

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE
TELECOMUNICACIÓN



“DESARROLLO DE UN GEOSERVICE
SOBRE NODO EDGE 5G PARA
ENTORNOS DE CONDUCCIÓN
AUTÓNOMA CONECTADA”

TRABAJO FIN DE GRADO

Septiembre - 2023

AUTOR: Cristina Nohemi Bravo Carrión

DIRECTOR/ES: Miguel Sepulcre Ribes

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE
ESCUELA POLITÉCNICA SUPERIOR DE ELCHE
GRADO EN INGENIERÍA DE TECNOLOGÍAS DE
TELECOMUNICACIÓN



“DESARROLLO DE UN GEOSERVICE
SOBRE NODO EDGE 5G PARA
ENTORNOS DE CONDUCCIÓN
AUTÓNOMA CONECTADA”

TRABAJO FIN DE GRADO

Septiembre - 2023

AUTOR: Cristina Nohemi Bravo Carrión

DIRECTOR/ES: Miguel Sepulcre Ribes



RESUMEN

En un mundo en constante evolución tecnológica, la digitalización y la movilidad están redefiniendo el sector del transporte. Esta transformación se ve impulsada por una creciente conciencia sobre la necesidad de sistemas de transporte más eficientes, seguros y sostenibles con el medio ambiente. En este contexto, los Sistemas de Transporte Inteligentes Cooperativos (C-ITS) están emergiendo como una solución clave, ofreciendo la posibilidad de optimizar el flujo de tráfico y responder adecuadamente a situaciones de peligro, lo cual requiere de una alta coordinación entre los diferentes elementos que componen el sistema de transporte.

Los sistemas C-ITS se basan en tecnologías de comunicaciones móviles e inalámbricas que permiten la comunicación directa entre vehículos (V2V, *Vehicle-to-Vehicle*) y entre vehículos e infraestructura (V2I, *Vehicle-to-Infrastructure*), o más conocido como V2X (*Vehicle-to-Everything*). Sin embargo, la generación de redes móviles 5G promete mejoras significativas en velocidad y latencias al combinarse con otras tecnologías como el *edge computing*. Estas comunicaciones son especialmente relevantes en el contexto de vehículos autónomos conectados, los cuales no solo están revolucionando la forma en que nos desplazamos, sino que también potencian la seguridad y la eficiencia en nuestras carreteras. La capacidad de estos vehículos para intercambiar información en tiempo real con otros vehículos e infraestructuras viales, facilitada por la tecnología 5G, es fundamental para prevenir accidentes y optimizar flujos de tráfico.

Frente a este panorama, el presente proyecto de fin de grado se centra en abordar un aspecto clave en el ámbito de los C-ITS y los vehículos autónomos conectados: la difusión geolocalizada y selectiva de mensajes de alertas de tráfico, concretamente de mensajes DENM (*Decentralized Environmental Notification Messages*). Por ello, el objetivo del proyecto es desarrollar un geoservice que permita una comunicación de alertas geolocalizada, contextualizada y estratégica. Por tanto, este servicio va a tener en cuenta la posición geográfica de los vehículos en relación con el evento de tráfico sucedido, asegurando que solo los vehículos directamente afectados o en riesgo inminente reciban la notificación pertinente. Este enfoque no solo optimiza la respuesta a situaciones críticas, sino también el volumen de tráfico en las redes de comunicación al eliminar mensajes redundantes o irrelevantes. De esta manera, el sistema contribuye a un uso más eficiente de los recursos de red y fortalece la confiabilidad en los vehículos autónomos conectados.

ABSTRACT

In a world in constant technological evolution, digitalization and mobility are redefining the transportation sector. This transformation is driven by a growing awareness of the need for more efficient, safer and environmentally sustainable transportation systems. In this context, Cooperative Intelligent Transportation Systems (C-ITS) are emerging as a key solution, offering the possibility to optimize traffic flow and respond appropriately to hazardous situations, which requires high coordination between the different elements that make up the transportation system.

C-ITS systems are based on mobile and wireless communications technologies that enable direct communication between vehicles (V2V, Vehicle-to-Vehicle) and between vehicles and infrastructure (V2I, Vehicle-to-Infrastructure), or better known as V2X (Vehicle-to-Everything). However, the 5G generation of mobile networks promises significant improvements in speed and latencies when combined with other technologies such as edge computing. These communications are especially relevant in the context of connected autonomous vehicles, which are not only revolutionizing the way we get around, but also boosting safety and efficiency on our roads. The ability of these vehicles to exchange information in real time with other vehicles and road infrastructures, facilitated by 5G technology, is essential to prevent accidents and optimize traffic flows.

Against this backdrop, this final degree project focuses on addressing a key aspect in the field of C-ITS and connected autonomous vehicles: the geolocalized and selective dissemination of traffic alert messages, specifically DENM (Decentralized Environmental Notification Messages) messages. Therefore, the objective of the project is to develop a geoservice that allows a geolocalized, contextualized and strategic communication of alerts. This service will therefore consider the geographical position of vehicles in relation to the traffic event, ensuring that only vehicles directly affected or at imminent risk receive the relevant notification. This approach not only optimizes the response to critical situations, but also the volume of traffic on the communication networks by eliminating redundant or irrelevant messages. In this way, the system contributes to a more efficient use of network resources and strengthens reliability in connected autonomous vehicles.

ÍNDICE

1. INTRODUCCIÓN.....	1
2. ESTADO DEL ARTE	5
2.1. SISTEMAS C-ITS	5
2.1.1. PILA DE PROTOCOLOS.....	8
2.1.2. MENSAJES C-ITS	10
2.2. REDES 5G.....	13
2.3. GEODIFUSIÓN.....	16
2.4. PROTOCOLOS PARA LA DIFUSIÓN DE MENSAJES C-ITS	18
2.4.1. MQTT (<i>Message Queuing Telemetry Transport</i>)	18
2.4.2. AMQP (<i>Advanced Message Queuing Protocol</i>)	22
2.4.3. ANÁLISIS	23
3. DISEÑO E IMPLEMENTACIÓN DEL GEOSERVICE	26
3.1. FINALIDAD DEL GEOSERVICE	26
3.2. FUNCIONAMIENTO INTERNO DEL GEOSERVICE	27
3.2.1. ASPECTOS TECNOLÓGICOS DE DISEÑO	28
3.2.2. SOLUCIÓN PROPUESTA	29
3.3. NIVEL DE ZOOM	35
3.4. INTEGRACIÓN DEL GEOSERVICE EN ARQUITECTURA EDGE	39
3.4.1. COMPONENTES QUE FORMAN EL NODO EDGE	39
3.4.2. INTERFAZ GEOSERVICE – V2X APP	43
3.5. IMPLEMENTACIÓN CLIENTE MQTT	44
4. EVALUACIÓN Y VALIDACIÓN	47
4.1. TEST DE SUSCRIPCIÓN A BROKER MQTT	47
4.2. FRECUENCIA DE SUSCRIPCIÓN	50
4.3. GENERACIÓN DE TÓPICOS	52

4.4. GESTIÓN DE TÓPICOS Y MENSAJES DENM EN CLIENTE MQTT	55
4.5. RECEPCIÓN DE MENSAJES DENM DESDE NODO EDGE.....	57
5. CONCLUSIONES.....	64
6. TRABAJOS FUTUROS	65
7. Bibliografía	68



ÍNDICE DE FIGURAS

Figura 2.1. Niveles de automatización. Fuente: Sociedad de Ingenieros Técnicos en Automoción, SAE.....	6
Figura 2.2. Pila de protocolos C-ITS.....	8
Figura 2.3. Estructura mensaje CAM.....	11
Figura 2.4. Estructura mensaje DENM.....	12
Figura 2.5. Escenarios de despliegue de 5G V2X - <i>Centralized</i>	14
Figura 2.6. Escenarios de despliegue de 5G V2X - <i>Core Central Office</i>	15
Figura 2.7. Escenarios de despliegue de 5G V2X – <i>Central Office</i>	15
Figura 2.8. Escenarios de despliegue de 5G V2X – <i>Cell Site</i>	15
Figura 2.9. Arquitectura del protocolo MQTT.....	21
Figura 2.10. Arquitectura del protocolo AMQP.....	22
Figura 3.1. Ejemplo ilustrativo de la difusión selectiva de mensajes de alerta.....	27
Figura 3.2. Estructura quadtree.....	30
Figura 3.3. Quadtrees afectados en función del área de relevancia.....	31
Figura 3.4. Quadkeys para nivel de zoom 1, 2 y 3.....	32
Figura 3.5. Ejemplo Coordenadas (X, Y) para nivel de zoom 3.....	33
Figura 3.6. Funciones internas del geoservice.....	34
Figura 3.7. Ejemplo Quadtrees para niveles de zoom 13-18.....	35
Figura 3.8. Ejemplo zoom Geoservice = zoom Cliente.....	37
Figura 3.9. Ejemplo zoom Geoservice > zoom Cliente.....	37
Figura 3.10. Ejemplo zoom Geoservice > zoom Cliente + wildcard.....	38
Figura 3.11. Ejemplo zoom Geoservice < zoom Cliente.....	38
Figura 3.12. Arquitectura nodo edge.....	40
Figura 3.13. Interfaz Geoservice - V2X APP.....	43
Figura 3.14. Ejemplo petición HTTP GET al Geoservice usando Postman.....	44

Figura 4.1. Ejemplo vehículo cambiando de quadtree durante su trayectoria.....	46
Figura 5.1. Numero de suscripciones soportadas por el broker MQTT	48
Figura 5.2. Visualización broker mediante MQTT Explorer	48
Figura 5.3. Especificaciones principales del PC de desarrollo.....	49
Figura 5.4. Frecuencia de suscripción a tópicos según nivel de zoom.....	50
Figura 5.5. Tiempo entre suscripciones en función de la densidad vehicular	51
Figura 5.6. Cálculo de tópicos para diferentes niveles de zoom	52
Figura 5.7. Mensaje de alerta del geoservice ante zoom inválido.....	53
Figura 5.8. Quadrees afectados en función del área de relevancia.....	54
Figura 5.9. Tópicos devueltos por el geoservice en función de los quadrees afectados	54
Figura 5.10. Escenario ejemplo cliente MQTT conectado a broker local.....	56
Figura 5.11. Traza de resultados ejemplo cliente MQTT conectado a broker local.....	57
Figura 5.12. Escenario ejemplo cliente MQTT conectado broker en nodo edge	58
Figura 5.13. Logs de la aplicación V2X.....	59
Figura 5.14. Número de secuencia observado desde MQTT Explorer	59
Figura 5.15. Traza de resultados ejemplo cliente MQTT conectado a broker en nodo edge	60
Figura 5.16. Escenario ejemplo varios clientes MQTT.....	61
Figura 5.17. Traza de resultados ejemplo múltiples clientes MQTT	61
Figura 5.18. Representación en Matlab de la traza de resultados.....	62

ÍNDICE DE TABLAS

Tabla 1. Valores distancia de relevancia 31

Tabla 2. Distancia de cobertura de un quadtree en función del nivel de zoom 35



GLOSARIO DE TÉRMINOS

ITS: Intelligent Transportation Systems.

C-ITS: Cooperative Intelligent Transportation Systems

V2X: Vehicle-to-Everything

C-V2X: Cellular Vehicle-to-Everything

V2I: Vehicle-to-Infrastructure.

V2V: Vehicle-to-Vehicle.

5G: 5 Generation

MQTT: Message Queuing Telemetry Transport

HTTP: Hypertext Transfer Protocol

OBU: On Board Unit

RSU: Roadside unit

ETSI: European Telecommunications Standards Institute

UE: Unión Europea

DSRC: Dedicated Short-Range Communications

ADAS: Advanced Driver Assistance Systems

GPS: Global Positioning System

REST: Representational State Transfer



1. INTRODUCCIÓN

En la actualidad, estamos asistiendo a un proceso de cambio acelerado que se percibe claramente en el ámbito tecnológico, en la globalización de la economía y en el impacto medioambiental. Las nuevas tecnologías, principalmente en ámbitos de digitalización y movilidad, acompañadas de nuevas sensibilidades sociales como la sostenibilidad ambiental, están impactando fuertemente en el sector del transporte. Esto se refleja claramente en la aparición de nuevos modelos basados en el desarrollo de una conectividad entre personas, vehículos y la propia infraestructura vial. La evolución hacia la conducción autónoma y conectada es una realidad que se impone día a día y a la que no debemos tardar en adaptarnos.

Los Sistemas Cooperativos de Transporte Inteligente (C-ITS, *Cooperative Intelligent Transport Systems*) son sistemas que permiten a los vehículos interactuar directamente entre sí y con la infraestructura. La disponibilidad de datos y su explotación en tiempo real permite incrementar la eficiencia de la vía, y en lo que respecta a la seguridad, una correcta reacción y actuación ante situaciones de peligro. Por ello, es vital una perfecta coordinación entre los distintos elementos que conforman el sistema de transporte, así como cubrir adecuadamente aspectos tan importantes como:

- La interoperabilidad: es esencial que los sistemas C-ITS sean interoperables, es decir, que puedan comunicarse entre sí independientemente del fabricante o del país en el que se utilicen. Esto requiere la adopción de estándares comunes a nivel internacional.
- La privacidad y seguridad de los datos: la compartición de datos entre vehículos e infraestructuras plantea preocupaciones sobre la privacidad y la seguridad de los datos. Es necesario establecer mecanismos robustos para garantizar que los datos compartidos sean seguros y se utilicen de manera responsable.
- Una buena infraestructura: la implementación de sistemas C-ITS requiere una infraestructura de comunicación robusta y fiable. Esto incluye la instalación de unidades a bordo (OBUs) en los vehículos, unidades de carretera (RSUs) en la infraestructura vial y centros de control de tráfico.

En lo que respecta a la comunicación entre los distintos componentes, los C-ITS engloban la comunicación de vehículo a vehículo (V2V), la comunicación de vehículo a infraestructura (V2I), así como la comunicación entre vehículos y peatones (V2P) o la red (V2N).

En definitiva, se trata de la comunicación de “vehículo a todo” (V2X), lo que permite contar con una amplia gama de servicios de información y cooperación, así como lograr grandes mejoras en ámbitos de:

- Seguridad vial: prevención de accidentes y choques mediante el intercambio de información sobre la velocidad, ubicación y dirección de los vehículos.
- Eficiencia del tráfico: mejora de la fluidez del tráfico y reduce la aparición de atascos.
- Reducción del impacto medioambiental: la comunicación V2X contribuye al ahorro de energía y a la reducción de emisiones contaminantes debidas al exponencial aumento de vehículos presentes en nuestras carreteras.
- Mayor confort para el usuario: al proporcionar información en tiempo real sobre el estado de la carretera y el tráfico, los sistemas C-ITS pueden ayudar a los conductores a tomar decisiones informadas sobre su ruta y mejorar su experiencia durante la conducción.

En los últimos años han surgido dos estándares principales de comunicación vehicular que utilizan la banda de 5.9 GHz asignada a aplicaciones ITS y que permiten la comunicación directa entre vehículos y entre vehículos e infraestructura sin necesidad de una infraestructura de comunicaciones (como por ejemplo la infraestructura de la red móvil). Por un lado tenemos el estándar IEEE 802.11p que se desarrolló para generar redes vehiculares y del que ha surgido diversas variantes como la desarrollada en EE.UU bajo el nombre DSRC, o la variante ITS-G5 desarrollada por ETSI (Instituto Europeo de Normas de Telecomunicaciones). Y por otro lado tenemos como alternativa a la tecnología IEEE 802.11p, la tecnología celular C-V2X (también denominada LTE-V2X) presente en las especificaciones del sistema 4G/LTE [1].

A medida que avanzamos hacia una mayor adopción de vehículos autónomos conectados, a parte de los estándares definidos, es fundamental contar con una red que pueda manejar grandes cantidades de datos en tiempo real, algo para lo que la tecnología 5G está especialmente diseñada.

Dicha tecnología promete ofrecer bajas latencias y un gran ancho de banda. Es decir, se centra en garantizar una respuesta casi inmediata (una latencia mínima), y en proporcionar un servicio con capacidad para conectar un mayor número de dispositivos simultáneamente sin degradar la calidad del mismo.

Esto es justo lo que requieren las comunicaciones V2X, una velocidad de respuesta rápida ante eventos relevantes de tráfico con el fin de evitar situaciones peligrosas en la vía. Sin embargo, no basta con la promesa de baja latencia que ofrece la red 5G, para conseguir las bajas latencias que se necesita en las comunicaciones V2X, es necesario el uso de un "edge computing node" [2]. Esto va a permitir llevar el poder de procesamiento lo más cerca posible de la fuente de datos o punto de consumo, en este caso, de los vehículos. Al evitar enviar datos a través de largas distancias para ser procesados en centros de datos, lo que se busca es tratarlos localmente, reduciendo la latencia y mejorando la eficiencia general.

Es importante también, que toda esa información que llega al nodo *Edge*, se difunda hacia los vehículos de la forma más eficiente posible, por ejemplo, evitando enviar notificaciones de alerta a vehículos para los que no es relevante dicha información.

Para asegurarnos de que efectivamente, solo son notificados aquellos vehículos ubicados en el área de interés del mensaje generado ante un evento de tráfico, se requiere de un servicio con capacidad de difusión de mensajes de alerta que tenga en cuenta la ubicación geográfica de sus nodos.

Sin embargo, las redes móviles actuales no están preparadas para realizar una difusión selectiva de mensajes por sí mismas teniendo en cuenta la localización de sus nodos. Esto significa que todos los vehículos recibirían la misma notificación de alerta, independientemente de si están directamente afectados por el evento de tráfico en cuestión o no. Lo cual puede dar lugar a respuestas ineficaces o innecesarias de los vehículos, y en última instancia, a una gestión ineficiente del tráfico.

Este es precisamente el objetivo que busca cumplir el presente trabajo de fin de grado: desarrollar un geoservicio que irá integrado en un nodo *Edge* y que permita una difusión de mensajes inteligente y contextualizada, teniendo en cuenta la localización de los vehículos y su relación con el evento de tráfico sucedido. De esta manera, solo los vehículos que están directamente afectados por el evento, o que podrían estarlo en un futuro cercano, recibirán la notificación de alerta correspondiente gracias a la capacidad que posee el geoservicio implementado para calcular las zonas afectadas por dicho evento. Además, este sistema ayuda a reducir latencias, carga en las redes de comunicación ya que se enviarían menos mensajes innecesarios y va a contribuir a adoptar una mayor confianza en los vehículos autónomos conectados.

Cabe destacar que, este trabajo de fin de grado se enmarca en un proyecto del laboratorio UWICORE, perteneciente al Departamento de Ingeniería de Comunicaciones, para el desarrollo de un nodo Edge computing para redes 5G destinado a vehículos autónomos conectados.

Para finalizar esta sección, se detallará la estructura que seguirá de aquí en adelante la presente memoria. En el capítulo 2 se abordará el estado del arte relevante para nuestra investigación, comenzando por los Sistemas C-ITS, profundizando en su pila de protocolos y en los tipos de mensajes utilizados. Se presentará una revisión sobre las Redes 5G, la importancia del concepto de la Geodifusión y se concluirá con un análisis de los protocolos que nos van a permitir el envío de esos mensajes C-ITS. En el capítulo 3, en cambio, se explicará cómo se ha llevado a cabo el diseño e implementación del *geoservice*, se indicará el papel que va a desempeñar dentro del nodo *Edge* en el que será implementado, así como su interacción con el resto de elementos que lo conforman. También se detalla en la sección aquellas funciones que se han programado para ser incorporadas en la OBU (*On Board Unit*) de los vehículos con el fin de recibir toda esa información enviada desde el nodo *Edge*. En el capítulo 4, se demostrará mediante la evaluación y validación de diferentes pruebas, el correcto funcionamiento del sistema implementado. Y por último, en los capítulos 5 y 6, se muestra las conclusiones obtenidas al respecto así como las posibles actualizaciones o trabajos futuros que se podrían realizar al diseño del *geoservice*.

2. ESTADO DEL ARTE

Los sistemas C-ITS han sido diseñados para mejorar la seguridad vial y la eficiencia del tráfico. Permiten la comunicación directa entre vehículos mediante el intercambio de mensajes como los CAM y DENM, y se basan en una pila de protocolos ITS. Con la llegada de la tecnología 5G, la industria se plantea la posibilidad de emplear la infraestructura celular para el intercambio de estos mensajes C-ITS y aprovechar el gran potencial que brinda 5G. Por lo que, en el siguiente capítulo se van a analizar los sistemas C-ITS (pila y mensajes), las redes 5G (incluyendo la importancia del *edge computing*), la geodifusión y los protocolos de mensajería más adecuados para el envío de los mensajes C-ITS.

2.1. SISTEMAS C-ITS

En los últimos diez años, se han producido novedosos avances en lo que respecta a las tecnologías de los C-ITS. En 2011, los fabricantes de vehículos de la UE unidos en el Consorcio de Comunicación CAR2CAR declararon su intención de iniciar una implementación a gran escala en 2015, año en que los sistemas estarían preparados tecnológicamente. Sin embargo, quedó claro que esto sólo sería posible si las principales partes interesadas seguían un planteamiento común para alcanzar dicho objetivo.

En 2014, la Comisión Europea creó una plataforma para la implementación de los sistemas cooperativos de transporte inteligente en la UE (plataforma C-ITS), en la que pudieran trabajar de manera conjunta.

A través de la plataforma C-Roads , en la que han colaborado dieciséis países de la UE, se han armonizado los servicios C-ITS V2I para hacerlos interoperables. Esto ha sido fruto de la colaboración entre esta plataforma y el Consorcio de Comunicación CAR2CAR, que ha mejorado la uniformidad de los mensajes y sistemas V2V y V2I.

En el año 2016, se creó la Asociación Automotriz 5G formada por entidades de los sectores de automoción y telecomunicaciones relacionadas con la tecnología para la movilidad conectada y automatizada, en la que se incluyen los servicios C-ITS. De esta asociación han surgido dos tecnologías diferentes para la comunicación de corto alcance con diferentes niveles de desarrollo: DSRC (*Dedicated Short Range Communications*) basada en la norma IEEE 802.11p, y C-V2X (*Cellular Vehicle-to-Everything*) basada en el estándar 3GPP LTE y 5G.

La Comisión Europea, dentro del conjunto de convocatorias dirigidas a la investigación e innovación en el transporte de carreteras para el periodo 2018-2020, entre otros, ha tenido en consideración el apoyo específico a las pruebas de conectividad con tecnología 5G en los nuevos servicios de movilidad y conducción semiautónoma.

Concretamente, en 2019 se define un reglamento en el que se adoptan nuevas reglas que aceleran el despliegue de los Sistemas C-ITS en las carreteras europeas, complementando así la Directiva 2010/40/UE. En dicho reglamento se establecen los requisitos legales mínimos para la interoperabilidad entre los diferentes sistemas cooperativos utilizados y posibilita la implementación a gran escala de los sistemas y servicios C-ITS [3].

Para que Europa convierta esta ambición en realidad, los agentes privados, los Estados miembros y las Administraciones regionales y locales tendrán que cooperar en el desarrollo de una visión común de la movilidad conectada y automatizada.

En el documento “En ruta hacia la movilidad automatizada: estrategia de la UE para la movilidad del futuro” [4] se propone el cumplimiento de una agenda clara, prospectiva y ambiciosa para que Europa pueda asumir el liderazgo mundial en el despliegue de la movilidad conectada y automatizada.

En términos técnicos, los niveles de automatización en la conducción que se han alcanzado y se busca alcanzar, son los que se muestran en la Figura 2.1:

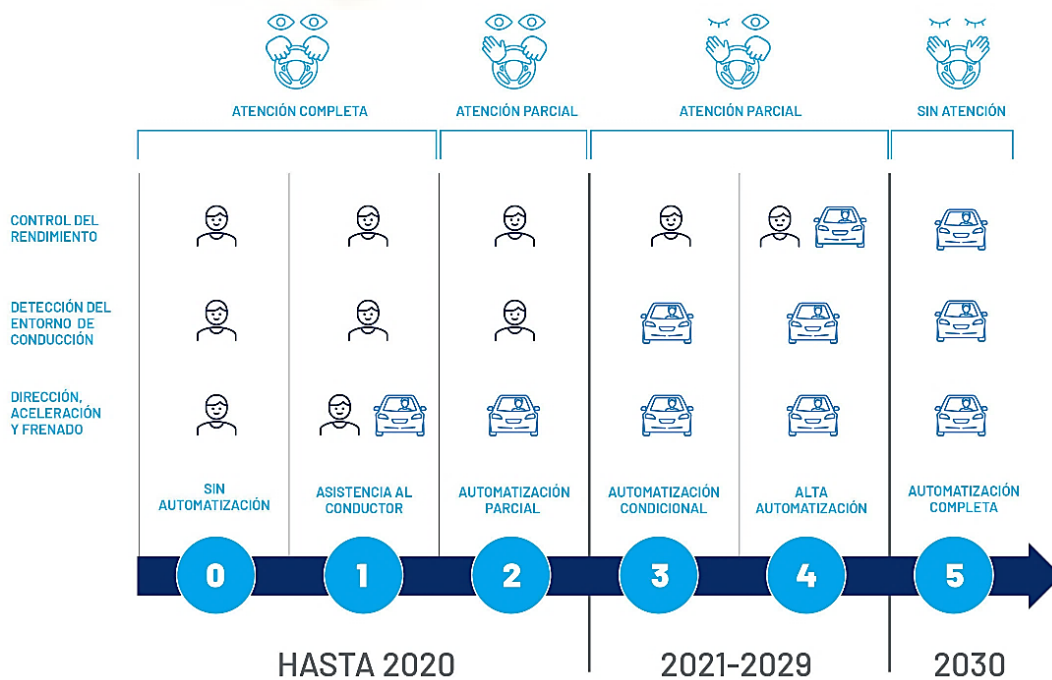


Figura 2.1. Niveles de automatización. **Fuente:** Sociedad de Ingenieros Técnicos en Automoción, SAE.

- Nivel 0 (Sin automatización): Un humano realiza todas las tareas de conducción sin la incorporación de sistemas complejos de ayuda. Este nivel sí admite la presencia de un asistente automatizado que avisa de la presencia de un vehículo en el ángulo muerto del retrovisor.
- Nivel 1 (Asistencia al conductor): La conducción depende únicamente del comportamiento humano, pero permite que un sistema de asistencia electrónico controle el movimiento lateral y longitudinal del vehículo, pero nunca los dos a la vez. Es decir, puede manejar la dirección para ayudar al aparcamiento o la aceleración y desaceleración mediante la programación de la velocidad. En la actualidad, muchos coches ya disponen de este tipo de asistencia a través del control de la velocidad de crucero.
- Nivel 2 (Automatización parcial): La responsabilidad última sigue siendo del humano, pero los asistentes electrónicos se encargan de la dirección y la aceleración y desaceleración. Por ejemplo, el sistema automatizado mantiene el coche en el centro del carril y al mismo tiempo controla la velocidad. En este nivel son esenciales los ADAS (Sistemas Avanzados de Asistencia al Conductor), sensores, cámaras y radares que evitan riesgos en la conducción y que están ya incorporados en una gran parte de los vehículos del mercado.
- Nivel 3 (Automatización condicional): El conductor ha de estar preparado para intervenir, pero ya no necesita supervisar la conducción. El sistema de automatización puede avisar al humano cuando no sepa responder ante una situación o haya un fallo técnico. La normativa europea ahora permite la circulación de estos vehículos, aunque la situación de escasez de semiconductores entre otros factores hace pensar a los expertos que hasta el año 2025 no se habrá generalizado su uso. Según la Asociación Española de Fabricantes de Automóviles y Camiones (ANFAC), el 23 % de la oferta ya incluye la tecnología para poder funcionar en este nivel "en determinadas condiciones".
- Nivel 4 (Alta automatización): Los sistemas automatizados están preparados para conducir de forma autónoma ante cualquier imprevisto y de forma sostenida en el tiempo. El humano ya no se responsabiliza de la conducción ya que el vehículo está 'entrenado' para reaccionar sin que intervenga un humano. Es una apuesta para entornos urbanos y velocidades bajas.

- Nivel 5 (Automatización completa): Los sistemas de conducción automatizada realizan toda la conducción al margen de limitaciones climatológicas o geográficas. En estos vehículos ya no hay volantes ni pedales y pueden moverse por cualquier carretera.

2.1.1. PILA DE PROTOCOLOS

Es fundamental adentrarse en los aspectos técnicos que permiten el funcionamiento eficiente de estos sistemas. Por ello, la pila de protocolos C-ITS es tan importante para la comunicación y operación de los mismos. Establece un conjunto de reglas y convenciones de comunicación que se utilizan para transmitir información entre los dispositivos y sistemas que conforman el ecosistema de los C-ITS. A continuación, examinaremos en detalle la estructura:

Las normas C-ITS siguen una arquitectura general, especificada en ETSI EN 302 665 e ISO 21217, con la estación ITS como elemento central. Para los C-ITS, se ha tenido que adaptar el modelo de referencia ISO OSI de modo que se definen tres capas de protocolo horizontales, dos entidades de protocolo verticales y las aplicaciones ITS en la parte superior, tal y como se muestra a continuación en la Figura 2.2:

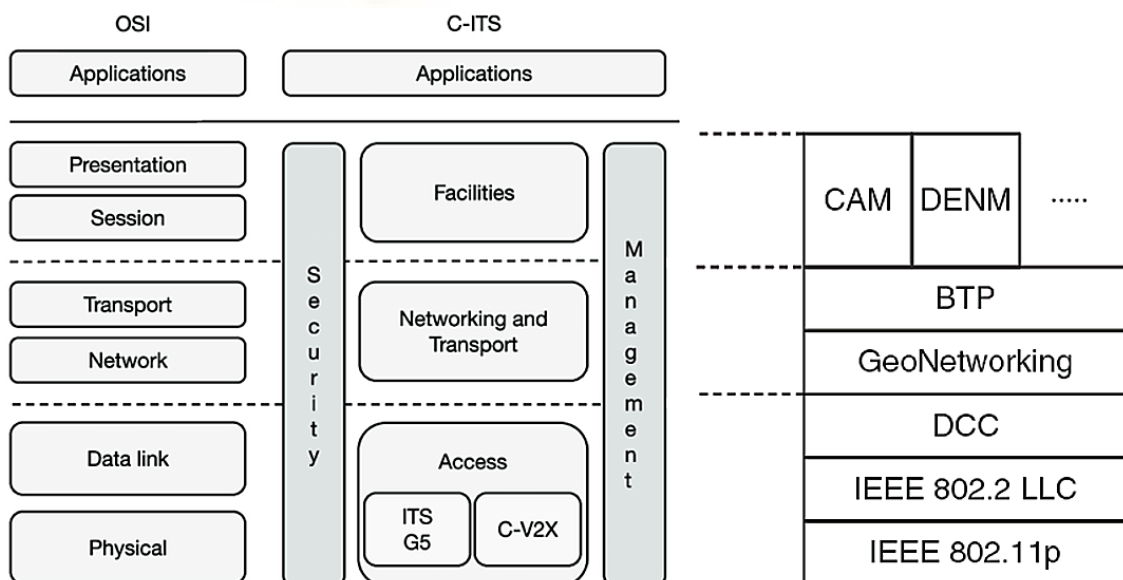


Figura 2.2. Pila de protocolos C-ITS

Veamos resumidamente en qué consisten las capas de protocolo horizontales:

Capa Tecnologías de Acceso

La capa de tecnologías de acceso abarca diversos medios de comunicación y protocolos relacionados con las capas física y de enlace de datos. Las tecnologías de acceso no están restringidas a un tipo específico de medio, aunque la mayoría de las tecnologías de acceso se basan en la comunicación inalámbrica. Se utilizan para la comunicación dentro de una estación ITS (entre sus componentes internos) y para la comunicación externa (por ejemplo, con otras estaciones ITS).

Capa de Red y Transporte

La capa de red y transporte comprende protocolos para la entrega de datos entre estaciones ITS y desde estaciones ITS a otros nodos de red, como por ejemplo, Internet. Los protocolos de red incluyen, en particular, el encaminamiento de datos de origen a destino a través de nodos intermedios y la difusión eficaz de datos en zonas geográficas. Los protocolos de transporte, por su parte, proporcionan la entrega de datos de extremo a extremo y, en función de los requisitos de las instalaciones y aplicaciones ITS, servicios adicionales, como la transferencia fiable de datos, el control de flujo y evitar las congestiones. Un protocolo particular de la capa de red y transporte de los ITS es el protocolo de Internet IP versión 6 (IPv6).

Capa de Servicios o *Facilities*

Los mensajes de concienciación cooperativa (CAM) y los mensajes de notificación ambiental descentralizada (DENM) son los dos protocolos presentes en esta capa que se han normalizado en ETSI EN 302 637-2 y ETSI EN 302 637-3, respectivamente. La capa de servicios o *facilities* ITS proporciona una colección de funciones para dar soporte a las aplicaciones ITS, proporcionando estructuras de datos para almacenar, agregar y mantener datos de distinto tipo y procedencia (por ejemplo, de sensores de vehículos y de datos recibidos mediante comunicación). En cuanto a la comunicación, esta capa permite varios tipos de direccionamiento a las aplicaciones, proporcionan manejo de mensajes específicos ITS y soportan el establecimiento y mantenimiento de sesiones de comunicación.

Capa de Aplicación

La capa de aplicación de los Sistemas ITS incluye las aplicaciones y servicios específicos que utilizan los datos de tráfico y la información de comunicación proporcionada por las capas subyacentes para ofrecer funciones que mejoren la seguridad, eficiencia y comodidad del transporte.

Respecto a las dos entidades verticales de la pila de protocolos, tenemos:

Capa de Gestión

La capa de gestión es responsable de la configuración de una estación ITS, del intercambio de información entre las distintas capas y otras tareas como coordinar y controlar el funcionamiento de los elementos de red y servicios de C-ITS, o proporcionar funciones de configuración, monitoreo y gestión de fallos para asegurar un funcionamiento adecuado y eficiente del sistema.

Capa de Seguridad

La capa de seguridad proporciona servicios de seguridad y privacidad, incluidos mensajes seguros en diferentes capas de la pila de comunicación, gestión de identidades y credenciales de seguridad, y aspectos para plataformas seguras (cortafuegos, pasarela de seguridad, hardware a prueba de manipulaciones).

2.1.2. MENSAJES C-ITS

Una vez comprendida la estructura y funcionamiento de la pila de protocolos C-ITS, es indispensable entender cómo se manejan los datos y la información que circulan a través de estos sistemas. En particular, en la capa '*facilities*' de la pila de protocolos, se encuentran dos tipos de mensajes cruciales para la operatividad de los C-ITS: los mensajes CAM (*Cooperative Awareness Messages*) y DENM (*Decentralized Environmental Notification Messages*). Estos mensajes permiten la transmisión de información sobre el estado y las condiciones del entorno de tráfico, así como la toma de decisiones en tiempo real para garantizar la seguridad y eficiencia en la circulación. A continuación, nos enfocaremos en analizar las características y la estructura de los mensajes CAM y DENM, puesto que son los mensajes C-ITS fundamentales.

CAM

CAM es un mensaje periódico que proporciona información de estado a las estaciones ITS vecinas. Su transmisión se activa cuando un vehículo se encuentra en un contexto relevante para la seguridad (básicamente, cuando el motor está en marcha). Un CAM se compone de una cabecera PDU y varios contenedores que agrupan los campos de datos según el papel del emisor y la frecuencia de su aparición en el mensaje, tal y como se observa en la figura

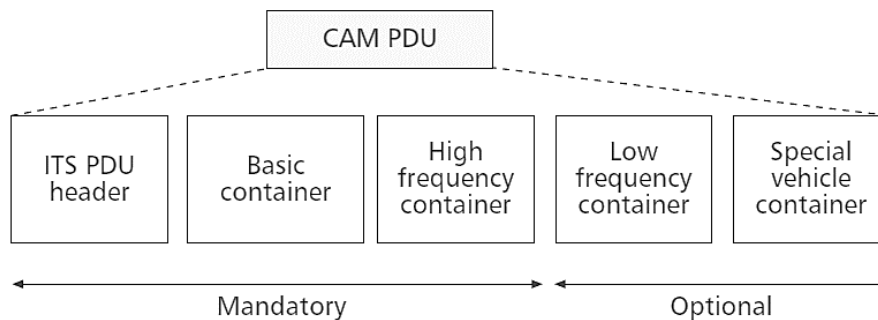


Figura 2.3. Estructura mensaje CAM

La cabecera de la PDU lleva la versión del protocolo, el tipo de mensaje y la dirección del remitente; el contenedor básico tiene el tipo de estación y su posición. Para reducir el tamaño del CAM, el contenedor de alta frecuencia contiene principalmente datos muy dinámicos (por ejemplo, rumbo, velocidad y aceleración del vehículo) y se envía en cada CAM. El contenedor de baja frecuencia contiene datos con menor relevancia para la seguridad (por ejemplo, el papel del vehículo), también puede tener un gran tamaño (por ejemplo, el historial de la trayectoria) y, por lo tanto, no siempre se añade al CAM. Los contenedores especiales para vehículos se añaden opcionalmente si son necesarios para la función del remitente, como en el caso del transporte público, las mercancías peligrosas, las obras en carretera o el rescate.

El concepto de contenedor garantiza un formato de mensaje flexible que puede adaptarse a las necesidades del vehículo emisor y receptor, al tiempo que minimiza la carga del canal inalámbrico.

DENM

DENM es un mensaje controlado por la aplicación y activado por eventos de seguridad. Cuando un nodo de infraestructura (como un vehículo) detecta una situación de peligro, el protocolo DENM asigna un identificador de acción que es único para la estación ITS que lo detecta. A diferencia de la difusión CAM a través de un único salto ITS-G5, al DENM se le asigna un área de relevancia para su difusión y puede transportarse a través de varios saltos ITS-G5 inalámbricos. Al igual que el CAM, el DENM se organiza en contenedores con una cabecera ITS PDU, tal y como se observa en la Figura 2.4. El contenedor de gestión – con campos para el identificador de la acción, la hora de detección, la posición del evento, etc. – es obligatorio, todos los demás contenedores se añaden opcionalmente si la aplicación lo necesita. El contenedor de situación tiene campos para describir el suceso mediante un código predefinido para el suceso causante, así como los sucesos relacionados (por ejemplo, sucesos vinculados o un historial de sucesos). El contenedor de localización lleva campos para la velocidad del evento, el rumbo y las trazas. Se puede añadir un contenedor a la carta para transmitir contenidos específicos de la aplicación, como para la posición en el carril, la reducción de impactos y las obras en la carretera, entre otros.

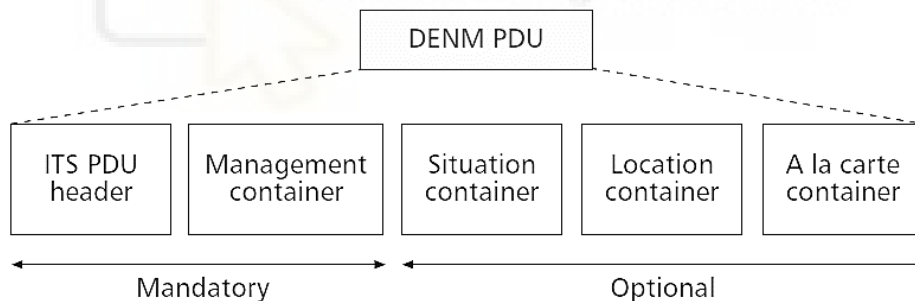


Figura 2.4. Estructura mensaje DENM

Dentro del contenedor de gestión de los mensajes DENM, encontramos 3 campos que posteriormente jugaran un papel importante en la implementación. Estos son los campos: *latitude*, *longitude* y *relevanceDistance*.

Es importante recalcar el papel vital que desempeñan los mensajes DENM en el contexto de eventos relevantes de tráfico. Estos mensajes son esencialmente aquellos que esperan recibir los vehículos cuando se produce una situación importante en la carretera, como un accidente, un atasco, o cualquier otro evento que pueda afectar la circulación. Los DENM

actúan por tanto, como alertas que se distribuyen de manera descentralizada para informar a los vehículos cercanos sobre el evento en cuestión. De esta manera, los vehículos que reciben un mensaje DENM pueden adaptar su comportamiento, como reducir la velocidad o cambiar de carril, para manejar la situación de la manera más segura y eficiente posible. Esta capacidad de reacción rápida y coordinada es fundamental para maximizar la seguridad en las carreteras y minimizar los tiempos de viaje.

2.2. REDES 5G

5G es la quinta generación de tecnología celular inalámbrica, que ofrece mayores velocidades de carga y descarga, conexiones más consistentes y una capacidad mejorada que las redes anteriores. El 5G es mucho más rápido y fiable que las redes 4G actualmente populares y tiene el potencial de transformar la forma en que utilizamos Internet para acceder a las aplicaciones, redes sociales e información [5]. Por ejemplo, tecnologías como los automóviles que se conducen solos, aplicaciones avanzadas de juegos y medios de transmisión en directo que requieren conexiones de datos muy fiables y de alta velocidad se beneficiarán enormemente de la conectividad 5G.

Como hemos dicho, la implementación de la tecnología 5G ha supuesto un salto cualitativo en términos de velocidad, latencia y capacidad de conexión. Esta tecnología utiliza bandas de frecuencia más altas y una infraestructura de red densa para proporcionar velocidades de transmisión de datos hasta 100 veces más rápidas que las redes 4G. Además, la latencia, o tiempo que tarda un paquete de datos en viajar desde su origen hasta su destino, se ha reducido significativamente, lo que permite la comunicación entre máquinas y aplicaciones en tiempo real.

Por esta razón, la industria ve una oportunidad en el 5G para la difusión eficiente de mensajes C-ITS, como los CAM y DENM, en lugar de intercambiarlos entre los propios vehículos. Utilizar el 5G para transmitir estos mensajes no sólo permitiría una entrega más rápida y confiable, sino que también proporcionaría un entorno de transporte más conectado e inteligente.

Sin embargo, a pesar de estos avances, las redes celulares actuales todavía presentan ciertas limitaciones que deben tenerse en cuenta. Una de las más notables es no poder brindar la latencia requerida por las comunicaciones V2X. Aunque las redes 5G prometen y, en muchos casos, ofrecen latencias bajas en comparación con sus predecesoras, las exigentes demandas de las comunicaciones V2X, especialmente en escenarios de alta

densidad vehicular o situaciones críticas de seguridad, pueden sobrepasar las capacidades actuales de la red. Por ello, es esencial considerar arquitecturas de red avanzadas que puedan trabajar en conjunto con el 5G para alcanzar esos estrictos requisitos.

En este sentido, la incorporación de nodos *edge computing* emerge como una solución necesaria y viable. Se trata de servidores que se encuentran ubicados más cercanos al usuario final (los vehículos en este caso), permitiendo que los datos se procesen más cerca de su origen y se reduzca el tiempo que tardan en viajar hasta los centros de datos, lo que conlleva una disminución muy significativa de la latencia. Este tipo de configuración potencia las capacidades de la red 5G, permitiendo alcanzar esas latencias ultra bajas que tan fundamentales son para las comunicaciones V2X [2].

A continuación, analizaremos los principales despliegues posibles de redes 5G considerados para dar soporte a servicios V2X con el fin de demostrar que traer el procesamiento de los datos al borde de la red, brinda esas mínimas latencias requeridas en este tipo de comunicaciones vehiculares [2].

- Centralized: Este despliegue centralizado considera que el servidor encargado de procesar los mensajes (AS V2X) está situado en la nube fuera del dominio de la red móvil (observar Figura 2.5). Este despliegue puede beneficiarse de los potentes recursos informáticos y de almacenamiento disponibles en la nube y facilitar el despliegue de servicios V2X. Sin embargo, este despliegue aumenta la latencia, ya que los paquetes tienen que atravesar toda la red y se requiere acceso a Internet para llegar al nodo de procesamiento.

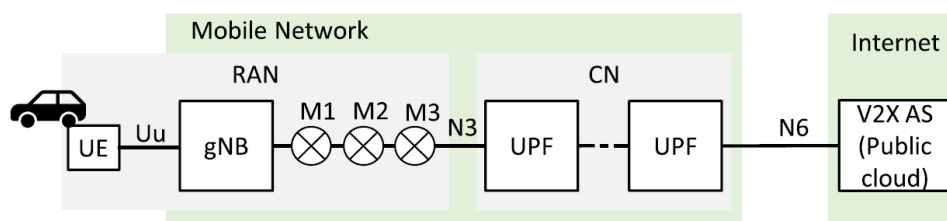


Figura 2.5. Escenarios de despliegue de 5G V2X - *Centralized*.

- Core Central Office: El servidor se ubica dentro de la red central (CN) 5G de la propia operadora como se muestra en la Figura 2.6, pero todavía está lejos de los vehículos ya que en este despliegue se requiere enrutar los paquetes V2X a través de la red de transporte.

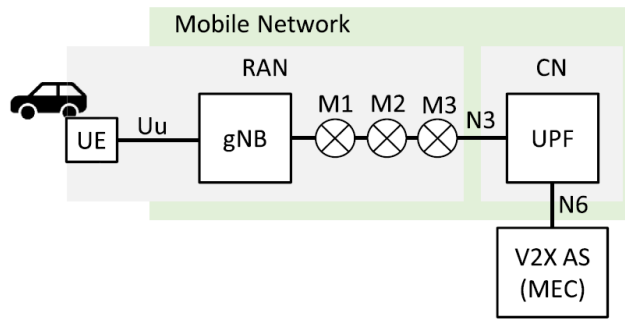


Figura 2.6. Escenarios de despliegue de 5G V2X - Core Central Office.

- Central Office: Aunque el servidor ya está más cerca, concretamente dentro de la red de acceso por radio (RAN), todavía podemos reducir más la distancia entre los vehículos el nodo de procesamiento de los datos.

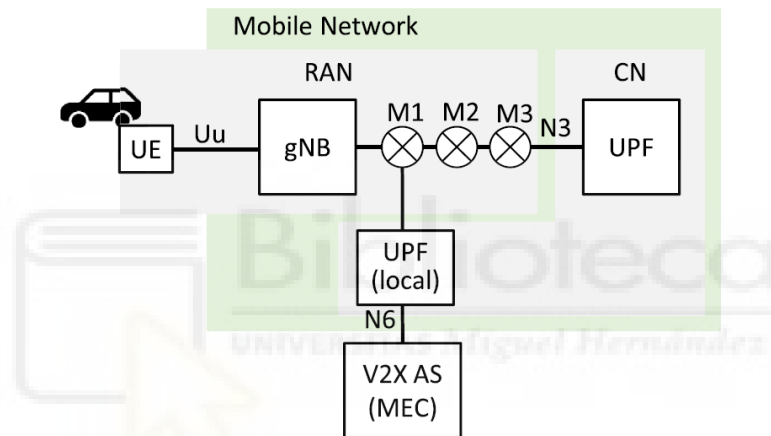


Figura 2.7. Escenarios de despliegue de 5G V2X - Central Office

- Cell site: En este último despliegue (Figura 2.8), el servidor se ubica prácticamente en la estación base (gNB), por lo que la distancia física que tienen que recorrer los mensajes C-ITS es claramente menor, y esto se va a ver reflejado en los tiempos de latencia.

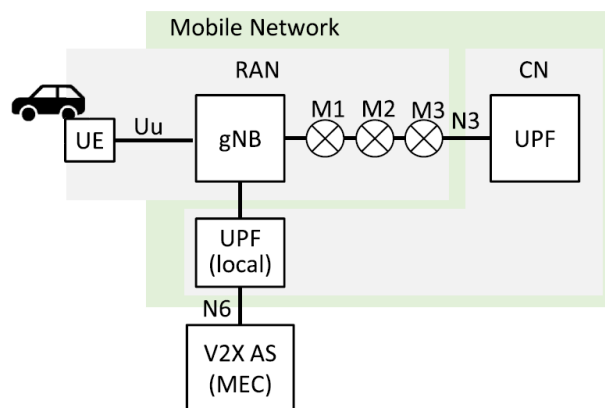


Figura 2.8. Escenarios de despliegue de 5G V2X - Cell Site

Como se puede observar, ante escenarios de comunicaciones vehiculares que requieran una rápida respuesta, la implantación de nodos *edge computing* en el borde de la red, facilita una comunicación en tiempo real, lo cual es esencial para responder a las dinámicas cambiantes del tráfico y garantizar la seguridad vial. Por lo que, la combinación de la tecnología 5G y el *edge computing* para el procesamiento de los datos, permite crear un ecosistema de comunicación vehicular mucho más rápido, seguro y eficiente.

2.3. GEODIFUSIÓN

Si se utiliza la red celular 5G para la difusión de mensajes C-ITS con el fin de aprovechar las grandes ventajas que ofrece, debemos tener en cuenta que el 5G no soporta el concepto de la geodifusión, lo cual es importante en el contexto de los sistemas de transporte inteligentes y la comunicación vehicular conectada para conocer la ubicación de sus nodos dentro de la red. Por tanto, aprovecharemos esta ventajosa característica para integrarla dentro del nodo *Edge*.

La geodifusión se refiere a la transmisión de información relevante a una región geográfica específica o a aquellos dispositivos que se encuentren dentro de dicha región. En un mundo donde la cantidad de datos generados es abrumadora, es esencial que los sistemas de comunicación sean capaces de filtrar y transmitir solo la información que es pertinente para un receptor en particular, en función de su ubicación geográfica.

Debemos pensar que un sistema que albergue información de todas las grandes ciudades europeas daría lugar a un flujo de datos inmanejable, por lo que la geodifusión debe garantizar que sólo se transmitan aquellos mensajes relevantes para la ubicación del receptor.

Una de las soluciones propuesta en lo que respecta a la geodifusión, fue el uso de los mensajes CAM enviados por los vehículos en combinación con una base de datos SIG (Sistema de Información Geográfica) como es PostGIS. Esta fue, la primera implementación propuesta por MOBiNET [6]. Sin embargo, esta solución presenta dos desventajas. La primera es que, al cargar los mensajes CAM, se introduce un riesgo innecesario para la privacidad de los usuarios que sólo desean información del sistema, ya que se les estaría exigiendo que carguen información en un servicio. La segunda es que el intercambio continuo de información CAM da lugar a muchas consultas a la base de datos del SIG, lo que hace que el sistema sea menos eficiente.

El proyecto CONVERGE [7] encontró una mejor solución introduciendo el concepto de cuadrícula o quadtree. Con este método, los usuarios no necesitan enviar datos CAM que incluyan su ubicación. Un vehículo simplemente se registra (suscribe) una vez y recibe los bordes de su cuadrícula actual. Una vez que abandona el quadtree, se comunica nuevamente con el servidor para obtener nuevas coordenadas. En este caso, la tarea se vuelve imposible en un sistema con muchos usuarios. Además, no utilizar CAMs tiene el beneficio adicional de no tener datos adicionales que puedan identificar al usuario, como la longitud y el ancho del vehículo.

Una planificación eficiente de los quadtrees podría minimizar esta desventaja. El principal inconveniente es que la difusión se limita a una combinación de quadtrees, lo que puede dar lugar a un área de difusión ligeramente mayor de lo previsto inicialmente. Pero aun así, el tener que enviar los bordes de esos quadtrees a cada vehículo cada vez que cambien de cuadrado, puede llegar a generar mucha carga de mensajes, por lo que tampoco es una opción viable para el caso del presente proyecto.

Sin embargo, en proyectos como 5G-CroCo [8], como solución a la ineficiencia del sistema de quadtrees propuesto en el proyecto del párrafo anterior, proponen no intercambiar las coordenadas de los bordes de los cuadrados con los vehículos, sino que ahora el vehículo obtiene su propia ubicación gracias al GPS (*Global Positioning System*), y la utiliza como base para conformar el identificativo o “quadkey” del quadtree en el que se encuentra. Para ello, se siguen las especificaciones establecidas por C-Roads en el documento “C-ITS IP Based Interface Profile Version 2.0.5 – Appendix A. Quadtrees” [9], donde indica cómo calcular esos quadkeys. Esto permite que cualquier sistema conozca los cuadrados relevantes para su ubicación actual.

En la búsqueda de soluciones innovadoras para la difusión de mensajes C-ITS en función de la localización geográfica de los vehículos, la idea que proponen en este último proyecto resulta de gran interés, concretamente, su enfoque en la eficiencia del sistema de quadtrees para la localización geográfica. Por lo tanto, nuestro proyecto busca desarrollar una solución inspirada en estos avances pero adaptada a nuestras propias especificaciones y requerimientos para abordar el desafío de difundir mensajes a nodos específicos en función de su localización. La estructura basada en quadtrees nos ayudará a reducir la carga de comunicación y eliminar la necesidad de actualizaciones periódicas de la posición de los usuarios finales a un servidor o base de datos.

2.4. PROTOCOLOS PARA LA DIFUSIÓN DE MENSAJES C-ITS

A continuación, se pretende elegir un protocolo adecuado para la difusión de mensajes a través de la red celular que sea capaz de adaptarse y soportar los diferentes contextos y escenarios vehiculares que se presenten. Debe garantizar una comunicación clara y sin pérdidas de información.

Los mensajes C-ITS, incluidos los CAM y DENM mencionados anteriormente, son los que nos van a proporcionar aquella información crítica sobre el estado y las condiciones del entorno de tráfico. Por lo tanto, estos mensajes se deben enviar y recibir correctamente para garantizar la seguridad y eficiencia del sistema de transporte.

Un protocolo de mensajería define las reglas y convenciones que deben seguirse para el envío y la recepción de mensajes. Esto incluye aspectos como el formato de los mensajes, la manera en que se deben enviar y recibir, o cómo se deben manejar los errores de transmisión.

El protocolo de mensajería elegido debe ser robusto y fiable, capaz de manejar grandes volúmenes de datos y garantizar que los mensajes se entreguen en tiempo real o con el mínimo retraso posible. Además, debe ser resistente y capaz de garantizar la integridad y la confidencialidad de los datos transmitidos. Esto es especialmente importante en el contexto de los C-ITS, donde la información transmitida puede ser crítica para la seguridad de los usuarios de la carretera.

También hay que tener en cuenta la escalabilidad del protocolo para la difusión de estos mensajes. Con el aumento de vehículos conectados y la creciente demanda de servicios de transporte inteligentes, es necesario que pueda manejar un creciente volumen de datos y usuarios sin degradar su rendimiento.

Vamos a analizar los 2 protocolos más utilizados en este contexto para poder conocer cual es el más conveniente para la implementación:

2.4.1. MQTT (*Message Queuing Telemetry Transport*)

MQTT es un protocolo de mensajería basado en estándares, o un conjunto de reglas, que se utiliza para el intercambio de información entre dos dispositivos [10]. Los sensores inteligentes, los dispositivos portátiles y otros dispositivos de Internet de las cosas (IoT) generalmente tienen que transmitir y recibir datos a través de una red con recursos restringidos y un ancho de banda limitado. Los dispositivos utilizan MQTT para la

transmisión de datos, ya que resulta fácil de implementar y puede comunicar datos de manera eficiente. También admite la mensajería entre dispositivos a la nube y la nube al dispositivo.

Este modelo de publicación/suscripción ligero y eficiente, lo hacen especialmente adecuado para aplicaciones en las que se requiere la entrega de mensajes en tiempo real o con el mínimo retraso posible, como es el caso de los C-ITS.

Ofrece grandes beneficios como:

- **Ligereza y eficiencia:** La implementación de MQTT requiere recursos mínimos, por lo que se puede usar incluso en pequeños microcontroladores. Por ejemplo, un mensaje de control MQTT mínimo puede tener tan solo dos bytes de datos. En general, son mensajes pequeños que minimizan la cantidad de datos transmitidos y el impacto en la red. Los encabezados de los mensajes MQTT también son pequeños para poder optimizar el ancho de banda de la red.
- **Escalabilidad:** MQTT puede manejar un gran número de clientes simultáneamente, lo que lo hace escalable y adecuado para entornos con un gran número de vehículos conectados y dispositivos de infraestructura de transporte. Además, su naturaleza descentralizada permite distribuir la carga entre diferentes servidores (brokers) y evitar cuellos de botella.
- **Fiabilidad:** MQTT ofrece diferentes niveles de calidad de servicio (QoS) que permiten adaptar la entrega de mensajes a las necesidades de cada aplicación, desde la entrega "al menos una vez" (0) hasta la entrega "exactamente una vez" (2). Esto garantiza que los mensajes críticos se entreguen de manera confiable y oportuna, incluso en redes no fiables.
- **Seguridad:** MQTT facilita a los desarrolladores el cifrado de mensajes y la autenticación de dispositivos y usuarios mediante protocolos de autenticación modernos, como OAuth, TLS1.3, certificados administrados por el cliente, etc.

Este protocolo se inventó en 1999 para su uso en la industria del petróleo y el gas. Los ingenieros necesitaban un protocolo para un ancho de banda mínimo y una pérdida de batería mínima para supervisar los oleoductos vía satélite. Inicialmente, el protocolo se conocía como transporte de telemetría de *Message Queue Server* debido al producto de IBM MQ Series que admitió por primera vez su fase inicial. En 2010, IBM lanzó MQTT

3.1 como un protocolo gratuito y abierto para que cualquiera pudiera implementarlo, que después, en 2013, se envió al organismo de especificación de la Organización para el Avance de Estándares de Información Estructurada (OASIS) para su mantenimiento. En 2019, OASIS lanzó una versión 5 de MQTT actualizada. Ahora MQTT ya no es un acrónimo sino que se considera el nombre oficial del protocolo [10].

En la comunicación de red tradicional, los clientes y servidores se comunican directamente entre sí. Sin embargo, MQTT utiliza un patrón de publicación – suscripción para desacoplar el remitente del mensaje (publicador) del receptor del mensaje (suscriptor). En lugar de ello, un tercer componente, denominado *broker*, controla la comunicación entre publicadores y suscriptores. El trabajo del broker consiste en filtrar todos los mensajes entrantes de los publicadores y distribuirlos correctamente a los suscriptores. Por lo tanto, el protocolo MQTT define una serie de entidades:

Cliente MQTT

Un cliente MQTT es cualquier dispositivo, desde un servidor hasta un microcontrolador, que ejecuta una biblioteca MQTT. Si el cliente envía mensajes, actúa como publicador, y si recibe mensajes, actúa como receptor. Básicamente, cualquier dispositivo que se comunique mediante MQTT a través de una red puede denominarse dispositivo cliente MQTT.

Broker MQTT

El broker MQTT es el sistema de *back-end* que coordina los mensajes entre los diferentes clientes. Las responsabilidades del broker incluyen recibir y filtrar mensajes, identificar a los clientes suscritos a cada tema (o tópico) y enviarles dichos mensajes. También se encarga de otras tareas como:

- La autorización y autenticación de clientes MQTT
- Pasar mensajes a otros sistemas para su posterior análisis
- El control de mensajes perdidos y sesiones de clientes

Conexión MQTT

Los clientes y los brokers comienzan a comunicarse mediante una conexión MQTT. Los clientes inician la conexión al enviar un mensaje *CONNECT* al broker MQTT. El broker confirma que se ha establecido una conexión al responder con un mensaje *CONNACK*. Los clientes nunca se conectan entre sí, solo con el agente.

Tema (o t3pico) MQTT

El t3rmino “tema” se refiere a las palabras clave que utiliza el agente MQTT a fin de filtrar mensajes para los clientes de MQTT. Los temas est3n organizados jer3rquicamente, de forma similar a un directorio de archivos o carpetas. Por ejemplo, considere un sistema dom3stico inteligente que opera en una casa de varios pisos que tiene diferentes dispositivos inteligentes en cada uno de ellos. En ese caso, es posible que el broker MQTT organice temas como:

Casa/planta0/sal3n/luz
Casa/planta1/habitacion2/temperatura

Publicaci3n MQTT

Los clientes MQTT publican mensajes que contienen el tema y los datos en formato de bytes. El cliente determina el formato de los datos, como datos de texto, datos binarios, archivos XML o JSON.

Suscripci3n MQTT

Los clientes MQTT env3an un mensaje “SUBSCRIBE” al broker MQTT para recibir mensajes sobre temas de inter3s. Contiene un identificador 3nico y una lista de suscripciones. Por ejemplo, para saber cu3ntas luces est3n encendidas en una casa, basta con que el cliente MQTT se suscriba al tema “*CasaID/Luz*”.

A continuaci3n, en la Figura 2.9 podemos observar un ejemplo de lo que ser3a la arquitectura de este protocolo de mensajer3a:

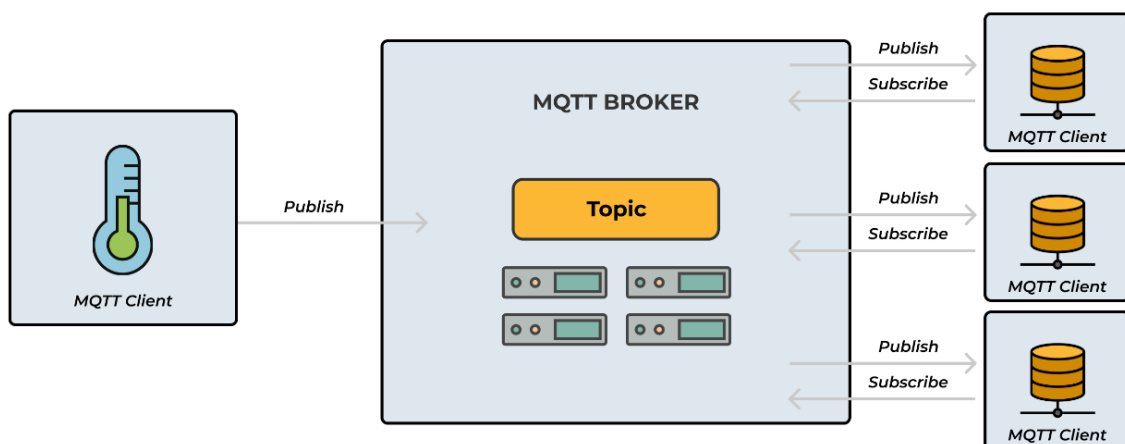


Figura 2.9. Arquitectura del protocolo MQTT

2.4.2. AMQP (*Advanced Message Queuing Protocol*)

AMQP es un protocolo de capa de aplicación diseñado para facilitar la comunicación entre diferentes sistemas de mensajería. Fue desarrollado por un consorcio de empresas líderes en la industria y se ha convertido en un estándar ampliamente aceptado en el ámbito de la mensajería empresarial.

El protocolo funciona sobre TCP/IP y proporciona una forma eficiente y confiable de enviar mensajes entre aplicaciones distribuidas. Utiliza un modelo de intercambio de mensajes basado en colas, donde los mensajes se envían a una cola y luego se entregan a los consumidores de manera ordenada. El proceso de comunicación en el AMQP involucra a tres componentes principales: el productor/publicador, el intermediario y el consumidor. El productor/publicador es la aplicación que envía mensajes, el intermediario (o broker) es el sistema de mensajería que se encarga de almacenar y enrutar los mensajes, y el consumidor es la aplicación que recibe los mensajes. En la Figura 2.10 podemos observar cómo sería la arquitectura que sigue:

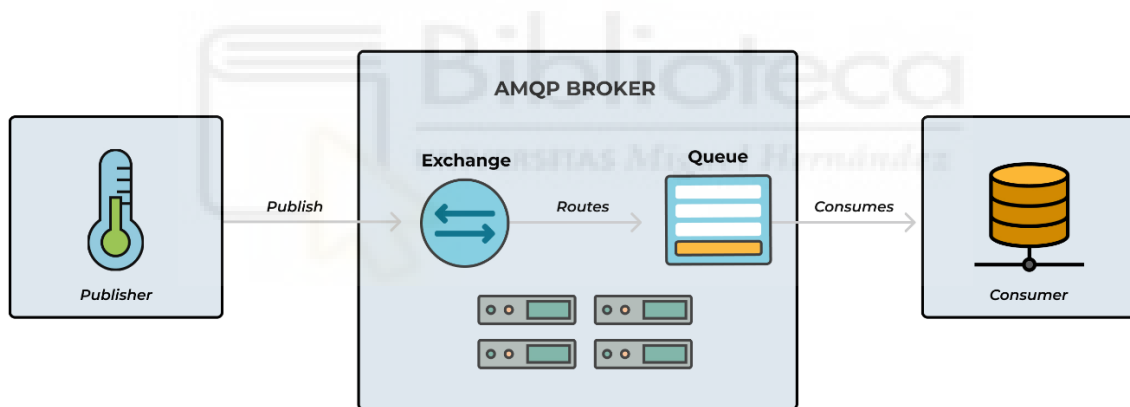


Figura 2.10. Arquitectura del protocolo AMQP

El AMQP utiliza una arquitectura cliente-servidor, donde los productores y consumidores se conectan al intermediario a través de conexiones TCP/IP. Los mensajes se envían a través de canales lógicos llamados sesiones, y cada sesión puede contener múltiples colas y mensajes. El protocolo AMQP define reglas para la estructura y el formato de los mensajes, así como para la gestión de colas, la entrega de mensajes y la confirmación de recepción. También proporciona mecanismos de seguridad y autenticación para garantizar la integridad y la confidencialidad de los mensajes.

El AMQP es altamente flexible y escalable, lo que lo convierte en una excelente opción para entornos empresariales donde se requiere una comunicación fiable y eficiente.

Permite la interoperabilidad entre diferentes sistemas de mensajería y proporciona una capa de abstracción que permite a las aplicaciones comunicarse sin preocuparse por los detalles de implementación subyacentes.

Respecto a las colas, cada una tiene su propio nombre que la identifica entre las demás entidades. El nombre puede ser definido por un cliente o por el broker de manera automática. Una cola es un fragmento de memoria que puede estar asociada con almacenamiento no volátil o en la memoria RAM del sistema. La versión no volátil asegura que, incluso después de reiniciar el broker, la cola persiste. Sin embargo, no garantiza que los mensajes se guarden de forma permanente, ya que depende de la configuración de los mensajes cómo serán tratados.

El mecanismo intrínseco de AMQP permite enviar mensajes a las colas y garantizar su procesamiento, ya que tiene un mecanismo de ACK - acuse de recibo - con el que es posible confirmar la recepción y procesamiento adecuado del mismo. Si un mensaje que tiene asociado un ACK no es confirmado por un consumidor - ya sea porque se perdió la conexión, porque se produjo un fallo de la aplicación al procesar, o cualquier otro problema - el broker vuelve a encolar el mensaje nuevamente, de manera tal que pueda ser consumido y procesado nuevamente. También es posible rechazar la aceptación de un mensaje - conocida como NACK - conscientemente. Esto puede ser útil cuando el procesamiento del mensaje no funciona. La respuesta del consumidor implica que el broker pueda borrar el mensaje o bien enviarlo a otras colas.

El protocolo AMQP fue definido entre 2004-2006 por el banco JP Morgan Chase e *iMatix Corporation* [11], quienes también desarrollaron implementaciones en C/C++ y Java. Y aunque tiene sus orígenes en la industria financiera, han logrado implementar un estándar abierto y neutral que se utiliza para facilitar la comunicación eficiente y confiable entre diferentes sistemas y aplicaciones en entornos distribuidos.

2.4.3. ANÁLISIS

En esta sección hemos visto que MQTT y AMQP son protocolos de mensajería diseñados para facilitar la comunicación entre sistemas distribuidos [12]. Ambos presentan sus similitudes y sus diferencias, y eso es lo que vamos a analizar a continuación con el objetivo de elegir aquel que sea más adecuado para la difusión de mensajes a través de las redes celulares 5G.

Similitudes

- Modelo de publicación-suscripción:

Tanto AMQP como MQTT admiten el patrón de mensajería publicación/suscripción. Un mensaje de un publicador puede difundirse a varios suscriptores. Aunque los dos protocolos difieren claramente en cómo implementan el mecanismo de publicación-suscripción (como veremos más adelante), al final, el patrón de mensajería se puede conseguir con ambos brokers.

- Mensajería asíncrona:

Tanto AMQP como MQTT soportan la comunicación asíncrona. En este patrón de comunicación, el publicador envía mensajes al broker sin esperar una respuesta de éste. Como resultado, tanto el publicador como el suscriptor no tienen que estar en línea al mismo tiempo para poder comunicarse. Esto es posible porque, en ambos protocolos, el broker se sitúa entre el editor y el suscriptor/consumidor - desacoplando totalmente las dos entidades.

- Calidad de servicio (QoS):

Ambos protocolos ofrecen varios niveles de calidad de servicio para garantizar una entrega fiable de los mensajes. Ofrecen opciones de semántica de mensaje "*at-most-once*", "*at-least-once*" y "*exactly-once*". Esto permite a los desarrolladores elegir el nivel de fiabilidad que mejor se adapte a los requisitos de su aplicación.

Diferencias

- Patrones de mensajería:

Como se ha mencionado anteriormente, el protocolo AMQP soporta un mecanismo de enrutamiento más sofisticado. Los mensajes van primero a un intercambiador (*exchange*) que, a su vez, enruta los mensajes a la cola correcta utilizando algunas reglas predefinidas. En cambio, el protocolo MQTT tiene un mecanismo de enrutamiento más sencillo. Los mensajes se publican en un tema y se difunden a todos los consumidores registrados en él. En MQTT no hay intercambios ni colas de espera. Aunque ambos se basan en un modelo de publicación/suscripción, el protocolo MQTT lo hace de manera más eficiente. Esto es cierto porque, para implementar el modelo publicación/suscripción con AMQP, en primer lugar, declara un intercambio y también varias colas, y después vincularlas al *exchange*. Esta capa adicional de uso

de *exchanges* y colas introduce una sobrecarga en este escenario que podría ser fácilmente minimizada si no eliminada con el protocolo MQTT.

- Flexibilidad y complejidad:

Además de su sofisticado mecanismo de enrutamiento, AMQP ofrece un rico conjunto de características. Entre ellas, la persistencia de mensajes y la transaccionalidad. Esto lo hace muy versátil, ya que puede utilizarse en una amplia gama de casos. Pero el inconveniente es que es más complejo de implementar y gestionar. MQTT, por otro lado, se centra en la simplicidad y la facilidad de uso, sacrificando algunas funciones avanzadas para agilizar la comunicación.

- Sobrecarga y eficiencia:

El rico conjunto de características de AMQP y su sofisticado mecanismo de enrutamiento proporcionan una funcionalidad avanzada a costa de una mayor sobrecarga. AMQP requiere una huella de red y unos recursos informáticos más importantes que MQTT. Esto es así porque MQTT está diseñado para ser ligero y eficiente, priorizando la optimización del ancho de banda y la energía.

Tras analizar cuidadosamente ambos protocolos y considerar los factores mencionados anteriormente, se ha seleccionado el protocolo MQTT como el protocolo de mensajería más adecuado para la implementación y difusión de mensajes C-ITS a través de la red celular. Esta decisión coincide con la tomada en el proyecto del departamento en el que se enmarca este trabajo.

3. DISEÑO E IMPLEMENTACIÓN DEL GEOSERVICE

En el siguiente capítulo, abordaremos el diseño e implementación del geoservicio, comenzando con su concepto y finalidad (apartado 3.1.), profundizaremos a lo largo del apartado 3.2. en su funcionamiento interno, analizando tanto los aspectos tecnológicos que fundamentan su diseño como la solución propuesta para la implementación. También se hablará de un aspecto muy relevante en el apartado 3.3. de cara al geoservicio, el nivel de zoom. En el apartado 3.4. se explicará cómo el *geoservice* se integra en la arquitectura del nodo *Edge*, detallando los componentes que lo conforman y la interfaz de comunicación entre el *geoservice* y uno de los elementos de dicho nodo, la “aplicación V2X”. Finalmente, en el apartado 3.5. se hablará sobre el cliente MQTT que se ha diseñado para poder recibir desde el nodo *Edge*, aquellos mensajes de alerta que sean relevantes para la posición geográfica de los vehículos.

3.1. FINALIDAD DEL GEOSERVICE

La función principal del *geoservice* es permitir una difusión eficiente y selectiva de mensajes de alerta sobre situaciones relevantes de tráfico en función de la posición geográfica de los nodos/vehículos a través de la red celular 5G, aprovechando las mínimas latencias y gran ancho de banda que ofrece al combinarse con la tecnología *edge computing*, lo cual garantizará la rápida respuesta que requieren las comunicaciones V2X.

Para lograr esto, proponemos utilizar un protocolo de mensajería ligero como es MQTT, conocido por su eficiencia y capacidad de manejar grandes volúmenes de datos, junto a la estructura jerárquica de datos quadtree, la cual nos va a permitir organizar y dividir recursivamente el espacio en cuadrantes, con el fin de representar de forma efectiva datos espaciales bidimensionales en un mapa.

En la actualidad, el número de vehículos en nuestras carreteras ha experimentado un crecimiento exponencial, lo que ha llevado a graves problemas como pueden ser las retenciones o los accidentes tráfico. Por esta razón, este tipo de tecnología puede ser un aliado crucial para mejorar la eficiencia y seguridad del transporte. Gracias a que el *geoservice* consiste en una herramienta que combina la información geoespacial con la comunicación y la conectividad vehicular, es posible mitigar este tipo de problemas enviando mensajes de alerta y notificaciones en tiempo real a aquellos vehículos que se encuentren dentro de la zona afectada por un determinado evento de tráfico, proporcionando así una difusión controlada de información.

La información geoespacial, se convierte así en un pilar fundamental para este servicio. Los dispositivos GPS incorporados en vehículos e infraestructuras de telecomunicación, nos van a permitir el acceso en tiempo real, a una gran cantidad de información precisa sobre la ubicación y el movimiento de los vehículos. Esta será utilizada para determinar los nodos a los que se deberá enviar la notificación pertinente ante un evento relevante de tráfico, como puede ser un accidente, un vehículo detenido en el arcén, una calle cortada por obras, un vehículo circulando a baja/alta velocidad, etc. En la Figura 3.1 podemos observar un esquema a modo ejemplo donde se intenta mostrar que sólo los vehículos ubicados en la zona afectada recibirán el mensaje:



Figura 3.1. Ejemplo ilustrativo de la difusión selectiva de mensajes de alerta

Esta difusión selectiva de mensajes que previenen a los conductores sobre determinadas alertas de tráfico es especialmente útil para mejorar la seguridad vial. Al enfocarse en grupos específicos de vehículos, se puede reducir significativamente la probabilidad de accidentes y congestión, a la vez que se optimiza el flujo de tráfico.

3.2. FUNCIONAMIENTO INTERNO DEL GEOSERVICE

Tras comprender la importancia y el propósito que busca satisfacer el *geoservice* en el contexto de la conectividad vehicular y la tecnología 5G, nos centraremos en su estructura y mecánica interna, desglosando su arquitectura y los componentes que le permiten llevar a cabo una difusión selectiva de mensajes con precisión y eficiencia. Así, el siguiente apartado, proporcionará una visión detallada de cómo este servicio opera e interactúa.

3.2.1. ASPECTOS TECNOLÓGICOS DE DISEÑO

Para garantizar una implementación eficaz y eficiente del geoservicio se han adoptado decisiones tecnológicas y arquitectónicas específicas. Estas decisiones se orientan no solo a cumplir con los objetivos propuestos, sino también a facilitar la integración, escalabilidad y mantenimiento del sistema en el futuro. Veamos en detalle los aspectos clave de este diseño:

Servicio REST

El *geoservice* ha sido diseñado como un servicio REST (*Representational State Transfer*). En términos prácticos, esto significa que el servicio utiliza una arquitectura software que se basa en el protocolo HTTP para el intercambio de datos a través de Internet con otros componentes ubicados dentro del nodo *Edge*.

Esta arquitectura se basa en el modelo cliente-servidor donde el cliente es el que solicita obtener los recursos o realizar alguna operación sobre dichos datos, mientras que el servidor es aquel ente que entrega o procesa dichos datos a solicitud del cliente. En este caso, el *geoservice* será quien actúe como servidor y aquel componente del nodo *edge* que consume su información, hará el papel de cliente.

Lenguaje Java

El lenguaje de programación Java ha sido el utilizado para el desarrollo del *geoservice*. Como bien es sabido, Java es un lenguaje de programación de propósito general que ha sido ampliamente adoptado en el desarrollo de software debido a su portabilidad, versatilidad y seguridad. Algunas ventajas clave de Java incluyen su capacidad para funcionar en diferentes plataformas sin necesidad de recompilar el código, su gran comunidad de desarrolladores y la disponibilidad de numerosas bibliotecas y marcos de trabajo que facilitan el desarrollo de aplicaciones complejas [13].

Framework Spring Boot

Para la implementación del *geoservice* se ha elegido Spring Boot como framework principal. Este consiste en un framework de código abierto basado en Java que facilita la creación de aplicaciones web y servicios REST de manera rápida y sencilla. Proporciona una configuración por defecto con una amplia gama de herramientas y complementos preintegrados, y una arquitectura lista para usar [13], lo que permite a los desarrolladores concentrarse en la lógica en lugar de ocuparse de detalles de infraestructura. Spring Boot además, favorece la modularidad y el mantenimiento lo que lo convierte en una elección

popular para el desarrollo ágil de aplicaciones y servicios. Por este motivo, el servicio desarrollado no solo es eficiente, sino también robusto y seguro, y está preparado para adaptarse y crecer según las necesidades del entorno vehicular.

Protocolo de mensajería MQTT

Tal y como ya se indicó, el protocolo MQTT ha sido elegido para gestionar el envío de mensajes debido a su diseño ligero y eficiente, siendo ideal para situaciones donde se requiere una entrega rápida y confiable de datos. Particularmente, el tipo de alerta que se espera recibir cuando se produzca un evento relevante de tráfico será un mensaje DENM.

Recordemos que dicho protocolo se basa en un modelo de publicación/suscripción. Esto significa que un dispositivo (cliente MQTT) puede suscribirse a un tópico (o tema) específico y otro dispositivo puede publicar un mensaje en ese tópico. Cuando esto ocurre, todos los dispositivos suscritos al tópico reciben el mensaje. Esta tecnología es la que va a aprovechar el geoservice para la entrega de mensajes DENM cuando se produzca una situación de tráfico que puedan afectar la seguridad vial.

3.2.2. SOLUCIÓN PROPUESTA

División del mapa en quadrees

En el ámbito de la conectividad vehicular, la difusión de mensajes es esencial para mantener informados a los conductores sobre eventos relevantes que puedan afectar su trayectoria o seguridad. Sin embargo, la difusión masiva de mensajes puede conducir a la saturación de la red y a la entrega de información no relevante para ciertos usuarios.

Para abordar este desafío, la solución propuesta introduce en primer lugar, la división del mapa o zona de interés en quadrees. Como ya indicamos, un quadtree es una estructura de datos jerárquica en la que cada nodo padre tiene exactamente cuatro hijos. Esta estructura se utiliza comúnmente para particionar un espacio bidimensional dividiéndolo de manera recursiva en cuatro cuadrantes o regiones iguales.

El tamaño de los quadrees vendrá determinado por el llamado “Nivel de Zoom”, donde cada uno de estos niveles incrementa el detalle del mapa, duplicando el número de mosaicos o hijos en cada dimensión. En páginas posteriores se aportará información más detallada sobre el nivel de zoom y su importante papel. De modo que, al aplicar este concepto, podemos segmentar el espacio vial en zonas afectadas o no afectadas por un evento de tráfico y evitar sobrecarga de información en áreas en las que no sea relevante.

En la siguiente imagen (Figura 3.2) podemos observar en color rojo lo que se llama “nodo padre”, que representa un nivel de zoom 0. Si aumentamos el nivel a 1, el nodo padre se divide en cuatro nodos hijos (cuadrados verdes). A medida que aumentemos el nivel de zoom, cada uno de los nodos hijos se convertirá en nodo padre y se subdividirá en otros 4 nodos hijo, y así sucesivamente hasta alcanzar el nivel de zoom deseado:

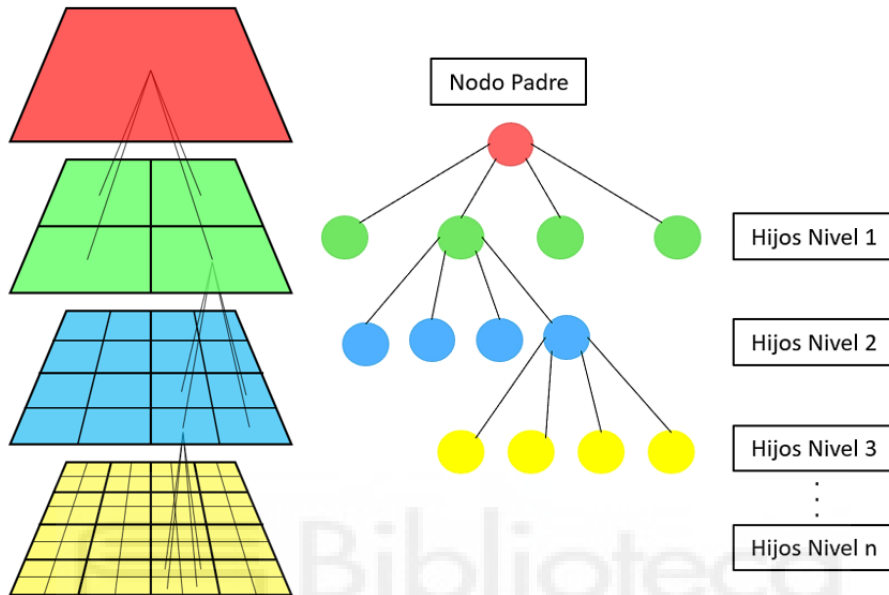


Figura 3.2. Estructura quadtree

Obtención de los tópicos afectados por un evento

Cada uno de esos quadtrees, o segmentos específicos del mapa, será identificado mediante un tópico MQTT único. Esta identificación no solo garantiza una referencia clara y singular para cada zona, sino que también facilita la organización de los datos relacionados con esa región.

El geoservice tiene integrado una serie de funciones y operaciones en su código que le permite calcular los tópicos afectados por un evento. Para ello, necesita de 3 datos: *latitud*, *longitud* y *distancia de relevancia*. Este conjunto de datos será proporcionado al geoservice para que pueda realizar su labor.

Latitud y longitud

Ambas coordenadas geográficas serán expresadas en grados decimales e indicarán la ubicación exacta del lugar donde se ha producido un determinado evento (accidente de tráfico, retenciones, vehículo averiado en la calzada, etc.). Por ejemplo:

(38.2681, -0.70819)

Distancia de relevancia

Cada evento, tiene asociada una distancia de relevancia, la cual indica la distancia máxima (en metros) a la que un vehículo puede estar del evento y aún considerarse relevante. Dicho dato será proporcionado al geoservice como un valor discreto entre 0 y 7 (así está establecido en el estándar ETSI), donde cada uno de esos valores tiene asociado un determinado valor en metros:

Tabla 1. Valores distancia de relevancia

<i>relevanceDistance</i>			
value	meters	value	meters
0	50	4	1.000
1	100	5	5.000
2	200	6	10.000
3	500	7	> 10.000

El geoservice utilizará este dato para crear un “área de relevancia”. Tomará el valor de la distancia de relevancia como un radio y creará alrededor del punto donde se ha producido un evento, un área circular a la que vamos a llamar “área de relevancia”. De modo que, el geoservice obtendrá los tópicos de aquellos quadrees que interseccionen con dicha área. En la Figura 3.3, en la izquierda podemos observar un área de relevancia interseccionando con dos quadrees, por lo que el geoservice sólo va a devolver los tópicos correspondientes a esas dos zonas. Con esto evitamos notificar a vehículos para los que no es relevante un determinado evento. En cambio, a la derecha de la figura, el área de relevancia es mayor, por lo que el evento ahora afecta a 4 quadrees, que serán a los que se dirijan las notificaciones de alerta pertinentes.

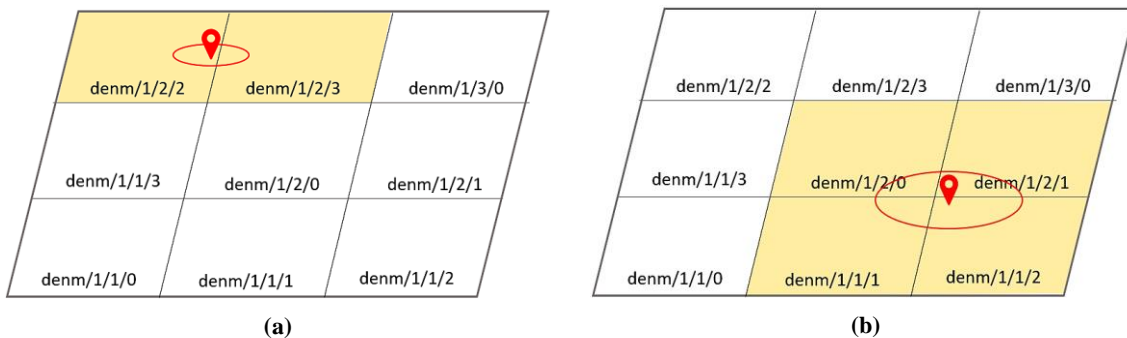


Figura 3.3. Quadrees afectados en función del área de relevancia

la escala y el área especialmente cerca de los polos debido a que representa las líneas de latitud (paralelos) como líneas horizontales equidistantes entre sí, se definirá los valores de latitud entre -85,05112878 y 85,05112878 grados y los valores de longitud entre -180 y +180 grados decimales, evitando así esa singularidad en los polos y logrando que el mapa proyectado sea cuadrado [14].

Para calcular esos quadkeys que identifican de manera única a cada quadtree, comenzamos convirtiendo las coordenadas geográficas del lugar donde se ha producido el evento a “Coordenadas de Quadtree”, las cuales se presentan como un par (X, Y):

- La coordenada X denota la posición horizontal del mosaico desde el extremo izquierdo del mapa en ese nivel de zoom.
- La coordenada Y denota la posición vertical del mosaico desde la parte superior del mapa en ese nivel de zoom.

De modo que van desde (0, 0) en la esquina superior izquierda a $(2^{zoom} - 1, 2^{zoom} - 1)$ en la esquina inferior derecha. Por ejemplo, para un nivel de zoom 2, las coordenadas del

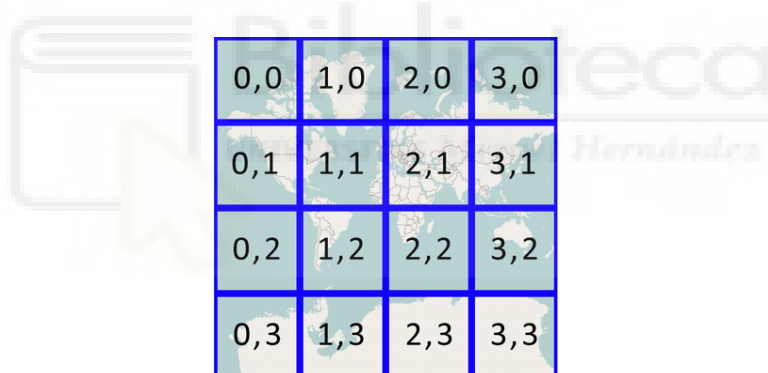


Figura 3.5. Ejemplo Coordenadas (X, Y) para nivel de zoom 3

mosaico van de (0, 0) a (3, 3) como se ve en la Figura 3.5:

Las operaciones a realizar para la conversión a coordenadas (X, Y) son las siguientes:

$$X = \left(\frac{\text{longitud} + 180}{360} \right) \times 2^{\text{zoom}}$$

$$Y = \left(0,5 - \frac{\log \left(\frac{1 + \text{sen} \left(\text{latitud} \times \frac{\pi}{180} \right)}{1 - \text{sen} \left(\text{latitud} \times \frac{\pi}{180} \right)} \right)}{4\pi} \right) \times 2^{\text{zoom}}$$

Ahora, convertiremos esas coordenadas (X, Y) en números binarios (X, Y)₂. Después se intercalarán los bits de Y con los de X, y el resultado se interpretará como un número en base 4 (donde se mantienen los ceros iniciales de modo que el número de bits sea igual al nivel de zoom), la cadena obtenida será el quadkey correspondiente del quadtree que contenga ese punto. Veamos un ejemplo para que se entienda correctamente, supongamos se ha producido un evento en las coordenadas (3,5) para un nivel de zoom 5:

$$\left. \begin{array}{l} X = 3 \rightarrow X_2 = 011 \\ Y = 5 \rightarrow X_2 = 101 \end{array} \right\} \begin{array}{l} X = 3 \rightarrow X_2 = \mathbf{00}011 \\ Y = 5 \rightarrow X_2 = \mathbf{00}101 \end{array}$$

Completamos con ceros a la izquierda para tener el mismo n° de bits que el nivel de zoom e intercalamos los bits de YX = 0000100111₂

Este número binario lo convertimos a base 4 y obtenemos: *quadkey* = 00213

- Transformación del Quadkey a tópico MQTT

Ya que los quadkeys actúan como un identificativo de cada quadtree, vamos a aprovechar esta ventaja y utilizarlos como tópicos MQTT de suscripción y publicación de mensajes DENM. Para ello, el geoservice cuenta con una función en su código para transformar esos quadkeys en tópicos. Por ejemplo:

$$\mathbf{quadkey = 00213 \rightarrow \text{tópico} = \text{denm}/0/0/2/1/3}$$

Todo lo explicado en el apartado 3.2.2, se puede resumir en el esquema de la Figura 3.6.

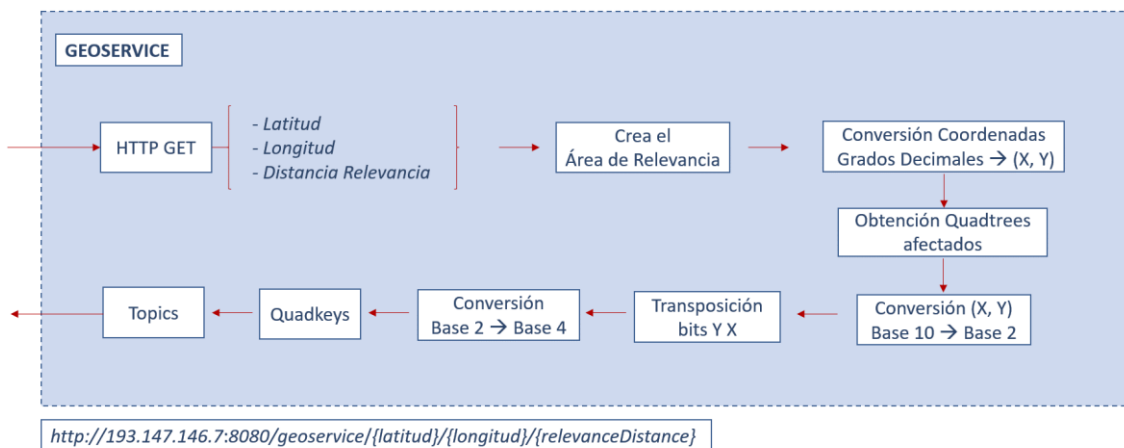


Figura 3.6. Funciones internas del geoservice

Por lo tanto, se utilizará esta funcionalidad que posee el geoservice para que se pueda publicar los mensajes DENM de alerta sólo a los tópicos que indique el servicio, logrando así esa discriminación a nivel de zonas que se desea alcanzar.

3.3. NIVEL DE ZOOM

El nivel de zoom es un parámetro fundamental que define la escala y la precisión con la que se representa y se analiza el entorno geoespacial. Tal y como se ha indicado anteriormente, un quadtree es una estructura de datos jerárquica que se utiliza para dividir un espacio bidimensional en regiones más pequeñas mediante subdivisiones recursivas, de modo que determina la profundidad de la jerarquía del quadtree.

Por lo que, con el nivel de zoom podemos establecer el tamaño de la superficie que cubrirá cada quadtree. Un nivel de zoom bajo producirá quadtrees con regiones más grandes, lo que reduce la resolución y el detalle del área de estudio, mientras que un nivel de zoom alto generará quadtrees con regiones más pequeñas, aumentando la resolución y el detalle. En la Figura 3.7 podemos observar un ejemplo del área que cubriría un quadtree de tamaño 13 a 18.

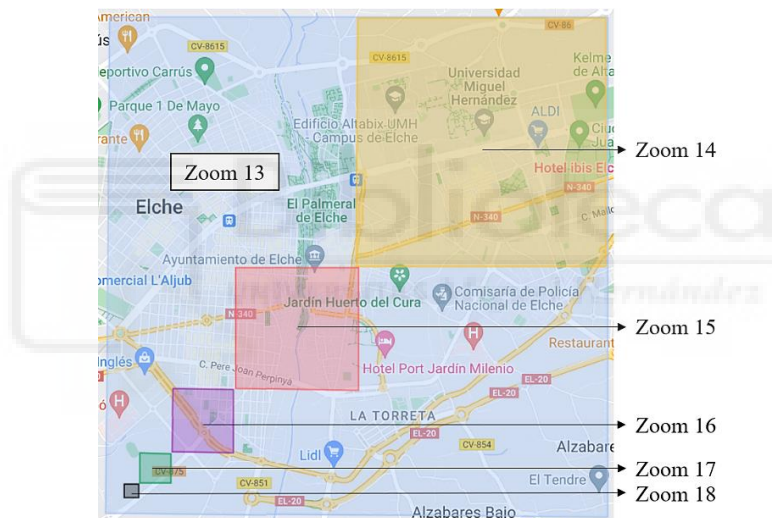


Figura 3.7. Ejemplo Quadtrees para niveles de zoom 13-18

En la Tabla 2 podemos ver la distancia en metros que tienen los quadtrees de lado en función del nivel de zoom.

Tabla 2. Distancia de cobertura de un quadtree en función del nivel de zoom

Nivel de zoom	Metros (lado)
13	4.000
14	2.000
15	1.000
16	500
17	250
18	125

Además, el nivel de zoom también tiene un impacto directo en los tópicos a los que se publicarán los mensajes y a los que se suscribirán los vehículos. Recordemos que el nivel de zoom es igual a la longitud de la quadkey. En el entorno definido, cada quadtree representa una zona geográfica específica, y los mensajes DENM se publicarán en los tópicos correspondientes a los quadtrees afectados por el evento. De igual manera, los vehículos se suscribirán a los tópicos asociados a los quadtrees en los que se encuentran, lo que permite que reciban únicamente la información relevante y oportuna para su ubicación. Es por ello que es muy importante que tanto el geoservicio como los clientes (vehículos) tengan una misma visión del mapa para recibir las alertas pertinentes en caso de producirse un evento.

Respecto a los vehículos, el nivel de zoom juega un papel importante en la frecuencia con la que se suscriben y anulan dichas suscripciones a medida que cambian de quadtree. Con un nivel de zoom alto, los quadtrees serán más pequeños y los vehículos cambiarían de quadtree con mayor frecuencia, lo que aumentaría la carga de mensajes y la cantidad de suscripciones y anulaciones de estas. Por otro lado, con un nivel de zoom bajo, los quadtrees serán más grandes, reduciendo la cantidad de suscripciones y su respectiva cancelación pero, podría ocasionar que algunos vehículos reciban información no relevante para su ubicación al notificar a zonas muy extensas.

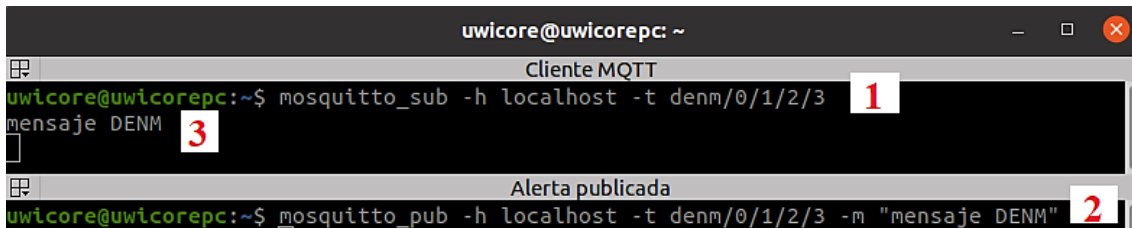
Pese a que C-Roads define una horquilla entre 13 y 18 como nivel de zoom recomendado para la notificación de mensajes DENM, no especifica un valor concreto. Esta falta de precisión representa una necesidad a tener en cuenta, ya que la elección del nivel de zoom puede tener un impacto significativo en la calidad y relevancia de la información proporcionada a los vehículos. A continuación, se muestra el estudio realizado sobre las diferentes situaciones que podrían darse suponiendo que el vehículo se encuentra dentro de un quadtree afectado por un evento:

Situación 1: Zoom Geoservice = Zoom Cliente

Tanto geoservice como vehículos utilizan el mismo nivel de zoom, es decir, ven los quadtrees en el mapa del mismo tamaño. Supongamos que tenemos un cliente suscrito al tópico denm/0/1/2/3 y sucede un evento que afecta justo a ese quadtree, ambos utilizan un nivel de zoom 4. Tal y como se muestra en la Figura 3.8, ocurriría lo siguiente:

1. Cliente se suscribe al tópico denm/0/1/2/3 (Nivel de zoom 4)
2. Se publica un mensaje DENM al tópico denm/0/1/2/3 (Nivel de zoom 4)

3. El cliente RECIBE el mensaje DENM debido a que los tópicos de suscripción y publicación coinciden.



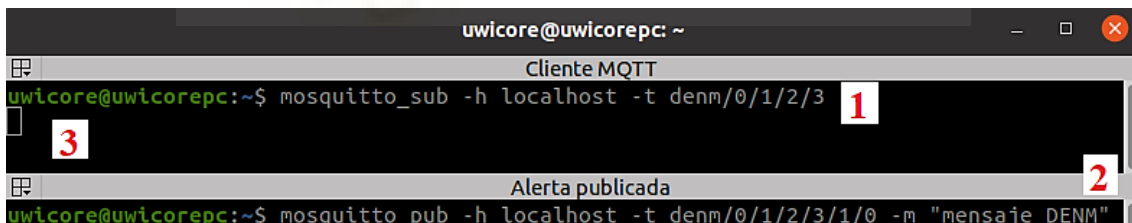
```
uwicore@uwicorepc: ~
Cliente MQTT
uwicore@uwicorepc:~$ mosquitto_sub -h localhost -t denm/0/1/2/3 1
mensaje DENM 3
Alerta publicada
uwicore@uwicorepc:~$ mosquitto_pub -h localhost -t denm/0/1/2/3 -m "mensaje DENM" 2
```

Figura 3.8. Ejemplo zoom Geoservice = zoom Cliente

Situación 2: Zoom Geoservice > Zoom Cliente

En este caso, el vehículo (cliente MQTT) trabaja con un nivel de zoom menor que el geoservice, es decir, los quadrees que el vehículo contempla cubren un área mucho mayor que este. Por lo que ocurriría lo siguiente (ver Figura 3.9)

1. Cliente se suscribe al tópico denm/0/1/2/3 (Nivel de zoom 4)
2. Se publica un mensaje DENM al tópico denm/0/1/2/3/1/0 (Nivel de zoom 6)
3. El cliente NO RECIBE el mensaje DENM debido a que los tópicos no coinciden, como si esa alerta no fuera para él, por lo que se está produciendo una pérdida de información a considerar.



```
uwicore@uwicorepc: ~
Cliente MQTT
uwicore@uwicorepc:~$ mosquitto_sub -h localhost -t denm/0/1/2/3 1
mensaje DENM 3
Alerta publicada
uwicore@uwicorepc:~$ mosquitto_pub -h localhost -t denm/0/1/2/3/1/0 -m "mensaje DENM" 2
```

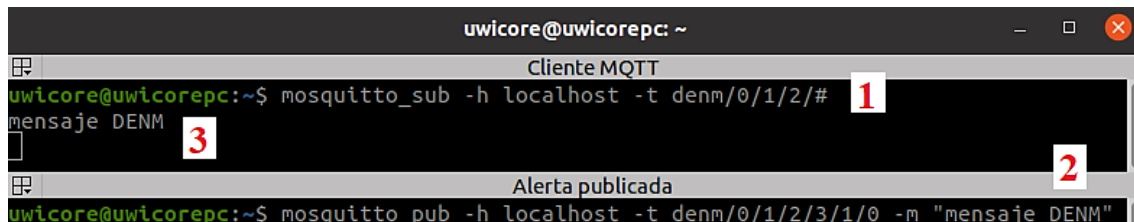
Figura 3.9. Ejemplo zoom Geoservice > zoom Cliente

Situación 3: Zoom Geoservice > Zoom Cliente + Wildcard (#)

El vehículo trabaja de nuevo con un nivel de zoom menor que el geoservice, pero utiliza un wildcard, es decir, un carácter especial del protocolo MQTT que permite suscribirse a varios niveles de zoom. De modo que puedes suscribirte a niveles de zoom superiores y recibir mensajes de niveles inferiores. Veamos el ejemplo de la Figura 3.10:

1. Cliente se suscribe al tópico denm/0/1/2/# (Nivel de zoom ≥ 2)
2. Se publica un mensaje DENM al tópico denm/0/1/2/3/1/0 (Nivel de zoom 6)

3. El cliente RECIBE el mensaje DENM porque está suscrito a un tópicos en el que puede recibir todos los mensajes que comiencen por “denm/0/1/2”, es decir, que sean igual o mayor a un nivel de zoom 2, y como el mensaje publicado es de nivel 6 recibe el mensaje, lo mismo ocurriría para un nivel de zoom 3, 5, 12... cualquier zoom que sea mayor o igual a 2.



```
uwicore@uwicorepc: ~
Cliente MQTT
uwicore@uwicorepc:~$ mosquitto_sub -h localhost -t denm/0/1/2/#
mensaje DENM
uwicore@uwicorepc:~$ mosquitto_pub -h localhost -t denm/0/1/2/3/1/0 -m "mensaje DENM"
```

Figura 3.10. Ejemplo zoom Geoservice > zoom Cliente + wildcard

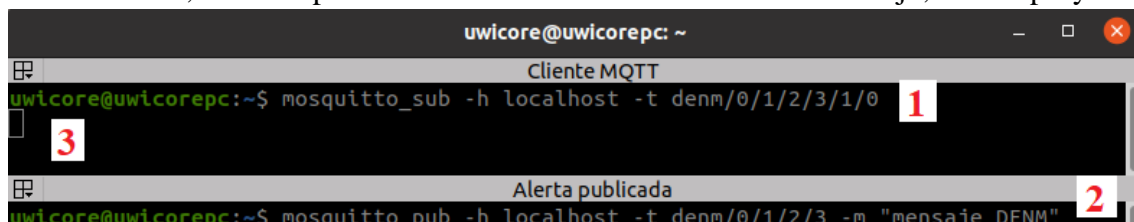
Con esta opción el vehículo no perdería información pero sin embargo, sí que recibiría alertas que no son relevantes para él, perdiendo ese nivel de discriminación que estamos buscando.

Situación 4: *Zoom Geoservice < Zoom Cliente*

En este caso, el vehículo (cliente MQTT) trabaja con un nivel de zoom mayor que el geoservice, es decir, los quadrees que el vehículo contempla cubren un área mucho menor que este. Por lo que ocurriría lo que se ve en la Figura 3.11:

1. Cliente se suscribe al tópicos denm/0/1/2/3/1/0 (Nivel de zoom 6)
2. Se publica un mensaje DENM al tópicos denm/0/1/2/3 (Nivel de zoom 4)
3. El cliente NO RECIBE el mensaje DENM debido a que los tópicos no coinciden, por lo que se produce una pérdida de información a considerar.

Por esta razón, es tan importante la elección del nivel de zoom de trabajo, en este proyecto



```
uwicore@uwicorepc: ~
Cliente MQTT
uwicore@uwicorepc:~$ mosquitto_sub -h localhost -t denm/0/1/2/3/1/0
mensaje DENM
uwicore@uwicorepc:~$ mosquitto_pub -h localhost -t denm/0/1/2/3 -m "mensaje DENM"
```

Figura 3.11. Ejemplo zoom Geoservice < zoom Cliente

se ha decidido utilizar el mismo nivel de zoom tanto para el Geoservice como para los vehículos (situación 1). Así evitamos esa posible pérdida de información relevante y mantener la coherencia y precisión en el procesamiento de los mensajes DENM sin poner en peligro a los vehículos y las personas que viajan en ellos o se encuentran alrededor.

Por otro lado, como se ha mencionado, C-Roads recomienda usar en este tipo de contextos, una horquilla definida entre 13 y 18 como valores para el nivel de zoom. Como meramente se limita a una recomendación, la implementación está preparada para soportar 22 niveles en total, de modo que si se define un nivel de zoom fuera de este rango [1 – 22] se mostrará un mensaje de error en el programa. La decisión tomada se debe a que un quadtree de nivel de zoom 22 apenas cubre poco más de 7 metros, distancia que resulta insignificante de cara a tener que dirigir una notificación debido a un evento, y por otro lado, la frecuencia con la que los vehículos deberían estar cambiando su tópico de suscripción es muy elevada, por lo que el rango de valores soportado por esta implementación está entre [1 – 22].

3.4. INTEGRACIÓN DEL GEOSERVICE EN ARQUITECTURA EDGE

Este trabajo se enmarca en un proyecto de mayor envergadura cuyo objetivo es la implementación de un nodo Edge para una red 5G que soporte servicios V2X. La aportación que se realiza con este trabajo es el diseño de un geoservice para ser integrado en dicho nodo. Esto da como resultado un nodo Edge capaz de proporcionar las mínimas latencias que requieren las comunicaciones V2X, con capacidad para ubicar geográficamente los nodos que componen la red y determinar qué vehículos deben ser notificados o no en función de lo relevante que sea el evento producido para su ubicación. Así, el propósito del siguiente apartado es detallar el papel que juega el geoservice dentro de este nodo Edge y cómo interactúa con el resto de componentes que hay presentes.

3.4.1. COMPONENTES QUE FORMAN EL NODO EDGE

El nodo Edge desarrollado por parte del equipo que forma parte del mencionado proyecto, se compone de diversas aplicaciones. Cada una de estas aplicaciones ha sido “*dockerizada*” (por ello, también se ha realizado esta tarea con el geoservice como parte del presente trabajo), lo que significa que se han encapsulado utilizando Docker [15], una plataforma que permite automatizar el despliegue de aplicaciones dentro de contenedores de software ligeros y estandarizados. Los contenedores Docker proporcionan un entorno aislado para las aplicaciones, garantizando que funcionen en cualquier sistema que soporte Docker, independientemente de las diferencias que pueda haber entre ellos. Esta tecnología es ampliamente reconocida por ofrecer flexibilidad, escalabilidad y consistencia en la ejecución de aplicaciones.

Una vez *dockerizadas*, estas aplicaciones se despliegan en un clúster gestionado por Kubernetes [16], que es un sistema de orquestación y gestión de contenedores de código abierto. Kubernetes permite automatizar el despliegue, escalado y operaciones de aplicaciones contenerizadas, y es ampliamente utilizado en infraestructuras de computación modernas debido a su robustez y versatilidad. En este clúster, las aplicaciones se implementan como "*pods*". Un *pod* es la unidad más pequeña y básica que se puede desplegar en Kubernetes. Aunque cada pod en el cluster actúa como una entidad contenerizada independiente, gracias a la naturaleza de Kubernetes y su capacidad de gestionar redes, los pods tienen la capacidad de comunicarse entre sí. Esto permite que las aplicaciones, aunque estén separadas y encapsuladas, colaboren y compartan datos según sea necesario, estableciendo un ecosistema cohesivo y eficiente.

A continuación se describe en la Figura 3.12, los elementos que componen el nodo Edge, dónde se sitúa el geoservice y como se produce su interacción con el resto de componentes.

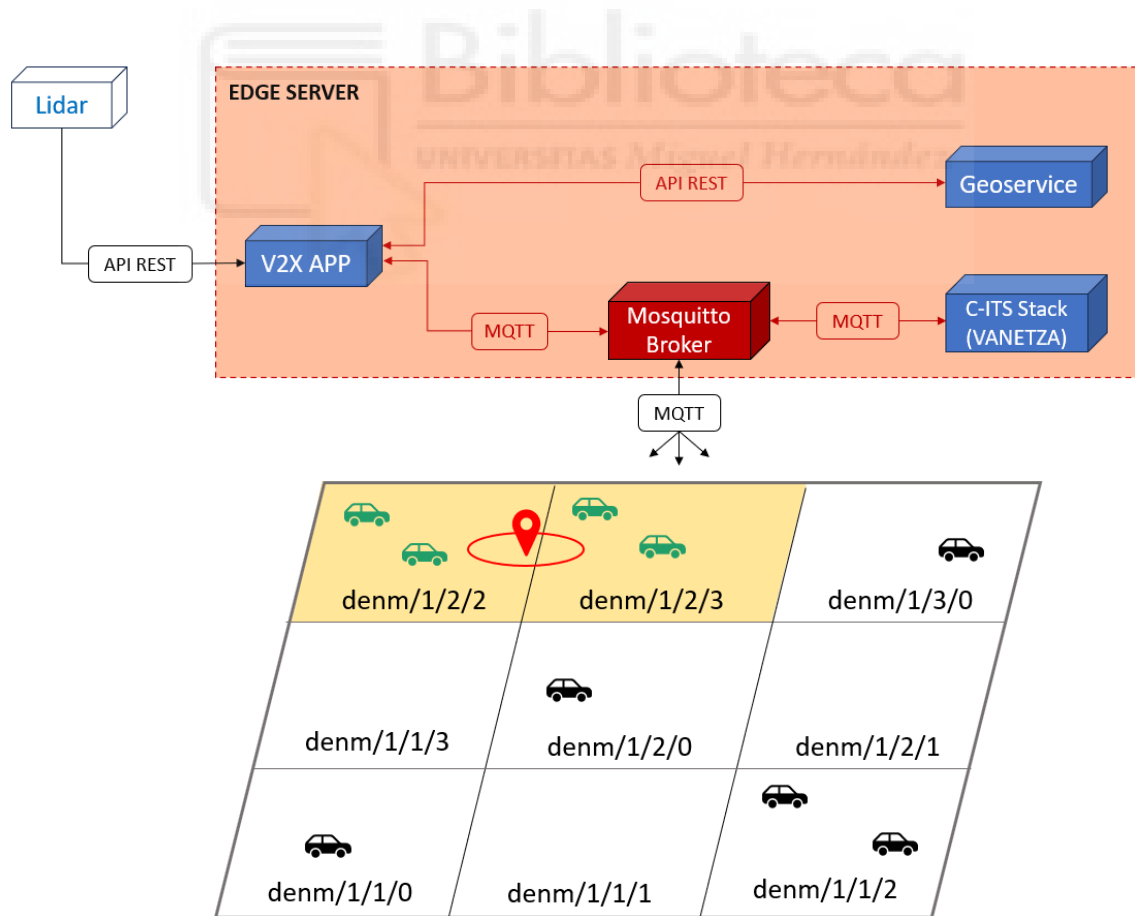


Figura 3.12. Arquitectura nodo edge

V2X APP

Se trata de una aplicación REST desarrollada con el framework Spring Boot y lenguaje de programación Java. Su principal función es actuar como intermediario entre el exterior y el interior del nodo edge. Se encargará de recibir toda la información que haya recogido un Lidar acerca del evento que se ha producido, y a continuación, consultar al geoservice los tópicos de los quadrees que se han visto afectados por ese evento concreto y después conformar un archivo JSON donde se organiza toda la información recibida del Lidar junto con los tópicos que haya calculado el geoservice.

De modo que llega a desempeñar hasta 3 roles en función de con quién tenga que comunicarse: cuando el intercambio de información se produzca entre Lidar y V2X APP (vía API REST), actuará como un servidor REST listo para recibir la información que le brinde el Lidar cada vez que se produzca un evento, mientras que si la comunicación es con el geoservice, actuará como un cliente REST que consumirá la información que este le proporcione. Por otro lado, cuando tenga que publicar ese JSON con toda la información recopilada a través del broker presente en el nodo edge, actuará como un cliente MQTT.

Broker MQTT

Consiste en un broker de mensajería MQTT de código abierto y altamente eficiente que va a actuar como servidor gestionando las suscripciones que hagan los diferentes clientes MQTT (vehículos), y retransmitiendo los mensajes que se realicen entre publicadores y suscriptores.

En este caso concreto, el broker deberá gestionar por un lado, el intercambio de mensajes que se produzca de la V2X APP a Vanetza, y de Vanetza a los vehículos suscritos a los tópicos afectados por un evento, y por otro lado, las suscripciones y cancelación de las mismas que vayan realizando los vehículos a medida que cambian de quadree.

Vanetza (C-ITS stack)

Vanetza es una implementación específica del C-ITS Stack que se ha desarrollado para el protocolo de comunicación vehicular V2X denominado "ITS-G5". Está diseñada para operar en redes de área vehicular utilizando el estándar IEEE 802.11p. Su objetivo es proporcionar una pila de protocolos completa y eficiente para habilitar las comunicaciones cooperativas en entornos V2X [17].

En este caso se encarga de recibir el archivo JSON proporcionado por la V2X APP que contiene toda la información relativa al evento y formar la estructura de los mensaje DENM que se van a publicar, siguiendo los estándares establecidos por la ETSI, y por último, se encarga de publicar la notificación de alerta sólo a los tópicos indicados por el geoservice.

Lidar

Por último, tenemos el Lidar (*Light Detection and Ranging*), que aunque se encuentra fuera del edge juega un papel crucial. Tiene la capacidad de reconocer un evento relevante de tráfico y recopilar toda la información relativa a ese evento. Luego proporcionará los datos al nodo edge a través de la V2X APP para que pueda realizar correctamente sus funciones.

Se trata de un sensor óptico remoto que utiliza pulsos láser para medir la distancia y crear mapas detallados en 3D del entorno circundante. Emitiendo rápidamente pulsos láser y midiendo el tiempo que tardan en rebotar en objetos cercanos, el Lidar puede determinar la distancia y la forma de los objetos con gran precisión. Gracias a esta capacidad, puede recopilar toda la información relevante a un evento y proporcionarla al edge.

- Interacción entre los componentes

Cada vez que el Lidar detecte un evento que sea relevante al tráfico, recopilará toda la información pertinente y se la enviará a la V2X APP vía API REST. Quien de toda esa información extraerá 3 datos: *latitud*, *longitud* y *distancia de relevancia*.

V2X APP utilizará estos datos para formular una petición de consulta, también vía API REST, al geoservice. El cual se encargará de calcular los tópicos de los quadrees a los que debe dirigirse el mensaje de alerta. Una vez conocidos los tópicos de interés, se envían de nuevo a la V2X APP, quien construye un JSON con toda la información proporcionada por el Lidar y los tópicos calculados por el geoservice. Esta información se envía a Vanetza vía MQTT.

Allí, Vanetza forma el mensaje DENM según el estándar ETSI y lo publica vía MQTT a los tópicos de interés. De modo que, solo recibirán el mensaje de alerta, aquellos vehículos que se encuentren dentro de alguno de los quadrees afectados por el evento.

3.4.2. INTERFAZ GEOSERVICE – V2X APP

Para lograr una implementación efectiva del geoservice se requiere de una estrecha colaboración entre los *pods* geoservice y V2X APP desplegados dentro del nodo Edge. Por lo que juega un papel importante la interfaz de comunicación definida entre ambos.

La interfaz está basada en la arquitectura REST, la cual hace uso de los verbos HTTP (GET, POST, PUT, DELETE) para realizar las operaciones requeridas sobre los recursos a través de una URL (*Uniform Resource Locators*). Esta elección ha sido tomada debido a su gran escalabilidad, eficiencia y flexibilidad a la hora de aprovechar los estándares y convenciones del protocolo HTTP.

En la Figura 3.13 podemos observar que cuando la V2X APP recibe información sobre un evento por parte del Lidar, en primer lugar se encarga de extraer los 3 datos que necesita el geoservice para realizar sus cálculos: la latitud y longitud del lugar exacto donde se ha producido el evento, y la distancia de relevancia correspondiente. A continuación, lo que hace es solicitar los tópicos de los quadrees afectados por esa alerta a través de una petición de tipo HTTP GET al geoservicio, quien en este caso, actúa como servidor REST. Para ello, lo que hace es formar una URL utilizando los datos extraídos mediante la cual se hará la petición. Tiene la siguiente estructura:

http://{IP}:{puerto}/geoservice/{latitud}/{longitud}/{relevanciaDistance}

La petición GET llega al geoservice a través de la interfaz API REST, quien extrae de la URL los datos de la posición del evento y la distancia de relevancia, realiza los cálculos ya explicados en apartados anteriores y obtiene los tópicos afectados por el evento en cuestión. Por último, el geoservice añade los tópicos en un JSON y lo utiliza como respuesta a la petición GET realizada por la V2X APP. El formato JSON (JavaScript Object Notation) se elige por su ligereza y facilidad de manejo en aplicaciones web, lo que es relevante dado el entorno en el que operan ambos componentes [18].

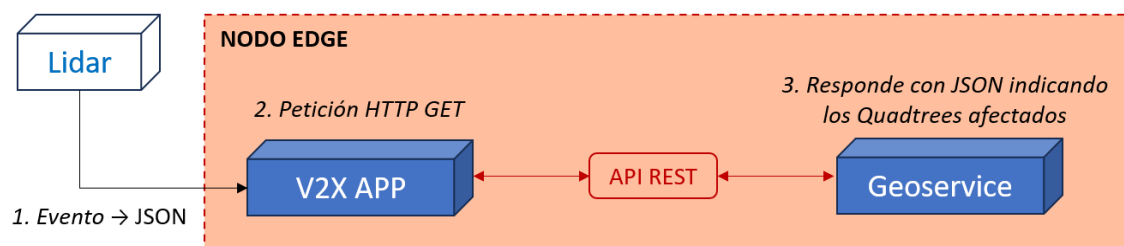


Figura 3.13. Interfaz Geoservice - V2X APP

En el siguiente ejemplo podemos observar cómo se realiza una petición de tipo GET usando la URL `http://localhost:8080/geoservice/38.26199/-0.71668/1` como punto de acceso al servicio. El geoservice, teniendo en cuenta los datos que contiene la petición, realiza los cálculos implementados en su código y obtiene, en este caso, que el tópico afectado por el supuesto evento es `denm/0/3/3/1/1/1/3/1/2/1/1/1/1/2/1`, por lo que se puede intuir que el nivel de zoom usado para este ejemplo es igual a 16. Para realizar la petición de ejemplo se ha utilizado la herramienta software *Postman*.

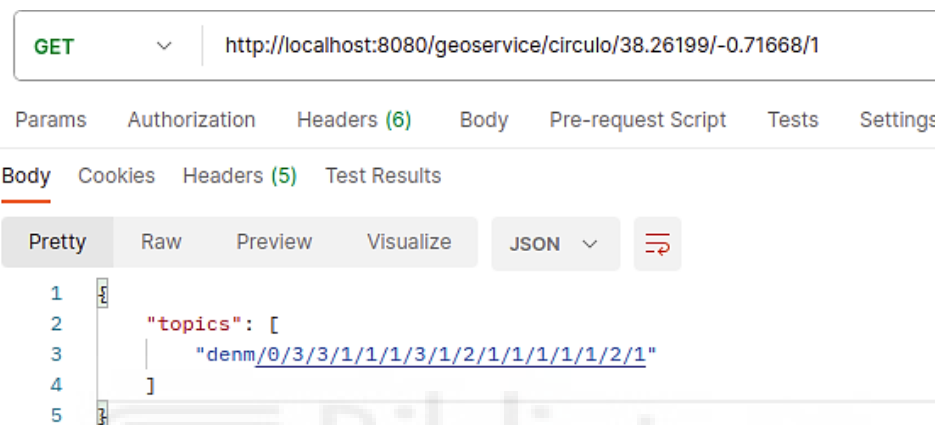


Figura 3.14. Ejemplo petición HTTP GET al Geoservice usando Postman

3.5. IMPLEMENTACIÓN CLIENTE MQTT

Para cumplir el objetivo perseguido en este proyecto y poder notificar solo a aquellos vehículos para los que sea relevante una determinada alerta, necesitamos que estos tengan la capacidad para suscribirse, así como de cancelar la suscripción a los tópicos de los quadtree en los que se encuentren conforme vayan avanzando en su trayectoria. Recordemos que el protocolo de mensajería MQTT utilizado para la notificación de mensajes se basa en un modelo de publicación/suscripción.

Por esta razón, se ha implementado un cliente MQTT con capacidad para realizar dicha función. La suscripción selectiva a partir de su ubicación permite al cliente recibir solo la información relevante para su área geográfica y evitar una sobrecarga innecesaria de mensajes. Para su desarrollo se ha utilizado el lenguaje de programación C y la biblioteca de código abierto "mosquitto.h" [19].

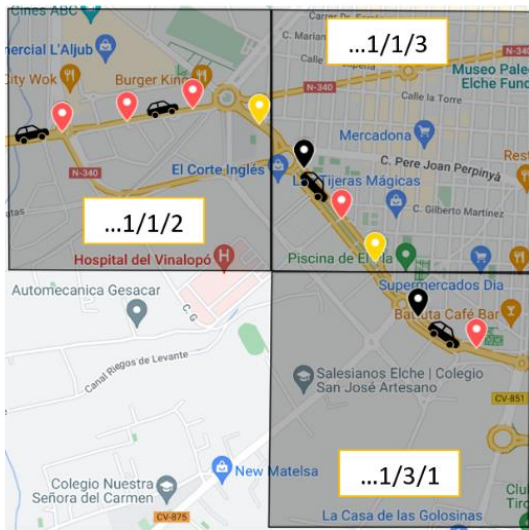
Una característica importante de este cliente es que ha sido implementado siguiendo las indicaciones y requerimientos establecidos por C-Roads, es decir, se ha utilizado las

mismas funciones y operaciones definidas en el geoservice para que el cliente (vehículo) pueda calcular en todo momento el tópico al que debe suscribirse en función de su ubicación actual. Esto asegura una alineación efectiva con las normas y estándares del sistema V2X, garantizando una correcta comunicación y coordinación entre geoservice y los vehículos.

Por razones de seguridad, el cliente cuenta con un "radio de proximidad a borde". Esta característica se basa en un valor en metros, definido según la consideración oportuna en el contexto del sistema V2X, por ejemplo el tipo de vía, la velocidad del vehículo, etc. Cuando un vehículo se encuentra cerca de un borde o línea de separación entre dos o más quadrees, se activa un período en el que el cliente estará suscrito a varios tópicos en lugar de uno. Esto se hace por razones de seguridad y para evitar que el cliente pierda información relevante en situaciones críticas o transiciones entre las regiones que cubre un quadtree. Durante este período de suscripción múltiple, el cliente puede recibir información de ambos tópicos, asegurando una cobertura completa y continua mientras se desplaza por la red de carreteras.

En el siguiente ejemplo se ha tenido en cuenta la velocidad del vehículo, el tipo de vía y el tiempo que falta para aproximarse a un borde (podría tenerse en cuenta otros factores que se consideren oportunos). Sabemos que la velocidad máxima en vías urbanas es de 50 km/h, si por ejemplo establecemos que el cliente realice una suscripción múltiple cuando se encuentre a 4 segundos de un borde para evitar que pueda perder información relevante, el radio de seguridad debería ser de 56 metros, por lo que si el cliente se encuentra a menos de 56 metros de un borde, permanecerá suscrito a los tópicos de los quadrees que haya a ambos lados de dicho borde. Supongamos que el cliente comienza la ruta a lo largo de la carretera marcada en la Figura 3.15. Lo primero que hace es suscribirse al tópico del quadtree en el que se encuentra (*denm/.../1/1/2*). En los puntos 2 y 3 lo que ocurre es que el vehículo como ya está suscrito a ese tópico únicamente mantiene dicha suscripción. En cambio en el punto 4 lo que ocurre es que, por un lado mantiene la suscripción a *denm/.../1/1/2* pero por otro, también se suscribe al tópico del siguiente quadtree (*denm/.../1/1/3*) debido a su proximidad al borde que marca el inicio y el fin de cada quadtree. Por otro lado, lo que ocurre en el punto 5 es una desuscripción del tópico *denm/.../1/1/2* debido a que el vehículo se encuentra a una distancia mayor que el "radio de proximidad a borde" definida en este ejemplo en 56 metros, es decir, se

encuentra lo suficientemente lejos como para considerar no relevantes las alertas que se produzcan en el quadtree que abandona.



Respuesta del cliente MQTT:

- 1. **Suscrito** al topic denm/.../1/1/2
- 2. Ya suscrito al topic
- 3. Ya suscrito al topic
- 4. Ya suscrito al topic
- 4. **Suscrito** al topic denm/.../1/1/3
- 5. Ya suscrito al topic
- 5. **Desuscrito** del topic denm/.../1/1/2
- 6. Ya suscrito al topic
- 7. Ya suscrito al topic
- 7. **Suscrito** al topic denm/.../1/3/1
- 8. Ya suscrito al topic
- 8. **Desuscrito** del topic denm/.../1/1/3
- 9. Ya suscrito al topic

Figura 3.15. Ejemplo vehículo cambiando de quadtree durante su trayectoria

Obtenemos así un cliente MQTT que tiene la capacidad de calcular el tópico al que debe suscribirse dado un determinado nivel de zoom y sin necesidad de comunicarse directamente con el geoservice. Esta autonomía en la toma de decisiones permite que el cliente se mantenga independiente y ágil en sus operaciones, reduciendo la carga de trabajo del sistema y agilizando las respuestas en tiempo real.

4. EVALUACIÓN Y VALIDACIÓN

La presente sección se centra en la evaluación y validación exhaustiva tanto del geoservice desarrollado como de las funcionalidades que se han diseñado con el objetivo de ser incorporadas en la OBU de los vehículos. Esto se hace para que adquieran esa capacidad de suscripción/desuscripción a tópicos MQTT en función de su posición geográfica y puedan recibir las alertas de tráfico pertinentes.

Para lograr este propósito y demostrar la funcionalidad, precisión y la confiabilidad que ofrece el geoservice, así como la correcta interacción entre dicho servicio y el vehículo (quien actuará como cliente MQTT), se han diseñado y ejecutado una serie de pruebas y escenarios representativos que abarcan diversas situaciones y casos de uso.

La evaluación y validación de este servicio y su interacción con el resto de elementos que componen el *edge server* no solo son esenciales para proporcionar una base sólida que garantice la efectividad y utilidad de la solución propuesta, sino que también va a permitir identificar posibles áreas de mejora y optimización del servicio desarrollado.

4.1. TEST DE SUSCRIPCIÓN A BROKER MQTT

La capacidad de un broker MQTT para gestionar un alto número de suscripciones es esencial para asegurar un servicio fiable y eficiente. El broker juega un papel crucial en la arquitectura de cualquier aplicación que utilice MQTT, ya que actúa como intermediario gestionando los mensajes entre los distintos clientes.

Si un broker no puede soportar la carga de suscripciones de todos los clientes conectados, la calidad del servicio se verá gravemente afectada, pudiendo dar lugar a la pérdida de mensajes y, en última instancia, a fallos en la aplicación. Por lo tanto, es de suma importancia realizar pruebas de carga que nos ayuden a averiguar el límite máximo de suscripciones que un broker puede manejar.

En este contexto, se han realizado pruebas para identificar la capacidad máxima de suscripciones de un broker MQTT. La realización de estas pruebas es fundamental para garantizar que el broker seleccionado sea capaz de soportar el número de clientes y suscripciones previstas en el entorno de producción, asegurando así la escalabilidad y robustez del sistema.

Para la realización de estas pruebas nos hemos apoyado en la herramienta SW “MQTT Explorer” [20], ya que permite observar lo que ocurre dentro de este y las gestiones que

hace con los mensajes que recibe. Se ha conectado varios clientes MQTT de forma simultánea, aumentando el número de estos de forma progresiva obteniendo los resultados de la Figura 4.1, donde se muestra un gráfico (obtenido de la herramienta MQTT Explorer) cuyo eje X indica el tiempo en minutos y el eje Y, el número máximo de clientes conectados. Inicialmente, en nuestro entorno de prueba, teníamos 8 clientes ya conectados, que incluían aplicaciones como V2X-APP, Vanetza o el propio MQTT Explorer. Para evaluar la capacidad máxima del broker, se realizaron pruebas incrementando el número de clientes MQTT que se conectan y suscriben al broker de manera simultánea, hasta alcanzar un total de 2000 clientes. Por lo tanto, en total, se obtiene un máximo de 2008 clientes conectados simultáneamente, incluidos los 8 clientes iniciales y los 2000 clientes adicionales utilizados para la prueba.

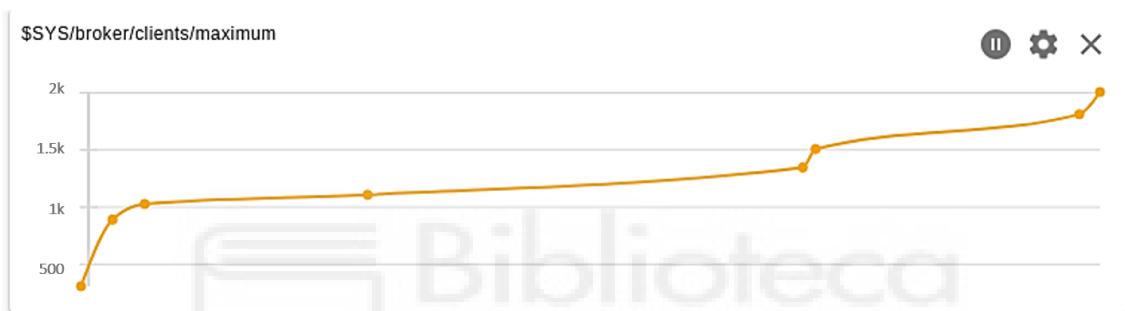


Figura 4.1. Numero de suscripciones soportadas por el broker MQTT

Este resultado se refleja en la Figura 4.2, donde se muestra una captura de MQTT Explorer en el momento en el que se conectan los 2008 clientes de forma simultánea al broker MQTT.

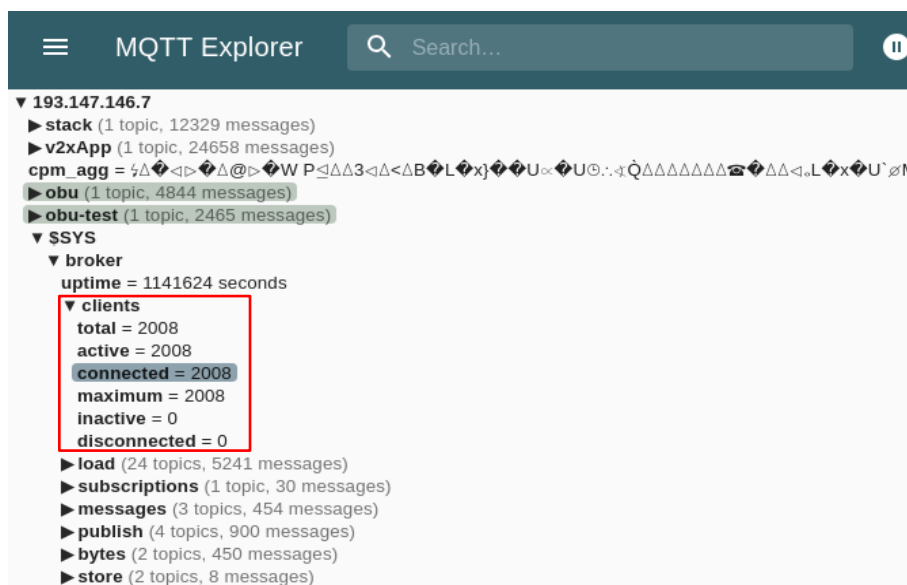


Figura 4.2. Visualización broker mediante MQTT Explorer

Lamentablemente, durante las pruebas, nos encontramos con una limitación en el equipo donde se estaban ejecutando. Al intentar lanzar más de 2000 clientes, el PC empezó a producir errores, lo que impidió probar una cantidad mayor de clientes MQTT. Este problema se debe a que, por cada cliente conectado, se ejecuta un subproceso adicional, lo que generaba una carga significativa en el PC y, eventualmente, lleva al sistema a un estado en el que no puede gestionar más subprocesos. Aunque el broker pueda ser capaz de manejar más suscripciones, las limitaciones del hardware utilizado para la prueba no permitieron evaluar su capacidad máxima real. En la Figura 4.3 se muestra las especificaciones principales del equipo en cuestión.

Nombre del dispositivo	uwicorePC >
Memoria	15,5 GiB
Procesador	Intel® Core™ i5-7200U CPU @ 2.50GHz × 4
Gráficos	Mesa Intel® HD Graphics 620 (KBL GT2)
Capacidad del disco	1,5 TB
Nombre del SO	Ubuntu 20.04.6 LTS
Tipo de SO	64 bits
Versión de GNOME	3.36.8
Sistema de ventanas	X11

Figura 4.3. Especificaciones principales del PC de desarrollo

Pese a esta limitación, es importante destacar que lograr 2000 suscripciones simultáneas sin alterar prácticamente el funcionamiento del broker MQTT es un resultado positivo. En el contexto de este proyecto, solo contamos con un nodo edge, es decir, un solo broker, por lo que manejar 2000 suscripciones simultáneas es un logro considerable y, de hecho, excede las expectativas iniciales.

Aunque siempre es deseable probar y conocer los límites máximos de cualquier sistema, en la práctica, es raro que un solo broker necesite manejar un número de suscripciones mucho mayor a 2000 de manera simultánea. Por lo tanto, aunque no pudimos determinar el límite absoluto de suscripciones que nuestro broker puede manejar, el resultado obtenido sugiere que este broker es adecuado para su rol dentro de la infraestructura planteada.

4.2. FRECUENCIA DE SUSCRIPCIÓN

Como ya hemos indicado a lo largo de esta memoria, el nivel de zoom es un componente central en la implementación de nuestro sistema, influyendo en aspectos como la frecuencia con la que un cliente se suscribe a nuevos quadrees. Este factor, aunque pueda parecer menor, tiene un impacto directo en la operatividad y rendimiento del sistema. Al cambiar el nivel de zoom, cambiamos el tamaño del área que cubren los quadrees. Este cambio en la cobertura geográfica afecta directamente a la frecuencia con la que un cliente necesita suscribirse a nuevos tópicos a medida que se desplaza por el mapa. A medida que aumenta el nivel de zoom, disminuye el tamaño de cada cuadrante y, en consecuencia, aumenta la frecuencia con la que un cliente debe suscribirse a nuevos cuadrantes.

Se ha realizado un estudio sobre la relación entre el nivel de zoom, el tamaño de los cuadrantes y la frecuencia de suscripción a nuevos quadrees. En la Figura 4.4, el eje horizontal representa la velocidad del vehículo en kilómetros por hora (km/h), mientras que el eje vertical indica cada cuanto tiempo se produce una nueva suscripción. El gráfico revela algunas tendencias interesantes y esperadas. Como era evidente, muestra que cuanto mayor es el nivel de zoom, menor es el tiempo entre suscripciones por parte de los clientes. Además, el gráfico también muestra que, a medida que aumenta la velocidad del vehículo, se reduce el tiempo de suscripciones a nuevos tópicos. Este comportamiento es consecuente con la naturaleza de nuestro sistema: a mayor velocidad, un vehículo cubrirá más terreno en menos tiempo, atravesando más cuadrantes y, por ende, necesitando suscribirse a nuevos tópicos con mayor frecuencia.

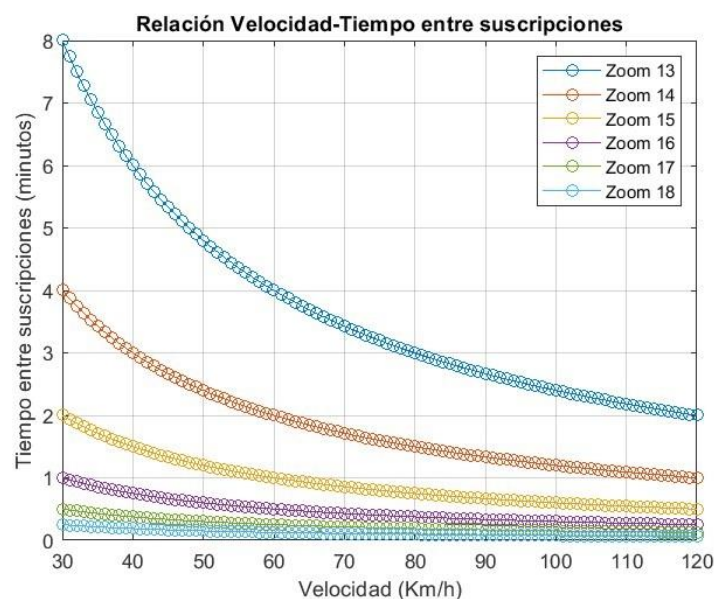


Figura 4.4. Frecuencia de suscripción a tópicos según nivel de zoom

Otra factor a tener en cuenta, es cómo afecta la densidad del tráfico que haya en las carreteras a esos tiempos entre suscripciones obtenidos, por lo tanto también se han obtenido unos resultados al respecto gracias a la utilización del modelo de Van Aerde [21]. En la Figura 4.5 podemos observar 3 ejemplos para diferentes números de carriles. A medida que aumenta el número de carriles, aumenta el número de vehículos por kilómetro. Pero también se puede ver que a medida que aumenta la densidad vehicular, aumentan los tiempos entre suscripciones para cada nivel de zoom.

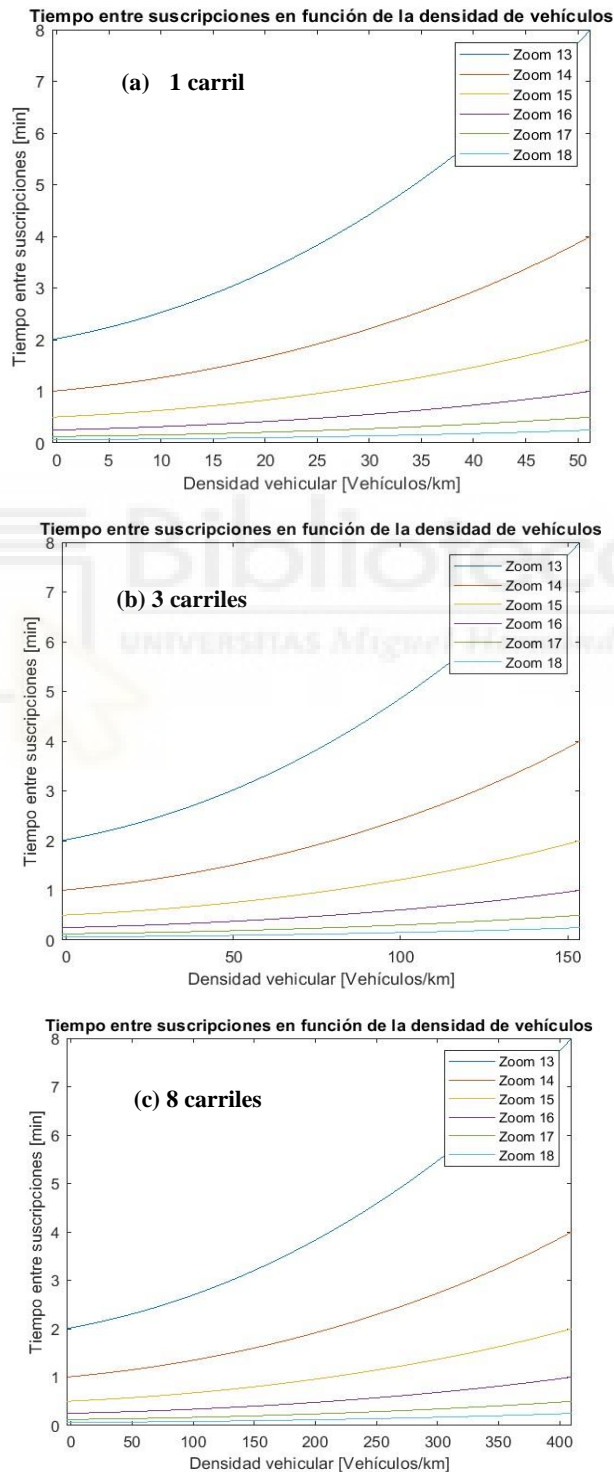


Figura 4.5. Tiempo entre suscripciones en función de la densidad vehicular

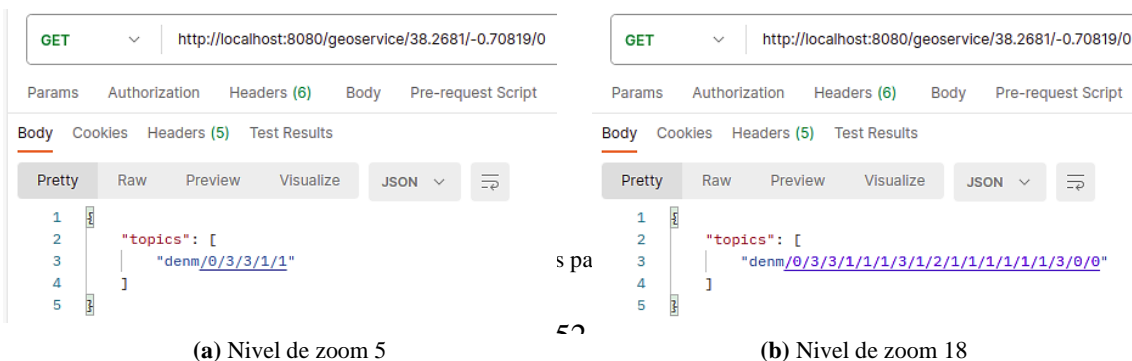
Esta relación entre la velocidad del vehículo y la frecuencia de suscripción es un factor importante a tener en cuenta en el diseño y optimización del sistema. Afecta directamente la carga en el broker MQTT y puede tener implicancias en la latencia y en la cantidad de datos transmitidos. Es fundamental considerar estas relaciones y su impacto en el rendimiento del sistema para asegurar un funcionamiento eficiente y efectivo en situaciones de uso real.

4.3. GENERACIÓN DE TÓPICOS

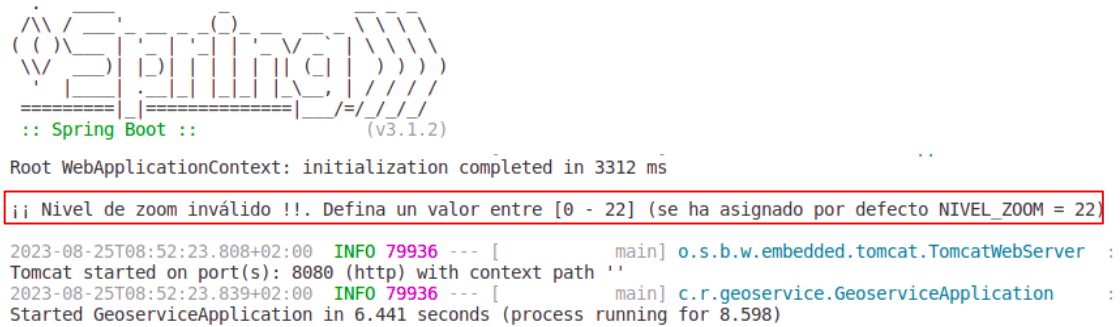
Cada nivel de zoom ofrece una perspectiva diferente, desde continentes hasta detalladas imágenes de calles y carreteras. Por lo tanto, la capacidad de generar tópicos específicos para cada nivel de zoom es esencial para el correcto funcionamiento del geoservice propuesto. Dado una ubicación específica, representada por su latitud y longitud, el servicio tiene la responsabilidad de discernir y proporcionar el tópico correspondiente del quadtree que contiene dicho punto, independientemente del nivel de zoom seleccionado.

Para demostrar esta capacidad que posee el geoservice, se ha realizado una serie de pruebas a nivel local que se comentaran a continuación. Es decir, se han realizado conectando el geoservice a un broker MQTT desplegado en el PC desde donde se están realizando los desarrollos, no al nodo Edge (esto se realizará en siguientes apartados) ya que primero se quiere verificar el correcto funcionamiento del geoservice.

Supongamos un punto cuyas coordenadas son “38.2681, -0.70819”, también fijaremos una distancia de relevancia de 50 metros con el fin de probar la capacidad de cálculo de tópicos MQTT. Introducimos esta información en la URL de la petición HTTP GET que realizaría la V2X APP al geoservicio (y que en este caso estamos simulando a nivel local con la herramienta software *Postman*), y observemos como devuelve el tópico correctamente según el nivel de zoom definido (5, 18 y 25 por ejemplo). Recordemos que dicho nivel influye en el tópico calculado tal y como se refleja en la respuesta que devuelve el geoservice en la Figura 4.6



Sin embargo, también se ha tenido en cuenta el caso en el que por error se intente acceder a niveles de zoom fuera del rango definido. Para ello, se ha incorporado una característica de protección. En situaciones donde se selecciona un nivel de zoom inválido, como puede ser por ejemplo un nivel de zoom 25, el sistema reasignará de forma predeterminada a 22 el valor del nivel de zoom. Y además, para mantener informado al usuario, se muestra un mensaje de alerta informándole de lo sucedido, tal y como se muestra en la Figura 4.7:



```

  ____
 /  _ \
/  / \  \
/  /  \  \
/_____\  \
:: Spring Boot ::
 (v3.1.2)

Root WebApplicationContext: initialization completed in 3312 ms

¡¡ Nivel de zoom inválido !!. Defina un valor entre [0 - 22] (se ha asignado por defecto NIVEL_ZOOM = 22)

2023-08-25T08:52:23.808+02:00 INFO 79936 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2023-08-25T08:52:23.839+02:00 INFO 79936 --- [           main] c.r.geoservice.GeoserviceApplication   : Started GeoserviceApplication in 6.441 seconds (process running for 8.598)

```

Figura 4.7. Mensaje de alerta del geoservice ante zoom inválido

Además de la correcta generación de tópicos en función del nivel de zoom definido, es vital reconocer otra pieza fundamental de este puzzle: el "área de relevancia". Este área es la que permite al geoservice saber el número de quadrees afectados por una determinada alerta de tráfico, y poder así calcular el tópico de cada uno de esos quadrees.

Imaginemos una situación donde una carretera principal en una ciudad tiene una interrupción inesperada. El impacto no se limita solo al tramo interrumpido, las carreteras y calles circundantes también se verán afectadas debido al desvío y acumulación del tráfico. De ahí la gran funcionalidad del área de relevancia implementada en el geoservice, ayudando a identificar el número de zonas afectadas o quadrees a notificar.

Veamos unos ejemplos (representados en la Figura 4.8) donde en función del tamaño del área de relevancia, el geoservice devolverá un número de tópicos diferente demostrando su correcto funcionamiento. En la Figura 4.9, podemos observar la respuesta del geoservice para cada ejemplo, en el caso del evento 1 el área de relevancia es de 50 metros, por lo que únicamente afecta a un solo quadtree, por ello el geoservice devuelve un solo tópico. En cambio, en el caso del evento 2, el área afectada cubre un área de 500 metros, por lo que serían cuatro los quadrees a los que se debe dirigir la alerta de tráfico correspondiente. Y por último, tenemos el ejemplo del evento 3, donde el área de relevancia se ha establecido en 200 metros, afectando en este caso a 2 quadrees.

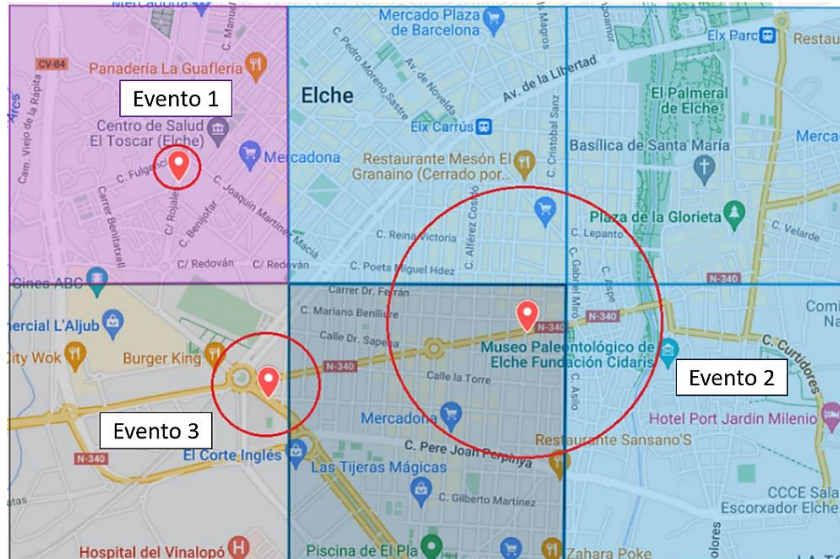


Figura 4.8. Cuadrees afectados en función del área de relevancia

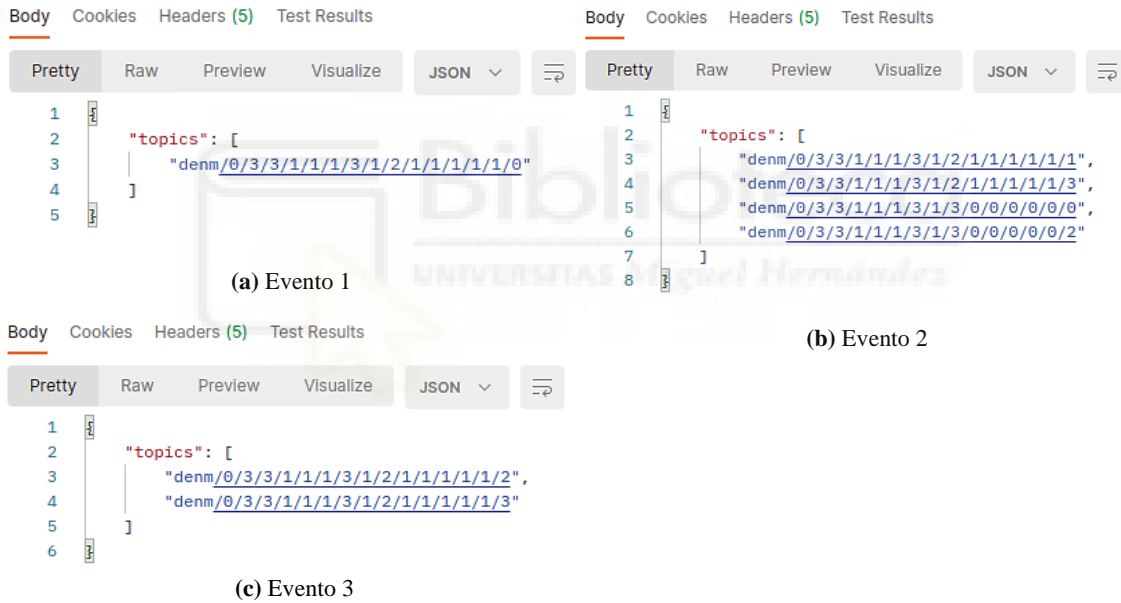


Figura 4.9. Tópicos devueltos por el geoservice en función de los cuadrees afectados

4.4. GESTIÓN DE TÓPICOS Y MENSAJES DENM EN CLIENTE MQTT

En el protocolo MQTT existen publicadores y suscriptores. En el caso de nuestro nodo Edge, Vanetza actúa como el publicador, utilizando los tópicos calculados por el geoservice en función de la ubicación del evento y su área de relevancia. Para que los vehículos puedan recibir los mensajes DENM, es necesario que estén suscritos a los tópicos de los quadtrees en los que se ubican y también, que vayan anulando la suscripción de los tópicos de aquellos quadtrees que dejan atrás para no recibir información innecesaria.

Por lo tanto, un aspecto crucial de esta solución propuesta pasa por asegurarse de que el cliente MQTT diseñado para ser incorporado en las Unidades de Bordo (OBU) de los vehículos, realice su labor correctamente. Este cliente tiene la responsabilidad de gestionar las suscripciones y cancelación de las mismas, a los tópicos correspondientes según la posición GPS del vehículo y, más importante aún, de recibir mensajes de alerta DENM cuando se encuentre dentro de un quadtree afectado por un evento de tráfico relevante.

Se han realizado pruebas específicas que implican simular un determinado trayecto a través del mismo mapa elegido para el ejemplo del apartado anterior (línea azul que transcurre desde el punto A al B), y sobre el cual se va a ir tomando la posición GPS del vehículo y verificar que efectivamente realiza las suscripciones y cancelación de estas de forma adecuada a medida que se mueve sobre la trayectoria definida. Esto implica tanto suscribirse a nuevos tópicos cuando entra en un nuevo quadtree, como eliminar la suscripción de aquellos que deja atrás.

Además, se muestra como el cliente recibe correctamente los mensajes DENM cuando se encuentra dentro de un quadtree afectado por un evento de tráfico. Esto implicaba simular la ocurrencia de un evento de tráfico, como un accidente o un atasco, y verificar que el cliente MQTT recibe el mensaje de alerta correspondiente. Para simular un publicador MQTT y enviar el mensaje DENM de alerta, se ha utilizado la herramienta de línea de comandos que forma parte del proyecto Mosquitto llamada “mosquitto_pub” [22]. Esta permite enviar mensajes a un broker para la realización de pruebas, en nuestro caso a nivel local (posteriormente se realizarán utilizando Vanetza, la cual recordemos que está desplegada en un nodo Edge).

Concretamente, el quadtree destacado en color amarillo va a ser el afectado por un evento justo cuando el vehículo se encuentra transcurriendo dentro de dicha zona (la marca roja indica la posición exacta del vehículo cuando recibe el mensaje). El mapa y los tópicos correspondientes de cada quadtree se muestran en la siguiente Figura 4.10 (se ha tomado un nivel de zoom 15 para el ejemplo):

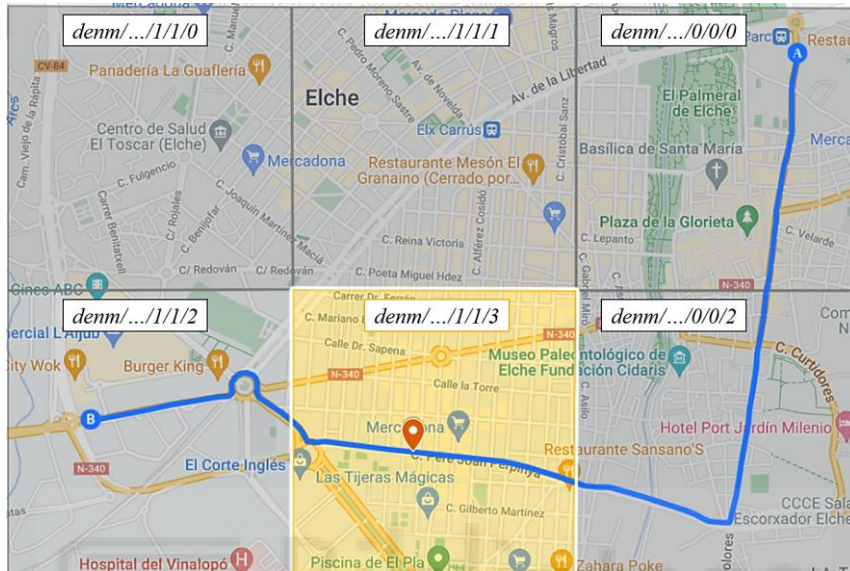


Figura 4.10. Escenario ejemplo cliente MQTT conectado a broker local

Lanzamos la simulación y generamos el siguientes fichero de resultados. Comenzamos analizando las diferentes columnas que conforman la Figura 4.11. En primer lugar encontramos el id del vehículo, dato que nos ayuda a identificar quién ha realizado las acciones MQTT, en este ejemplo, han sido realizadas por el mismo vehículo.

Las columnas de latitud y longitud marcan el punto exacto donde se ha producido la acción en cuestión. La cuarta columna, llamada *Evento*, indica el tipo de acción MQTT que ha realizado el cliente: 1 si se trata de la suscripción a un nuevo tópico, 2 en caso de la desuscripción del tópico de un quadtree que ya ha abandonado, y 3 cuando el vehículo recibe un mensaje DENM alertándolo que se ha producido un evento relevante de tráfico y que toma las acciones y decisiones oportunas. El vehículo inicia su ruta en el punto A (38.271170, -0.694630), por lo que procede a suscribirse al tópico correspondiente, en este caso el acabado en ".../0/0/0". Cuando está próximo al borde de cambio de quadtree (38.264330, -0.695680), lo cual se detecta gracias al "área de proximidad a borde" que tiene implementada el cliente MQTT, se suscribe también al tópico del próximo quadtree al que va a ingresar (".../0/0/2"). Sin embargo, cuando dicha área de seguridad pasa de

coincidir con dos quadrees en el borde a coincidir con solo uno, anula su suscripción del tópico correspondiente al quadtree que ha dejado atrás (“.../0/0/0”). Esto significa que el vehículo ya está lo suficientemente lejos del borde y que los eventos que suceden dentro de dicha zona ya no son relevantes para ese vehículo.

En el punto 38.258130, -0.702890, vuelve a ocurrir lo mismo, antes de ingresar al siguiente quadtree, por seguridad y evitar pérdidas de información, se suscribe al tópico “.../1/1/3”. Cuando está dentro de ese quadtree, se produce un determinado evento, por lo que se publica vía MQTT la correspondiente alerta. Aquellos clientes que se ubiquen dentro de esta zona y estén suscritos a su tópico asociado, recibirán el mensaje DENM. Por esta razón, en la línea 7 del fichero de resultados se puede ver reflejado la recepción del mensaje cuando el cliente se ubicaba en la posición 38.259070, -0.708270. Las 2 últimas líneas muestran la suscripción al tópico del quadtree al que va a ingresar (“.../1/1/2”) y la desuscripción del tópico “.../1/1/3”.

id	Latitud	Longitud	Evento	Topic	Mensaje
1	38.271170	-0.694630	1	denm/0/3/3/1/1/1/3/1/3/0/0/0/0/0/0	-
2	38.264330	-0.695680	1	denm/0/3/3/1/1/1/3/1/3/0/0/0/0/0/2	-
3	38.261670	-0.696220	2	denm/0/3/3/1/1/1/3/1/3/0/0/0/0/0/0	-
4	38.258130	-0.702890	1	denm/0/3/3/1/1/1/3/1/2/1/1/1/1/1/3	-
5	38.258940	-0.706920	2	denm/0/3/3/1/1/1/3/1/3/0/0/0/0/0/2	-
6	38.259070	-0.708270	3	denm/0/3/3/1/1/1/3/1/2/1/1/1/1/1/3	DENM
7	38.260010	-0.713920	1	denm/0/3/3/1/1/1/3/1/2/1/1/1/1/1/2	-
8	38.260180	-0.721900	2	denm/0/3/3/1/1/1/3/1/2/1/1/1/1/1/3	-
9					
10					

Figura 4.11. Traza de resultados ejemplo cliente MQTT conectado a broker local

Con este ejemplo, se ha podido verificar que el cliente MQTT implementado gestiona correctamente las suscripciones y cancelación a tópicos en función de la posición GPS del vehículo y recibe los mensajes de alerta DENM cuando se encuentra dentro de un quadtree afectado.

4.5. RECEPCIÓN DE MENSAJES DENM DESDE NODO EDGE

Recordemos que hasta ahora, las pruebas y ejemplos explicados se han realizado conectando el geoservice al PC donde se ha llevado a cabo todo el desarrollo. Este enfoque ha sido útil para realizar pruebas iniciales y asegurarse de que cada componente individual del sistema funciona correctamente en un entorno controlado.

Sin embargo, para la siguiente demostración, se utilizarán los componentes desplegados en el nodo Edge, los cuales se encuentran en un PC de la universidad UMH. Por lo tanto, conectaremos el geoservice al broker MQTT que hay allí disponible. Esta configuración es más representativa de cómo funcionará el sistema en un entorno real y, por lo tanto, es crucial para validar que el sistema completo funciona tal y como se espera.

Supongamos el escenario que se muestra en la Figura 4.12. Tenemos un vehículo siguiendo la trayectoria marcada por la línea azul del mapa inferior. En este caso hemos seleccionado una zona de Bologna – Italia y un nivel de zoom 15. El punto de color rojo marcará el lugar en el que se encontraba el cliente cuando recibe el mensaje DENM y el punto negro indica el lugar donde se ha producido el evento en cuestión. En la parte superior de los quadrees se pueden ver los tópicos correspondientes a dichos quadrees, el de la derecha tiene un color amarillo indicando que ese va a ser el quadtree a notificar.

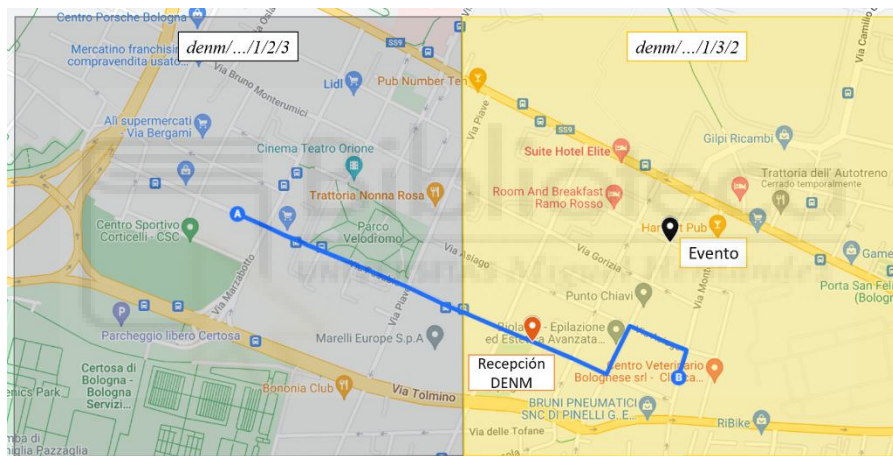
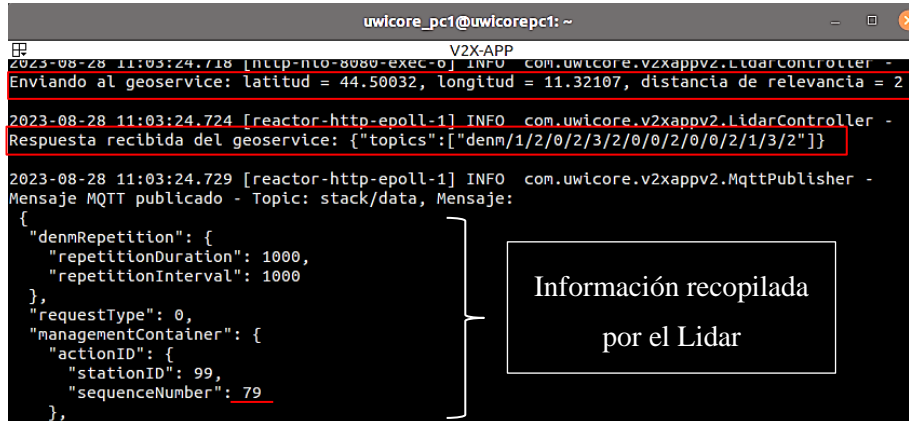


Figura 4.12. Escenario ejemplo cliente MQTT conectado broker en nodo edge

El proceso comienza con la detección de un evento relevante al tráfico por parte del sensor Lidar. Este recopila toda la información relativa al evento y la envía vía API REST al nodo Edge. Allí la V2X APP recibe los datos, los analiza y extrae aquellos que son relevantes para el geoservice, es decir, la ubicación exacta donde se ha producido el evento (44.50032, 11.32107) y la distancia de relevancia asociada (2 = 200 metros). Con estos datos realiza la petición HTTP GET solicitando al geoservice que le indique los tópicos de relevancia para el evento producido. Por su parte, el geoservice obtiene aquellos quadrees que interseccionan con el área de relevancia del evento, calcula sus respectivos tópicos (en este caso “denm/.../1/3/2”) y los envía como respuesta a la V2X APP. Esto mismo se puede observar en la Figura 4.13 que contiene los logs de la aplicación V2X. La última tarea que realiza V2X APP es enviar vía MQTT a Vanetza

toda la información recogida por el Lidar junto con los tópicos indicados por el geoservice. Una vez recibe esta información, Vanetza basándose en los estándares correspondientes, conforma el mensaje DENM y lo publica a los tópicos afectados por la alerta.



```
uwicore_pc1@uwicorepc1: ~  
V2X-APP  
2023-08-28 11:03:24.718 [http-nto-8080-exec-6] INFO com.uwicore.v2xappv2.LidarController -  
Enviando al geoservice: latitud = 44.50032, longitud = 11.32107, distancia de relevancia = 2  
2023-08-28 11:03:24.724 [reactor-http-epoll-1] INFO com.uwicore.v2xappv2.LidarController -  
Respuesta recibida del geoservice: {"topics":["denm/1/2/0/2/3/2/0/0/2/0/0/2/1/3/2"]}  
2023-08-28 11:03:24.729 [reactor-http-epoll-1] INFO com.uwicore.v2xappv2.MqttPublisher -  
Mensaje MQTT publicado - Topic: stack/data, Mensaje:  
{  
  "denmRepetition": {  
    "repetitionDuration": 1000,  
    "repetitionInterval": 1000  
  },  
  "requestType": 0,  
  "managementContainer": {  
    "actionID": {  
      "stationID": 99,  
      "sequenceNumber": 79  
    }  
  },  
}
```

Figura 4.13. Logs de la aplicación V2X

Cuando Vanetza consigue publicar de forma correcta el mensaje DENM, responde vía MQTT con un mensaje que indica el número de secuencia del evento junto con un *true* que envía a V2X APP. Con la herramienta MQTT Explorer podemos visualizar y analizar los mensajes intercambiados en esta red MQTT, por ello es posible observar ese mensaje de confirmación por parte de Vanetza a V2X APP del que hablamos (“79 – true”) en la Figura 4.14, donde 79 es el número de secuencia en este caso, confirmando que Vanetza ha publicado el mensaje DENM que contiene la información relacionada con el evento que ha detectado el Lidar.

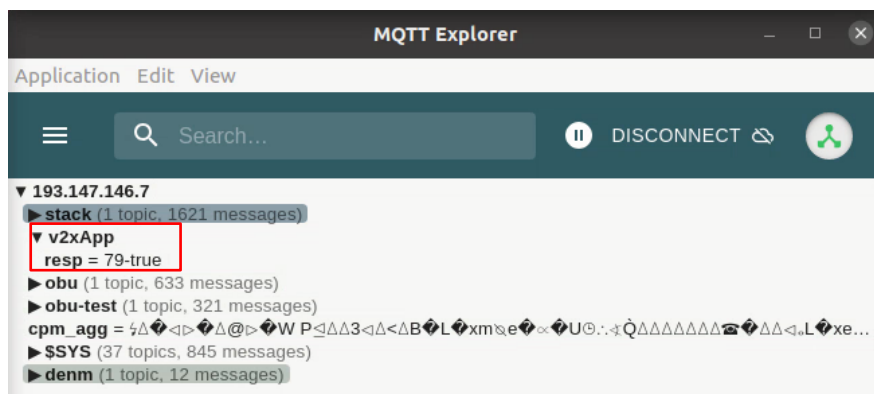


Figura 4.14. Número de secuencia observado desde MQTT Explorer

Una vez realizada la publicación de la alerta, todos aquellos clientes que estén suscritos al tópico afectado, recibirán el mensaje DENM. Este es el caso del vehículo que está

siguiendo la trayectoria de la Figura 4.14, quien en el momento en el que se produce la notificación del evento se encuentra dentro del quadtree “denm/.../1/3/2”, tal y como lo refleja su fichero de resultados generado en la Figura 4.15. Inicialmente podemos observar la suscripción al tópicico “denm/.../1/2/3” y después al tópicico “denm/.../1/3/2”, una vez el vehículo abandonó el primer quadtree y estaba lo suficientemente lejos, procedió a la desuscripción de su tópicico, y por último, cuando se encontraba en el punto (44.49857, 11.31763) recibe el mensaje DENM publicado al tópicico “denm/.../1/3/2” alertándolo del evento producido para que tome las acciones pertinentes.

id	Latitud	Longitud	Evento	Topic	Mensaje
1	44.500830	11.310280	1	denm/1/2/0/2/3/2/0/0/2/0/0/2/1/2/3	-
2	44.499160	11.315630	1	denm/1/2/0/2/3/2/0/0/2/0/0/2/1/3/2	-
3	44.498830	11.316710	2	denm/1/2/0/2/3/2/0/0/2/0/0/2/1/2/3	-
4	44.498570	11.317630	3	denm/1/2/0/2/3/2/0/0/2/0/0/2/1/3/2	DENM

Figura 4.15. Traza de resultados ejemplo cliente MQTT conectado a broker en nodo edge

Vemos así que la prueba realizada para validar el funcionamiento del sistema completo en un entorno que simula una situación real, asegurándose de que 1) el geoservice desplegado en el nodo edge es capaz de calcular los tópicos correctos usando los datos proporcionados por V2X APP, 2) el broker MQTT de esta red es capaz de gestionar las suscripciones de los clientes y los mensajes que se publican a sus tópicos, y 3) los vehículos son capaces de recibir estos mensajes de alerta, funciona correctamente.

El siguiente paso es realizar una prueba que involucre a varios clientes. El objetivo es demostrar que el sistema no solo es capaz de notificar a un solo cliente cuando ocurre un evento, sino que también es capaz de manejar múltiples clientes simultáneamente y notificar a cada uno de ellos de manera adecuada si están suscritos al tópicico afectado por un evento. En dicha prueba, se simularán varios vehículos moviéndose por diferentes áreas del mapa, se utilizará unas trazas GPS que corresponden a trazas reales obtenidas en la ciudad de Bolonia [23]. Los vehículos irán tomando su dispositivo GPS para determinar su posición exacta y, por lo tanto, conocer en todo momento el quadtree en el que se encuentra a medida que sigue su trayectoria. Como se ha explicado anteriormente, se suscribirán automáticamente a los tópicos correspondientes a su ubicación actual y cancelaran su suscripción de los tópicos de las áreas que van dejando atrás, además de recibir mensajes DENM si son relevantes para su posición en el mapa.

Concretamente, tenemos 6 clientes cada uno de ellos siguiendo sus respectivas trayectorias a través del mapa. La marca de color negro de la Figura 4.16, se sitúa el punto exacto (44.49596, 11.32422) donde se ha producido el determinado evento, el cual afecta a un área de relevancia de 100 metros (datos obtenidos de V2X APP). En cambio, los puntos rojos indican el lugar en el que se encontraba el cliente cuando recibe el mensaje DENM. Los quadtree con fondo amarillo representan los quadtree a los que se debe dirigir la alerta, en la parte superior de cada quadtree se puede ver su respectivo tópicos.

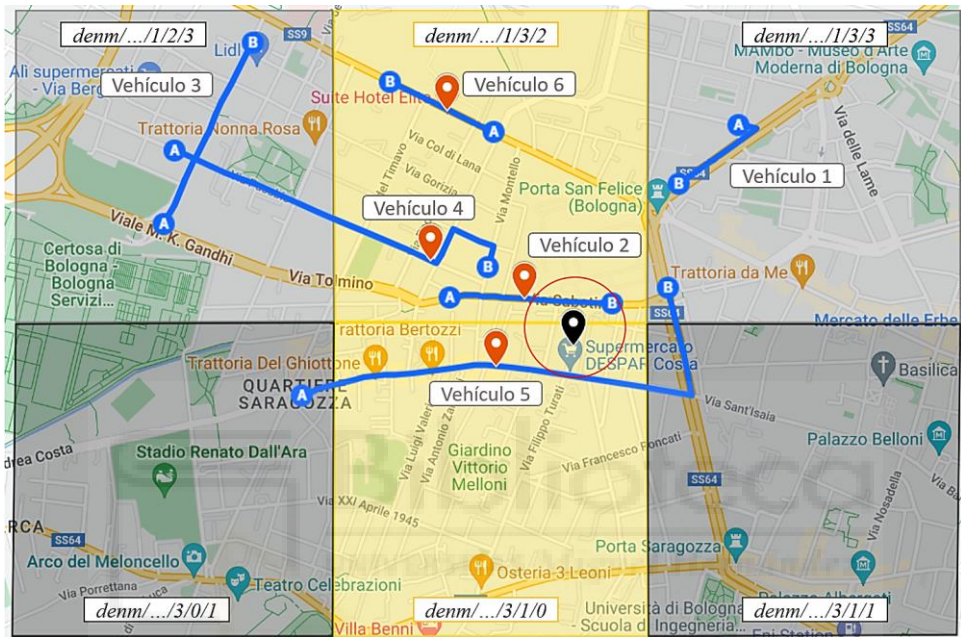


Figura 4.16. Escenario ejemplo varios clientes MQTT

El fichero de resultados generado por los vehículos de este escenario, se muestran en la Figura 4.17. Se puede observar que solo aquellos clientes dentro de los quadtree afectados recibirán el mensaje DENM (2, 4, 5 y 6).

id	Latitud	Longitud	Evento	Topic	Mensaje
1	44.501335	11.330155	1	denm/1/2/0/2/3/2/0/0/2/0/0/2/1/3/3	-
2	44.497128	11.320168	1	denm/1/2/0/2/3/2/0/0/2/0/0/2/1/3/2	-
3	44.497115	11.322510	3	denm/1/2/0/2/3/2/0/0/2/0/0/2/1/3/2	DENM
4	44.498691	11.309637	1	denm/1/2/0/2/3/2/0/0/2/0/0/2/1/2/3	-
6	44.500734	11.310504	1	denm/1/2/0/2/3/2/0/0/2/0/0/2/1/2/3	-
7	44.499279	11.315316	1	denm/1/2/0/2/3/2/0/0/2/0/0/2/1/3/2	-
8	44.498877	11.316626	2	denm/1/2/0/2/3/2/0/0/2/0/0/2/1/2/3	-
9	44.498144	11.319041	3	denm/1/2/0/2/3/2/0/0/2/0/0/2/1/3/2	DENM
10	44.494409	11.314218	1	denm/1/2/0/2/3/2/0/0/2/0/0/2/3/0/1	-
11	44.494854	11.315219	1	denm/1/2/0/2/3/2/0/0/2/0/0/2/3/1/0	-
12	44.495103	11.316628	2	denm/1/2/0/2/3/2/0/0/2/0/0/2/3/0/1	-
13	44.495441	11.321538	3	denm/1/2/0/2/3/2/0/0/2/0/0/2/3/1/0	DENM
14	44.494954	11.326205	1	denm/1/2/0/2/3/2/0/0/2/0/0/2/3/1/1	-
15	44.494812	11.327619	2	denm/1/2/0/2/3/2/0/0/2/0/0/2/3/1/0	-
16	44.496048	11.328028	1	denm/1/2/0/2/3/2/0/0/2/0/0/2/1/3/3	-
17	44.502334	11.318252	1	denm/1/2/0/2/3/2/0/0/2/0/0/2/1/3/2	-
18	44.501829	11.319809	3	denm/1/2/0/2/3/2/0/0/2/0/0/2/1/3/2	DENM

Figura 4.17. Traza de resultados ejemplo múltiples clientes MQTT

A diferencia de los ejemplos anteriores, vamos a utilizar dicho archivo de resultados para representarlos gráficamente en *Matlab* y poder explicar los acontecimientos sucedidos de forma más visual empleando la Figura 4.18. Esta figura muestra las posiciones (latitud y longitud) de los diferentes vehículos. En color negro se muestra las trayectorias que siguen los vehículos, igual que en la Figura 4.16. Por cada suscripción nueva, es decir, a un tópico diferente, se representará sobre el recorrido de los vehículos un círculo de colores diferentes. Las cancelaciones a tópicos se representan mediante una equis, y se le asigna el mismo color que el círculo que representa la suscripción a ese mismo tópico.

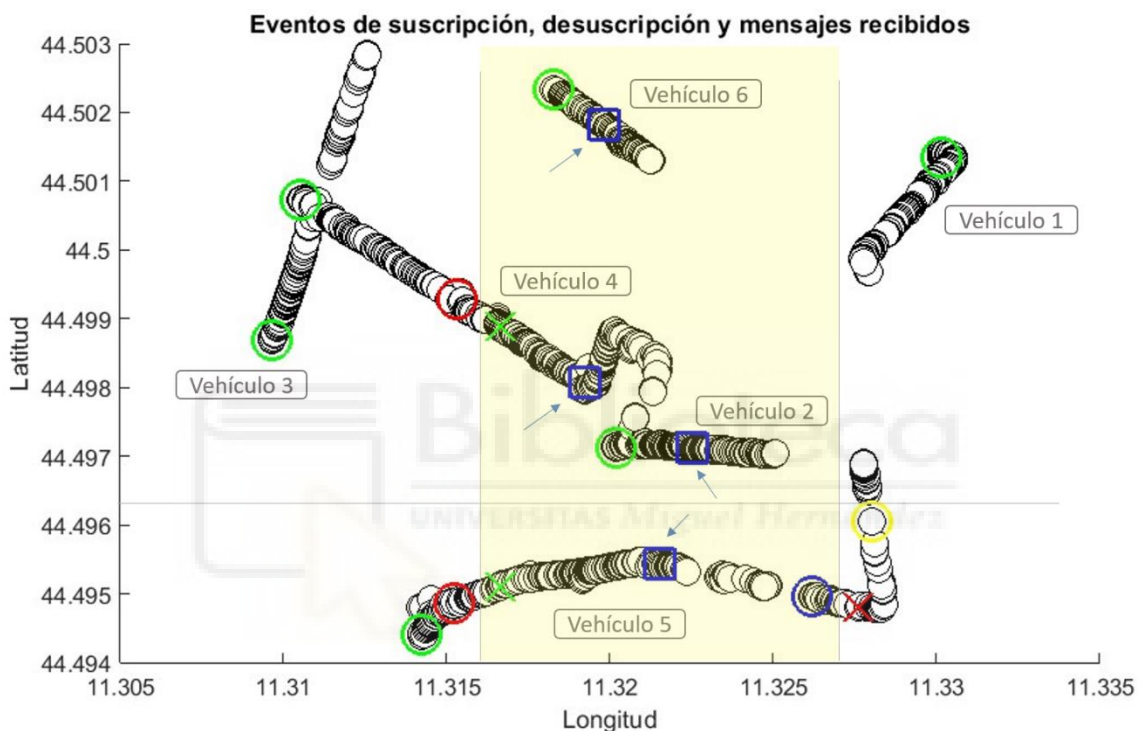


Figura 4.18. Representación en Matlab de la traza de resultados

Por último, los cuadrados de color azul indican la ubicación de los clientes en el momento en el que reciben el mensaje DENM.

Para que se comprenda mejor la Figura 4.18, vamos a analizar el caso del *Vehículo 5*. Este comienza su ruta en el quadtree izquierdo inferior realizando su respectiva suscripción al tópico “denm/.../3/0/1” (señalizado con un círculo de color verde). Unos metros más adelante, como resulta que está próximo a un borde, se suscribe también al tópico del siguiente quadtree al que va a adentrarse (“denm/.../3/1/0”), representado con un círculo rojo. La siguiente cruz de color verde que encontramos sobre la trayectoria del vehículo representa la desuscripción del tópico al que se suscribió inicialmente, si nos fijamos es también de color verde, de modo que mediante círculos representamos el

momento en el que se produce una suscripción a un determinado tópico y con las cruces del mismo color que el círculo de la suscripción, representamos la desuscripción del cliente de ese mismo tópico.

Luego encontramos un cuadrado de color azul (se ha señalado con un flecha azul), este símbolo indica la posición en la que se encontraba el vehículo en el momento en el que recibe el mensaje DENM de alerta publicado desde el nodo Edge. Todos los vehículos que se encuentren dentro de los quadrees afectados por la alerta, es decir, aquellos que estén suscritos a los tópicos “denm/.../1/3/2” y “denm/.../3/1/0” serán alertados (como es el caso del vehículo 5 y también de los vehículos 2, 4 y 6). Sin embargo, los que estén fuera de estas zonas, no recibirán nada debido a que no es un evento relevante para su posición. Todo esto se refleja también en la traza de resultados de la Figura 4.17. Traza de resultados ejemplo múltiples clientes MQTT

El siguiente símbolo que encontramos es un círculo de color azul que indica una nueva suscripción, concretamente al tópico “denm/.../3/1/1” debido a que se encuentra próximo a dicho quadtree. Luego se muestra una cruz de color rojo, lo que significa que ha anulado su suscripción al tópico “denm/.../3/0/1”. Por último, encontramos una nueva suscripción al tópico “denm/.../1/3/3” debido a su proximidad al siguiente quadtree.

Con el resto de los vehículos ocurre lo mismo. Los círculos simbolizan nuevas suscripciones, las cruces del mismo color que los círculos indican la desuscripción de ese mismo tópico, y por último, los cuadrados de color azul indican el momento en el que se recibe un mensaje DENM publicado por Vanetza.

Por lo tanto, concluimos este ejemplo en el que hemos puesto a prueba la implementación con varios clientes (vehículos) conectados a la red MQTT, simulando una situación más parecida a la vida real.

5. CONCLUSIONES

Este trabajo se enmarca en los sistemas C-ITS, donde gracias al uso de la tecnología de *edge computing*, se logra proporcionar las exigentes latencias que requieren las comunicaciones V2X, y que las redes celulares 5G por sí solas no son capaces de abordar. De modo que, en lugar de realizar el intercambio de mensajes C-ITS entre los propios vehículos, se ha aprovechado esta ventaja que ofrece el nodo Edge desarrollado por el departamento, para la difusión de información a través de las redes celulares.

El principal objetivo de este trabajo ha sido desarrollar un geoservicio que le otorgue a dicho nodo Edge, la capacidad de realizar esa difusión de mensajes a través de la red celular de forma eficiente. Para ello, el geoservice tiene en cuenta la posición geográfica de los nodos/vehículos que hay en la red y evita que se les envíe mensajes que no son relevantes para su posición.

A lo largo de esta memoria, detallamos tanto su concepto y finalidad como su funcionamiento interno, prestando especial atención a aspectos como el nivel de zoom y la interfaz de comunicación entre el geoservicio y la aplicación V2X.

La implementación del cliente MQTT ha sido otro aspecto crucial del proyecto, ya que ha permitido la gestión efectiva de los tópicos y la recepción de mensajes DENM.

Se han llevado a cabo varias pruebas para evaluar la solución propuesta, incluida la generación de tópicos para diferentes niveles de zoom, la gestión de tópicos y recepción de mensajes DENM en el cliente MQTT, y la validación integral del sistema, que incluyó la recepción de mensajes DENM desde un nodo edge.

El proceso de evaluación y validación de la solución propuesta nos ha permitido confirmar la efectividad del sistema en la práctica. Los resultados demuestran no solo la viabilidad sino también el valor añadido que nuestra solución podría aportar al ecosistema de los C-ITS. Por lo que podemos concluir que este proyecto ha desarrollado una solución integral que aborda una necesidad crítica en el ámbito de los C-ITS: la difusión eficiente de mensajes a vehículos en función de su localización geográfica. Se ha diseñado un geoservicio innovador, se ha implementado un cliente MQTT eficiente y se ha validado la solución en su totalidad. Aunque se han superado muchos desafíos, se espera que este trabajo sirva como base para futuras investigaciones y desarrollos en este campo, contribuyendo así a una movilidad más segura y eficiente.

6. TRABAJOS FUTUROS

A pesar de los logros obtenidos y las soluciones propuestas en este proyecto, se reconoce que siempre hay margen para la mejora y el desarrollo adicional. La dinámica en constante cambio de la tecnología y las infraestructuras de transporte implica que siempre habrá nuevas oportunidades y desafíos que abordar. Con esto en mente, es importante señalar que este proyecto, aunque ambicioso y exhaustivo, representa solo un paso en un camino continuo hacia la mejora de los sistemas C-ITS. Hay varias áreas identificadas durante el desarrollo de este proyecto que se consideran importantes para futuras investigaciones y desarrollos. Aunque el geoservicio desarrollado y la implementación del protocolo MQTT han mostrado ser soluciones efectivas para superar algunos de los desafíos actuales, hay múltiples facetas de este problema que aún no han sido exploradas completamente.

Uno de los posibles trabajos futuros más relevantes sería el desarrollo de una interfaz gráfica, en forma de una aplicación, que permita a los conductores visualizar en tiempo real en las pantallas de sus vehículos la información relevante sobre eventos de tráfico. Actualmente, el geoservicio desarrollado en este proyecto se encarga de difundir mensajes relevantes a nodos específicos en función de su localización. Sin embargo, una extensión natural de este servicio sería proporcionar una manera más intuitiva y accesible para que los conductores interpreten esta información. La aplicación podría incluir un mapa en tiempo real que marque el lugar donde se ha producido el evento en cuestión. Por ejemplo, en caso de un accidente de tráfico, la aplicación podría mostrar la ubicación exacta del accidente, permitiendo a los conductores tomar decisiones informadas sobre si deben buscar rutas alternativas. Además, la aplicación podría integrarse con sistemas de navegación existentes para proporcionar rutas alternativas automáticamente. Este desarrollo no solo mejoraría la experiencia del usuario, sino que también podría tener un impacto positivo en la gestión del tráfico, ayudando a prevenir congestiones y mejorando la seguridad en la carretera. Sin embargo, es importante tener en cuenta que el desarrollo de esta aplicación también plantearía nuevos desafíos, como garantizar la privacidad y seguridad de los datos de los usuarios, y asegurarse de que la aplicación sea accesible y fácil de usar para todos los conductores.

Otra área importante para futuros trabajos sería escalar la implementación actual para utilizar múltiples nodos edge que puedan comunicarse entre sí, en lugar de depender de un solo nodo. Esto implicaría desarrollar un sistema de coordinación y comunicación

entre los nodos edge para asegurarse de que los mensajes se distribuyan de manera eficiente y efectiva. A medida que aumenta el número de vehículos y dispositivos conectados en la red, es fundamental tener una infraestructura que pueda manejar este aumento de carga sin comprometer el rendimiento.

Además, es crucial realizar pruebas en un entorno real y no solo en un entorno simulado. Aunque las simulaciones son una herramienta valiosa para el desarrollo y la validación de sistemas, hay ciertos aspectos y variables del mundo real que no pueden ser completamente replicados en un entorno simulado. Por lo tanto, es necesario realizar pruebas en un entorno real para validar completamente el sistema y asegurarse de que funcione correctamente en todas las situaciones posibles. Realizar pruebas en un entorno real también permitiría identificar y abordar cualquier problema o limitación no anticipada que pueda surgir, y proporcionaría una evaluación más precisa de cómo el sistema afecta la experiencia del conductor y la gestión del tráfico en situaciones reales.

Otra opción sería que debido a que los vehículos tienen que suscribirse y anular dichas suscripciones a los quadrees relevantes cada vez que cambian de ubicación, con el fin de mejorar la eficiencia de este proceso, se podría implementar un sistema de predicción de trayectoria que permita anticipar los cambios de ubicación de los vehículos y gestionar las suscripciones de manera más eficiente. Este sistema podría utilizar algoritmos de predicción de movimiento basados en datos históricos y en tiempo real para estimar la trayectoria futura de un vehículo y suscribirlo automáticamente a los quadrees relevantes en su camino. De esta manera, el vehículo podría recibir los mensajes relevantes con anticipación, sin tener que gestionar sus suscripciones de manera continua.

Por último, se propone una idea de futura mejora para el nodo Edge, la cual sugiere incorporar un servicio de análisis de datos en tiempo real en el nodo edge. Actualmente, el nodo edge tiene una aplicación V2X que recibe información de un lidar, solicita al geoservicio los tópicos de los quadrees afectados y los envía vía MQTT a Vanetza, servicio que se encarga de conformar según los estándares los mensajes DENM y publicarlos a los tópicos que indica el geoservicio. Sin embargo, se podría mejorar aún más la eficiencia y la utilidad del sistema incorporando un servicio que pueda analizar los datos recibidos en tiempo real para identificar patrones, predecir eventos y proporcionar recomendaciones. Por ejemplo, el servicio podría analizar los datos de tráfico para predecir posibles congestiones o accidentes antes de que ocurran y enviar recomendaciones a los vehículos cercanos para evitar estas áreas. Este servicio también

podría incorporar aprendizaje automático y análisis predictivo para mejorar continuamente su precisión y utilidad con el tiempo. Además, los datos analizados podrían ser compartidos con otros servicios y aplicaciones, como sistemas de navegación y aplicaciones de tráfico, para mejorar la gestión del tráfico en general.

En conclusión, hay varias áreas de mejora y expansión para el sistema de transporte inteligente cooperativo desarrollado en este proyecto. La implementación de una interfaz gráfica para conductores, la escalabilidad a múltiples nodos edge, o la optimización de la gestión de suscripciones a través de un sistema de predicción de trayectoria son solo algunos ejemplos de posibles trabajos futuros que podrían ayudar a mejorar la eficiencia, seguridad y usabilidad del sistema. La tecnología y las necesidades de los usuarios están en constante evolución, y es importante seguir investigando y desarrollando nuevas soluciones para mantenerse al día y ofrecer los mejores servicios posibles.



7. Bibliografía

- [1] C. B. Abellón, «Las comunicaciones V2X son esenciales para una conducción más segura y eficiente,» 25 Febrero 2020. [En línea]. Available: <https://www.gmv.com/es-es/media/blog/automocion/las-comunicaciones-v2x-son-esenciales-para-una-conduccion-mas-segura-y>.
- [2] B. Coll-Perales, «End-to-End V2X Latency Modeling and Analysis in 5G Networks,» *IEEE Transactions on Vehicular Technology*, vol. 72, nº 4, pp. 5094-5109, Abril 2023.
- [3] Parlamento de la Unión Europea, «Directiva 2010/40/UE por la que se establece el marco para la implantación de los sistemas de transporte inteligentes en el sector del transporte por carretera y para las interfaces con otros modos de transporte.,» 2010.
- [4] Comisión Europea, «COMUNICACIÓN DE LA COMISIÓN AL PARLAMENTO EUROPEO, AL CONSEJO, AL COMITÉ ECONÓMICO Y SOCIAL EUROPEO Y AL COMITÉ DE LAS REGIONES En ruta hacia la movilidad automatizada: estrategia de la UE para la movilidad del futuro.,» 2018.
- [5] Amazon, «¿Qué es el 5G?,» [En línea]. Available: <https://aws.amazon.com/es/what-is/5g/>.
- [6] U. Noyer, T. Schlaug, P. Cercato y L. Mikkelsen, «MOBiNET – architecture overview of an innovative platform for European,» 2015.
- [7] M. Fünfroeken, «Deliverable D4.3 Architecture of the Car2X,» 30 Junio 2018.
- [8] 5. PPP, «Deliverable D2.1 Test Case Definition and Trial site Description,» 31 julio 2020.
- [9] C-Roads, «C-ITS IP Based Interface Profile,» 4 Octubre 2022.

- [10] Amazon, «¿Qué es MQTT?,» febrero 2022. [En línea]. Available: <https://aws.amazon.com/es/what-is/mqtt/>.
- [11] iMatix. [En línea]. Available: <https://imatix.com/>.
- [12] N. Clement, «AMQP vs MQTT: Messaging protocols compared,» 20 Junio 2023. [En línea]. Available: <https://www.cloudamqp.com/blog/amqp-vs-mqtt.html>.
- [13] IBM, «¿Qué es Java Spring Boot?,» 2019. [En línea]. Available: <https://www.ibm.com/es-es/topics/java-spring-boot>.
- [14] Microsoft, «Bing Maps Tile System,» 6 Agosto 2022. [En línea]. Available: <https://learn.microsoft.com/en-us/bingmaps/articles/bing-maps-tile-system>.
- [15] Docker. [En línea]. Available: <https://www.docker.com/>.
- [16] Kubernetes, «Orquestación de contenedores para producción,» [En línea]. Available: <https://kubernetes.io/es/>.
- [17] O. VANETZA, «Vanetza - Your open-source ETSI C-ITS protocol stack,» [En línea]. Available: <https://www.vanetza.org/>.
- [18] Wikipedia, «JSON,» 5 Agosto 2023. [En línea]. Available: <https://es.wikipedia.org/wiki/JSON>.
- [19] E. Mosquitto, «mosquitto.h,» [En línea]. Available: <https://mosquitto.org/api/files/mosquitto-h.html>.
- [20] T. Nordquist, «MQTT Explorer,» [En línea]. Available: <http://mqtt-explorer.com/>.
- [21] M. V. Aerde, «Single regime speed-flow-density relationship for congested and uncongested highways,» de *Proc. 74th TRB Annual*, Washington DC, 1995.
- [22] M. Eclipse, «mosquitto_pub man page,» [En línea]. Available: https://mosquitto.org/man/mosquitto_pub-1.html.

- [23] J. Gozalvez, M. S. y R. Bauza, «IEEE 802.11p Vehicle to Infrastructure Communications in Urban Environments,» *IEEE Communications Magazine*, vol. 50, nº 5, pp. 176-183, 2012.
- [24] T. Fioreze y G. Heijenk, «Extending DNS to support geocasting: towards VANETs: a proposal», IEEE Vehicular Networking Conference, USA, Diciembre 2010.
- [25] IBM, «Soluciones de Edge Computing,» 2020. [En línea]. Available: <https://www.ibm.com/es-es/edge-computing?>
- [26] J. M. R. Moncayo, «Qué es REST: Conoce su potencia,» 17 Mayo 2018. [En línea]. Available: <https://openwebinars.net/blog/que-es-rest-conoce-su-potencia/>.
- [27] RedHat, «¿Qué es la arquitectura del edge computing?,» 14 Mayo 2021. [En línea]. Available: <https://www.redhat.com/es/topics/edge-computing/what-is-edge-architecture>.
- [28] KeepCoding, «¿Qué es el estándar AMQP (Advanced Message Queuing Protocol)?,» 16 Agosto 2023. [En línea]. Available: <https://keepcoding.io/blog/que-es-el-estandar-amqp/>.