad-hoc. Por lo cual finalmente se optó por crear un punto de acceso en la tarjeta de red con un servidor DHCP propio, de esta manera cualquier teléfono se puede conectar automáticamente sin necesidad de una configuración adicional.

Una vez resuelto el problema de la comunicación inalámbrica se pasó a la siguiente fase.

4.3.1.2. Trayectoria del vehículo

Para poder predecir la trayectoria de un vehículo y estimar si este es un peligro potencial es necesario tener un modelo matemático del comportamiento del vehículo. Existen diversos modelos de la física de los vehículos para los cuales se necesitan muchos datos y complejas ecuaciones para la estimación de la trayectoria. Dadas las limitaciones de computo de los *smartphones*, la complejidad de transmitir estos datos al dispositivo móvil y que el sistema opere en tiempo real, se optó por el modelo matemático más sencillo posible, tanto en datos como en cálculos, y que de unos resultados suficientemente aceptables. La solución fue el modelo de la bicicleta.

4.3.1.2.1. Modelo de bicicleta

El modelo simplificado de bicicleta transforma la complejidad de cálculo del giro de un vehículo dado que cada rueda tiene una velocidad angular diferente, a un modelo donde hay solamente dos ruedas, una directriz y otra fija. En la Figura 4.9 se puede observar la simplificación, con ello se ha conseguido simplificar las ecuaciones del coche al modelo de bicicleta el cual es más simple.

Con este modelo simplificado calcular la posible trayectoria del vehículo se facilita enormemente. En la Figura 4.10 se puede observar el modelo de la bicicleta de Akerman

FIGURA 4.9: Simplificación modelo de coche

FIGURA 4.10: Simplificación modelo de coche

que consta de un sistema de coordenadas (OV, XV, YV) que está centrado en el neumático trasero, en base a las coordenadas y a la información disponible (velocidad $V = |V_G|$ y ángulo de giro (ω)), se deduce el ángulo de Akerman (el ángulo de las ruedas del vehículo, ϵ_r) y el Centro Instantáneo de Rotación (CIR) $(I_S(X_S, Y_S))$ con las ecuaciones 4.6

$$\begin{aligned} R_S &= \frac{V}{\omega}, & \epsilon_r &= \frac{L}{R_S} \\ X_S &= 0, & Y_S &= R_S \cdot \cos\left(\frac{\epsilon_r}{2}\right) \end{aligned} \quad (4.6)$$

donde:

- R_S es el radio de giro del vehículo.
- V es la velocidad medida en $m \setminus s$.
- ε_r es el ángulo de Akerman.
- L es la longitud del vehículo en metros.
- X_S y Y_S definen el punto central de la rotación del vehículo.

4.3.1.2.2. Área de Destino Geográfica (ADG)

Una vez que se tiene un modelo de la cinemática del vehículo sencillo hay que calcular la trayectoria que tendrá, con las ecuaciones 4.7 se puede calcular la trayectoria del vehículo en el tiempo.

$$\begin{aligned} X(t) &= R_S \cdot \sin(\omega \cdot t + \varepsilon_r) + X_S \\ Y(t) &= R_S \cdot \sin(\omega \cdot t + \varepsilon_r) + Y_S \end{aligned} \quad (4.7)$$

donde:

- t es el tiempo transcurrido.
- ω es la velocidad angular medida en $rad \setminus s$.

Dado que tanto el peatón como el vehículo están en movimiento y es un sistema en constante cambio, con las ecuaciones 4.8 se define un Área de Destino Geográfico (ADG) que se puede observar gráficamente en la Figura 4.11

$$\begin{aligned} XI(t) &= R_{SI}(t) \cdot \sin(\omega \cdot t + \varepsilon_r) + X_S \\ YI(t) &= R_{SI}(t) \cdot \sin(\omega \cdot t + \varepsilon_r) + Y_S \\ I &= [R, L] \\ R_{SR}(t) &= \frac{|\omega|}{\omega} \cdot \sqrt{(R_S + \frac{W}{2} + P_V \cdot t)^2 + L^2} \\ R_{SL}(t) &= \frac{|\omega|}{\omega} \cdot \sqrt{(R_S - \frac{W}{2} - P_V \cdot t)^2 + L^2} \end{aligned} \quad (4.8)$$

donde:

- I define si es la ecuación es de la línea que delimita el ADG por la derecha (R) o por la izquierda (L).

- W es el ancho del vehículo medido en metros.
- P_V es la velocidad del peatón en $m \setminus s$.

FIGURA 4.11: Posible área de destino

4.3.1.3. Aplicación móvil de demostración

Una vez conocida el área de interés se puede calcular si las trayectorias del peatón y del vehículo entran en conflicto y alertar al conductor o al peatón.

Los accidentes por atropello se están incrementándose porque los usuarios se distraen mirando su dispositivo móvil. En una ciudad el número de peatones es muy superior al de vehículos. Los módulos ITS contienen la información del resto de módulos ITS colindantes, la gran mayoría de estos son vehículos en movimiento, por lo cual son potenciales peligros para el peatón. El *smartphone* del peatón solo se puede conectar a un dispositivo ITS a la vez. Se decidió que el módulo ITS avisara al dispositivo móvil del peatón de todos los peligros colindantes. Como primer paso se decidió desarrollar una aplicación básica para probar la validez del sistema de alerta.

En la interfaz se mostraban los datos de los vehículos colindantes y sus características de posicionamiento, velocidad y dirección. Mediante un código de colores se indicaba si el dispositivo, transportado por el peatón, estaba fuera de peligro por encontrarse a la espalda del vehículo, verde, estaba en frente del vehículo pero fuera de ADG, azul, o en alerta por encontrarse en el interior del ADG, rojo. en la Figura 4.12 se puede observar la interfaz desarrollada para ver la viabilidad del sistema.

FIGURA 4.12: Primera interfaz de AAV

4.3.2. Segunda fase: Desarrollo de la aplicación móvil

Viendo que la interfaz gráfica era mejorable, su posible uso era limitada y la utilidad para avisar de posibles peligros era poca se ha decidido desarrollar una nueva aplicación para *smartphone* más acorde a todas estas exigencias. Cuando se desarrolló la idea inicial los módulos DSRC-UPM-INSIA no habían sido desarrollados aún, por lo cual se ha optado por reproducir los mismos algoritmos de colisión en la nueva aplicación con el añadido de la versatilidad de los nuevos módulos DSRC-UPM-INSIA y de la mejora sustancial producida en los dispositivos móviles.

4.3.2.1. Adquisición de los datos

Para conseguir los datos de los vehículos colindantes el *smartphone* se conecta a la red abierta que crean los módulos DSRC-UPM-INSIA, y a través de ella se conecta al GNES. A través de peticiones consigue la información de todos los vehículos que consta de tiempo de captura de los datos, posición (latitud y longitud), velocidad y dirección (ángulo respecto del norte). Con estos datos puede aplicar las formulas, desarrolladas durante la primera fase de la aplicación, y calcular si un vehículo puede estar en trayectoria de colisión con el peatón y avisar a este.

4.3.2.2. Interfaz gráfica

La primera decisión y más importante fue que la aplicación tuviera el mínimo texto posible, fuera lo más simple e intuitiva posible y que avisara al usuario mediante sonido y vibraciones. El desarrollo de ésta interfaz gráfica se realizó mediante un “Trabajo Final de Grado” (Masvidal, 2015).

FIGURA 4.13: Diferentes pantallas de la aplicación

En la nueva aplicación se utiliza el mismo código de colores para avisar de forma visual de si el peatón esta fuera de peligro, en aviso o en alerta, verde, azul y rojo respectivamente. En el caso de alerta se avisa al peatón adicionalmente mediante vibración y sonido. en las Figuras 4.13(a), 4.13(b), 4.13(c) se puede observar como se muestra la interfaz en cada caso.

Adicionalmente se ha desarrollado un *widget* (ver Figura 4.13(d)) con la misma funcionalidad por si el usuario solo le interesa tenerlo como un sistema de aviso sin la necesidad de tener abierta la aplicación mostrándola por la pantalla.

4.3.3. Pruebas

El sistema ha sido probado en condiciones reales controladas donde se simulaba el cruce indebido de un peatón en una calle en curva donde la visibilidad es limitada. Éste es un tipo de accidente muy común en las ciudades.

En la figura 4.14 se puede observar la trayectoria del peatón y del vehículo, en esta prueba el peatón estaba quieto y no cruza por el aviso del teléfono. En la gráfica 4.14(a) está representada la trayectoria del vehículo y del peatón, en la gráfica están marcados los

instantes que sirven de referencia para conocer la trayectoria y también relacionarlos con la gráfica 4.14(b). En la gráfica 4.14(b) está representada la distancia del vehículo respecto al peatón, siendo positiva si el peatón está en frente del vehículo y negativa si el peatón está en la parte trasera, la distancia al ADG así como los valores del aviso de presencia y el de aviso de colisión. También está representado el tiempo a la posible colisión.

FIGURA 4.14: Resultados de prueba de la aplicación de aviso de peatones

Si se observa la Figura 4.14(a) aparentemente el sistema no ha funcionado correctamente dado que la aplicación no da ningún aviso de presencia y que directamente pasa a alertar de posible colisión, T3, esto es debido a que el vehículo al empezar a trazar la curva hace crecer el ADG rápidamente provocando que el peatón se sitúe en su interior. Adicionalmente el dispositivo ha dejado de avisar al peatón cuando el vehículo está a su lado. El motivo es que el vehículo está en las proximidades con una trayectoria lineal y el peatón está quieto por lo cual la forma del ADG es muy estrecha y la posición del peatón queda situada en su exterior. Esto queda más claro si se observa la tercera gráfica de la figura 4.14(b) donde en el instante 11 la distancia al ADG es mayor que la distancia de aviso.

En la Figura 4.15 se ve otra prueba en la que el peatón está caminando en dirección contraria al vehículo. En la gráfica 4.15(a) está representada la trayectoria del vehículo y del peatón, en la gráfica están marcados los instantes que sirven de referencia para conocer la trayectoria y para como guía en la gráfica 4.15(b). En la gráfica 4.14(b) está representada la distancia del vehículo respecto al peatón, siendo positiva si el peatón está en frente del vehículo y negativa si el peatón está en la parte trasera, la distancia al ADG así como los valores del aviso de presencia y el de aviso de colisión. También está representado el tiempo a la posible colisión.

En la Figura 4.15(a) si hay una fase de aviso de presencia (T7) y posteriormente de aviso de colisión (T8-T10). Esto es debido a la velocidad del vehículo en ese instante y la trayectoria del peatón que no es de cruce con el vehículo sino paralela a este. Se puede observar claramente en la gráfica 4.15(b) que la distancia al ADG en el instante T8 está situado entre los valores de aviso de colisión y de presencia.

FIGURA 4.15: Resultados de prueba de la aplicación de aviso de peatones

4.4. ***BikeWarn: Detección de bicicletas mediante iBeacons***

En ciudades se están promoviendo medios alternativos al vehículo privado para solucionar los grandes problemas de contaminación y circulación. Uno de los medios de transporte que se están impulsando últimamente con más fuerza y que está teniendo un incremento constante a lo largo del tiempo es la bicicleta, ya sea por la implantación de servicios públicos de bicicletas o el uso de bicicletas privadas. El uso de bicicletas en ciudad entre semana es principalmente para el transporte y en fines de semana como uso recreativo o deportivo. Estas circunstancias están provocando un incremento considerable de accidentes en ciudades.

El aviso a los conductores de la presencia de una bicicleta en la calzada es vital pues es de los vehículos más frágiles. La instalación de los módulos desarrollados sería una solución muy conveniente para ayudar a reducir la siniestralidad de las bicicletas, lamentablemente esta implementación no es viable en un vehículo de estas características:

- Autonomía: La bicicleta al no poseer de un sistema de alimentación propio sería necesario la adición de baterías.
- Volumen: El módulo aunque de reducidas dimensiones para una bicicleta es demasiado aparatoso, si a esto le sumamos las baterías necesarias harían que el volumen necesario aumente considerablemente resultando excesivo para las características del vehículo.
- Peso: Como se ha explicado en el punto anterior la necesidad de añadirle baterías al módulo, el peso del sistema aumenta considerablemente.
- Precio: Los módulos desarrollados tienen un precio comedido que no implica un sobre coste apreciable a la hora de comprar un vehículo motorizado, pero en una bicicleta puede llegar a alcanzar un sobre coste en la bicicleta que pocos usuarios estarían dispuestos a asumir.

Una posible solución sería desarrollar una aplicación parecida a la de aviso a agentes vulnerables (Sección 4.3) pero especializada para bicicletas, esto conlleva un inconveniente, depende que el usuario de la bicicleta decida activarla si lleva un *smartphone* encima, filosofía adoptada por la DGT. Además el dispositivo encargado de la comunicación no está en el vehículo, sino que está en el ciclista. También implica que el responsable de evitar la colisión recae en la bicicleta la cual tiene menor margen de maniobra que un vehículo, aparte de que el ciclista no puede estar atento al *smartphone*, al tráfico y mantener el equilibrio al mismo tiempo. Lo fundamental es avisar al conductor del vehículo, que es el que va a colisionar con la bicicleta.

La otra opción es equipar a la bicicleta con un medio de comunicaciones de bajo coste y pasivo, que le permita comunicarse sin obligar al ciclista a hacer ninguna acción que le distraiga de la circulación.

Por estas razones se ha pensado y desarrollado una solución de bajo coste y con unas características físicas que sean viables para la implementación en una bicicleta. En un principio se pensó en un sistema RFID pero el módulo no tiene un receptor RFID y su implementación implicaría aumentar los costes de este. Finalmente se ha decidido utilizar emisores intermitentes inalámbricos de bajo coste, *iBeacons*, por sus características:

- Frecuencia de emisión: es configurable y puede tomar valores de una emisión cada 100ms. a una vez cada 60s.
- Alcance: el rango de recepción de la señal emitida varía de unos pocos centímetros hasta 50 metros en función de la potencia con que se configure.
- Autonomía: en función de la frecuencia y potencia de emisión la batería de estos dispositivos puede durar hasta dos años sin necesidad de cambio.
- Precio: pese a ser una tecnología reciente el precio de *iBeacons* comerciales es de unos pocos euros.
- Volumen: un *iBeacon* no ocupa prácticamente nada y sería muy sencillo colocarlo en la bicicleta y su presencia pasaría desapercibida (Figura 4.16).
- Peso: el peso de un *iBeacon* es de unos pocos gramos que no implicaría ningún sobreesfuerzo ni para el ciclista más exigente.

Para el desarrollo de la aplicación cooperativa se han desarrollado dos aplicaciones diferentes, una utilidad para el módulo DSRC-UPM-INSIA y una aplicación móvil para el aviso de bicicletas cercanas.

4.4.1. Utilidad: detección de *ibeacons*

Dado que el módulo ya poseía previamente un módulo BLE que puede detectar estos *iBeacons* y que los *iBeacons* emiten periódicamente una trama donde indican: su MAC y con que potencia están emitiendo (TX-Power), entre otros datos.

Se ha desarrollado una aplicación en *Python* la cual está conectada al GUES y se encarga de monitorizar el dispositivo BLE para capturar los datos retransmitidos por los *iBeacons* instalados en las bicicletas, junto con la potencia recibida (RSSI) por el dispositivo BLE. Cada vez que llega una actualización de los datos de un *iBeacon* extrae los datos de este,

FIGURA 4.16: iBeacons utilizados

los encapsula y le comunica al GNES la presencia de una bicicleta en los alrededores para que esta información sea incluida en la BD_Local del GNES y sea accesible. Para que todas las aplicaciones de alto nivel puedan acceder a estos datos y realizar las acciones que consideren oportunas. Cuando el GNES recibe la información del *iBeacon* de la bicicleta estos datos son encapsulados con el TST y la posición (latitud y longitud) GPS del módulo y guardados en la BD_Local.

4.4.2. Aplicación móvil

Una vez se ha implementado la aplicación en el módulo, y se ha testado su correcto funcionamiento, cualquier aplicación de alto nivel que esté conectada con GNES será capaz de conocer las bicicletas cercanas (Figura 4.17) con una simple consulta.

Con tal fin se ha desarrollado una pequeña aplicación "*BikeWarn*" donde se le indica al conductor la cercanía de una bicicleta con un código de colores en función de si existe una bicicleta en las cercanías y la distancia de esta al vehículo. Si la pantalla esta verde, no hay ninguna bicicleta a su alrededor, si la pantalla esta en naranja, se ha detectado una bicicleta en las cercanías, si la pantalla esta en rojo, significa que hay una bicicleta a menos de 5 metros y que tiene que tener cuidado.

FIGURA 4.17: Funcionamiento de *BikeWarn*

Para calcular las distancias entre la bicicleta y el vehículo se ha utilizado la ecuación 4.9 deducida a partir de (Benkic et al., 2008).

$$Distancia = 10^{\frac{TxPower - RSSI}{10 \cdot n}} \quad (4.9)$$

Donde $n = 2$ para espacios abiertos.

La limitación del sistema radica en que mediante los *iBeacons* no se puede dar una posición relativa de la bicicleta respecto del vehículo. Solo se puede indicar al conductor la distancia a la que se encuentra la bicicleta respecto al vehículo sin darle información de si se encuentra delante, detrás, a la derecha o a la izquierda. Para intentar paliar este problema se ha utilizado GNES donde los diferentes vehículos comparten los usuarios que no tienen posibilidad de geocomunicación por no poseer un módulo DSRC-UPM-INSIA. Así un vehículo puede acotar con más precisión la posición de la bicicleta (Figura 4.18).

4.4.3. Pruebas y resultados

Para probar el sistema se hicieron varios experimentos de detección de bicicletas en entornos controlados y en entornos urbanos con tráfico real en la figura 4.19 se puede observar uno de ellos donde se produce el adelantamiento de un vehículo a una bicicleta en un tramo del campus sur de la UPM situado en Vallecas donde hay poca visibilidad debido al trazado de la carretera y que frecuentemente hay coches, camiones y autobuses estacionados en segunda fila.

El sistema detectó la bicicleta por primera vez a una distancia de 57 metros y la mantuvo localizada todo el tiempo durante el adelantamiento y posteriormente hasta que el vehículo

FIGURA 4.18: Acotamiento de la bicicleta

FIGURA 4.19: Trayectoria de la bicicleta y del vehículo durante la prueba

se alejó de ella 50 metros. Este experimento demuestra que el sistema cooperativo *BikeWarn* es efectivo, su funcionamiento es correcto con unas distancias adecuadas para entornos urbanos y carreteras comarcales con curvas pronunciadas que eviten la visibilidad del ciclista.

4.5. ***MotoWarn*: Aviso de motocicletas en ángulos muertos**

Este sistema cooperativo es una novedosa solución a las situaciones más comunes que conducen a accidentes con motocicletas, que son causadas por las características de este tipo de vehículo, el tamaño, la movilidad y el tipo de movimientos que son capaces de realizar. Los casos típicos de accidente donde está implicada una motocicleta son:

- La motocicleta está situada en uno de los puntos ciegos de un vehículo: Un accidente típico se produce cuando la motocicleta se encuentra en el ángulo muerto del espejo retrovisor de un vehículo e intenta adelantar a vehículo gracias a su mejor aceleración, ya que el conductor del vehículo no tiene conocimiento de la presencia de la motocicleta se produce el accidente al realizar el vehículo un cambio de carril o un giro.
- La motocicleta circula en el espacio entre dos vehículos en carriles adyacentes: Teniendo en cuenta las características de las motocicletas, en las grandes ciudades a menudo viajan entre dos carriles para evitar el tráfico intenso, por lo que situaciones como la apertura de una puerta o un cambio de carril pueden provocar un accidente, dado que los conductores de vehículos no suelen tener en cuenta dicha situación.
- En las carreteras de calzada única con curvas cerradas y con baja visibilidad: Son bastante comunes en las carreteras de montaña esta situación dado que son carreteras con poca visibilidad. En este tipo de carretera, una motocicleta y otro vehículo en movimiento en la dirección opuesta tienden a moverse hacia el centro de la carretera, por lo que puede ocurrir un accidente. Dadas las reducidas dimensiones de las motocicletas, los conductores de los vehículos no son capaces de percibir la motocicleta a tiempo para ajustar la velocidad y posición del vehículo en la carretera para mantener las condiciones de seguridad, lo que provoca el accidente por colisión o en el intento de esquivar a la motocicleta.

Los módulos desarrollados tienen una gran capacidad de comunicación pero no tienen ninguna interfaz adecuada para comunicarse con el conductor. Por lo cual se ha decidido implementar la aplicación de seguridad sobre una plataforma externa al módulo de comunicaciones, para se ha conectado a través de GNSS para obtener la información de la red vehicular que es necesaria para detectar si una motocicleta se encuentra en la zona con el fin de advertir al conductor y evitar un accidente. La aplicación se puede implementar en diversos dispositivos tales como la consola del vehículo, un ordenador o incluso un navegador de coche, porque todos estos dispositivos son capaces de conectarse al módulo de comunicaciones.

Finalmente se ha desarrollado una aplicación en un *smartphone*, la razón de esta elección es que estos dispositivos los tienen prácticamente la mayoría de los conductores y también tienen una conectividad compatible con los módulos de comunicaciones, por lo que sólo es necesario descargar e instalar una aplicación en el móvil para implementar la aplicación de prevención de accidentes, esta aplicación no necesita un permiso especial para trabajar en el dispositivo. Esta aplicación ha sido desarrollada para la plataforma *Android* y se llama *MotoWarn*. La plataforma *Android* ha sido seleccionada porque a día de hoy, es el sistema operativo más extendido en los teléfonos inteligentes. La aplicación se conecta al módulo DSRC a través de la red WiFi específica del módulo y por medio de GNET es capaz de obtener datos de los vehículos conectados a la red, estos datos del módulo se almacenan en el teléfono para poder hacer un seguimiento más preciso de los objetos. El módulo DSRC proporciona a la aplicación móvil los datos de cada vehículo conocido en la red, así como datos del propio módulo: Identificador (GN_Addr), posición GPS, velocidad y dirección.

Con estos datos, *MotoWarn* puede determinar qué vehículos de su alrededor, se están acercando y cuáles se están alejando. La aplicación filtra los vehículos quedándose con las motocicletas, que están en los alrededores o acercarse al vehículo, ya sea porque el vehículo se acerca a la motocicleta, el vehículo supera a la motocicleta o a la motocicleta se aproxima al vehículo, los objetos con interés se almacenan en una base de datos interna para ser analizados con más cuidado y advertir al conductor si es necesario.

4.5.1. Interfaz desarrollada

Se han probado diferentes interfaces de usuario con el fin de determinar cuál es la que produce un nivel más bajo de distracción para el conductor, y aun así es eficaz en situaciones de peligro. Para ello se han utilizado estímulos sonoros y visuales, pop-ups, texto e imágenes únicas. Tras el estudio de la bibliografía relacionada con las distracciones de las HMI en conducción ((Jiménez et al., 2012), (Brooke et al., 1996), (Zijlstra, 1993), (Pérez-Moreno et al., 2015)) se llegó a las siguientes conclusiones:

- El uso exclusivo de texto se ha descartado debido a que provoca grandes distracciones.
- El uso del sonido se ha reducido solo a los casos de peligro inminente para evitar la saturación del conductor.
- Pop-ups se utilizan sólo durante el inicio de la aplicación, ya que son una fuente de distracción innecesaria.
- El uso de imágenes es el más adecuado siempre que sea comedido y no intrusivo.

Por estas razones, se decidió utilizar imágenes casi exclusivamente. Por lo cual se ha desarrollado una interfaz basada en áreas de interés, donde la pantalla se divide en una cuadrícula de 5x5 y el vehículo se coloca en la rejilla central (3.3), la parte delantera del espacio de vehículo está representado por las filas 1 y 2, y las posiciones traseras están representadas por las filas 4 y 5. Estas rejillas tienen diferentes dimensiones dependiendo de la posición de la motocicleta respecto al vehículo. También tienen un comportamiento diferente dependiendo de si la motocicleta se acerca al vehículo desde la parte delantera o la parte trasera.

En el caso de una motocicleta y un vehículo tanto que circula en la misma dirección, el área de interés se encuentra en la parte trasera del vehículo, ya que, la situación de peligro se produce porque la motocicleta se encuentre muy probablemente en un punto ciego del vehículo, con la posibilidad de un accidente cuando la moto intenta adelantar al vehículo. En el caso de que la motocicleta este acercarse al vehículo en sentido contrario a la marcha del vehículo, el área de interés se encuentra por delante del vehículo.

Diferenciando estos dos casos, los tamaños de la rejilla adquieren diferentes dimensiones dependiendo del dirección relativa de la motocicleta y el vehículo:

- Ambos vehículos circulan en la misma dirección: el área de interés se encuentra en la parte posterior del vehículo, por lo cual, las celdas traseras (filas 4 y 5) abarcan hasta 50 metros y las celdas delanteras (filas 1 y 2) sólo 30 metros (Figura 4.20a).
- Vehículos que se mueven en direcciones opuestas: el área de interés se encuentra en la parte delantera del vehículo, por lo cual, las celdas delanteras (filas 1 y 2), abarcan a 150 metros, mientras que las celdas traseras (filas 4 y 5) no tienen prácticamente ningún interés y tienen un valor simbólico de 15 metros (Figura 4.20b).

De esta forma no se requiere que el conductor preste atención a la imagen de la motocicleta en la pantalla para determinar su significado. La selección de las distancias depende de las características del vehículo, el tipo de carretera y la notificación requerida para asegurar que el sistema funciona correctamente.

4.5.2. Posicionamiento en la interfaz

La parte más importante es conocer la posición de la motocicleta respecto al vehículo y mostrarla adecuadamente en la pantalla del dispositivo. Lo primero que hay que conocer es la distancia existente entre la motocicleta y el vehículo. La distancia se calcula mediante las formulas de la “Distancia de Haversine” que se pueden observar en la ec. 4.10.

 FIGURA 4.20: Diagrama de la pantalla de *MotoWarn*

$$\begin{aligned}
 Distance_{Haversine} &= R \cdot c \\
 R &= Earthradius \\
 \Delta Lat &= Lat_{Car} - Lat_{Bike} \\
 \Delta Long &= Long_{Car} - Long_{Bike} \\
 a &= \sin^2\left(\frac{\Delta Lat}{2}\right) + \cos(Lat_{Bike}) \cdot \cos(Lat_{Car}) \cdot \sin^2\left(\frac{\Delta Long}{2}\right) \\
 c &= 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a})
 \end{aligned} \tag{4.10}$$

El siguiente paso es conocer el ángulo relativo que tiene la motocicleta respecto al vehículo, esto lo conseguimos utilizando las ecuaciones de ec. 4.11.

$$\begin{aligned}
 Ang_{rel} &= Azimut - Ang_{Car} \\
 Azimut &= \text{atang}\left(\frac{\Delta p}{Lat_{Bike} - Lat_{Car}}\right) \\
 \Delta p &= (Long_{Bike} - Long_{Car}) \cdot |\cos(Lat_{Car} + Lat_{Bike})|
 \end{aligned} \tag{4.11}$$

Una vez que podemos situar la motocicleta respecto al vehículo en unos ejes de coordenadas conocidos tenemos que determinar en qué posición de la cuadrícula hay que pintar la motocicleta. Con la ec. 4.12 determinamos las coordenadas respecto al vehículo.

$$\begin{aligned}
 Vertical &= Distance \cdot \cos(Ang_{rel}) \\
 Horizontal &= Distance \cdot \sin(Ang_{rel})
 \end{aligned} \tag{4.12}$$

Si tenemos en cuenta que dependiendo de si la motocicleta lleva una trayectoria de cruce o de adelantamiento respecto al vehículo las cuadrículas toman un tamaño diferente hay que determinar esta, para ello utilizamos el algoritmo 3.

Algoritmo 3 Cruce o adelantamiento

Entrada: Ángulo Motocicleta, Ángulo vehículo.

Salida: Estado relativo

```

1: if  $Ang_{Bike} \in [Ang_{Car} - 90, Ang_{Car} + 90]$  then
2:   return Cruce
3: else
4:   return Adelantamiento
5: end if

```

Una vez que conocemos si la motocicleta está cruzándose con el vehículo o adelantándolo podemos determinar en qué cuadrícula situarlo para ello se utiliza el algoritmo 4. Dependiendo del resultado del algoritmo 3 los Vértices Horizontales y Verticales cambian los valores, para adaptarse a las condiciones de las cuadrículas anteriormente descritas en la sección 4.5.1.

Algoritmo 4 Determinar cuadrícula

Entrada: Horizontal, Vertical, Vértices Horiz. = $\{h_0, \dots, h_6\}$, Vértices Vert. = $\{v_0, \dots, v_6\}$.

Salida: fila, columna

```

1: for  $i = 0; v_0$  hasta  $v_5; i++$  do
2:   if vertical >  $v_i$  then
3:     fila = i
4:   end if
5: end for
6: if vertical >  $v_6$  then
7:   fila = -1
8:   return fila, columna
9: end if
10: for  $j = 0; h_0$  hasta  $h_5; j++$  do
11:   if horizontal >  $h_j$  then
12:     columna = j
13:   end if
14: end for
15: if horizontal >  $h_6$  then
16:   columna = -1
17: end if
18: return fila, columna

```

4.5.3. Pruebas de la aplicación

La aplicación ha sido probada tanto en ambientes controlados como en carreteras convencionales con condiciones de circulación reales.

4.5.3.1. Pruebas controladas

Los resultados que se presentan a continuación son de experimentos realizados con condiciones controladas, en las instalaciones del INSIA, dada la sencillez para realizar algunas maniobras y la localización de los puntos conflictivos.

Se han realizado cuatro experimentos diferentes: seguimiento del vehículo a la motocicleta, adelantamiento de la motocicleta al vehículo, cruce de la motocicleta y el vehículo en recta y por ultimo cruce en una curva sin visibilidad de la motocicleta y el vehículo.

En las figuras que se muestran de las pruebas se puede observar las trayectorias seguidas por la motocicleta y el vehículo, para poder observarse la posición en cada instante, se han puesto marcadores cada cierto tiempo para que sea más fácil de mostrar la dinámica del experimento. También está representada la distancia calculada por la aplicación de *MotoWarn*. En esta distancia la aplicación pudo calcular la posición relativa de la motocicleta respecto al vehículo y poder mostrarla al conductor del vehículo, siendo esta positiva si la motocicleta se encuentra delante del vehículo y negativa si se encuentra detrás.

4.5.3.1.1. Seguimiento

Este experimento consistió en simular el caso de que un vehículo se aproxima a la motocicleta en ciudad o autopista a gran velocidad y debe frenar porque la motocicleta circula a menor velocidad y posterior seguimiento dada la imposibilidad de adelantar.

(a) Trayectoria del vehículo y de la motocicleta

(b) Distancia relativa calculada por la aplicación

FIGURA 4.21: Seguimiento del vehículo a la motocicleta

Los resultados obtenidos en el experimento se pueden observar en la Figura 4.21 donde se muestra la trayectoria 4.21(a) y la distancia 4.21(b). Se puede observar que durante el experimento la aplicación estuvo en todo momento avisando al conductor del vehículo de la presencia de una motocicleta porque la distancia nunca fue superior a 30 metros. Entre los instantes T2 y T3 la distancia se redujo a 5 metros que es muy poca distancia.

4.5.3.1.2. Adelantamiento

Este experimento consistió en simular el adelantamiento de una motocicleta, gracias a la mayor aceleración que tiene, a un vehículo en una carretera convencional de doble sentido o en una vía rápida, produciéndose una situación potencialmente peligrosa si el conductor no tiene conocimiento de la presencia de la motocicleta.

FIGURA 4.22: Adelantamiento de la motocicleta al vehículo

Los resultados obtenidos durante la prueba se pueden observar en la Figura 4.22 donde se muestra la trayectoria 4.22(a) y la distancia 4.22(b). El adelantamiento se produce entre los instantes T2 y T3, se puede observar en la distancia relativa entre el vehículo y la motocicleta

de la Figura 4.22(b) que pasa de ser negativa a positiva y en la Figura 4.22(a) que se observa que hay dos puntos azules consecutivos. También se observa que la motocicleta entre los instantes T_0 y T_2 mantiene una distancia segura y cuando ve que las condiciones son buenas para el adelantamiento lo realiza con seguridad. El problema en estas situaciones radica en que para que el motorista vea si existe la posibilidad de adelantamiento se coloca en el ángulo muerto del conductor del vehículo. El sistema en todo momento está informando al conductor del vehículo de la presencia de la motocicleta en su parte trasera, de ésta forma la tiene presente en caso de querer hacer una maniobra.

4.5.3.1.3. Cruce en recta

Este experimento se diseñó para ver si el sistema era capaz de tratar adecuadamente la interacción de la motocicleta y el vehículo aproximándose en direcciones contrarias. Este experimento pone a prueba la fiabilidad del sistema dado que la velocidad relativa entre ambos vehículos es mayor que en otros experimentos.

FIGURA 4.23: Cruce del vehículo y la motocicleta en recta

Los resultados obtenidos en el experimento se pueden observar en la Figura 4.23 donde se muestra la trayectoria 4.23(a) y la distancia 4.23(b). Durante el experimento se pudo corroborar el funcionamiento de la red vehicular y del sistema *MotoWarn*. Observando la Figura 4.23 se puede observar que el sistema hace un seguimiento adecuado de la motocicleta a más de 150 metros sin problema y en todo momento se conoce la posición de la motocicleta. El sistema funciona adecuadamente en caso de cruce avisando al conductor a distancias menores de 150 metros si la motocicleta se aproxima de frente y dejándolo de

hacer una vez que se ha producido el cruce, en las inmediaciones del instante T4, pasados 15 metros.

4.5.3.1.4. Cruce en curva

Este experimento se diseñó para simular las situaciones que se producen principalmente en carreteras de montaña donde hay una serie de curvas sin visibilidad y el encontrarse una motocicleta en sentido contrario supone un gran problema si aparece de repente en el campo visual del conductor. Para simularlo se eligió una curva cerrada donde la visibilidad queda obstaculizada por la presencia de un edificio.

(a)

(b)

FIGURA 4.24: Seguimiento del vehículo a la motocicleta

Los resultados obtenidos durante la prueba se pueden observar en la Figura 4.24 donde se muestra la trayectoria 4.24(a) y la distancia 4.24(b). En este experimento se observa que el aviso de motocicleta se produce entre los instantes T1 y T2 cuando el vehículo se

encuentra a unos 30 metros de distancia, esto es debido a que el sistema no considera que la motocicleta está en la misma carretera. Gracias a la información proporcionada por *MotoWarn* el conductor se abrió al tomar la curva para dejar espacio suficiente para pasar a la motocicleta. Pese a haber un edificio en medio el sistema funcionó correctamente durante todo el tiempo.

4.5.3.2. Pruebas en carreteras convencionales

Los resultados que se presentan a continuación son de experimentos realizados en carreteras convencionales. Concretamente en los alrededores de Madrid en un tramo de la carretera M-315, la cual es una carretera de un único carril por sentido y con una baja densidad de tráfico. Las características de este tramo son ideales para probar el sistema porque contiene tramos de curvas cerradas donde la visibilidad es muy limitada. Adicionalmente la orografía y elementos colindantes a la carretera limitan la visibilidad (Figura 4.25). Las características descritas anteriormente hacen que la velocidad adecuada ronde aproximadamente los 40 *km/h*, mientras que algunos tramos permiten velocidades seguras de 60 *km/h*, aunque el límite de velocidad genérico es de 90 *km/h*.

FIGURA 4.25: Muestra de las características de visibilidad de la M-315

Se han realizado dos experimentos diferentes: seguimiento de la motocicleta al vehículo y posterior adelantamiento, cruce de la motocicleta y el vehículo en una curva con baja visibilidad.

4.5.3.2.1. Seguimiento y adelantamiento

En esta prueba se simuló el caso en el que un vehículo circula por el carril izquierdo de la carretera y una motocicleta se acerca desde la parte trasera a una velocidad más alta, alcanzando el vehículo y posteriormente lo adelanta cuando tiene oportunidad porque el tráfico lo permite.

FIGURA 4.26: Seguimiento y adelantamiento de la motocicleta al vehículo

En la Figura se muestran los resultados de la prueba. En la Figura 4.26(a) se muestran las trayectorias de la motocicleta y del vehículo. Los ejes X e Y representan, respectivamente, las coordenadas GPS en el sistema de coordenadas UTM, también están representados los puntos de interés de la ruta seguida por los dos vehículos que están involucrados en la prueba. En la Figura 4.26(b) se muestra la evolución de la distancia existente entre la motocicleta y el vehículo, las distancias negativas significan que la motocicleta está detrás del vehículo. Así como los instantes en los que la aplicación MotoWarn comunica al conductor

del vehículo de la presencia de una motocicleta en sus alrededores. La Figura 4.26(c) representa la evolución de la velocidad de los vehículos durante la prueba, expresada en kilómetros por hora.

Se puede observar que la transmisión de datos a través de la red de comunicaciones vehicular funcionó correctamente, siempre proporcionó la información necesaria para determinar la presencia de la motocicleta, posición y la velocidad, al vehículo. La aplicación MotoWarn se mantuvo constantemente en comunicación con el módulo DSRC-INSIA-UPM intercambiando información, posibilitando el control de la posición relativa de la motocicleta respecto al vehículo y la generación de los mensajes correspondientes si fuera necesario. En este caso, la advertencia se puso en marcha en el instante T0 de la prueba, ya que como se ha indicado anteriormente, la distancia entre los dos vehículos es inferior a 50 metros. La Figura 4.26(a) muestra la diferente posición de la motocicleta durante la aproximación y el adelantamiento, parando la advertencia a los entre los instantes T4 y T5, por encontrarse la motocicleta más alejada de 30 metros.

En la Figura 4.27 se muestra la secuencia de imágenes que representa esta prueba, incluyendo la diferente posición de la motocicleta y el automóvil, que detalla la maniobra de adelantamiento, cuando la motocicleta se coloca en los puntos ciegos del vehículo durante la operación.

FIGURA 4.27: Secuencia de imágenes de seguimiento y adelantamiento

Para comprobar la fiabilidad del sistema, se realizaron 5 repeticiones de esta prueba, con una distancia de transmisión máxima promedio de 300 metros, el alcance máximo del DSRC, con un retardo de transmisión máximo de 100 ms (la resolución natural de la reloj GPS a 10 Hz) y donde la visualización de las advertencias funciona sin retrasos, con un error máximo de la posición 5 metros, el error natural de los receptores GPS utilizados en esta ocasión con los módulos

4.5.3.2.2. Cruce con baja visibilidad

En esta prueba se simuló el caso en que el vehículo y la motocicleta se aproximan en direcciones opuestas coincidiendo en una curva sin visibilidad o curva ciega.

(a)

(b)

(c)

(d)

FIGURA 4.28: Cruce de la motocicleta y el vehículo en curva sin visibilidad

En la Figura 4.28 se muestran las gráficas resultantes del experimento. En la Figura 4.28(a) se muestra la trayectoria de la motocicleta y del vehículo durante la prueba, mediante la posición GPS de la ruta seguida, con puntos de referencia en los mismos instantes. En la Figura 4.28(b) se representa la distancia existente entre ambos vehículos, siendo positiva

si la motocicleta se encuentra en frente del vehículo y negativa si está detrás de este, en esta gráfica se puede observar cuando el sistema avisa al conductor de la presencia de la motocicleta, cómo se explicó anteriormente el sistema avisa cuando la motocicleta se encuentra a menos de 150 metros y se aproxima al vehículo. En la Figura 4.28(c) se muestra el tiempo que existe hasta que los dos vehículos se crucen y en la Figura 4.28(d) se representa la velocidad de ambos vehículos. Las gráficas de la distancia, tiempo de cruce y velocidad están estrechamente relacionadas.

La advertencia que se transmite al conductor del vehículo produce que este reduzca la velocidad y se acerque a la parte derecha de la carretera, y de esta manera evitará un posible impacto con la motocicleta, al aumentar la separación lateral. También, se puede ver que la moto entra campo visual del conductor del vehículo cuando la distancia entre ellos es de sólo 10 metros (T2), lo que demuestra la eficacia del sistema y la ventaja de tener un horizonte visual ampliado gracias a los módulos DSRC-UPM-INSIA capaz de generar advertencias de seguridad.

FIGURA 4.29: Secuencia de imágenes de cruce

Por último, en la Figura 4.29 se muestra la secuencia de imágenes que representa esta prueba, donde los vehículos circulan en sentidos opuestos en una curva con baja visibilidad. En la segunda imagen se presenta en el instante en que la motocicleta no es visible, pero el sistema de advertencia informa al conductor sobre su presencia en las proximidades del vehículo.

Capítulo 5

CONCLUSIONES Y TRABAJOS FUTUROS

Con la realización del presente documento se concluye el periodo de investigación del programa de doctorado, en este último capítulo está centrado en las conclusiones obtenidas durante la realización de esta tesis doctoral y las líneas futuras de investigación y trabajos que se podrían derivar como consecuencia de la realización de esta tesis doctoral.

5.1. Conclusiones

A lo largo del presente documento se han ido cumpliendo las premisas planteadas al inicio del doctorado: solucionar la problemática que hay en el uso de VANETs en el desarrollo de sistemas cooperativos. Para lo cual había que solucionar problemas tanto a nivel hardware como software, mediante el desarrollo de una tecnología propia de comunicaciones que cumpliera todos los requisitos de estandarización y que además fuera robusta, fiable, interoperable y de bajo coste.

En la memoria se han explicado los retos planteados para la consecución de la tesis en el capítulo 1, como se encontraba la tecnología y las investigaciones precedentes en el capítulo 2. En el capítulo 3: “COMUNICACIONES V2X: METODOLOGÍA Y APORTACIONES” se ha explicado extensamente los pasos seguidos para la consecución de los objetivos marcados en el inicio de la tesis, así como el cambio de tecnología efectuado en el transcurso de la misma. En el capítulo 4: “APLICACIONES DE SISTEMAS COOPERATIVOS” se describen extensamente las aplicaciones ADAS más significativas desarrolladas a lo largo de este periodo.

Durante el periodo de la tesis se utilizaron diferentes tecnologías, esto fue debido a que cuando se inició ésta, la tecnología específica de soporte de IEEE 802.11p aún no estaba disponible comercialmente y no había capacidad para desarrollarla por nuestra cuenta. Por esto se decidió abordar el tema mediante el uso de tecnología WSAW la cual usa el estándar IEEE 802.15.4 y posteriormente cuando salieron al mercado las primeras tarjetas de comunicación que soportaban el estándar IEEE 802.11p se decidió migrar a esta para crear nuestros módulos DSRC-UPM-INSIA, que era un objetivo del doctorado.

Durante el periodo de la tesis que se estuvo utilizando la tecnología WSAW, nos centramos en el desarrollo de algoritmos de enrutamiento geográfico, esta decisión nos ayudó a identificar la problemática de la comunicación vehicular, continuos cambios en la red, identificación de bucles, problemas con la saturación radioeléctrica, etc. Todo este esfuerzo terminó con el desarrollo de un algoritmo de enrutamiento geográfico en árbol llamado GCTP, el cual es robusto y que se amolda ágilmente a las circunstancias de la vía, teniendo tasas de pérdidas de paquetes muy aceptables para este tipo de redes de comunicaciones, 5%.

El algoritmo GCTP se ha utilizado en diversas aplicaciones de seguridad vial, donde se puede destacar:

- Una aplicación desarrollada junto al grupo LSI de la Universidad Carlos III de Madrid de reconocimiento de obstáculos en la vía mediante un telémetro láser y una cámara de infrarrojos, la información de la posición de estos obstáculos se transmiten mediante el algoritmo GCTP usando tecnología WSAW a otro vehículo para que actúe

en consecuencia y así evitar un posible accidente. A esta aplicación se la denominó “*Sistema Integrado de evitación automática de colisiones*” y gracias a la cual fuimos galardonados en el año 2014 con el “*Premio a la investigación en el campo de la automoción*” de la *Fundación Eduardo Barreiros*.

- Usando el algoritmo GCTP y una aplicación para la localización de vehículos se hizo una demostración del sistema en el congreso *IEEE Intelligent Vehicles Symposium (IV)* del año 2012 en Alcalá de Henares.

Con la aparición de tarjetas de comunicaciones capaces de soportar el estándar ETSI ITS-G5, la liberación del estándar provisional de geocomunicaciones por parte de la ETSI y dadas las limitaciones de computo, ancho de banda y transferencia de datos detectadas en los módulos WSA se decidió buscar una posible alternativa a estos. Tras adquirir un par de tarjetas de comunicaciones y ver que era posible la comunicación entre ambas se buscó un módulo capaz de contenerlas, creándose los módulos DSRC-UPM-INSIA.

Durante el desarrollo de estos módulos DSRC-UPM-INSIA, se ha indagado acerca de la implementación de las normas de comunicación en módulos comerciales. Por ello se ha tenido que profundizar en el estudio de la normativa de la familia ETSI EN 302 636 la cual abarca las geocomunicaciones vehiculares. Durante el desarrollo e implantación de estas se publicaron las guías definitivas las cuales tenían cambios sustanciales por lo cual hubo que comenzar de nuevo con todos los desarrollos llevados hasta esa fecha. Durante el estudio y desarrollo de los estándares para su posterior implementación se fueron detectando fallos y falta de concreción en los mismos, los cuales se reportaron a ETSI. Dado que no ha habido contestación por parte de ETSI y a fecha de 1 de diciembre de 2015 no hay una nueva versión hemos tenido que implementar nuestras soluciones para que fueran operativos. Para intentar que estos algoritmos desarrollados en base a la normativa de ETSI fueran lo más exportables posibles se decidió realizarlos utilizando *Java SDK* por sus cualidades de multiparadigma y multiplataforma.

Una vez desarrollados e implementados los estándares de geocomunicaciones de la capa de red nos encontramos con la problemática de conseguir que los algoritmos desarrollados pudieran acceder al nivel de enlace de datos sin intermediario. Tras barajar varias opciones y dadas las limitaciones de *Java* para desarrollar el nivel de enlace, se tuvo que abandonar el lenguaje de programación *Java* y desarrollarlo mediante C++ el cual cumple los principios del multiparadigma y multiplataforma pero se tiene que compilar para cada máquina, lo que no pasa con *Java*.

Una vez desarrollados en los módulos DSRC-UPM-INSIA los niveles de enlace de datos y de red se tenía que testear el correcto funcionamiento de todas las características implementadas en estos para ello hubo que desarrollar una aplicación sencilla a tal fin.

Gracias a esta se ha podido comprobar que el sistema es robusto y fiable, las pérdidas de paquetes rondan el 0.5 % en posiciones estáticas y un 2 % cuando todos los módulos están en movimiento y los tiempos de latencia muy ajustados de 5 milisegundos por salto.

Una vez comprobado que sus niveles de enlace de datos y de red eran correctos se procedió a implementar el nivel de transporte que es reflejado en ETSI EN 302 636-5-1 (636-5-1, 2014). Posteriormente se procedió a la implementación de todos los niveles definidos por los estándares ETSI se procedió al desarrollo de una aplicación ADAS sencilla de testeo. Esta aplicación consistía en reflejar los vehículos de alrededor de otro vehículo, al intentar implementarla nos dimos cuenta que para conseguir tal fin había que inundar la red de mensajes del tipo TSB para dar conocimiento a todos los vehículos colindantes de la posición de cada vehículo, lo cual conllevaría la saturación de la VANET creada. Después del análisis de varias opciones y conociendo la arquitectura del sistema implementado donde existe una base de datos con la información de todos los módulos colindantes, se decidió crear un nuevo módulo que fuera capaz de acceder a esta base de datos e hiciera de intermediario para las aplicaciones. De esta manera se creó el GNES (*Geonetworking Extension and Support*), esta decisión se tomó para evitar que muchas aplicaciones accedieran simultáneamente a la capa de red y la saturaran. Con esta decisión la aplicación inicial de prueba no necesitaba inundar de mensajes simplemente hacer una petición al GNES.

Una vez desarrollada la aplicación inicial para utilizar con los módulos DSRC-UPM-INSIA, se comprobó que era necesario ampliar las características del GNES haciendo que emita cada cierto tiempo un mensaje de tipo SHB, esto se realiza para que el resto de módulos actualice con más frecuencia la base de datos, ya que el periodo de actualización definido por el estándar es insuficiente para un sistema ADAS. Posteriormente se ha dado servicio a sistemas sin conectividad IEEE 802.11p la cual se proporciona a través de GNES por lo cual se ha implementado una base de datos interna en GNES para almacenar los datos de estos usuarios y compartirlos con otros módulos, aumentando de esta forma la información accesible y la capacidad de creación de aplicaciones cooperativas.

Con toda la arquitectura desarrollada se pasó a implementar aplicaciones ADAS que den servicio a usuarios vulnerables de la vía, peatones, ciclistas y motoristas. Para lo cual se ha desarrollado una aplicación específica para cada tipo, obteniéndose unos resultados notables que pueden ayudar a la prevención de accidentes.

- Aviso a peatones en vías urbanas. Es una aplicación para smartphone que gracias al GNES conoce la posición y trayectoria de todos los vehículos colindantes. Con esta información puede avisar al peatón de un potencial peligro.

- *BikeWarn*. Es una aplicación que avisa al conductor de un vehículo de la presencia de una bicicleta en los alrededores gracias a unos *iBeacons* instalados en las bicicletas.
- *MotoWarn*. Es un sistema que alerta al conductor de un vehículo de la presencia de una motocicleta mostrándole mediante una interfaz gráfica, sencilla y no invasiva, la posición relativa de la motocicleta respecto al vehículo.

En el documento “*Guide about technologies for future C-ITS service scenarios*” (ERTICO, 2015) se sugiere la introducción de la tecnología móvil G5 como solución tecnológica para las VANETs descartando la tecnología DSRC, este impulso en esta dirección es para que las compañías de telecomunicaciones participen en el desarrollo de los sistemas cooperativos. A nuestro parecer esta estrategia no es la adecuada porque el uso único de tecnología G5 provoca la dependencia de un elemento externo para establecer las comunicaciones entre vehículos mientras que usando DSRCs no es necesario ningún agente externo. La implantación de las antenas necesarias para dar servicio será muy costosa para poder dimensionar la red para dar servicio a los móviles de los usuarios, los vehículos y todos los elementos conectados, la ventaja de los DSRC es que no se necesita una estructura fija y que la red se autorregula. En el documento dicen que la latencia será inferior a 15ms y de un rango de 300-500m, se ha demostrado a lo largo de esta tesis que los módulos DSRC-UPM-INSIA igualan e incluso mejoran estos números. Por lo cual no estamos nada de acuerdo con esta filosofía y se defenderá la utilización de VANETs abiertas y gratuitas para la creación de sistemas cooperativos que ayuden a cumplir con los objetivos de reducción de la siniestralidad en las carreteras y ciudades, y la reducción de la contaminación.

5.2. Aportaciones fundamentales

A continuación se enumeran las principales aportaciones que se han conseguido durante el doctorado.

- Se ha creado un novedoso algoritmo para la recolección de la información de los vehículos capaz de adaptarse a los cambios de las características del tráfico.
- Se ha analizado pormenorizadamente las normas de comunicaciones entre vehículos recientemente publicadas por la ETSI para su posterior implementación. En este proceso se han descubierto erratas y fallos que han sido comunicados al organismo correspondiente.
- Se ha desarrollado un sistema de comunicación intervehicular, tanto a nivel hardware como software. Este sistema de comunicaciones es de reducidas dimensiones y de bajo costo, ideal para la implantación en el actual parque automovilístico europeo.
- Se ha demostrado que la tecnología está lo suficientemente madura para empezar a la implantación de sistemas de comunicación en vehículos.
- Se ha comprobado que los sistemas cooperativos son un potencial medio para reducir la siniestralidad en las carreteras y ciudades.

5.3. Trabajos futuros

Para terminar con esta última sección de la tesis doctoral se enumerarán una serie de líneas de investigación que se deberían explorar para seguir avanzando en los sistemas cooperativos y la creación de VANETs.

- **Implementación de los estándares de seguridad:** Dado que durante la realización de esta tesis los estándares de seguridad definidos en los documentos seguidos no eran definitivos y ya ocurrió que entre los provisionales y los definitivos había mucha diferencia, se decidió que no se implementaran en los módulos DSRC-UPM-INSIA. Por ello en el momento que el estándar definitivo de seguridad sea publicado debería implementarse con la mayor brevedad posible.
- **Interoperabilidad:** Dado que no hay muchos sistemas DSRC accesibles en el mercado que sigan los estándares de la ETSI no se ha podido comprobar la interoperabilidad de los módulos DSRC-UPM-INSIA. Por lo cual sería interesante comprobar si estos son capaces de comunicarse con otros módulos DSRC y solucionar los problemas si los hubiera.
- **Vehículos autónomos:** Uso de los módulos desarrollados en vehículos autónomos para el control de los parámetros de estos vehículos a distancia, teleguiado o en sistemas cooperativos como pueden ser trenes de carretera.
- **Desarrollo de nuevos ADAS:** Usar los módulos implantados en vehículos como herramienta para la creación de nuevos ADAS, algunas ideas preliminares sin haber estudiado a fondo la viabilidad de estas podrían ser: asistentes de velocidad adecuada a la vía en función de las condiciones meteorológicas, aviso de accidente, aviso del estado del tráfico.
- **Implantación de RSU en túneles:** Si se implantaran los módulos DSRC-UPM-INSIA a modo de RSU en los diferentes túneles de la red vial se podría dar cobertura de geoposicionamiento a todos los vehículos que circularan por ellos. Gracias a este proyecto se estudiaría el comportamiento de los módulos en condiciones especiales, se podría estudiar si las condiciones del túnel influyen en el rango de transmisión, la latencia y descubrir problemas asociados.

Bibliografía

- 636-1, E. E. . (2014). *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 1: Requirements*. European Telecommunications Standards Institute, 650 Route des Lucioles F-06921 Sophia Antipolis Cedex - FRANCE, v1.2.1 edition.
- 636-2, E. E. . (2007). *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 2: Scenarios*. European Telecommunications Standards Institute, 650 Route des Lucioles F-06921 Sophia Antipolis Cedex - FRANCE, v1.2.1 edition.
- 636-3, E. E. . (2014). *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 3: Network Architecture*. European Telecommunications Standards Institute, 650 Route des Lucioles F-06921 Sophia Antipolis Cedex - FRANCE, v1.2.1 edition.
- 636-4-1, E. E. . (2014). *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality*. European Telecommunications Standards Institute, 650 Route des Lucioles F-06921 Sophia Antipolis Cedex - FRANCE, v1.2.1 edition.
- 636-5-1, E. E. . (2014). *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 5: Transport Protocols; Sub-part 1: Basic Transport Protocol*. European Telecommunications Standards Institute, 650 Route des Lucioles F-06921 Sophia Antipolis Cedex - FRANCE, v1.2.1 edition.
- 663, E. E. . (2009). *Intelligent Transport Systems (ITS); European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band*. European Telecommunications Standards Institute, 650 Route des Lucioles F-06921 Sophia Antipolis Cedex - FRANCE, v1.1.0 edition.
- 931, E. E. . (2014). *Intelligent Transport Systems (ITS); Vehicular Communications; Geographical Area Definition*. European Telecommunications Standards Institute, 650 Route des Lucioles F-06921 Sophia Antipolis Cedex - FRANCE, v1.1.1 edition.

- Agency, N. G.-I. (2014). Office of geomatics: World geodetic system 1984 (wgs 84). <http://earth-info.nga.mil/GandG/wgs84/>.
- Al-Sultan, S., Al-Doori, M. M., Al-Bayatti, A. H., and Zedan, H. (2014). A comprehensive survey on vehicular ad hoc network. *Journal of Network and Computer Applications*, 37:380–392.
- Anaya, J., Merdrignac, P., Shagdar, O., Nashashibi, F., and Naranjo, J. (2014). Vehicle to pedestrian communications for protection of vulnerable road users. In *Intelligent Vehicles Symposium Proceedings*, pages 1037–1042. IEEE.
- Aparicio, F., Arenas, B., Gómez, A., Jiménez, F., López, J., Martínez, L., and Páez, J. (2008). *Ingeniería del Transporte*. CIE.
- Bachir, A. and Benslimane, A. (2003). A multicast protocol in ad hoc networks inter-vehicle geocast. In *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, volume 4, pages 2456–2460 vol.4.
- Benkic, K., Malajner, M., Planinsic, P., and Cucej, Z. (2008). Using rssi value for distance estimation in wireless sensor networks based on zigbee. In *Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on*, pages 303–306.
- Bhoi, S. and Khilar, P. (2014). Ijs: An intelligent junction selection based routing protocol for vanet to support its services. *nternational Scholarly Research Notices*, 2014:14.
- Biswas, S., Tatchikou, R., and Dion, F. (2006). Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety. *IEEE Communications Magazine*, 44(1):74–82.
- Blum, J., Eskandarian, A., and Hoffman, L. (2003). Mobility management in ivc networks. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pages 150–155.
- Boleng, J., Camp, T., and Tolety, V. (2001). Mesh-based geocast routing protocols in an ad hoc network. In *Proceedings of the 15th International Parallel & Distributed Processing Symposium, IPDPS '01*, pages 184–, Washington, DC, USA. IEEE Computer Society.
- Briesemeister, L., Schafers, L., Hommel, G., and Ag, D. (2000). Disseminating messages among highly mobile hosts based on inter-vehicle communication. In *In IEEE Intelligent Vehicles Symposium*, pages 522–527.
- Brooke, J. et al. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7.
- Broughton, J. (2003). The benefits of improved car secondary safety. *Accident Analysis & Prevention*, 35(4):527 – 535.

- Camp, T. and Liu, Y. (2003). An adaptive mesh-based protocol for geocast routing. *Journal of Parallel and Distributed Computing*, 63(2):196 – 213. Routing in Mobile and Wireless Ad Hoc Networks.
- Celes, C., Braga, R., de Oliveira, C., Andrade, R., and Loureiro, A. (2013). Geospin: An approach for geocast routing based on spatial information in vanets. In *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*, pages 1–6.
- Chauvel, C. and Haviotte, C. (2010). Automatic emergency calls in france. In *4th International Conference ESAR (Expert Symposium on Accident Research)*.
- Coates, M., Nowak, R., et al. (2000). Network loss inference using unicast end-to-end measurement. In *Proc. ITC Conf. IP Traffic, Modeling and Management*, pages 28–1.
- connected Vehicles, I. (30/09/2015). First toyota cars to include v2v and v2i communication by the end of 2015. <http://sites.ieee.org/connected-vehicles/2015/09/30/first-toyota-cars-to-include-v2v-and-v2i-communication-by-the-end-of-2015/>.
- Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1.
- DGT (25-11-2015). Comobity, la app “segura” para ciclistas. <http://revista.dgt.es/es/reportajes/2015/11NOVIEMBRE/1113Comobity-la-App-segura-para-ciclistas.shtml#.Vx22vIR94Xo>.
- Dobbs, R. (2013). *Infrastructure productivity: how to save 1 trillion dolar a year*. McKinsey.
- Doshi, A. and Trivedi, M. (2010). Communicating driver intents: A layered architecture for cooperative active safety applications. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 373–378.
- Durresi, M., Durresi, A., and Barolli, L. (2005). Emergency broadcast protocol for inter-vehicle communications. In *Parallel and Distributed Systems, 2005. Proceedings. 11th International Conference on*, volume 2, pages 402–406.
- ERTICO (2015). Guide about technologies for future c-its service scenarios. Technical Report Final report, ITS Europe.
- ERTICO (2016). Connected and automated driving (c-its). Technical Report Final report, European Union.
- europaean Commission (2015). ecall: Time saved = lives saved. <https://ec.europa.eu/digital-single-market/en/ecall-time-saved-lives-saved>.

- F. García; F. Jiménez; J.E. Naranjo; J.G. Zato; F. Aparicio; J.M. Armingol; de la Escalera, A. (2012). Environment perception based on lidar sensors for real road applications. *Robotica*, 30:185–193.
- Fang, S. and Luo, T. (2013). A novel two-timer-based broadcast routing algorithm for vehicular ad-hoc networks. In *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCoM), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, pages 1518–1522.
- Festag, A. (2014). Cooperative intelligent transport systems standards in europe. *Communications Magazine, IEEE*, 52(12):166–172.
- for Information technology, I. S. (2006). 802.15.4-2006 - ieee standard for information technology– local and metropolitan area networks– specific requirements– part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (wpans). Technical report, IEEE.
- for Information technology, I. S. (2010). 802.11p local and metropolitan area networks– specific requirements– part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Wireless access in vehicular environments. Technical report, IEEE.
- for life, V. (19-12-2014). Volvo cars and poc to demonstrate life-saving wearable cycling tech concept at international ces 2015. <https://www.media.volvocars.com/us/en-us/media/pressreleases/155565/volvo-cars-and-poc-to-demonstrate-life-saving-wearable-cycling-tech-concept-at-international-ces-201>.
- García, F., Jiménez, F., Anaya, J. J., Armingol, J. M., Naranjo, J. E., and de la Escalera, A. (2013). Distributed pedestrian detection alerts based on data fusion with accurate localization. *Sensors*, 13(9):11687.
- García, F., Jiménez, F., Puertas, E., Naranjo, J. E., Armingol, J. M., and Fernández, J. (2013). Pre-collision systems for urban environment accidents avoidance. *Securitas Vialis*, pages 1–9.
- Garcia, F., Musleh, B., de la Escalera, A., and Armingol, J. (2011). Fusion procedure for pedestrian detection based on laser scanner and computer vision. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1325–1330.
- Geyer, S., Hakuli, S., Winner, H., Franz, B., and Kauer, M. (2011). Development of a cooperative system behavior for a highly automated vehicle guidance concept based on the conduct-by-wire principle. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 411–416.

- Handmann, U., Kalinke, T., Tzomakas, C., Werner, M., and Seelen, W. (2000). An image processing system for driver assistance. *Image and Vision Computing*, 18(5):367 – 376.
- Honda (2013). Honda demonstrates advanced vehicle-to-pedestrian. <http://www.hondanews.ca/en/news/release/Honda-Demonstrates-Advanced-VehicletoPedestrian-and-VehicletoMotorcycle-Safety-Technologies->---
- Huang, Q., Miller, R., McNeille, P., Dimeo, D., and Roman, G.-C. (2002). Development of a peer-to-peer collision warning system. *Ford Techincal Journal*, 5(2).
- Jerbi, M., Senouci, S.-M., Rasheed, T., and Ghamri-Doudane, Y. (2009). Towards efficient geographic routing in urban vehicular networks. *Vehicular Technology, IEEE Transactions on*, 58(9):5048–5059.
- Jiménez, F. (Madrid, 2012). *Libro Verde de los Sistemas Cooperativos*. ITS España.
- Jiménez, F., Liang, Y., and Aparicio, F. (2012). Adapting isa system warnings to enhance user acceptance. *Accident Analysis & Prevention*, 48:37–48.
- Johnson, D. and Maltz, D. (1996). Dynamic source routing in ad hoc wireless networks. In Imielinski, T. and Korth, H., editors, *Mobile Computing*, volume 353 of *The Kluwer International Series in Engineering and Computer Science*, pages 153–181. Springer US.
- Karp, B. and Kung, H. T. (2000). Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom '00, pages 243–254, New York, NY, USA. ACM.
- Ko, Y.-B. and Vaidya, N. (1999). Geocasting in mobile ad hoc networks: location-based multicast algorithms. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pages 101–110.
- Korkmaz, G., Ekici, E., Özgüner, F., and Özgüner, U. (2004). Urban multi-hop broadcast protocol for inter-vehicle communication systems. In *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, VANET '04, pages 76–85, New York, NY, USA. ACM.
- Krajzewicz, D., Erdmann, J., Behrisch, M., and Bieker, L. (2012). Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138.
- Lagana, M., Feiri, M., Sall, M., Lange, M., Tomatis, A., and Papadimitratos, P. (2012). Secure communication in vehicular networks-preserve demo. *IEEE Communications Society*.
- Li, F. and Wang, Y. (2007). Routing in vehicular ad hoc networks: A survey. *Vehicular Technology Magazine, IEEE*, 2(2):12–22.

- Liao, W.-H., Sheu, J.-P., and Tseng, Y.-C. (2001). Grid: A fully location-aware routing protocol for mobile ad hoc networks. *Telecommunication Systems*, 18(1-3):37–60.
- Liao, W.-H., Tseng, Y.-C., Lo, K.-L., and Sh, J.-P. (2000). Geogrid: A geocasting protocol for mobile ad hoc networks based on grid. *Journal of Internet Technology*, 1:23–32.
- Liu, G., Lee, B.-S., Seet, B.-C., Foh, C., Wong, K., and Lee, K.-K. (2004). A routing strategy for metropolis vehicular communications. In *International Conference on Information Networking (ICOIN)*, page 134–143.
- Lochert, C., Hartenstein, H., Tian, J., Fussler, H., Hermann, D., and Mauve, M. (2003). A routing strategy for vehicular ad hoc networks in city environments. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pages 156–161.
- Lochert, C., Mauve, M., Füßler, H., and Hartenstein, H. (2005). Geographic routing in city scenarios. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(1):69–72.
- Maihofer, C. (2004). A survey of geocast routing protocols. *Commun. Surveys Tuts.*, 6(2):32–42.
- Maihofer, C. and Eberhardt, R. (2004). Geocast in vehicular environments: caching and transmission range control for improved efficiency. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 951–956.
- Maihöfer, C., Leinmüller, T., and Schoch, E. (2005). Abiding geocast: Time-stable geocast for ad hoc networks. In *Proceedings of the 2Nd ACM International Workshop on Vehicular Ad Hoc Networks, VANET '05*, pages 20–29, New York, NY, USA. ACM.
- Masvidal, D. G. (2015). Detección de usuarios vulnerables de la carretera mediante smartphones. Trabajo final de grado sin publicar.
- McCall, J. and Trivedi, M. (2006). Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *Intelligent Transportation Systems, IEEE Transactions on*, 7(1):20–37.
- Mehar, S., Senouci, S., and Remy, G. (2012). Dissemination protocol for heterogeneous cooperative vehicular networks. In *Wireless Days (WD), 2012 IFIP*, pages 1–6.
- Moriarty, D. E. and Langley, P. (1998). Learning cooperative lane selection strategies for highways. In *AAAI/IAAI*, pages 684–691. Citeseer.
- Namboodiri, V., Agarwal, M., and Gao, L. (2004). A study on the feasibility of mobile gateways for vehicular ad-hoc networks. In *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks, VANET '04*, pages 66–75, New York, NY, USA. ACM.

- Navas, J. C. (2001). *Geographic Routing in a Datagram Internetwork*. PhD thesis, Rutgers, The State University Piscataway, NJ. AAI3027993.
- Navas, J. C. and Imielinski, T. (1997). Geocast—geographic addressing and routing. In *Proceedings of the 3rd Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom '97, pages 66–76, New York, NY, USA. ACM.
- Ni, S.-Y., Tseng, Y.-C., Chen, Y.-S., and Sheu, J.-P. (1999). The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom '99, pages 151–162, New York, NY, USA. ACM.
- Nissan Motor CO., L. (07/05/2015). Sky project - intersection collision avoidance. http://wiki.fot-net.eu/index.php/SKY_Project_-_Intersection_Collision_Avoidance.
- nxp (2015). Wireless connectivity for automotive. <http://www.nxp.com/products/automotive-products/interface-and-connectivity/wireless-connectivity:WIRELESS-CONNECTIVITY-AUTO>.
- Olaverri-Monreal, C., Gomes, P., Fernandes, R., Vieira, F., and Ferreira, M. (2010). The see-through system: A vanet-enabled assistant for overtaking maneuvers. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 123–128.
- Ooi, C.-C. and Fisal, N. (2004). Implementation of geocast-enhanced aodv-bis routing protocol in manet. In *TENCON 2004. 2004 IEEE Region 10 Conference*, volume B, pages 660–663 Vol. 2.
- Papadimitratos, P., La Fortelle, A., Evenssen, K., Brignolo, R., and Cosenza, S. (2009). Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation. *Communications Magazine, IEEE*, 47(11):84–95.
- para la Seguridad del Automóvil, F. I. T. (2007). *Colección 2006: Evidencias científicas de la eficacia de las Tecnologías*. Editado por FITSA.
- Patrascu, D. (2010). New ivc unit from oki. <http://www.autoevolution.com/news/new-ivc-unit-from-ok-27213.html>.
- Pérez-Moreno, E., Hernández-Lloreda, M. J., Gallego-Largo, T. R., and Castellanos, M. Á. (2015). Impulsive driving: definition and measurement using the i-driving scale (ids). *The Spanish journal of psychology*, 18:E93.
- Perkins, C. and Royer, E. (Feb. 1999). Ad-hoc on demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, page 90–100.

- Qualcomm (2015). Advancing telematics and v2x for tomorrow. <https://www.qualcomm.com/products/automotive/telematics>.
- Rao, S. D. and Murthy, C. S. R. (2005). Distributed dynamic qos-aware routing in wdm optical networks. *Computer Networks*, 48(4):585 – 604.
- Rooke, A. (2015). About heero. <http://www.heero-pilot.eu/view/en/heero.html>.
- Saleet, H., Langar, R., Naik, K., Boutaba, R., Nayak, A., and Goel, N. (2011). Intersection-based geographical routing protocol for vanets: A proposal and analysis. *Vehicular Technology, IEEE Transactions on*, 60(9):4560–4574.
- Santos, R. A., Edwards, A., Edwards, R. M., and Seed, N. L. (2005). Performance evaluation of routing protocols in vehicular ad-hoc networks. *Int. J. Ad Hoc Ubiquitous Comput.*, 1(1/2):80–91.
- Stojmenovic, I., Ruhil, A., and Lobiyal, D. (2003). Voronoi diagram and convex hull based geocasting and routing in wireless networks. In *Computers and Communication, 2003. (ISCC 2003). Proceedings. Eighth IEEE International Symposium on*, pages 51–56 vol.1.
- Sun, M.-T., chi Feng, W., Lai, T.-H., Yamada, K., Okada, H., and Fujimura, K. (2000). Gps-based message broadcast for adaptive inter-vehicle communications. In *Vehicular Technology Conference, 2000. IEEE-VTS Fall VTC 2000. 52nd*, volume 6, pages 2685–2692 vol.6.
- Tan, H.-S. and Huang, J. (2006). Dgps-based vehicle-to-vehicle cooperative collision warning: Engineering feasibility viewpoints. *Intelligent Transportation Systems, IEEE Transactions on*, 7(4):415–428.
- Tinyos (2015). Collection. [Internet; ultima consulta agosto-2015].
- tno (2015). Safe mobility. <https://www.tno.nl/en/focus-area/urbanisation/mobility-logistics/safe-mobility/>.
- Tonguz, O., Wisitpongphan, N., and Bai, F. (2010). Dv-cast: A distributed vehicular broadcast protocol for vehicular ad hoc networks. *Wireless Communications, IEEE*, 17(2):47–57.
- Unex (2015). Unex: V2x and v2x fusion solution. <http://www.unex.com.tw/v2x>.
- Viriyasitavat, W., Bai, F., and Tonguz, O. (2010). Uv-cast: An urban vehicular broadcast protocol. In *Vehicular Networking Conference (VNC), 2010 IEEE*, pages 25–32.
- Zijlstra, F. R. H. (1993). *Efficiency in work behaviour: A design approach for modern tools*. TU Delft, Delft University of Technology.

Apéndice A

Estándares de comunicación

A.1. Estándar de GeoNetworking

En el estándar ETSI EN 302 636-4-1 636-4-1 (2014) es donde se describe el funcionamiento de GeoNetworking. A continuación se encuentra un resumen de este.

A.1.1. Elementos del estándar

A continuación se hace una breve descripción de los elementos básicos de GeoNetworking, así como sus características y usos particulares.

A.1.1.1. “*GeoNetworking address format*”

En las redes de comunicación usuales la forma de identificar a las maquinas es mediante un número de identificación único, “Control de Acceso al Medio”, MAC por sus siglas en inglés: “Media Access Control”, es un identificador de 48 bits que corresponde de forma única a una tarjeta o dispositivo de red. En GeoNetworking aunque esto sería suficiente para identificar cada módulo, se podría llegar a saturar el sistema de asignación de MAC’s, dado el elevado número de elementos susceptibles de portar un módulo ITS, por lo cual se ha tomado la decisión de incluir nuevas variables que hagan que la duplicidad de identificadores prácticamente sea nula. Para ello se ha definido el “*GeoNetworking address format*” (GNADDR), se pueden observar sus campos en la Figura A.1.

Donde los campos de la Figura A.1 son:

- M: indica si GNADDR ha sido creado automáticamente o manualmente.

- ST: es un identificador del tipo de módulo ITS (*Roadside*, peatón, bicicleta, ciclomotor, motocicleta, vehículo de pasajeros, autobús, camión ligero, camión pesado, tráiler, vehículo especial o tranvía).
- SCC: es el código del país al que pertenece el módulo ITS.
- MID: es el número MAC de la tarjeta de red que se utiliza para acceder a la red de GeoNetworking.

GeoNetworking necesita una forma precisa de conocer la posición exacta en el globo terráqueo de todos los elementos que componen la red de comunicación para poder determinar dónde está el propio módulo y el resto de módulos para poder crear los caminos de enrutamiento. Para ello los módulos tienen que retransmitir de alguna manera los datos de posicionamiento, para tal menester se utiliza el “*Short Position Vector*” (SPV) ver la Figura A.2 y el “*Long Position Vector*” (LPV) ver la Figura A.3.

	0								1								2								3							
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	GN_ADDR																															
4																																
8																																
12																																
16	TST																															
	Lat																															
	Long																															

Donde los campos de la Figura A.2 son:

- A- II

FIGURA A.3: *Long Position Vector*

Donde los campos de la Figura A.3 son:

- SPV: son los campos del “*Short Position Vector*”.
- PAI: indica si los datos contenidos en el paquete tienen validez dependiendo la precisión del GPS. (0 datos no validos por insuficiencia de precisión, 1 datos correctos).
- S: es la velocidad medida en $\frac{cm}{s}$.
- H: es el ángulo azimutal respecto del Norte medido en $\frac{1}{10}^{\circ}$.

A.1.1.3. Almacenamiento de datos

Como todo protocolo de red, GN necesita tener conocimiento del medio que le rodea por lo cual tiene una serie de bases de datos y buffers donde almacenar la información.

A.1.1.3.1. “*Location Table*”

Las características del estándar obligan a que cada módulo tenga una base de datos de los módulos conocidos y que este actualizada, esta base de datos se la conoce como “*Location Table*” (LocT). La LocT es esencial para el funcionamiento de los módulos pues se basan en ella para el enrutamiento de paquetes en la red, si la LocT esta desactualizada o con datos erróneos se producen problemas de comunicación dejando inservible la red.

Como se ha comentado el mantenimiento de esta base de datos es crucial por lo cual cada cierto periodo de tiempo se recorre completamente comprobando la última actualización de los datos y si el tiempo transcurrido desde la recepción de dichos datos y el actual sobrepasa un límite este módulo es eliminado de la LocT. El límite es una constante descrita en 636-4-1 (2014).

A.1.1.3.2. “*Local Position Vector*”

El “*Local Position Vector*” (LocPV) es una parte esencial de GN, es donde se almacenan todos los datos referidos al posicionamiento geográfico del módulo. Todos los submódulos

de GN acceden a esta información, por lo cual tiene que ser flexible, rápido y eficiente a la hora de utilizarlo, tanto para consultas de los datos como para actualización de estos.

A.1.1.3.3. “*Buffers*”

También existen diferentes *buffers* donde almacenar paquetes que por diferentes causas no se pueden tratar en ese preciso instante para no bloquear el protocolo y que pueda ir atendiendo diferentes peticiones. Estos *buffers* están situados en los lugares críticos del estándar, están implementados con una filosofía FIFO, y son de tamaño limitado para evitar que se necesite mucha memoria.

A.1.2. Cabeceras del estándar

Como se ha comentado anteriormente GeoNetworking sustituye al protocolo IP, por lo cual, como todo protocolo tiene sus cabeceras propias, en la Figura A.4 se puede observar como es un paquete retransmitido por una red de GN.

<i>MAC Header</i>	<i>LLC Header</i>	<i>GeoNetworking Header</i>	<i>Payload (optional)</i>

FIGURA A.4: Estructura de los paquetes GeoNetworking

En la Figura A.5 se puede observar como la cabecera de GeoNetworking se divide en tres subcabeceras, esto es debido a dos factores diferentes:

<i>MAC Header</i>	<i>LLC Header</i>	<i>Basic Header</i>	<i>Common Header</i>	<i>Extender Header (optional)</i>	<i>Payload (optional)</i>

FIGURA A.5: Estructura de los paquetes GeoNetworking con detalle de *GN Header*

- Al protocolo se le puede incorporar seguridad entonces las cabeceras se codifican como se puede ver en la Figura A.6 manteniéndose una parte sin codificar, esta es la “*Basic Header*” y el resto se codifica.

<i>MAC Header</i>	<i>LLC Header</i>	<i>Basic Header</i>	<i>Secured Packet with: Common Header, Extender Header (optional) and Payload (optional)</i>

FIGURA A.6: Estructura de los paquetes GeoNetworking con seguridad

- El protocolo puede comportarse de diferente manera como se ha visto en los apartados anteriores la información a enviar en los paquetes varía por lo cual hay que tener varios

paquetes diferentes, para ello se diferencian dos partes el “*Common Header*” que como indica es una parte común a todos los paquetes y el “*Extender Header*” que es la parte específica de los diferentes tipos de paquetes.

A.1.2.1. “Basic Header”

En la Figura A.7 se pueden observar los campos de la “Basic Header” del protocolo de GeoNetworking.

FIGURA A.7: Basic Header

Donde los campos de la Figura A.7 son:

- *Version*: hace referencia a la versión del protocolo de GeoNetworking.
- *NH*: indica el tipo de la siguiente cabecera contenida en el paquete podemos ver los posibles valores en la Tabla A.1.
- *Reserved*: es un campo reservado para futuras versiones del protocolo.
- *LT*: es el tiempo de validez del paquete antes de su eliminación por antiguo, se puede ver sus especificaciones en la Figura A.8.
- *RHL*: es un limitador de saltos, cuando el paquete pasa por un nodo se decrementa el valor, si llega a cero se descarta este.

TABLA A.1: Campo NH en *GeoNetworking Basic Header*

<i>Next Header (NH)</i>	<i>Encoding</i>	<i>Description</i>
ANY	0	<i>Unspecified</i>
<i>Common Header</i>	1	<i>GeoNetworking Common Header</i>
<i>Secured Packet</i>	2	<i>GeoNetworking Secured Packet</i>

La codificación del tiempo de validez del paquete no es lineal, para ello se ha codificado utilizando la ecuación A.1 en la Tabla A.2 podemos observar la codificación que sigue la base de tiempos.

$$LT = LT_{Multiplier} \cdot T_{Base} \quad (A.1)$$

0	1	2	3	4	5	6	7
<i>Multiplier</i> [0 to 63]						Base (ver tabla A.2)	

FIGURA A.8: Composición del campo LT

Donde los campos de la Figura A.8 son:

- *Multiplier*: es un número.
- Base: es en la base que se expresa el “*Multiplier*” (Ver Tabla A.2).

TABLA A.2: Codificación del campo LT

Valor	LT_{Base}
0	50 ms
1	1 s
2	10 s
3	100 s

A.1.2.2. “Common Header”

Como se ha comentado anteriormente la “*Common Header*” es una cabecera que aparece en todos los paquetes de datos de GeoNetworking, en la Figura A.9 se puede ver los campos de esta.

FIGURA A.9: *Common Header*

Donde los campos de la Figura A.9 son:

- NH: indica el tipo de la siguiente cabecera contenida en el payload de GeoNetworking los posibles valores están en la Tabla A.3.
- *Reserved*: es un campo reservado para futuras versiones del protocolo.
- HT: Identifica el tipo de paquete de GeoNetworking, los posibles valores están en la Tabla A.4.
- HST: Identifica el subtipo de paquete de GeoNetworking, los posibles valores están en la Tabla A.4.
- TC: representa los requerimientos del paquete en la capa de enlace.
- *Flags*: el bit 0 de este campo indica si el módulo es fijo o móvil.
- PL: es la longitud del Payload contenido en el paquete.
- MHL: representa el máximo número de saltos.

TABLA A.3: Campo NH en *GeoNetworking Common Header*

<i>Next Header (NH)</i>	<i>Encoding</i>	<i>Description</i>
ANY	0	<i>Unspecified.</i>
BTP-A	1	<i>Transport protocol BTP-A for interactive packet transport.</i>
BTP-B	2	<i>Transport protocol BTP-B for non-interactive packet transport.</i>
IPv6	3	<i>IPv6 header.</i>

TABLA A.4: GeoNetworking Header Types and Header Sub-Types

Header Type (HT)	Header Sub-type (HST)	Encoding	Description
ANY		0	Unspecified
	UNSPECIFIED	0	Unspecified
BEACON		1	Beacon
	UNSPECIFIED	0	Unspecified
GEOUNICAST		2	GeoUnicast
	UNSPECIFIED	0	Unspecified
GEOANYCAST		3	Geographically-Scoped Anycast (GAC)
	GEOANYCAST_CIRCLE	0	Circular area
	GEOANYCAST_RECT	1	Rectangular area
	GEOANYCAST_ELIP	2	Ellipsoidal area
GEOBROADCAST		4	Geographically-Scoped broadcast (GBC)
	GEOBROADCAST_CIRCLE	0	Circular area
	GEOBROADCAST_RECT	1	Rectangular area
	GEOBROADCAST_ELIP	2	Ellipsoidal area
TSB		5	Topologically-scoped broadcast (TSB)
	SINGLE_HOP	0	Single-hop broadcast (SHB)
	MULTI_HOP	1	Multi-hop TSB
LS		6	Location service (LS)
	LS_REQUEST	0	Location service request
	LS_REPLY	1	Location service reply

A.1.2.3. “GeoNetworking Extender Header”

En esta sección se describen brevemente los posibles paquetes que se utilizan en el protocolo de GeoNetworking que anteriormente se han listado en la Tabla A.4, indicando los campos que van en la cabecera “*GeoNetworking Extender Header*”.

A.1.2.3.1. GeoUnicast

En la Figura A.10 se pueden observar los campos adicionales que hay que rellenar para mandar un paquete de tipo GeoUnicast (GUC).

Donde los campos de la Figura A.10 son:

- SN: es el número de secuencia del paquete, cada paquete GUC debe tener un número único de identificador, sirve para detectar paquetes duplicados.
- *Reserved*: es un campo reservado para futuras versiones del protocolo.

FIGURA A.10: *GeoUnicast packet header*

- SO PV (LPV): es la posición del módulo emisor con un formato de “*Long Position Vector*” (Ver Fig. A.3).
- DE PV (SPV): es la posición del módulo destino con un formato de “*Short Position Vector*” (Ver Fig. A.2).

A.1.2.3.2. *Topological Scoped Broadcast*

En la Figura A.11 se pueden observar los campos adicionales que hay que rellenar para mandar un paquete de tipo “*Topological Scoped Broadcast*” (TSB). Este tipo de paquete están destinados a los n-vecinos colindantes.

FIGURA A.11: Topological Scoped Broadcast packet header

Donde los campos de la Figura A.11 son:

- SN: es el número de secuencia del paquete, cada paquete TSB debe tener un número único de identificador, sirve para detectar paquetes duplicados.
- *Reserved*: es un campo reservado para futuras versiones del protocolo.
- SO PV (LPV): es la posición del módulo emisor con un formato de “*Long Position Vector*” (Ver Fig. A.3).

A.1.2.3.3. *Single-Hop Broadcast packet header*

Anteriormente se han mencionado los paquetes del tipo “*Single-Hop Broadcast*” (SHB) son un tipo específico de TSB en el cual solo se avisa a los vecinos de alrededor en visión directa, por lo cual tiene su propio tipo de paquete que se puede observar en la Figura A.12.

FIGURA A.12: *Single-Hop Broadcast packet header*

Donde los campos de la Figura A.12 son:

- SO PV (LPV): es la posición del módulo emisor con un formato de “*Long Position Vector*” (Ver Fig. A.3).
- *Reserved*: es un campo reservado para operaciones del medio.

A.1.2.3.4. *Geographical Scoped AnyCast/BroadCast*

La cabecera que se observa en la Figura A.13 es la misma para paquetes del tipo “*Geographical Scoped AnyCast*” (GAC) y “*Geographical Scoped BroadCast*” (GBC), estos son paquetes destinados a hacer una difusión en un área determinada. En la sección A.1.5 se hace una explicación más detallada de las características de las áreas de difusión de información.

FIGURA A.13: *Geographical Scoped AnyCast/BroadCast packet header*

Donde los campos de la Figura A.13 son:

- SN: es el número de secuencia del paquete, cada paquete GAC/GBC debe tener un número único de identificador, sirve para detectar paquetes duplicados.
- *Reserved*: es un campo reservado para futuras versiones del protocolo.
- SO PV (LPV): es la posición del módulo emisor con un formato de “*Long Position Vector*” (Ver Fig. A.3).
- *GeoAreaPos latitude*: Es el valor de la Latitud en $\frac{1}{10}\mu^\circ$ del centro del área.
- *GeoAreaPos longitude*: Es el valor de la Latitud en $\frac{1}{10}\mu^\circ$ del centro del área.

- *Distance a*: es la distancia “a” en metros del área.
- *Distance b*: es la distancia “b” en metros del área.
- *Angle*: es el ángulo en grados medidos desde el Norte que tiene el área.

A.1.2.3.5. Beacon

En la figura A.14 se puede observar los campos de la cabecera “*Beacon*” es un tipo de paquete interno de GN que es enviado por los módulos para que el resto de módulos colindantes conozcan la existencia de este.

FIGURA A.14: Beacon packet header

Donde los campos de la Figura A.14 son:

- SO PV (LPV): es la posición del módulo emisor con un formato de “Long Position Vector” (Ver Fig. A.3).

A.1.2.3.6. *Location Service*

Cuando un módulo se quiere comunicar exclusivamente con otro mediante un GUC y no conoce su localización emite un tipo de paquete “*Location Service Request*” (LS_Req) (ver Figura A.15) para localizarlo, cuando un módulo recibe un paquete del tipo LS_Req emite un paquete del tipo “*Location Service Reply*” (LS_Rep) (ver Figura A.16), dando a conocer su localización.

FIGURA A.15: *Location Service Request packet header*

Donde los campos de la Figura A.15 son:

- SO PV (LPV): es la posición del módulo emisor con un formato de “*Long Position Vector*” (Ver Fig. A.3).
- *Request GN_ADDR*: es la “*GeoNetworking address format*” del módulo buscado.

FIGURA A.16: *Location Service Reply packet header*

Donde los campos de la Figura A.16 son:

- SO PV (LPV): es la posición del módulo emisor con un formato de “*Long Position Vector*” (Ver Fig. A.3).
- DE PV (SPV): es la posición del módulo destino con un formato de “*Short Position Vector*” (Ver Fig. A.2). El módulo emisor del LS_Req.

A.1.3. Operaciones de comunicación

En este apartado se analizan los procedimientos que realizan los módulos cuando se manda, se recibe o se reenvía un paquete. Mediante unos datagramas se explicarán estos algoritmos.

En la Figura A.17 se puede observar como GN se relaciona con los diferentes protocolos de la torre OSI, mediante los “Servicios de acceso al protocolo” o en inglés “*Service Access Protocol*” (SAP). GN tiene relación con la capa inferior “*Link Layer Protocol*” (LLP) o “Capa de enlace” mediante IN_SAP, con los protocolos de transporte (PT) mediante GN_SAP, con el protocolo de mantenimiento mediante GN_Mgmt_SAP y con la unidad de seguridad que es la encargada de codificar y decodificar los mensajes por medio de Sec_GN_SAP. Estos servicios de acceso al protocolo son esenciales para que el protocolo de GeoNetworking se comuniquen con el resto de protocolos.

El GN_SAP a su vez tiene tres protocolos de comunicación *Request* (ver Fig. 3.23), *Confirm* y *Indication* (ver Fig. 3.24). Con estos protocolos la capa superior de la torre OSI se puede comunicar con GN.

FIGURA A.17: Interacciones de GeoNetworking.

- PT – > GN: se utilizan los protocolos “*Request*” y “*Confirm*”. La capa superior emite un mensaje del tipo “*Request*” y GeoNetworking le contestara con un mensaje “*Confirm*” indicando a PT el resultado de la operación.
- GN – > PT: se utiliza el protocolo “*Indication*” en el que se le transmite la información al PT.

El IN_SAP a su vez tiene dos protocolos de comunicaciones el LLC_IN y el LLC_OUT. En donde LLC_OUT es la salida del LLP hacia GN y LLC_IN es la entrada de GN en LLP.

Una vez aclaradas las entradas y salidas del protocolo de GeoNetworking en los siguientes apartados se explicaran los procedimientos seguidos cuando se recibe una comunicación por alguna de sus entradas. A partir de ahora en adelante denominaremos “envío” a los mensajes que entran por GN_SAP y “recepción” a los mensajes que entran por IN_SAP.

A.1.3.1. Operaciones de envío

En esta sección se explican brevemente los pasos que realizan los paquetes desde que se recibe la orden de “enviar” un paquete hasta que se retransmite por la red.

En la Figura A.18 se puede ver el flujograma que se realiza cuando se quiere enviar un paquete por GeoNetworking. En esta figura hay que destacar una pequeña sección donde hay un bucle autónomo que manda mensajes a la red, porque como se ha explicado anteriormente es necesario que todos los módulos tengan conocimiento de los módulos vecinos y sus posiciones para poder crear la red, esto implica que hay que retransmitir la posición cada cierto periodo de tiempo, este periodo viene determinado por una constante que se encuentra en 636-4-1 (2014) más una cierta aleatoriedad, esa función la realiza el manejador de *beacons* (“*Beacon PH (Packet Handling)*”).

FIGURA A.18: Esquema de generación de paquetes.

Una vez observado el comportamiento general del generador de paquetes en los siguientes subapartados se explican más detalladamente cada uno de los generadores específicos para cada tipo de paquete.

A.1.3.1.1. GUC sender

En la Figura A.19 se puede observar los pasos que se realizan para enviar un paquete del tipo GeoUnicast. Se puede observar que si no hay conocimiento en la LocT de la posición del módulo de destino se inicia una búsqueda de este módulo mediante un paquete del tipo LS_Req y se almacena el paquete en un *buffer* hasta que pueda ser atendida la petición.

FIGURA A.19: Esquema de *GeoUnicast Sender*.

A.1.3.1.2. GAC y GBC sender

En la Figura A.20 (a) se pueden observar los pasos que se realizan para generar un paquete del tipo “*Geographical Scoped Anycast*”.

En la Figura A.20 (b) se pueden ver los pasos que se realizan cuando se genera un paquete del tipo “*Geographical Scoped Broadcast*”.

FIGURA A.20: Esquemas de GAC y GBC Sender.

A.1.3.1.3. TSB y SHB sender

En la Figura A.21 (a) se puede observar los pasos que se realizan cuando se quiere generar un paquete del tipo “*Topological Scoped Broadcast*” en el flujograma se destaca que si no hay nodos cercanos se almacena el mensaje en un *buffer* a la espera de que aparezca algún módulo en las proximidades.

En la Figura A.21 (b) se observan los pasos que se realizan cuando se quiere generar un paquete del tipo “*Single-Hop Broadcast*” en el flujograma se destaca que si no hay nodos cercanos se almacena el mensaje en un *buffer* a la espera de que aparezca algún módulo en las proximidades.

FIGURA A.21: Esquemas de TSB y SHB Sender.

A.1.3.2. Operaciones de recepción

Cuando el protocolo recibe un paquete por medio de la red LLC_OUT tiene que analizar el paquete y en función de las características de este realizar una serie de operaciones para determinar que realizar con el paquete, retransmitirlo y/o comunicárselo al PT, en la Figura A.22 se ve un datagrama con los pasos detallados.

FIGURA A.22: Esquema de recepción de paquetes.

A.1.3.2.1. Detector de Paquetes Duplicados

En la Figura A.23 se puede observar el Detector de Paquetes Duplicados “DDBM” de sus siglas en inglés “*Duplicated Detectro Buffer Manage*” este módulo es muy importante en todo el algoritmo pues es llamado prácticamente por todos los subprocesos.

FIGURA A.23: Esquema del detector de Paquetes duplicados

Primero comprueba que el paquete no sea duplicado valiéndose del SN y el identificador del creador del paquete, estos datos están contenidos en las cabeceras. Lo siguiente es

comprobar que no hay una duplicidad de identificadores en la red, si lo hay se disparan una serie de mecanismos que cambian los identificadores de los módulos automáticamente, entonces se extrae la información del módulo emisor y del módulo destino, si la hubiera, para actualizar la LocT. Por último se comprueba si se estaba buscando al módulo emisor para retransmitir todos los posibles mensajes almacenados en los buffers de *Location Service* y GeoUnicast.

A.1.3.2.2. *Beacon y GUC receive*

En la Figura A.24 se puede observar los pasos que se realizan cuando se recibe un paquete de los tipos “*Beacon*” (a) y “GeoUniCast (GUC)” (b). Cuando se recibe un paquete del tipo “*Beacon*” se comprueba que no esté duplicado dicho paquete mediante el “DDBM” y se actualiza la LocT.

FIGURA A.24: Esquemas de recepción de paquetes del tipo Beacon y GUC.

Cuando se recibe un paquete del tipo “GUC” lo primero que se hace es desempaquetar el mensaje y los diferentes campos de la cabecera para chequear que todo está bien esto se realiza mediante el “GUC PH” (*GUC Packet Handling*), se comprueba si el paquete es para el módulo receptor y dependiendo de esto se actúa en consecuencia, se chequea que el paquete no este duplicado por medio del “DDBM” y si es para el módulo se avisa al PT indicándole

los datos que correspondientes del paquete, sino se busca el próximo salto y se retransmite por medio del LLC_IN.

A.1.3.2.3. GAC y GBC receive

En la Figura A.25 se pueden observar los pasos que se realizan cuando se recibe algún paquete de los tipos “*Geographical Scoped AnyCast (GAC)*” (a) y “*Geographical Scoped BroadCast (GBC)*” (b). Tanto en el caso de que el paquete sea GAC o GBC se desempaqueta el mensaje para chequear que todos los parámetros estén bien por medio del “GAC/GBC PH” (*GAC/GBC Packet Handling*), se analiza mediante el “DDBM”.

FIGURA A.25: Esquemas de recepción de paquetes del tipo GAC y GBC.

En el caso de que el paquete sea GAC se comprueba si el módulo está dentro del área de recepción (ver A.1.5), si es así el módulo avisa al PT indicándole los datos que correspondientes del paquete, si no está en el interior se busca el próximo salto y se retransmite por medio del LLC_IN.

En el caso de que el paquete sea GBC se busca el próximo salto y se retransmite por medio del LLC_IN, si el módulo se encuentra en el área de recepción indicada en la cabecera (ver A.1.5) el módulo avisa al PT indicándole los datos que correspondientes del paquete.

A.1.3.2.4. TSB y SHB receive

En la Figura A.26 se puede observar los pasos que se realizan cuando se recibe un paquete de los tipos “Topological Scoped Broadcast (TSB)” (a) y “Single-Hop Broadcast (GBC)” (b).

FIGURA A.26: Esquemas de recepción de paquetes del tipo TSB y SHB.

Cuando se recibe un paquete del tipo “TSB” se desempaqueta el mensaje mediante el “TSB PH” (*TSB Packet Handling*) para comprobar que todos los datos de la cabecera están correctos, se analiza mediante el “DDBM” y se avisa al PT indicándole los datos que correspondientes del paquete, adicionalmente se comprueba si hay vecinos cercanos y se retransmite el mensaje en modo *broadcast*, en el caso de no haber vecinos se almacena el mensaje en un buffer a la espera de que haya vecinos disponibles.

Cuando se recibe un paquete del tipo “SHB” se desempaqueta el mensaje mediante el “SHB PH” (*SHB Packet Handling*) para chequear que todos los parámetros estén bien, se analiza mediante el “DDBM” y se avisa al PT indicándole los datos que correspondientes del paquete.

A.1.3.2.5. LS Request y LS Reply receive

En la Figura A.27 se puede observar los pasos que se realizan cuando se recibe un paquete de los tipos “*Location Service Request (LS_Req)*” (a) y “*Location Service Reply (LS_Rep)*” (b).

FIGURA A.27: Esquema de recepción de paquetes del tipo *Location Service*

Cuando se recibe un paquete del tipo “LS_Req” se desempaqueta el mensaje mediante el “LS_Req PH” (*LS_Req Packet Handling*) para comprobar que todos los datos de la cabecera están correctos, se analiza mediante el “DDBM”, se chequea que el mensaje sea para el módulo receptor, si es así se crea un mensaje LS_Rep y se finaliza el algoritmo. Si no es para el módulo el mensaje se difunde entre todos los vecinos, si los hay mediante un mensaje en modo “*broadcast*”, si no existen vecinos se almacena el mensaje en un *buffer* a la espera de que haya módulos vecinos disponibles.

Cuando se recibe un paquete del tipo “LS_Rep” se desempaqueta el mensaje mediante el “LS_Rep PH” (*LS_Rep Packet Handling*) para comprobar que todos los datos de la cabecera están correctos, se analiza mediante el “DDBM”, se chequea que el mensaje sea para el módulo receptor, si es así se deja de buscar al módulo que ha originado el “LS_Rep” parando

la emisión de mensajes del tipo “LS_Req”. Si el mensaje no es para este módulo se busca cual es el siguiente salto y se retransmite por medio del LLC_IN.

A.1.4. Protocolo de acceso al servicio de geoposicionamiento (GN-SAP)

En los anexos del estándar ETSI EN 636-4-1 636-4-1 (2014) se especifican los campos que son necesarios para la comunicación entre los PT y GeoNetworking. Después GN los extrae y los trata para mandar los paquetes adecuados.

A.1.4.1. Request

Los paquetes del tipo “Request” son los que van desde los diferentes PT al GN. Por cada “Request” existirá un “confirm” indicando si el paquete ha sido aceptado o no, si no es aceptado se devuelve una breve explicación de por qué no ha podido ser aceptado el mensaje.

Los campos del “Request” son:

- *Upper protocol entity*: este campo especifica de que PT proviene el mensaje para poder tratar adecuadamente los datos del paquete.
- *Packet transport type*: este campo especifica el tipo de paquete de GN se debe utilizar, usando la tabla A.4 como codificador.
- *Destination*: dependiendo del tipo de paquete que se desee enviar a través de GN puede ser la posición geográfica del área de interés si es un paquete del tipo GAC o GBC, o ser un GNADDR en cualquier otro caso.
- *Communication profile*: este valor va dirigido a LLC para que configure los parámetros del medio de comunicaciones.
- *Security profile*: campo opcional, indica a GN si debe encriptar el mensaje y como.
- *Maximum packet lifetime* (opt): vida máxima del paquete.
- *Repetition interval* (opt): intervalo de repetición del paquete, entre diferentes retransmisiones.
- *Maximum repetition time* (opt): tiempo máximo de repetición del paquete.
- *Maximum hop limit* (opt): número máximo de saltos que puede efectuar el paquete.
- *Traffic class*.

- *Length*: longitud de los datos a retransmitir contenidos en el campo “Data”.
- *Data*: datos a retransmitir.
- *opt*: campo opcional, en el caso de que el campo este vacío se tomaran los valores por defecto que están definidos en el estándar.

A.1.4.2. Indication

Los paquetes del tipo “Indication” son los paquetes que van desde el Geonetworking a los diferentes PT. Estos datos se producen cuando un mensaje llega al GN y es necesario transmitírselo a la instancia superior.

Los campos del “*Indication*” son:

- *Upper protocol entity*: este campo especifica de que PT proviene el mensaje para que los diferentes PT puedan decidir si es para ellos o no.
- *Packet transport type*: este campo especifica el tipo de paquete de GN que ha llegado al módulo, usando la tabla A.4 como codificador.
- *Destination* (opt): dependiendo del tipo de paquete que se desee enviar a través de GN puede ser la posición geográfica del área de interés si es un paquete del tipo GAC o GBC, o ser un GNADDR en cualquier otro caso.
- *Source position vector*: es la posición del módulo emisor en formato LPV.
- *Security report* (opt): campo que contiene el resultado de la descryptación del mensaje si hubiera sido necesario.
- *Certificate id* (opt): campo que contiene el certificado de descryptación.
- *Permissions* (opt): campo que contiene los permisos necesarios para la descryptación.
- *Traffic class*.
- *Remaining packet lifetime* (opt): indica cuanto tiempo de validez tenía el paquete al recibirlo por el GN.
- *Remaining hop limit* (opt): indica cuantos saltos quedaban antes de ser descartado el paquete.
- *Length*: longitud de los datos recibidos que están contenidos en el campo “Data”.
- *Data*: datos a retransmitir.

- opt: campo opcional, si existe se rellena, si no hay suficiente información en el paquete de GN se deja en blanco.

A.1.5. Definición de las áreas geográficas para GeoBroadcast y GeoAnycast

En el estándar ETSI EN 363-4-1 636-4-1 (2014) a la hora de mandar paquetes de datos a determinadas áreas de interés se utilizan los tipos de paquetes llamados “GeoBroadcast” o “GeoAnycast”. En el estándar se enumeran tres posibles tipos de áreas: Circular, rectangular o elipsoidal. Estas áreas vienen definidas en el estándar ETSI EN 302 931 931 (2014), donde se especifican las características de las áreas así como sus ecuaciones. También se definen las variables necesarias para describir las áreas:

- $P_0(x, y)$: posición del punto central del área.
- Distancia a: distancia entre el punto central y el punto central del lado mayor del rectángulo o el semieje mayor de la elipse.
- Distancia b: distancia entre el punto central y el punto central del lado menor del rectángulo o el semieje menor de la elipse.
- r: Radio de la circunferencia.
- θ : ángulo azimutal: Ángulo entre el Norte y el lado mayor del rectángulo o el semieje mayor de la elipse.
- φ : ángulo cenital: Ángulo entre el Este y el lado mayor del rectángulo o el semieje mayor de la elipse. ...

TABLA A.5: Elementos necesarios para definir el área

	Punto central	Distancia a	Distancia b	r	θ	φ
Circunferencia	x			x		
Rectángulo		x	x		x	
Elipse		x	x		x	

En la Tabla A.5 se puede comprobar que variables son esenciales para definir precisamente la circunferencia (Figura A.28), el rectángulo (Figura A.29) y la elipse (Figura A.30). El estándar 931 (2014) también define la función A.2 para determinar si un punto dado en coordenadas cartesianas $P(x,y)$ está en el interior o no del área.

La función $F(x, y)$ varía en función del área quedando:

$$F(x, y) = \begin{cases} = 1 & \text{si } P(x, y) \text{ es el punto central del área} \\ > 0 & \text{si } P(x, y) \text{ está dentro del área} \\ = 0 & \text{si } P(x, y) \text{ está en el borde del área} \\ < 0 & \text{si } P(x, y) \text{ está fuera del área} \end{cases} \quad (\text{A.2})$$

En la ecuación A.3 se puede observar como es $F(x, y)$ para la circunferencia.

$$F(x, y) = 1 - \left(\frac{x}{r}\right)^2 - \left(\frac{y}{r}\right)^2 \quad (\text{A.3})$$

FIGURA A.28: Elementos de la circunferencia.

En la ecuación A.4 se puede observar como es $F(x, y)$ para el rectángulo.

$$F(x, y) = \text{Min} \left(1 - \left(\frac{x}{a}\right)^2, 1 - \left(\frac{y}{b}\right)^2 \right) \quad (\text{A.4})$$

FIGURA A.29: Elementos del rectángulo.

En la ecuación A.5 se puede observar como es $F(x, y)$ para la elipse.

$$F(x, y) = 1 - \left(\frac{x}{a}\right)^2 - \left(\frac{y}{b}\right)^2 \quad (\text{A.5})$$

FIGURA A.30: Elementos de la elipse.

A.2. Estándar Protocolo básico de transporte (*Basic Transport Protocol, BTP*)

Este protocolo de transporte es el más básico de todos que se comunica con GN, en la Figura A.31 se puede ver la relación que tiene con los elementos de su alrededor. El GN-SAP son los la comunicación con GeoNetworking explicada en A.1.4.

FIGURA A.31: Protocolo BTP.

La Figura A.32 representa la estructura de cabeceras que contendría un paquete enviado con la capa de transporte BTP.

MAC Header	LLC Header	GeoNetworking Header	BTP Header	Payload Header

FIGURA A.32: Estructura de los paquetes BTP

Existen dos subtipos de BTP, el BTP-A que es para una comunicación interactiva y BTP-B que es para una comunicación no interactiva. Ambos subtipos tienen cabeceras diferentes que se pueden observar en la Figura A.33.

FIGURA A.33: BTP header

Donde los campos de la Figura A.33 son:

- *Destination Port*: este campo es común en los dos subtipos de BTP, determina el puerto de destino de la capa de sesión.
- *Source Port*: este campo pertenece únicamente al subtipo BTP-A, determina el puerto de origen del paquete, es decir que elemento de la capa de sesión lo ha enviado.
- *Destination port Info*: este campo pertenece únicamente al subtipo BTP-B, da información adicional sobre la capa de sesión.

A.2.1. Servicio de acceso al protocolo BTP (BTP-SAP)

En los anexos del estándar ETSI EN 636-5-1 636-5-1 (2014) se especifican los campos que son necesarios para la comunicación entre la capa de sesión y BTP. Después BTP los extrae y los trata para comunicarlo aguas arriba (capa de sesión) mediante un “indication” o aguas abajo (hacia GN) mediante un “request”.

A.2.1.1. Request

Los paquetes del tipo “Request” son los que van desde las diferentes aplicaciones al BTP.

Los campos especificados en el estándar necesarios para enviar un “Request” son:

- *BTP type*: indica el subtipo de BTP.
- *Source port*: campo obligatorio para el subtipo BTP-A. Si es BTP-B se ignorará. Puerto de origen de la capa de sesión.
- *Destination port*: puerto de destino de la capa de sesión.
- *Destination port info*: campo opcional para el subtipo BTP-B. Si es BTP-A se ignorará.
- Campos con GN: Responden a las mismas características que en GN. Ver A.1.4.
- *Length*: longitud de los datos.
- *Data*: datos enviados por la capa de sesión.

A.2.1.2. Indication

Los “Indication” son los mensajes que transmite el BTP a la capa de sesión indicando la recepción de un nuevo mensaje proveniente de la capa inferior.

Los campos especificados en el estándar necesarios para enviar un “Indication” son:

- *BTP type*: indica el subtipo de BTP que se ha recibido.
- *Source port*: campo obligatorio para el subtipo BTP-A. Si es BTP-B se ignorará. Puerto de origen de la capa de sesión.
- *Destination port*: puerto de destino de la capa de sesión.
- *Destination port info*: campo opcional para el subtipo BTP-B. Si es BTP-A se ignorará.
- Campos con GN: responden a las mismas características que en GN. Ver A.1.4.
- *Length*: longitud de los datos.
- *Data*: datos recibidos por GN.