



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Universitat Politècnica de València

Departamento de Informática de Sistemas y Computadores

Solución para la gestión de tráfico de Valencia en base a criterios ambientales

TRABAJO DE FIN DE MÁSTER

Máster Universitario en Ingeniería de Computadores y Redes

Autor

Marcos Terol Lloret

Director

Dr. Carlos Miguel Tavares de Araujo Cesariny Calafate

Fecha

4 de Diciembre de 2020

Resumen

Existen multitud de planificadores de rutas que tratan de proporcionar la ruta más rápida o la mejor ruta al usuario teniendo en cuenta la congestión del tráfico, la estimación de este a través de datos históricos, o previendo la ruta que van a tomar los vehículos.

Lo que se trata en este estudio es, además de ofrecer la mejor ruta por los criterios estándar, como pueden ser tiempo de trayecto o distancia recorrida, ofrecer una alternativa capaz de atender también a criterios ambientales, es decir, según la contaminación de ciertos tramos, desviar el tráfico para minimizar la circulación por esas zonas, y así poder descongestionar las áreas más afectadas de la ciudad.

Se ha realizado un análisis de la reducción de velocidad adecuada para disminuir el número de vehículos en diferentes tipos de vías. Junto a esto se ha establecido como afecta el corte de ciertas vías al tráfico general de la ciudad y así poder comprobar si es eficiente la limitación del tráfico por motivos medioambientales.

Palabras clave: Redes Vehiculares; gestión de tráfico; SUMO; ABATIS.

Abstract

There are many route managers who try to provide the fastest route or the best route to the user, taking into account traffic congestion, estimating it through historical data, or anticipating the route that vehicles will take.

What is discussed in this study is, in addition to offering the best route according to standard criteria, such as travel time or distance traveled, to offer an alternative capable of also meeting environmental criteria, that is, according to the contamination of certain sections. , deviate traffic to minimize circulation in these areas, and thus be able to decongest the most affected areas of the city.

An analysis of the adequate speed reduction has been carried out to reduce the number of vehicles on different types of roads. Along with this, it has been established how the cutting of certain roads affects the general traffic of the city and thus be able to check if the limitation of traffic for environmental reasons is efficient.

Keywords: Vehicle Networks; traffic management; SUMO; ABATIS.

Resum

Hi ha multitud de planificadors de rutes que tracten de proporcionar la ruta més ràpida o la millor ruta a l'usuari tenint en compte la congestió de trànsit, l'estimació d'aquest a través de dades històriques, o preveient la ruta que van a prendre els vehicles. El que es tracta en aquest estudi és, a més d'oferir la

millor ruta pels criteris estàndard, com poden ser temps de trajecte o distància recorreguda, oferir una alternativa capaç d'atendre també a criteris ambientals, és a dir, segons la contaminació de certs trams, desviar el trànsit per minimitzar la circulació per aquestes zones, i així poder descongestionar les àrees més afectades de la ciutat.

S'ha realitzat una anàlisi de la reducció de velocitat adequada per disminuir el nombre de vehicles en diferents tipus de vies. Ademés d'això s'ha establert com afecta el tall de certes vies al trànsit general de la ciutat i així poder comprovar si és eficient la limitació del trànsit per motius mediambientals.

Paraules clau: Xarxes Vehiculares; gestió de trànsit; SUMO; Abatis.

Índice general

Índice de figuras	iii
1 Introducción	1
1.1 Trabajo Previo	1
1.2 Planteamiento de problema	1
1.3 Objetivos	2
1.4 Estructura	2
2 Estado del Arte	4
2.1 Vehicular Traffic Management based on Traffic Engineering for Vehicular Ad Hoc Networks	4
2.2 DIVERT	5
2.3 MGRM	6
2.4 Deep learning support for intelligent transportation systems	7
2.5 Vehicular Crowd Management	7
3 Herramientas utilizadas	9
3.1 ABATIS	9
3.1.1 ABATIS Tools	11
3.2 SUMO	12
3.2.1 Paradigmas básicos y características	13
3.3 OMNeT++	14
3.3.1 Componentes	16
3.3.2 Modelos	17
3.3.3 Plataformas	17
3.4 INET	17
3.5 TraCI	18
3.5.1 sumo-launchd	18

4	Implementación de la solución propuesta	20
4.1	TraCI	21
4.2	LinkABATIS	23
4.3	Regulación dinámica del flujo de tráfico	27
5	Resultados	33
6	Conclusiones	39
6.1	Conclusiones	39
6.2	Trabajos futuros	40
	Bibliografía	41

Índice de figuras

2.1	Diseño Re-Route.	5
3.1	Respuesta del servidor ABATIS	10
3.2	Simulación gráfica SUMO	13
3.3	Jerarquía de módulos Omnet++	15
3.4	Interfaz Gráfica Tkenv	16
3.5	Protocolo de comunicación entre TraCI y SUMO.	18
4.1	Arquitectura de simulación.	20
4.2	Fragmento archivo .rou	21
4.3	Fragmento archivo .add	22
4.4	Comunicación Omnet++ - SUMO	23
4.5	Comunicación OMNeT++ - linkABATIS - ABATIS	24
4.6	Ejemplo de archivo CSV usado en el proyecto.	25
4.7	Funcionamiento herramientas de ABATIS	25
4.8	Índice AEMET de contaminación del aire.	27
4.9	Gran Vía del Marqués del Turia	28
4.10	Gráfica reducción tráfico Gran Vía Marqués del Turia	29
4.11	Calle Dénia	29
4.12	Calle Almirante Cadarso	30
4.13	Gráfica de reducción del tráfico: Calle Denia - Calle Almirante Cadarso.	30
4.14	Calle Cádiz	31
4.15	Calle Sueca	31
4.16	Gráfica de reducción del tráfico: Calle Sueca - Calle Cádiz.	32
5.1	Calle del Almirante Cadarso.	34
5.2	Gráfica velocidad al cortar calle del Almirante Cadarso	35
5.3	Ruta Avinguda Peris i Valero - Calle Hernan Cortés.	37
5.4	Desviación Ruta Avinguda Peris i Valero - Calle Hernan Cortés.	38

Capítulo 1

Introducción

1.1 Trabajo Previo

Este trabajo se basa en un proyecto previo realizado por Jorge Zambrano [1] en su tesis doctoral. En dicha tesis se desarrolló una Ecuación de Predicción de Tráfico con la que se puede predecir las condiciones futuras del tráfico. Con ese trabajo consiguió mejoras globales en términos de velocidad promedia y tiempo de trayecto, además de obtener indicadores de una reducción en la congestión y fluidez del tráfico. En este trabajo se propone extender dicha solución para permitir regular el tráfico en diferentes zonas atendiendo a criterios ambientales.

1.2 Planteamiento de problema

Si tomamos los datos de la Organización Mundial de la Salud podemos observar que a causa de la polución aérea mueren cada año cerca de 4,2 millones de personas, y que el 91 % de la población mundial vive en zonas que exceden los límites recomendados relativos a calidad del aire [2]. Estos datos son sin duda alarmantes, y deben hacernos cuestionar qué modelo de sociedad estamos construyendo.

Las áreas más perjudicadas por la contaminación son, sin duda, las grandes ciudades, las cuales sufren por emisiones gaseosas de industrias cercanas y por la emisiones de los vehículos en circulación. Algunas ciudades como Londres ya han tomado medidas, por ejemplo restringiendo el tráfico en el centro de la ciudad [3], o como Valencia, apostando por un transporte sostenible creando el llamado anillo ciclista [4], que consiste en la creación de una red de carriles bici que conectan toda la ciudad.

En general, se podría decir que son soluciones acertadas y necesarias, pero no suficientes. En este proyecto se intenta dar una solución adicional que sirva como complemento a las medidas que ya se han tomado. Concretamente, busca des-

congestionar las áreas más perjudicadas por la contaminación, y esto se consigue reduciendo la cantidad de gases que se vierten en esas zonas. Sabiendo los índices de contaminación de cada distrito de la ciudad se pueden ofrecer a los conductores rutas alternativas que alivien la polución en esas áreas más perjudicadas.

1.3 Objetivos

El principal objetivo del proyecto es modificar el servidor de rutas ABATIS, sistema centralizado para gestionar todo el tráfico en una ciudad, de manera que tenga en cuenta los datos de contaminación, o zonas específicas como pueden ser colegios, ya que los niños son más sensibles a este tipo de contaminación, lo que puede generarles enfermedades cardiorrespiratorias de por vida.

Este Trabajo de Fin de Máster tiene como objetivos adicionales:

- Desarrollar un algoritmo que tenga en cuenta datos de contaminación, número de vías, cantidad de vehículos y su velocidad.
- Realizar pruebas en diferentes escenarios para comprobar la viabilidad del sistema.
- Analizar los resultados obtenidos en las diferentes pruebas realizando modificaciones en las características de la vía y/o vehículos.
- Definir trabajos futuros.

1.4 Estructura

Este TFM está estructurado de la siguiente manera: este primer capítulo incluye una presentación e introducción al TFM, detallando los objetivos y motivación del mismo.

A continuación, en el capítulo 2, se presenta el estado del arte donde se hará referencia a varios trabajos relacionados.

En el capítulo 3 se describen y detallan las herramientas usadas para alcanzar los objetivos del TFM.

Después de haber descrito qué herramientas se han usado, en el capítulo 4 trataremos de explicar la implementación, modelo y funcionamiento del proyecto, y se explicarán las principales modificaciones realizadas, incluyendo los inconvenientes y las soluciones para llevarlas a cabo.

En el capítulo 5, y partiendo de una implementación ya funcional, se presentarán los experimentos y el análisis de los mismos.

Por último, en el capítulo 6, se presentarán las principales conclusiones en base a los resultados y al funcionamiento de la aplicación. Además, se definirán futuros trabajos en la misma línea.

Capítulo 2

Estado del Arte

En este capítulo vamos a analizar algunos trabajos académicos que guardan relación con nuestro proyecto, aunque su objetivo principal no sea la reducción de gases de efecto invernadero mediante reducción de los niveles de tráfico.

2.1 Vehicular Traffic Management based on Traffic Engineering for Vehicular Ad Hoc Networks

En primer lugar vamos a tratar el proyecto presentado por Guidoni et al. [5] en el que presentan una solución a la congestión del tráfico en grandes ciudades. Estos autores toman en consideración otros proyectos basados en comunicación V2V, pero descartan este tipo de comunicación por la cantidad excesiva de mensajes que deben producirse para una predicción de tráfico efectiva, además de los inconvenientes de que, si se pierden los mensajes de cualquier vehículo, se produce una pérdida considerable de la precisión de los datos.

Tienen también en cuenta estudios con comunicaciones V2I y, aunque creen que puede ser una buena estrategia, la descartan por el alto coste de crear una infraestructura capaz de soportar una alta demanda del servicio.

Teniendo en cuenta la literatura, crean un servicio de control de tráfico llamado Re-RouTe. Los autores analizan las ventajas y desventajas de usarlo como un servicio centralizado o descentralizado, pero no concretan cual de las dos sería más idónea, por lo que sugieren usar su sistema de ambas formas.

Re-Route se basa principalmente en cuatro módulos, como podemos ver representados en la figura 2.1:

- Location Information: Encargado de recibir la información de los vehículos.
- Network Representation: Crea un grafo con los pesos teniendo en cuenta la Location Information.

- Network Classification: Clasifica las calles según estén congestionadas o no.
- Route Suggestion: Verifica las rutas alternativas para vehículos que estén en una situación de congestión.

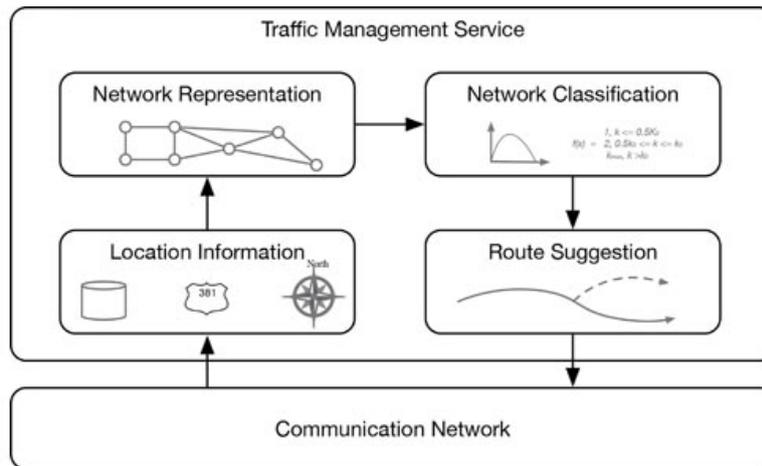


Figura 2.1: Diseño Re-Route.

A diferencia de otros estudios, este se basa únicamente en la densidad de los vehículos cuando ya se sabe que hay puntos de congestión. Es efectivo según muestran sus resultados, pero les queda trabajo por delante prediciendo cuáles van a ser las zonas más propensas al bloqueo.

2.2 DIVERT

Los autores de DIVERT (A Distributed Vehicular Traffic Re-Routing System for Congestion Avoidance) [6] consideran que, aunque los estudios que se hacen con sistemas centralizados tienen buenos resultados, sufren de dos problemas graves. El primero es la escalabilidad, ya que el servidor central es el que tiene que realizar todo el cómputo y comunicación; en segundo lugar la privacidad, ya que los vehículos comparten toda su información. Además, consideran que los sistemas completamente descentralizados no son adecuados para un sistema de redireccionamiento proactivo. Por ejemplo, al crear redes vehiculares ad-hoc (VANET), los vehículos pueden intercambiar información mediante la comunicación de varios saltos y, por lo tanto, pueden detectar signos de congestión en regiones pequeñas, al tiempo que preservan su privacidad. Sin embargo, las VANET no permiten que los vehículos obtengan una vista precisa del tráfico global de la red de carreteras,

lo que da como resultado decisiones de redireccionamiento incorrectas, o al menos subóptimas. Además, en una arquitectura totalmente distribuida, debido a la falta de un coordinador, los vehículos no pueden realizar acciones sincronizadas al mismo tiempo, lo que hace que sea imposible tomar decisiones colaborativas en tiempo real.

Por estos motivos, proponen un sistema híbrido en el que siguen manteniendo un servidor, y la diferencia fundamental es que la carga computacional la realizan en gran medida los vehículos. Además, siguiendo con la privacidad, los datos enviados por los vehículos al servidor son anónimos para poder proteger a los usuarios y, a la vez, percatarse de congestiones en tiempo real. Esto hace que el servidor desconozca los datos origen-destino de los usuarios.

Para tomar decisiones colaborativas de redireccionamiento, los vehículos intercambian mensajes a través de una VANET. Han optimizado la difusión de datos VANET para permitir un cálculo de redireccionamiento distribuido y eficiente. Además, el sistema equilibra la privacidad del usuario con la eficacia del redireccionamiento. DIVERT, según los autores, aumenta sustancialmente la privacidad del usuario, mientras que la efectividad del enrutamiento se ve mínimamente afectada.

2.3 MGRM

Los autores de MGRM (An Adaptive Mechanism for Congestion Control in Smart Vehicular Network) [7], en base a un análisis de la literatura, creen que los mecanismos de reedireccionamiento actuales no son eficientes porque lo que consiguen es que, cuando una carretera está colapsada, se desvíe el tráfico a otra carretera vecina, por lo que la congestión también aumentará en esta paulatinamente cuando se desvíen los vehículos.

Lo que proponen para solucionarlo es un mecanismo de enrutamiento adaptativo. En primer lugar detectando la congestión con parámetros multimétricos, como el tamaño de la carretera, ancho del vehículo, tiempo de viaje, etc. Luego dividen el tipo de congestión según sea alta, moderada o baja. Esta información es tenida en cuenta para saber el número de vehículos que se deben desviar, y si es necesaria más de una ruta de desvío, lo que evitaría el problema planteado.

La metodología de este protocolo se basa en cinco apartados:

- **Vehicle Initialization:** Recolectan la información de cada vehículo, como su posición o preferencias de destino.
- **Collection of Information related to Vehicles:** Recolecta información de vehículos en ruta, como velocidad media, consumo o emisiones.

- Traffic states determination: Analiza la congestión a través de parámetros como el número de vehículos o tamaño de la vía.
- Traffic congestion analysis: Determina el tipo de congestión a partir del apartado anterior.
- Congestion Control: Reconoce la congestión, y ofrece rutas alternativas para solucionarla.

Según los resultados, los autores de este trabajo consiguen que su protocolo reduzca el tiempo medio de las rutas, así como una reducción considerable del consumo, y por tanto menos emisiones vertidas al medio ambiente.

2.4 Deep learning support for intelligent transportation systems

Los autores de este estudio [8] introducen las nuevas tecnologías que han ido apareciendo en los últimos años, como son las nuevas redes de comunicaciones 4G y 5G, además de una mayor integración en los vehículos de sistemas de conectividad, lo que ha hecho posible la creación de protocolos de comunicación nuevos y más eficientes.

El estudio se basa en la incorporación de Deep Learning en Sistemas Inteligentes de Transporte (ITS), igual que ya se ha integrado en otros sectores como la agricultura, la manufactura, sistemas de seguridad, y domótica.

La integración de Deep Learning en ITS la realizan con el objetivo de que el sistema sea capaz de realizar la detección de elementos móviles, teniendo la capacidad de detectar múltiples elementos al mismo tiempo, la identificación de estos como clases de objetos (coches, motocicletas, peatones...), el recuento de estos objetos, y señalar otros objetos que un conductor debería reconocer.

Con estos datos pretenden poder predecir el flujo de tráfico, y así mejorar la distribución de este, calcular el riesgo de accidentes, automatizar el control de los semáforos, detección de líneas, y reducir el consumo de combustible.

2.5 Vehicular Crowd Management

Este estudio [9] usa de caso de estudio eventos deportivos como el Mundial de fútbol para optimizar la navegación y reducir la congestión.

Con el uso de la tecnología IoT los autores proponen un nuevo sistema de control de congestiones. IoT proporciona la integración de diferentes tecnologías y diferentes paradigmas de comunicación, incluida la identificación y rastreo, el

uso de sensores y actuadores, protocolos de comunicación y el uso de dispositivos inteligentes. Su sistema usa diferentes sensores como vehículos inteligentes, smartphones, cámaras de tráfico y actuadores como los conductores de vehículos autónomos.

El sistema se basa principalmente en cuatro secciones.

- **Data Collection and Processing Module:** Recolecta información del tráfico a través de vehículos conectados, cámaras de tráfico y información de usuario de los smartphones.
- **The Optimization Module:** Es el director que controla y coordina los módulos de salida y navegación de los vehículos.
- **Navigation Module:** Es el responsable de proporcionar las rutas a los vehículos que lo soliciten.
- **Departure Control Module:** Su objetivo es determinar el momento en que tiene que salir cada vehículo de un evento importante en el que se produzca congestión en la salida.

Con este modelo consiguieron reducir los tiempos de salida en caso de una evacuación o un gran evento.

Capítulo 3

Herramientas utilizadas

En este capítulo procederemos a explicar las herramientas utilizadas, sus principales características, y las razones de su uso.

3.1 ABATIS

ABATIS es un servidor de rutas basado en la Open Source Routing Machine (OSRM) [10], y está implementado en C++, siendo open-source. Como datos de entrada usa los archivos generados por OpenStreetMap, tanto en el formato PBF como en OSM XML.

Se ha instalado este servidor en un equipo Linux. Para acceder a la información de ruta se hace mediante peticiones HTTP con este formato. El detalle de los parámetros utilizados se describirá más adelante en el proyecto.

```
GET /route/v1/{profile}/{coordinates}?alternatives= {true|false|number}
    &steps={true|false} &geometries={polyline|polyline6|geojson}
    &overview={full|simplified|false}&annotations={true|false}
```

La respuesta a esta petición se divide en tres puntos:

- **code:** si la solicitud se realizó correctamente aparece “Ok”. De lo contrario, mostrará uno de los códigos de estado general y dependiente del servicio.
- **waypoints:** Array de objetos Waypoint que representa todos los waypoints en orden.
- **routes:** Array de objetos tipo Route, ordenados por rango de recomendación descendente.

La figura 3.1 nos muestra un ejemplo de una respuesta del servidor ABATIS en formato JSON al hacer una petición GET solicitando una ruta desde la “Calle dels Tomassos” hasta el “Carrer de Císcar”.

La petición es:

```
http://158.42.55.97:5000/route/v1/driving/-0.3727876,39.4616726;-0.3653993,39.4656817.json?steps=true&overview=full&geometries=geojson
```

The screenshot displays a JSON response in a viewer interface. At the top, there are tabs for 'JSON', 'Datos sin procesar', and 'Cabeceras'. Below the tabs are actions: 'Guardar', 'Copiar', 'Contraer todo', 'Expandir todo', and 'Filtrar JSON'. The JSON content is as follows:

```
{
  "code": "Ok",
  "waypoints": [
    {
      "hint": "GSgAgP__38AAAAUwAAAAA_gLMT_r_KSNaAgAAvxBMM0II",
      "distance": 0,
      "location": {
        "0": -0.372788,
        "1": 39.461673,
        "name": "Calle dels Tomassos"
      }
    },
    {
      "hint": "OioAgP__39FAAAAiQAAAAA_gKpbPr_0jJaAgAAzxRMM0II",
      "distance": 0,
      "location": {
        "0": -0.365399,
        "1": 39.465682,
        "name": "Carrer de Císcar"
      }
    }
  ],
  "routes": [
    {
      "duration": 107.3,
      "distance": 1146.4,
      "weight_name": "routability",
      "weight": 107.3,
      "geometry": {...},
      "legs": [...]
    }
  ]
}
```

Figura 3.1: Respuesta del servidor ABATIS

Se puede comprobar que la petición nos la ha devuelto de forma satisfactoria. Como “waypoints” nos ha devuelto el punto de salida y el punto destino. Por

último, solo nos ha ofrecido una ruta en la que nos indica la duración y distancia estimadas por el servidor.

3.1.1 ABATIS Tools

En este apartado vamos a comentar las principales herramientas de ABATIS que se han usado en este proyecto.

osrm-extract

Esta herramienta del servidor ABATIS obtiene información del área que extraigamos desde OpenStreetMap. En nuestro caso obtiene la información de la ciudad de Valencia. Con estos datos crea tres archivos necesarios para los cálculos de rutas.

Los archivos son los siguientes:

- Un archivo con sufijo `.osrm` que contiene los datos para el cálculo de rutas.
- Otro archivo con sufijo `osrm.restriction` que contiene las restricciones del sentido de las calles.
- Un último archivo que posee los nombres de todos los caminos con sufijo `osrm.names`.

Además, esta herramienta hace uso de un archivo de configuración denominado `extractor.ini`, el cual permite especificar el perfil que el servidor tomará como referencia para el cálculo de las rutas, como `car.lua`, `foot.lua`, y `bycycle.lua`.

osrm-prepare

Esta herramienta genera archivos necesarios para el cálculo, como son: `.osrm.edge`, `.osrm.fileIndex`, `.osrm.hsgr`, `osrm.nodes`, `.osrm.ramIndex`, y `.osrm.timestamp`, a partir de los archivos generados por `osrm-extract`.

osrm-customize

Esta herramienta se usa para modificar el peso de ciertas vías, lo que nos permite actualizar el mapa según nuestros intereses.

osrm-datastore

Esta herramienta se complementa con la anterior, ya que es la que nos permite, sin tener que parar y volver a ejecutar el servidor, guardar las modificaciones que le hemos indicado.

osrm-routed

Esta herramienta es la que ejecuta el servidor ABATIS. También nos permite especificar 2 tipos de procesamiento diferentes.

- Jerarquías de contracción (CH) que se adapta mejor a los casos de uso donde el rendimiento de la consulta es clave, especialmente para matrices de gran distancia.
- Multi-Level Dijkstra (MLD) que se adapta mejor a los casos de uso asociados a menores distancias, y en los que el rendimiento de las consultas aún debe ser muy bueno, y además se deben realizar actualizaciones en vivo de los datos, por ejemplo para actualizaciones periódicas de tráfico.

En nuestro caso, puesto que nos interesan las actualizaciones en vivo, se ha decidido usar la versión MLD.

3.2 SUMO

Simulation of Urban MObility (SUMO) [11] es un paquete de simulación de tráfico multimodal continuo, microscópico, altamente portátil y de código abierto, diseñado para manejar grandes redes. Fue desarrollado en 2001 por el German Aerospace Center (DLR), y disponibilizado como software Open Source en el 2002. Esto se debe principalmente al deseo de que fuera accesible a otras instituciones públicas, y a la necesidad de la comunidad de tener un simulador de tráfico capaz de ayudar a la realización de estudios de investigación.

Entre sus características destacan, a parte de las mencionadas, la comunicación con simuladores de redes tales como pueden ser OMNet++ o ns-3, y la capacidad de importar redes de carreteras en formatos como OpenStreetMap o OpenDrive.

Sumo es básicamente un simulador de tráfico microscópico. Cada vehículo es representado explícitamente, definido por un identificador, tiempo de salida, y la ruta a seguir. Si se desea se pueden añadir más detalles como la velocidad, la posición o parámetros de llegada. Además de esto, es posible asignar variables de contaminación sonora o polución. También es interesante el poder modificar la apariencia del vehículo en el entorno gráfico del simulador. Vemos un ejemplo de simulación gráfica en la figura 3.2.

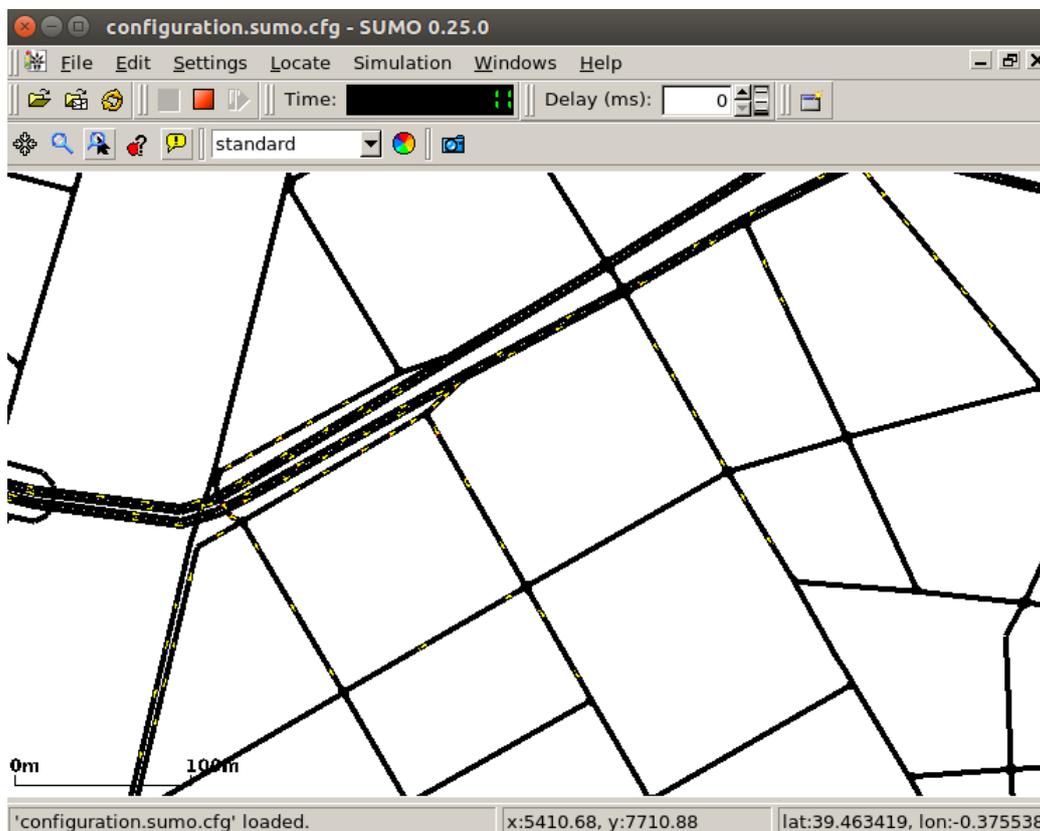


Figura 3.2: Simulación gráfica SUMO

La definición de los vehículos puede ser generada usando diferentes fuentes. Para escenarios de gran escala se suelen usar matrices origen/destino (O/D). Estas describen el movimiento del tráfico indicando el número de vehículos en una zona, y en qué tiempo.

SUMO es capaz de generar varios archivos de salida por cada simulación. Proporciona datos desde anillos de inducción hasta las posiciones de cada vehículo en cada momento, además de datos más complejos como la información de cada viaje, como pueden ser tiempo o distancia. SUMO se actualizó recientemente, y ahora proporciona también datos de emisiones y de consumo de combustible.

3.2.1 Paradigmas básicos y características

SUMO tiene el potencial de simular el tráfico de la red de carreteras de una ciudad, con la ventaja de que esta implementación puede ser multimodal, incluyendo no solo el movimiento de vehículos dentro de la ciudad, pero también el sistema de transporte público, la red ferroviaria, carriles bici y rutas de peatones. Esta ventaja

se traduce en la posibilidad de la creación de múltiples rutas, que a su vez pueden ser descompuestas en subrutas, y así describir una única modalidad de tráfico.

El flujo del tráfico se simula microscópicamente, y esto significa que cada vehículo está localizado en un punto con información propia, como tiempo de salida, velocidad, y la ruta que va a seguir. La simulación respetará la velocidad máxima de la calle y los demás atributos de esta.

Las funcionalidades principales de SUMO son:

- Simulación microscópica de vehículos, peatones y transporte público.
- Simulación multimodal del tráfico.
- Tiempo de semáforos que pueden ser generados por SUMO o importadas de otras fuentes.
- Aceptación de formatos externos como por ejemplo OSM, VISUM o Open-Drive
- Completamente portátil.

3.3 OMNeT++

OMNeT++ [12] es una biblioteca y un framework de simulación C++ extensible, modular, y basado en componentes, principalmente para la construcción de simuladores de red. El concepto de “Red” se entiende en un sentido más amplio, incluyendo redes de comunicación inalámbricas y cableadas, redes en chip, redes de cola, etc. La funcionalidad específica del dominio, así como el soporte para redes de sensores, redes inalámbricas ad-hoc, protocolos de Internet, modelado de rendimiento, redes fotónicas, etc., es proporcionada por marcos modelo, desarrollados como proyectos independientes. OMNeT++ ofrece un IDE basado en Eclipse, un entorno de tiempo de ejecución gráfico, y una serie de otras herramientas. Hay extensiones para simulación en tiempo real, emulación de red, integración de base de datos, integración de SystemC, y varias otras funciones. OMNeT++ se distribuye bajo la Academic Public License.

Aunque OMNeT++ no es un simulador de red en sí mismo, ha ganado una gran popularidad como plataforma de simulación de red en la comunidad científica, así como en entornos industriales, y ha creado una gran comunidad de usuarios.

OMNeT++ proporciona una arquitectura de componentes para módulos. Los componentes (modules) se programan en C++, y luego se ensamblan en componentes y módulos más grandes utilizando un lenguaje de alto nivel (NED).

La figura 3.3 representa gráficamente la jerarquía de módulos en los que partiendo de módulos básicos se forman módulos compuestos. Estos se comunican

entre ellos a través de puertas de enlace tanto dentro de los módulos compuestos como al exterior. La principal utilidad de unir ciertos módulos es darles características y parámetros en común lo que puede facilitar la programación.

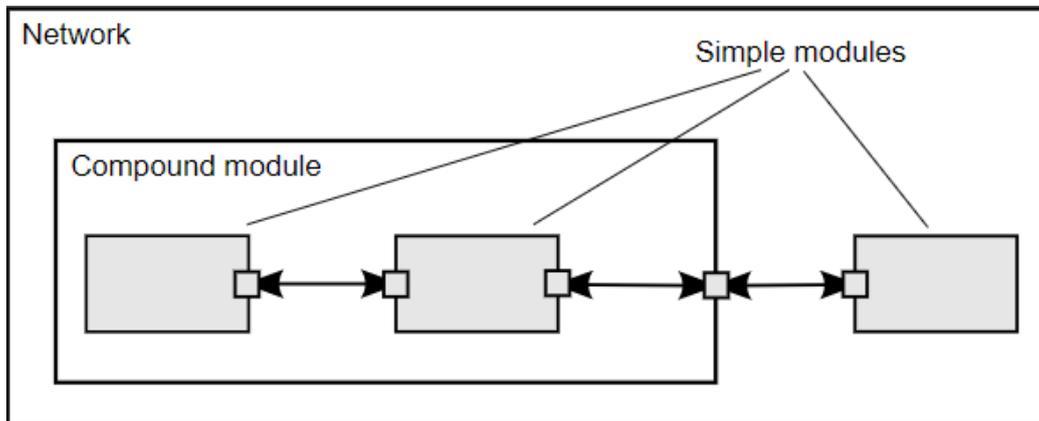


Figura 3.3: Jerarquía de módulos Omnet++

OMNeT++ tiene un amplio soporte de GUI y, debido a su arquitectura modular, el kernel de simulación (y los módulos) se pueden integrar fácilmente en sus aplicaciones.

Una de las interfaces de usuario más usados en OMNet++ es Tkenv que es el que hemos usado en este proyecto. Vemos un ejemplo de una simulación en la figura 3.4.

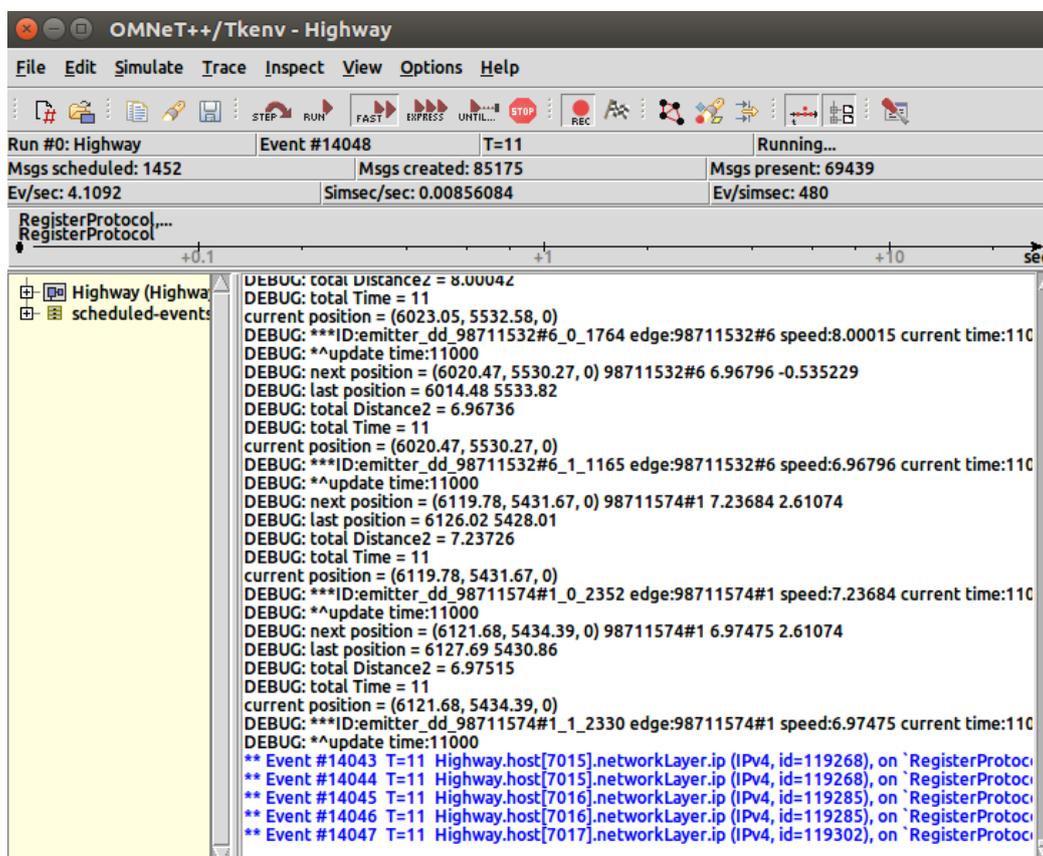


Figura 3.4: Interfaz Gráfica Tkenv

3.3.1 Componentes

Los elementos principales de OMNeT++ son:

- Biblioteca del kernel de simulación (C++).
- El lenguaje de descripción de topología de NED.
- IDE de simulación basado en la plataforma Eclipse.
- GUI de tiempo de ejecución de simulación interactiva (Qtenv).
- Interfaz de línea de comandos para la ejecución de la simulación (Cmdenv).
- Utilidades (makefile creation tool, etc.).
- Documentación, simulaciones de muestra, etc.

3.3.2 Modelos

Durante los años en los que OMNeT++ ha estado disponible, investigadores en diversas áreas han escrito innumerables modelos de simulación y frameworks de modelos para él: queuing, modelado de recursos, protocolos de Internet, redes inalámbricas, LAN conmutadas, redes peer-to-peer, transmisión de medios, redes ad-hoc, redes de malla, redes de sensores inalámbricos, redes vehiculares, NoC, redes ópticas, sistemas HPC, computación en la nube, SAN y otros más. La mayoría de estos frameworks son de código abierto, se desarrollan como proyectos independientes, y siguen sus propios ciclos de lanzamiento.

3.3.3 Plataformas

El kernel de simulación OMNeT++ es C++ estándar y se ejecuta básicamente en todas las plataformas donde hay disponible un compilador C++ moderno. El IDE de simulación requiere Windows, Linux o macOS.

3.4 INET

INET Framework puede considerarse la biblioteca de modelos de protocolo estándar de OMNeT++. El Framework INET provee protocolos, agentes y otros modelos para investigadores y estudiantes que trabajan con redes de comunicación.

INET usa la mayoría de protocolos de comunicación desde modelos para la pila de Internet como Transport Control Protocol (TCP), User Datagram Protocol (UDP), Internet Protocol en la versión 4 (IPv4) y en la versión 6 (IPv6), Open Shortest Path First (OSPF), Border Gateway Protocol (BGP), etc. También utiliza protocolos de capa de enlace cableado e inalámbrico como Ethernet, Point-to-Point Protocol (PPP), IEEE 802.11, etc. Además da soporte para protocolos de movilidad como protocolos Mobile Ad hoc Network (MANET), Differentiated Services (DiffServ), MultiProtocol Label Switching (MPLS) con Label Distribution Protocol (LDP) y Resources Reservation Protocol (RSVP-TE), varios modelos de aplicación y componentes.

Algunos simuladores toman INET como base y lo amplían en direcciones específicas, como redes vehiculares (Veins, CoRE), redes superpuestas / peer-to-peer (OverSim) o LTE (SimuLTE).

3.5 TraCI

La API Traffic Control Interface (TraCI) [13] nos permite la comunicación entre los simuladores OMNeT++ y SUMO. TraCI usa comunicación TCP basada en la arquitectura cliente/servidor. Concretamente, SUMO hace de servidor mientras que OMNeT++ hace de cliente. Podemos ver cómo actúa el protocolo TCP en la siguiente figura.

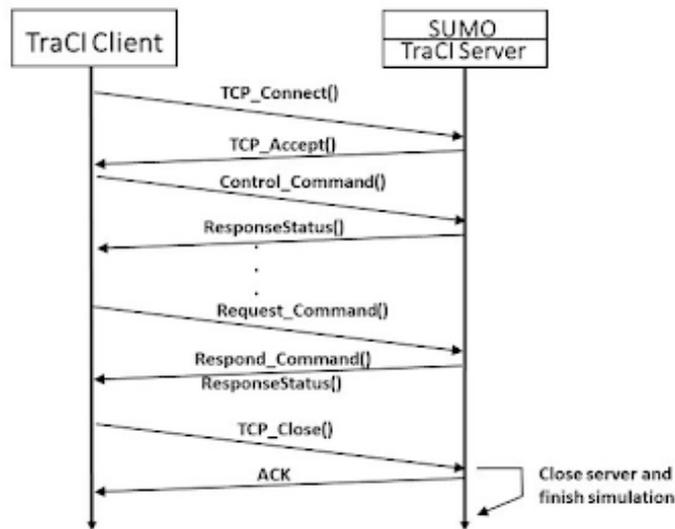


Figura 3.5: Protocolo de comunicación entre TraCI y SUMO.

Una vez la conexión TCP está establecida, el simulador de red, en este caso Omnet+, controla el simulador vehicular, en nuestro caso SUMO. Esto habilita el cambio de movimientos de cada vehículo lo que permite un posible ajuste de cada nodo gracias a la información generada en la red VANET.

Para garantizar la sincronización entre los simuladores, el simulador de red envía periódicamente un comando para el simulador de tráfico que contiene el tiempo real de simulación más un paso posterior de esta.

El simulador vehicular realiza el siguiente paso de la simulación y envía las posiciones de los vehículos de vuelta al simulador de red. Esto es posible porque SUMO va un paso por delante que Omnet++.

3.5.1 sumo-launchd

TraCI trae consigo un daemon llamado sumo-launchd para la comunicación entre SUMO y OMNeT++. Está diseñado para ejecutarse en background escuchando

las peticiones entrantes. Para cada petición de conexión entrante recibe un *setup* de la simulación en formato XML, y en ese momento lanza una instancia diferente de SUMO y un proxy entre OMNeT++ y SUMO.

Todo el control de los puertos de la red está controlado por el *daemon*. Las instancias son creadas y destruidas según la necesidad, y esto consigue simplificar la ejecución de las simulaciones.

Las respuestas a la ejecución del *daemon* pueden ser:

- SUMO se ejecuta correctamente:
 - exit code: 0
 - status: Done.
- SUMO no ha podido empezar:
 - exit code: -1
 - status: Could not start SUMO (/tmp/sumo -c erlangen.sumo.cfg): [Error 2] The system cannot find the file specified.
- SUMO fue iniciado, pero fue interrumpido: (al tratar de ejecutar comandos no disponibles para la versión)
 - exit code: 1
 - status: Exited with error code 1
 - stderr: Error: Requested command not implemented (0xcc): Command not implemented in sumo
- SUMO fue iniciado pero fue interrumpido: (por un error en el archivo de red)
 - exit code: 1
 - status: Exited with error code 1
 - stderr: Error: expected end of tag “net” In file ‘my.net.xmlt’ At line/-column 10948/3. Quitting (on error).

Capítulo 4

Implementación de la solución propuesta

En este capítulo describiremos la relación entre las diferentes herramientas. Para ello se ha seguido la siguiente estructura mostrada en la figura 4.1, que detalla la relación entre los diferentes elementos software involucrados en los estudios de simulación del tráfico.

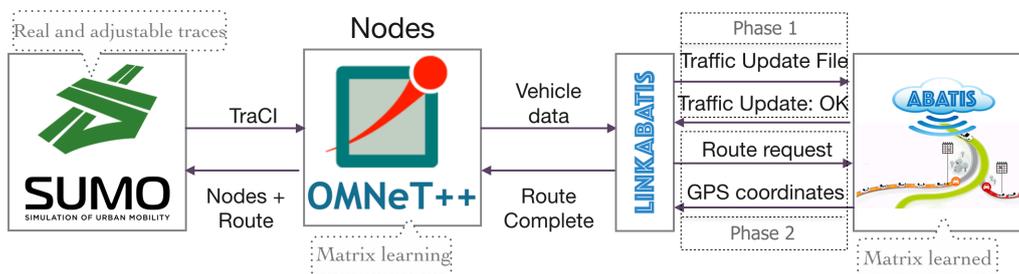


Figura 4.1: Arquitectura de simulación.

Como se ha descrito anteriormente, el encargado de la comunicación entre SUMO y OMNeT++ va a ser TraCI, mientras que la comunicación con el servidor de rutas ABATIS va a realizarse a través de nuestra aplicación linkABATIS que, con los datos proporcionados por OMNeT++, los traduce para que ABATIS sea capaz de tratarlos, y así proporcionarnos información sobre las rutas solicitadas. LinkABATIS, a su vez, también traduce los datos devueltos en formato JSON por ABATIS a los archivos add y rou para que OMNeT++, junto a SUMO, pueda interpretarlos, y así obtener datos relevantes como tiempo de trayecto o distancia recorrida, entre otros.

4.1 TraCI

Como se ha comentado en capítulo anterior TraCI es el encargado de la comunicación entre los dos simuladores. En este apartado se explicará de una forma más detallada y aplicada al proyecto dicha comunicación.

En primer lugar deberemos ejecutar sumo-launchd con un comando similar a este:

```
~ /sumo-launchd.py -vv -c sumo
```

También necesitará un archivo XML de configuración con el siguiente formato:

```
<?xml version="1.0"?>
<launch>
  <copy file="valencia.net.xml" />
  <copy file="valencia.rou.xml" />
  <copy file="valencia.sumo.cfg" type="config" />
</launch>
```

Como podemos observar, es necesario el archivo de la red (net), el archivo de rutas (rou) y el archivo de configuración de sumo (sumo.cfg). El archivo net nos da toda la información del escenario descargado, esta se divide en:

- Cada calle (edge) como una colección de carriles, incluida la posición, la forma y el límite de velocidad de cada carril.
- Lógica de los semáforos referenciados a los cruces(junctions).
- Junctions, incluidas sus definiciones de derecho de paso.
- Conexiones entre los edges y las junctions.

En el archivo rou está toda la información de las rutas posibles que los vehículos pueden seguir. Como podemos ver en la figura 4.2 cada ruta tiene una id y el id de los edges que la conforman.

```
<routes>
<route id="4693573#1_to_98711574#1" edges="4693573#1 4693573#2 4693573#3 98711556 98711574#0 98711574#1"/>
<route id="4693573#1_to_98711543#2" edges="4693573#1 4693573#2 6277183#0 98711543#1 98711543#2"/>
<route id="4693573#1_to_6277192#3" edges="4693573#1 4693573#2 4693573#3 98711556 6277192#0 6277192#1 6277192#2 6277192#3"/>
<route id="4693573#1_to_41998660#0" edges="4693573#1 4693573#2 4693573#3 98711556 6277192#0 6277192#1 6277192#2 41998660#0"/>
<route id="4693573#1_to_52092240" edges="4693573#1 4693573#2 6277183#0 6277183#1 11993589#0 11993589#1 11993589#2 52092240"/>
...
</routes>
```

Figura 4.2: Fragmento archivo .rou

El archivo de configuración de sumo que hemos usado en nuestro proyecto tiene el siguiente formato.

```

<configuration>
  <input>
    <net-file value="valencia.net.xml"/>
  <additional-files value="valencia.rou.valenciaATA.xml
valencia.add.valenciaATA.xml"/>
  </input>
  <output>
    <tripinfo-output value="valencia.sumo.tripinfo.xml"/>
    <vehroute-output value="valencia.sumo.vehroute.xml"/>
  </output>
  <report>
    <verbose value="true"/>
    <log value="valencia.sumo.log.txt"/>
    <error-log value="valencia.sumo.error.txt"/>
  </report>
  <time>
    <begin value="0"/>
    <end value="900"/>
  </time>
</configuration>

```

Como se puede observar se le indica los archivos necesarios para la simulación como el archivo net y rou descritos además del archivo add. También de forma opcional se le indica los archivos donde queremos que nos extraiga la información de la simulación y el registro de incidencias. Por último hay que indicar el tiempo inicial y final de la simulación.

Como se ha indicado para nuestra simulación también es necesario el archivo add, este contiene la información de los vehículos junto a sus rutas. Vemos un ejemplo de esto en la figura 4.3.

```

<additional>
  <routeDistribution id="dd_4693573#1_0">
    <route refId="4693573#1_to_98711574#1" probability="0.07"/>
    <route refId="4693573#1_to_98711543#2" probability="0.07"/>
    ...
  </routeDistribution>
  <vehicle id="emitter_dd_4693573#1_0_0" depart="0.00" departLane="0" departPos="25.00" departSpeed="5.56" route="4693573#1_to_9281469#1"/>
  <vehicle id="emitter_dd_4693573#1_0_588" depart="0.59" departLane="0" departPos="25.00" departSpeed="5.56" route="4693573#1_to_52092239"/>
  <vehicle id="emitter_dd_4693573#1_0_1176" depart="1.18" departLane="0" departPos="25.00" departSpeed="5.56" route="4693573#1_to_98711574#1"/>
  ...
</additional>

```

Figura 4.3: Fragmento archivo .add

Datos adicionales

Los datos proporcionados por TraCI no son suficientes para nuestra simulación, por eso se ha editado el módulo TraCIScenarioManagerLaunchd, que es el encar-

gado de crear y mover los vehículos controlados por TraCI.

El primer dato obtenido con esta nueva implementación es la ID de los vehículos que han llegado a su destino; también obtenemos los datos del tiempo de salida y llegada junto al tiempo total. Por último obtenemos datos sobre la distancia recorrida por cada vehículo, y su correspondiente velocidad media del trayecto. Estos datos son vertidos a un archivo que se le envía a OMNET++ como parámetro de configuración. Esto queda ilustrado en la figura 4.4.

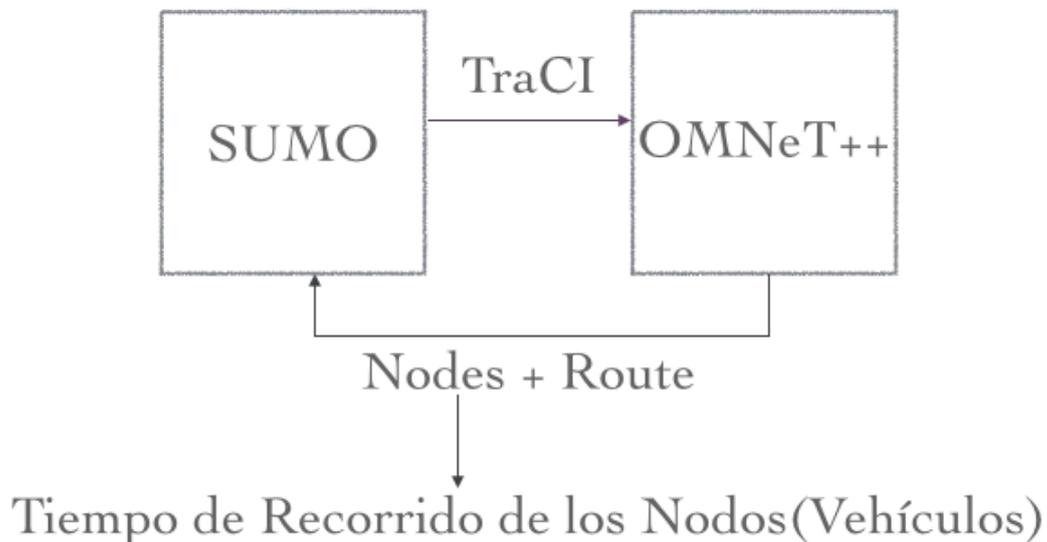


Figura 4.4: Comunicación Omnet++ - SUMO

4.2 LinkABATIS

La aplicación linkABATIS es la parte fundamental del proyecto. Está desarrollada plenamente en Python. Entre otras funciones, se encarga de la comunicación entre el servidor de rutas ABATIS y el simulador OMNeT++.

Para comunicarse realiza una petición HTTP del tipo GET a ABATIS teniendo en cuenta la dirección IP del servidor y el puerto donde está escuchando. Junto a esto se le pasa las coordenadas geográficas de inicio y fin del trayecto deseado. Un ejemplo de solicitud sería:

```
http://158.42.55.97:5000/route/v1/driving/-0.372507,39.468762;-  
0.350881,39.473339.json?steps=true&overview=full&geometries=geojson
```

Como es sabido, el simulador OMNeT++ no es capaz de interpretar coordenadas geográficas. LinkABATIS, antes de hacer la solicitud GET, hace una traducción de edges a coordenadas y, cuando tenga la respuesta del servidor, hará el

proceso inverso para así ser capaz de mantener una comunicación bidireccional entre los simuladores y el servidor de rutas. Este proceso se ilustra en la figura 4.5.

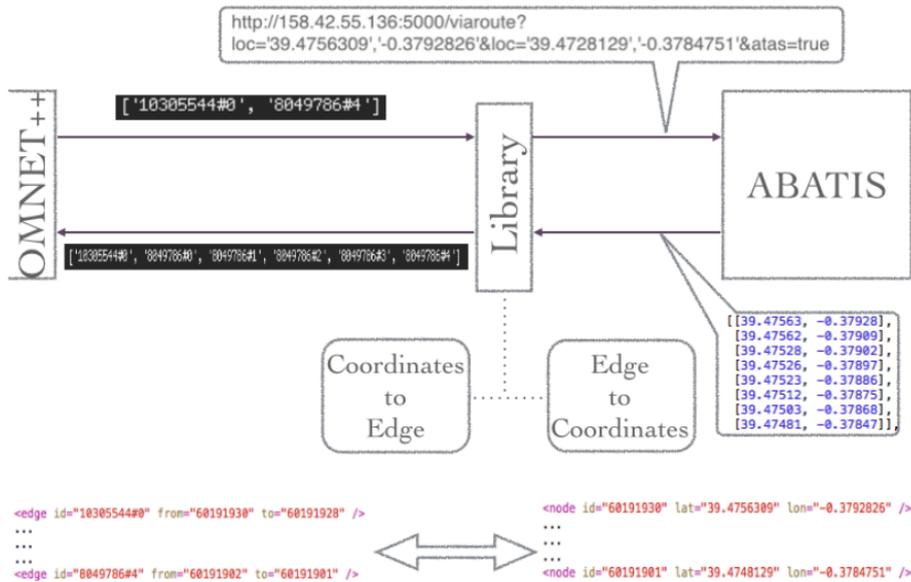


Figura 4.5: Comunicación OMNeT++ - linkABATIS - ABATIS

Además de la comunicación en sí, en nuestro experimento se ha implementado un algoritmo para restringir el tráfico de ciertas zonas de la ciudad y comprobar el efecto que esto tendría en la circulación general de Valencia. Esto no sería posible sin las herramientas descritas en el capítulo anterior de ABATIS.

Cuando el servidor de ABATIS ya está en funcionamiento, antes de solicitar las rutas de los vehículos, creamos un archivo CSV. Los archivos Comma-Separated Values (CSV) se basan en almacenar información por líneas separadas por comas, creando una especie de tabla con filas y columnas. En nuestro proyecto la primera columna corresponde al nodo de origen, la segunda al nodo destino y por último la tercera al peso de los nodos. Un ejemplo de ello lo podemos ver en la figura 4.6.

```
updateTraffic.csv x
1 112821862,113352662,3
2 96029971,1307522742,3
3 1307522742,112820963,3
4 112821659,713072792,3
5 713072792,112821750,3
6 112821750,112821862,3
7
```

Figura 4.6: Ejemplo de archivo CSV usado en el proyecto.

Cuando ya tenemos el archivo CSV generado, nuestra aplicación linkABATIS procederá a editar el servidor ABATIS en un primer momento con la herramienta osrm-customize, y la herramienta osrm-datastore para confirmar la modificación.

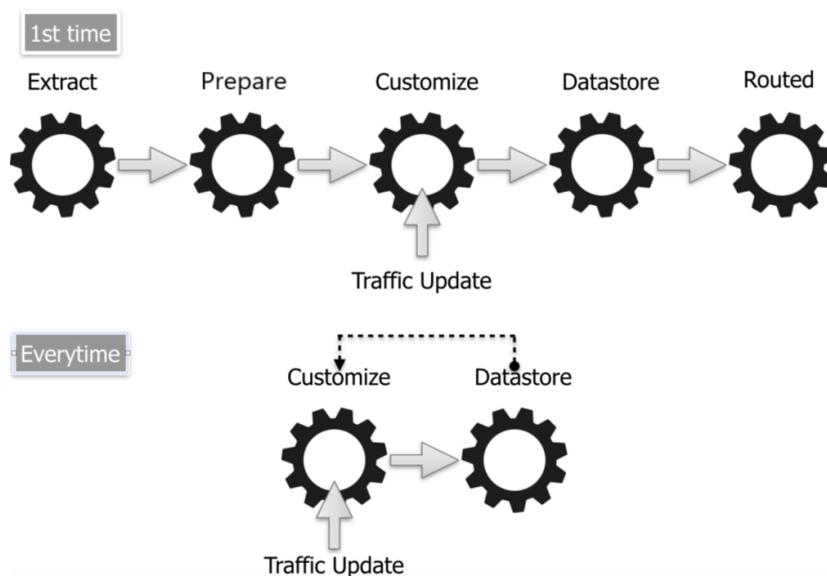


Figura 4.7: Funcionamiento herramientas de ABATIS

Como podemos observar en la figura 4.7, solo si el servidor no está iniciado y no tenemos los archivos necesarios de configuración, deberemos realizar todo el proceso. Todo esto está contemplado por linkABATIS, el cual comprueba en qué estado está el servidor, y realiza las funciones necesarias.

Ahora procederemos a explicar el algoritmo usado para la modificación de los

pesos descrito anteriormente.

Algorithm 1: Eco algorithm.

```
Input: OSM file, road Name
Output: Traffic update file, Eco Route
1 firstTimeInABATIS = true;
2 while ABATIS server route is actived do
3   if firstTimeInABATIS is true then
4     Extract module;
5     Prepare module;
6     Customize module;
7     Datastore module;
8     Routed module;
9     firstTimeInABATIS = false;
10  else
11    for way in OSM file do
12      for attributes in way do
13        if idWAY == road Name then
14          speedUpdated = Function ECO;
15          updateSpeedFromEdge(speedUpdated);
16        end
17      end
18    end
19    Traffic Update File;
20    ABATIS server route;
21    Customize module;
22    Datastore module;
23  end
24  for edges in vectorEdge do
25    Update BBDD with new routes
26  end
27  Writte additional and routes files
28 end
```

En la primera parte del algoritmo, desde la línea 1 hasta la 9, lo que se realiza es una comprobación de que el servidor ABATIS está en funcionamiento. En caso de no estarlo, ejecutamos las herramientas descritas anteriormente en el orden visto en la figura 4.7. Si está funcionando correctamente, el algoritmo comprueba que la calle que nosotros le pidamos existe realmente en el mapa de la simulación. Si existe actualizará el peso de la vía, y solicitaremos la nueva ruta a ABATIS

ya con el peso actualizado, y esta información la integraremos en los archivos necesarios para la simulación “rou” y “add”, además de actualizar nuestra BBDD para tener registro de ello.

4.3 Regulación dinámica del flujo de tráfico

En este apartado vamos a proceder a explicar los parámetros usados para la regulación de los niveles de tráfico reduciendo artificialmente la velocidad permitida.

En primer lugar se toma en cuenta el nivel de contaminación que tenga la zona en la que vamos a actuar. Este nivel de contaminación [14] se basa en la medición de micropartículas (PM2.5 y PM10), ozono (O3), dióxido de nitrógeno (NO2), emisiones de dióxido de azufre (SO2) y monóxido de carbono (CO).

El parámetro de contaminación viene regulado según el nivel que vemos en la figura 4.8. Si la calidad del aire es "Muy buena/Buena", el índice de contaminación no afectará a la ecuación. Si es "Moderada/Mala", habrá una penalización de la velocidad, mientras que si es "Muy Mala" apostamos por el corte de la vía, si es posible, ya que supone un gran riesgo para la salud.

	Índice AEMET medias ($\mu\text{g}/\text{m}^3$)	O ₃ móvil 8h	NO ₂ 1h	SO ₂ 1h	PM10 móvil 24h	PM2.5 móvil 24h
Muy Buena	1	0-80	0-40	0-100	0-20	0-10
Buena	2	80-120	40-100	100-200	20-35	10-20
Moderada	3	120-180	100-200	200-350	35-50	20-25
Mala	4	180-240	200-400	350-500	50-100	25-50
Muy Mala	5	>240	>400	>500	>100	>50

Figura 4.8: Índice AEMET de contaminación del aire.

Además de la contaminación, hay que tener en cuenta otros criterios. Debemos ser conscientes si una de las vías que deseamos regular es una arteria de la ciudad, ya que el corte de una vía principal puede causar el colapso de toda la ciudad, lo que provocaría justo el efecto contrario al buscado, que es la reducción de la polución.

Otro parámetro importante es el tipo de vía, ya que no afecta del mismo modo una reducción a una calle que a una avenida, por ejemplo. Se tienen que analizar sus detalles como la velocidad máxima previa a la modificación, la longitud, o el número de carriles.

Por último, se han tenido en cuenta las zonas sensibles. Estas zonas son las que están próximas a grupos que pueden ser de riesgo: espacios cercanos a hospitales, residencias o colegios.

Se han realizado diferentes experimentos para comprobar como varía la fracción de velocidad según el tipo de vía. Los resultados nos muestran tres tipos diferentes de vías.

Los datos de las gráficas que mostramos a continuación son consecuencia de la utilización de la fórmula 4.1 para cambiar los pesos de las vías; donde α es la fracción por la que dividimos la velocidad V .

$$f(x) = \frac{V}{\alpha} \quad (4.1)$$



Figura 4.9: Gran Vía del Marqués del Turia

La primera es Marqués del Túria; la diferencia con el resto, como se puede ver en la figura 4.9, es que tiene tres carriles, es una avenida por donde pasa habitualmente una cantidad más grande de vehículos, y al ser una vía principal cabe la posibilidad de que una reducción de su velocidad pueda afectar en menor medida los resultados, ya que posiblemente los vehículos sigan usando esta ruta.

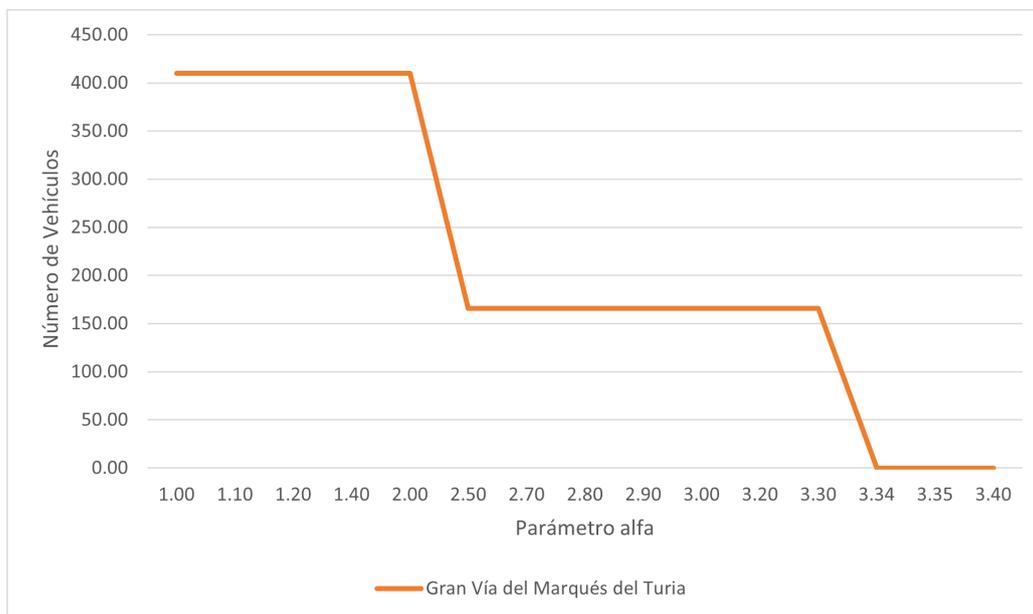


Figura 4.10: Gráfica reducción tráfico Gran Vía Marqués del Turia

Los resultados de la gráfica 4.10 nos muestran que, para que disminuya la cantidad de vehículos que circulan por la avenida, hay que reducir la velocidad a la mitad, y solo cuando se reduce en más de un tercio la velocidad a los vehículos les conviene coger rutas alternativas.

Como segundo tipo tenemos las calles Dénia y Almirante Cadarso. Podemos observar en las figuras 4.11 y 4.12 que son vías residenciales de un solo sentido.



Figura 4.11: Calle Dénia



Figura 4.12: Calle Almirante Cadarso

Como se observa en la gráfica 4.13, al poco de reducir la velocidad de estas calles, los vehículos buscan una ruta alternativa con una reducción de más de la mitad de los vehículos, y con una bajada de velocidad adicional pasan a cero.

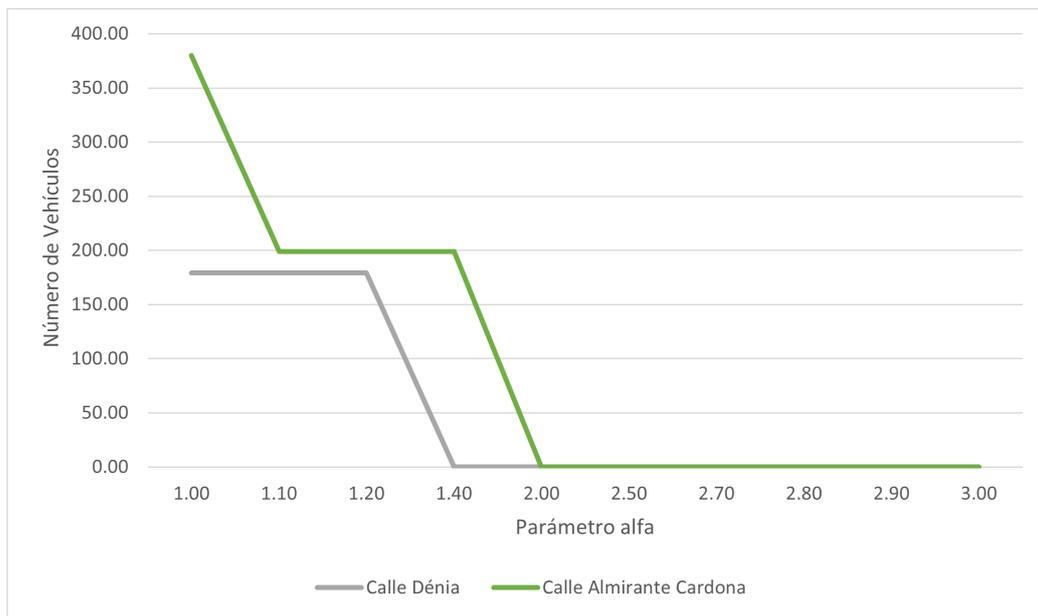


Figura 4.13: Gráfica de reducción del tráfico: Calle Denia - Calle Almirante Cardarso.

Por último tenemos la calle Cádiz y la calle Sueca que, como vemos en las figuras 4.14 y 4.15, también son vías residenciales con un único carril, pero tienen

la particularidad que son calles de conexión directa entre dos de las grandes avenidas de Valencia, haciendo una especie de anillo entre ellas. Son calles usadas con frecuencia porque no hay rutas alternativas relativamente próximas a ellas.

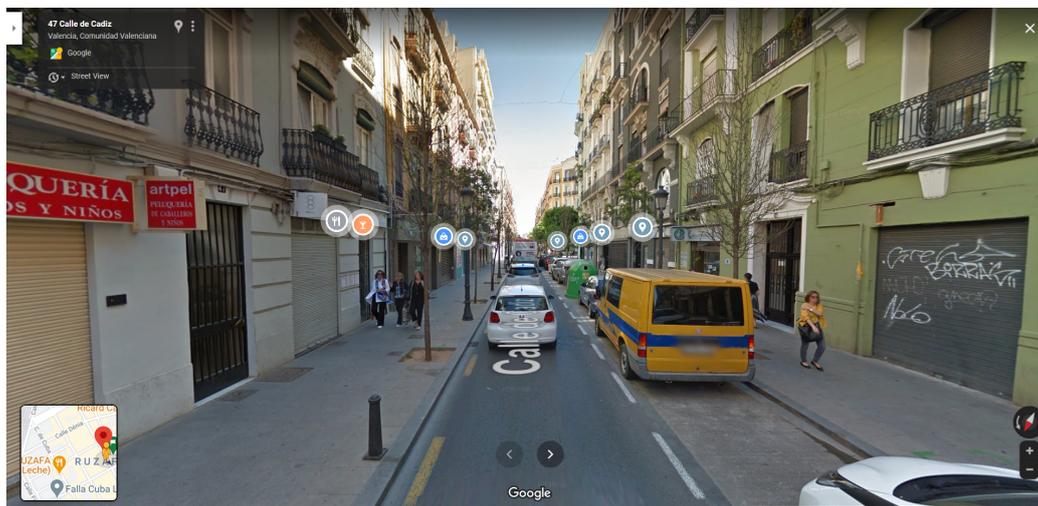


Figura 4.14: Calle Cádiz

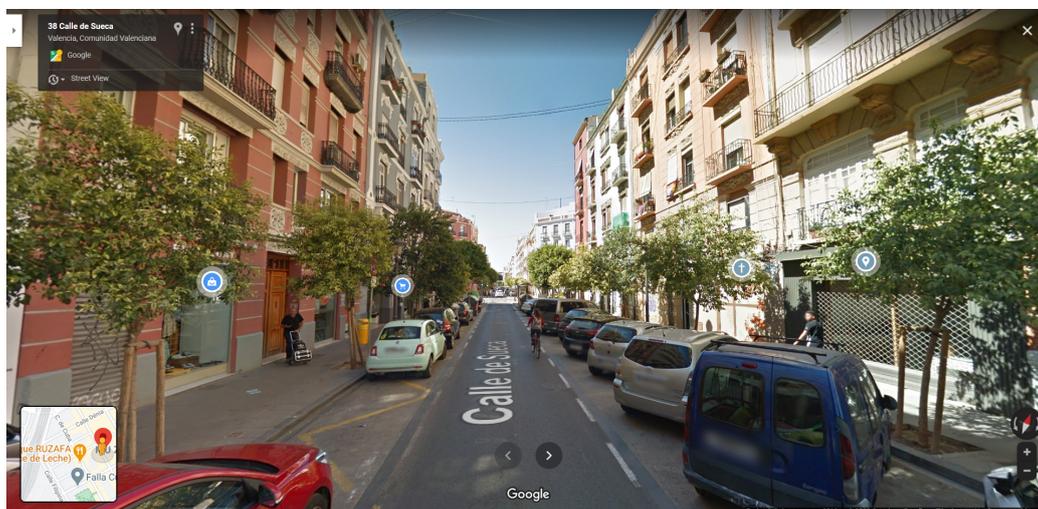


Figura 4.15: Calle Sueca

Los resultados que vemos en la gráfica 4.16 nos muestran que, para ver una reducción del tráfico en estas vías, es necesaria una reducción de velocidad drástica, o lo que es lo mismo, el corte de las calles, porque como vemos, aún reduciendo más de tres veces la velocidad de la vía, siguen circulando vehículos por la misma.

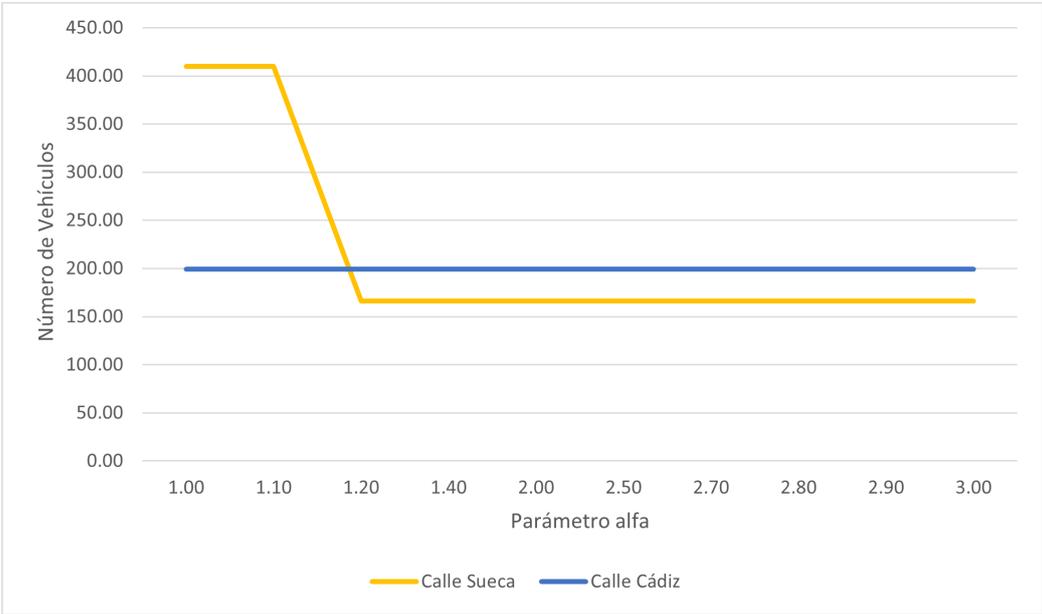


Figura 4.16: Gráfica de reducción del tráfico: Calle Sueca - Calle Cádiz.

Capítulo 5

Resultados

Para estudiar la efectividad de la solución propuesta, se han realizado diferentes pruebas para ver las consecuencias que tiene el corte de algunas de las vías más céntricas de Valencia, y qué supondría dicho corte para el tráfico de la ciudad.

Antes de proceder a explicar los resultados en sí, describiremos los parámetros utilizados, los cuales se presentan en la tabla 5.1, así como las variables que han afectado a las simulaciones. Los parámetros utilizados por OMNeT++ son definidos dentro del archivo de configuración `omnetpp.ini`.

Tabla 5.1: Parámetros Simulación

Parámetro	Valor	Descripción
host	localhost	Máquina donde se ejecuta
port	9999	Puerto de comunicación
module Type	valenciaNew.Car	Módulo Base de la simulación
module Name	host	Nombre del módulo
update Interval	1.0 s	Tiempo intervalo TraCI
Launch Config	launch.sumo.xml	Archivo para lanzar SUMO
Config File SUMO	configuration.sumo.cfg	Archivo de configuración SUMO
pathAdditionalFile	valencia.add.xml	Path file de los vehículos
pathDataBase	coor_node_edges.db	Base de Datos de Valencia
pathOSM	valencia.osm	Archivo OSM de Valencia
netFile	valencia.net.xml	Archivo net
pathRouteFile	valencia.rou.xml	Rutas de Valencia
additionalStatistics	true	Activación de estadísticas
pathStadisticalFile	valenciaNew.sta.csv	Ruta del archivo de estadísticas

Vamos a tomar como ejemplo los resultados de la Calle del Almirante Cadarso

para ver como afecta al tráfico en general el corte de una calle concreta. Esta calle tal, y como se observa en la figura 5.1, se encuentra entre la Gran Vía del Marqués del Turia y la Avinguda del Regne de València, y da acceso a un centro educativo.

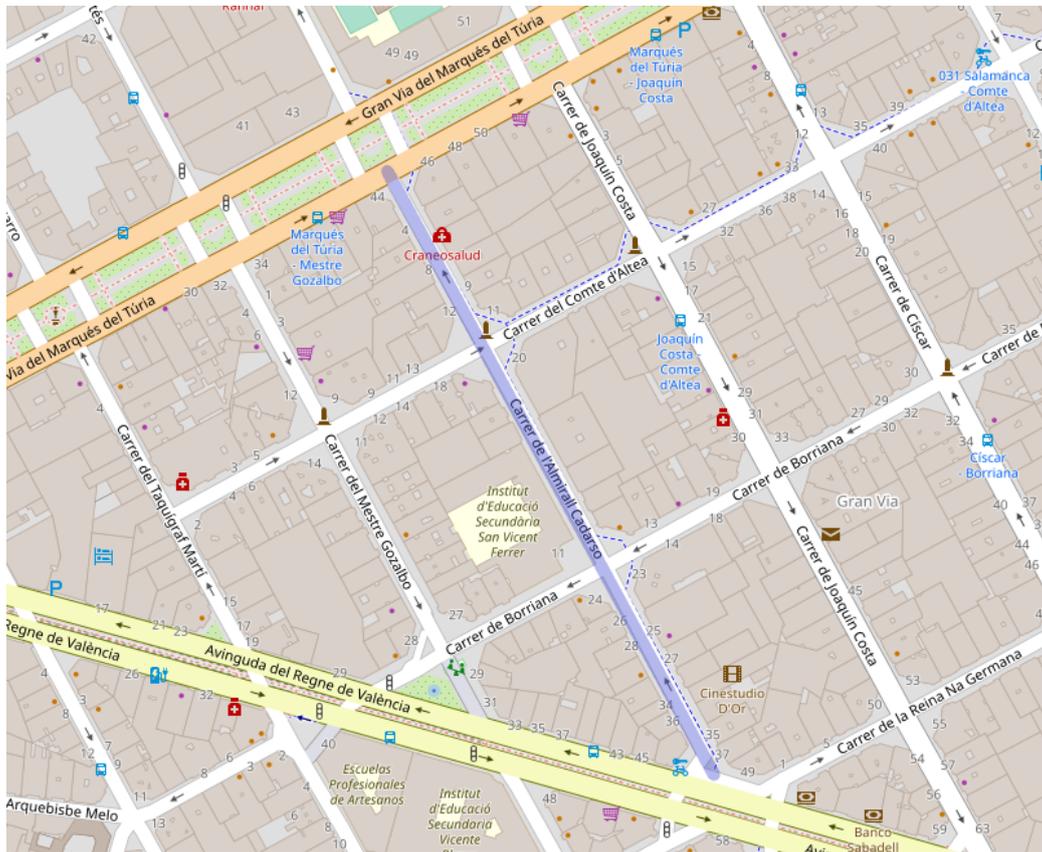


Figura 5.1: Calle del Almirante Cadarso.

Para nuestro estudio se procede a simular un corte de esta calle, lo cual se consigue mediante un ajuste del parámetro α , el cual tomará un valor suficientemente elevado como para impedir que cualquier tráfico circula por dicha calle. Después se simula el tráfico que circula por la ciudad en una situación normal, obligando a que todo el tráfico que inicialmente pasaba por esa calle se desvía a otras calles cercanas.

Tabla 5.2: Datos del tráfico en Valencia al cortar la Calle Almirante Cadarso.

Métricas	Situación normal	Calle cortada
Average Speed	8.5 Km/h	8.7 Km/h
Average Distance	3566 m	3580 m
Average Time	559.37 s	531.56 s

En términos de resultados experimentales, como se puede comprobar en la tabla 5.2, la velocidad media de los vehículos, distancia y el tiempo de trayecto no tienen variaciones significativas en el cómputo total. De hecho, comprobamos que sí aumenta ligeramente la distancia media recorrida por los vehículos, pero este aumento no es substancial. Por otro lado, comprobamos que existen pequeñas diferencias de velocidad en los vehículos afectados por el corte, lo cual también explica por qué el tiempo de trayecto bajó ligeramente, pero en el cómputo general podemos concluir que estas ligeras variaciones no suponen consecuencias significativas en la fluidez del tráfico de la ciudad al existir varias vías alternativas para la circulación.

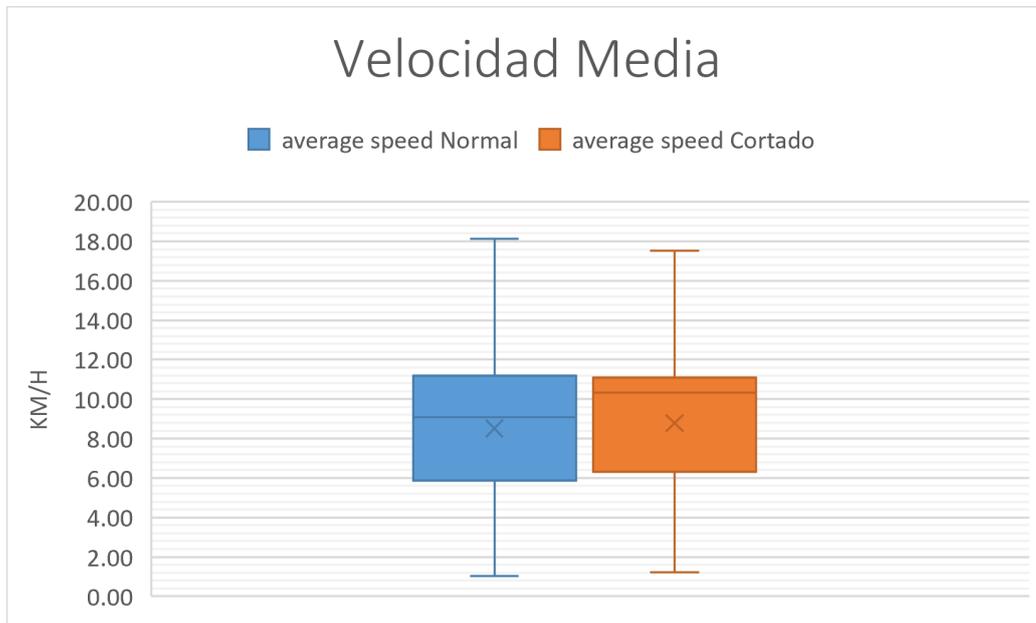


Figura 5.2: Gráfica velocidad al cortar calle del Almirante Cadarso

El Ayuntamiento de Valencia, cuando se realizó este estudio, proporciona datos de tráfico [15] hasta 2019 en el cual el ayuntamiento registra un tráfico medio diario próximo a 57.000 vehículos en la zona en la que hemos realizado el corte.

En nuestro experimento, basándose en estos datos, se han usado 1.050 vehículos en una franja de 25 minutos.

En la figura 5.2 se muestra una gráfica con la distribución de la velocidad media de los vehículos en Valencia durante ese período de 25 minutos usado para nuestro estudio, donde se aprecia un cierto nivel de congestión, aunque bajo. Si para ese mismo nivel de tráfico cortamos la calle Almirante Cadarso, nuestro objetivo es averiguar cómo afecta a la velocidad de los vehículos que tienen que desplazarse por otras rutas.

Como vemos, en una situación normal, la velocidad media máxima es superior, pero ambas rozan los 18 Km/h. Puede parecer que esta cifra es algo reducida pero, como se ha dicho antes, no se alcanza una velocidad superior porque el número de vehículos en el experimento es algo alto, ya que se ha tratado de usar datos reales de densidad de tráfico usando referencias de horas punta. Lo mismo sucede con la velocidad media mínima en la que no es apreciable una diferencia considerable. También se puede comprobar que la velocidad media es ligeramente superior cuando el tráfico debe desviarse por otras rutas, como pasa con la velocidad máxima y mínima, aunque la diferencia entre ambas es despreciable.

Vistos los resultados de una forma global, vamos a analizar los resultados de una forma atómica en el mismo escenario.

En una situación habitual, mostrada en la ilustración 5.3, un coche con origen en la Avinguda Peris i Valero y destino la calle Hernan Cortés, circularía por la avenida, giraría a la derecha en Carrer del Duc de Calàbria, atravesaría la Avinguda del Regne de Valencia, recorrería la calle del Almirante Cadarso, atravesaría la Gran Via del Marqués del Turia, pasando a la calle de Isabel la Católica, y por último girando en calle Colon para llegar a su destino, tal y como se ilustra en la figura 5.3.

Capítulo 6

Conclusiones

6.1 Conclusiones

En este proyecto se ha descrito la problemática que afrontan las grandes ciudades con índices cada vez más altos de contaminación atmosférica, y como algunas de ellas han tratado de remediarlo. También se han visto algunas de las soluciones propuestas en la literatura.

Tras la problemática se han explicado las diferentes herramientas usadas durante el trabajo, los componentes usados en cada una de ellas, y si ha sido necesario una modificación de las mismas. Cuando se ha dejado claro el uso de OMNeT++ como simulador de tráfico de datos, SUMO como simulador de redes vehiculares, y ABATIS como servidor de rutas, se ha procedido a explicar la interacción entre los tres elementos, y las aplicaciones que hemos usado para ello.

La herramienta de más importancia en el proyecto ha sido linkABATIS, usada para la conexión entre los dos simuladores y el servidor de rutas. Se ha descrito como interaccionan, y se ha detallado el protocolo implementado para la reducción de tráfico.

Nos hemos centrado en un caso donde se estudia la efectividad de reducir la velocidad de cinco calles de Valencia ubicadas en la zona centro. Se han determinado los factores que afectan a esta velocidad, y la ecuación empleada para ello. Igualmente se ha visto cual es la reducción del número de vehículos según el factor de reducción. Los resultados muestran diferentes tipos de vías que, en conclusión, se pueden dividir entre avenidas con mayor longitud y más de un carril, las vías que podemos llamar comunes, y por último las vías que, aún siendo de un solo carril, por su localización y dirección pasan una gran cantidad de vehículos en comparación a otras del mismo estilo.

Finalmente se han mostrado los resultados del tráfico general de la ciudad al cortar una de las calles céntricas por una zona bastante concurrida. Se puede con-

cluir que el corte de una sola calle no afecta al tráfico urbano, ya que es posible redirigir el tráfico por vías alternativas. Se demuestra que el sistema es lo suficiente eficiente para lograr esta redistribución de tráfico sin una repercusión significativa en tiempo, distancia o velocidad media, los cuales se mantienen mayormente inalterados.

6.2 Trabajos futuros

A partir de este proyecto se puede seguir analizando el impacto de realizar cortes de tráfico más generalizados y en diferentes situaciones, tanto de contaminación como otro tipo de restricciones (festividades, grandes eventos, etc.).

Además, la elección de las calles a cortar no se ha basado en ningún criterio específico. En un futuro trabajo se debería recolectar, tratar y usar datos reales para proporcionar una imagen más fidedigna en los resultados, y así poder aplicarlos en un escenario real.

Bibliografía

- [1] J. L. Z. Martínez, “Efficient traffic management in urban environments,” Ph.D. dissertation, Universitat Politècnica de València, 2019.
- [2] W. H. Organization. (2018) Air pollution. Last visited: 2020-11-10. [Online]. Available: <https://www.who.int/airpollution/ambient/en/>
- [3] UE. (2020) Reglamento de acceso urbana en europa. Last visited: 2020-11-03. [Online]. Available: <https://es.urbanaccessregulations.eu/countries-mainmenu-147/united-kingdom-mainmenu-205/london>
- [4] “El anillo ciclista de valència, la infraestructura con mejor reputación de la ciudad,” <https://valenciaextra.com/es/anillo-ciclista-infraestructura-mejor-reputacion-valencia/>, 2020, accessed: 2020-11-10.
- [5] D. L. Guidoni, G. Maia, F. S. H. Souza, L. A. Villas, and A. A. F. Loureiro, “Vehicular traffic management based on traffic engineering for vehicular ad hoc networks,” *IEEE Access*, vol. 8, pp. 45 167–45 183, 2020.
- [6] J. Pan, I. S. Popa, and C. Borcea, “Divert: A distributed vehicular traffic re-routing system for congestion avoidance,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 1, pp. 58–72, 2017.
- [7] G. Singh, R. Batth, and S. Egerton, “Mrgm: An adaptive mechanism for congestion control in smart vehicular network,” *International Journal of Communication Networks and Information Security*, vol. 12, pp. 273–280, 08 2020.
- [8] J. Guerrero-Ibañez, J. Contreras-Castillo, and S. Zeadally, “Deep learning support for intelligent transportation systems,” *Transactions on Emerging Telecommunications Technologies*, p. e4169, 2020. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4169>

- [9] A. Elbery, H. S. Hassanein, and N. Zorba, "Vehicular crowd management: An iot-based departure control and navigation system," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.
- [10] D. Luxen and C. Vetter, "Real-time routing with openstreetmap data," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '11. New York, NY, USA: ACM, 2011, pp. 513–516. [Online]. Available: <http://doi.acm.org/10.1145/2093973.2094062>
- [11] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. [Online]. Available: <https://elib.dlr.de/124092/>
- [12] (2020) Omnet++. Last visited: 2020-11-20. [Online]. Available: <https://www.omnetpp.org>
- [13] A. Wegener, M. Piorkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "Traci: An interface for coupling road traffic and network simulators," *Proceedings of the 11th Communications and Networking Simulation Symposium, CNS'08*, 04 2008.
- [14] "Índice diario de calidad del aire previsto por aemet - agencia estatal de meteorología - aemet. gobierno de españa," http://www.aemet.es/es/eltiempo/prediccion/composicion_quimica_atmosfera/indice_previsto, 2020, accessed: 2020-10-15.
- [15] A. de València. (2019) Oficina de estadística. Last visited: 2020-10-03. [Online]. Available: <http://www.valencia.es/ayuntamiento/catalogo.nsf/IndiceAnuario?readForm&lang=1&capitulo=3&tema=9&bdOrigen=ayuntamiento/estadistica.nsf&idApoyo=58FB3C7A3D56E414C1257DD40057EB6C>