



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



*Trabajo de Máster en Ingeniería de Computadores*

# **Sistema centralizado para la optimización del tráfico mediante un servidor de rutas**

*Realizado por:*

**Jorge Luis Zambrano Martínez**

*Dirigido por:*

**Carlos Tavares Calafate**

*Universidad Politécnica de Valencia*

*Departamento de Informática de Sistemas y Computadores*

*Grupo de Redes de Computadores*

*Valencia - España*

*Abril 2015*

**Jorge Luis Zambrano Martinez**  
*Estudiante del Master Universitario en Ingeniería de Computadores*  
*Departamento de Informática de Sistemas y Computadores*  
*Universidad Politécnica de Valencia*  
*jorgeluis004@hotmail.com*

## RESUMEN

---

Las soluciones de planificación de rutas prometen lograr mejoras en términos del tiempo recorrido de una ruta si son conocedores del estado de la congestión del tráfico en tiempo real. Con las actuales soluciones, la mayoría toman como base, y de manera estática, el coste de los segmentos de la ruta, siendo el primer reto obtener estimaciones de la congestión de tráfico fiables para las vías de una ciudad en distintas horas del día, y para los distintos días del año. En general, calcular esta cantidad de datos llevará mucho tiempo, y tendrá un elevado coste computacional al evaluar las rutas óptimas en base a dependencias.

En este trabajo se estudiará la viabilidad de integrar un sistema de recomendación de rutas con un simulador de red y un simulador de tráfico. Concretamente, se desarrolla la integración de OMNeT++ y SUMO, ambas herramientas muy conocidas y utilizadas, con el servidor de rutas ABATIS como solución a la planificación de rutas, permitiendo así investigar cuales son los diferentes efectos sobre los flujos de vehículos en zonas urbanas densamente pobladas. Para eso se ajustan los patrones de tráfico, siguiendo modelos de conducción en tiempo real que van desde cambios de velocidad a cambios en la selección de rutas. Sin embargo, se desconoce la cantidad de vehículos que tienen que ser introducidos en la simulación y, por lo tanto, se desarrolló una heurística para ajustar la cantidad de vehículos a inyectar partiendo de la cantidad de vehículos que pasan por los detectores colocados en las distintas vías de Valencia, España. Resultados iniciales muestran que el servidor de rutas propuesto ofrece rutas con calidad similar a la herramienta integrada en SUMO, y que el uso de rutas alternativas es propenso a introducir cambios notables en los tiempos de recorrido de los vehículos.

## ABSTRACT

---

The route planning solutions promise improvements in terms of the travel time for a route if they are familiar with the traffic congestion state in real-time. With current solutions, where most are static and based on, the cost of road segments, the first challenge is to obtain reliable estimates of traffic congestion for a city at different times of day, and for different days of the year. Overall, calculating this amount of data will take a long time, and have a high computational cost when evaluating the optimal routes based on dependencies.

In this paper, the feasibility of integrating a route recommendation system, a network simulator and a traffic simulator is studied. In particular we study the integration of OMNeT++ and SUMO, both widely known and used tools, with the ABATIS route server as a solution for route planning, allowing to investigate what are the different effects on vehicle flows densely populated urban areas. To achieve that goal the traffic patterns adopted are adjusted, following driver models ranging from real-time speed changes to changes in route selection. However, the number of vehicles that have to be introduced in the simulation is unknown, and thus we developed a heuristic to adjust the number of vehicles to be injected based on the number of vehicles detected on the different sensors placed along the main streets of Valencia, Spain. Initial results show that the proposed route server offers routes with similar quality compared to the tool integrated in SUMO, and that the use of alternative routes is likely to introduce significant changes on the travel times of vehicles.

## ÍNDICE GENERAL

---

1	INTRODUCCIÓN Y OBJETIVOS	1
1.1	Planteamiento del Problema	1
1.2	Objetivos	1
1.3	Estructura de la memoria	2
2	SIMULADORES PARA REDES VEHICULARES	3
2.1	Introducción	3
2.1.1	Modelos de tráfico vehicular	3
2.1.2	Modelos de movilidad	4
2.2	Simuladores de tráfico vehicular	6
2.3	Simuladores de red	6
3	ABATIS CON OMNET Y SUMO	9
3.1	Simulador de tráfico SUMO	9
3.1.1	Introducción	9
3.1.2	Paradigmas Básicos	9
3.1.3	Características	10
3.1.4	Herramientas Adicionales	10
3.2	Simulador de red OMNeT++	15
3.2.1	Introducción	15
3.2.2	Estructura Modular	15
3.2.3	Paradigmas de Simulación	16
3.2.4	Paquete Inet	17
3.3	Servidor de rutas ABATIS	19
3.3.1	Introducción	19
3.3.2	API del Servidor	20
3.3.3	Formato de Salida	20
3.3.4	Herramientas ABATIS	21
3.4	Interfaz linkABATIS	21
3.4.1	Situación Actual	21
3.4.2	Objetivos	22
3.4.3	Propuestas de arquitecturas	23
3.4.4	Estructura de la Interfaz	24
4	HEURÍSTICA DE AJUSTE DE FLUJOS DE TRÁFICO	27
4.1	Análisis del Tráfico para la ciudad de Valencia	27
4.2	Procesos de Ajuste	28
4.2.1	Cálculo del Ajuste	30
4.3	Resultados	30
5	PRUEBAS Y SIMULACIONES	33
5.1	Descripción de los parámetros de simulación	33
5.2	Resultados	34
5.2.1	Distribución de probabilidad ABATIS vs DFROUTER	34
5.2.2	Estrategia estándar	34
5.2.3	Estrategia opuesta	35
5.2.4	Media de Tiempos ABATIS vs DFROUTER	35
6	CONCLUSIONES	43
	BIBLIOGRAFÍA	44

## ÍNDICE DE FIGURAS

---

Figura 1	Comportamiento promedio del Lunes en Valencia [15].	1
Figura 2	Simuladores por categoría.	3
Figura 3	Modelos Vehicular(desde la izquierda macroscópico, microscópico, sub-microscópico).	4
Figura 4	Clasificación de Modelos de Movilidad	5
Figura 5	Simulación de tráfico multimodal.	10
Figura 6	Conversión de Red plana a una descripción completa.	11
Figura 7	Ejemplo de Archivo de Estadística.	13
Figura 8	Archivo de detectores.	13
Figura 9	Archivo de rutas.	14
Figura 10	Archivo de flujos.	14
Figura 11	Archivo de emisores.	14
Figura 12	Archivo de Configuración DFROUTER.	15
Figura 13	Módulos Simples y Compuestos.	16
Figura 14	Formato del Mensaje TraCI.	18
Figura 15	Maquina de Estados OMNeT++ y SUMO.	18
Figura 16	Esquema del tiempo de llegada de Vehiculos.	19
Figura 17	Dirección de petición al Servidor ABATIS.	20
Figura 18	Respuesta de ABATIS.	21
Figura 19	Archivo de Configuración de ABATIS.	22
Figura 20	Arquitectura actual.	22
Figura 21	Primera Hipótesis Propuesta.	23
Figura 22	Segunda hipótesis propuesta.	23
Figura 23	Arquitectura linkABATIS.	24
Figura 24	Funcionamiento linkABATIS	24
Figura 25	Corrección de rutas de ABATIS.	25
Figura 26	Promedio de volumen de tráfico en Valencia por mes[15].	28
Figura 27	Promedio del volumen del tráfico en Valencia en una semana.	28
Figura 28	Comportamiento promedio diario para diferentes días de la semana[15].	29
Figura 29	Extracto de información de los Códigos ATA en Valencia.	30
Figura 30	Ajustes de Vehículos en Valencia.	31
Figura 31	Diferentes ajuste de vehículos en Valencia.	31
Figura 32	Funcionamiento de la Simulación con Porcentaje variables de vehículos que consulta su ruta a ABATIS.	34
Figure 33	Distribución de probabilidad para el tiempo de llegada ABATIS y DFROUTER	36
Figure 34	Distribución de probabilidad para el tiempo de llegada para un reparto 1-99%.	36
Figura 35	Distribución de probabilidad para el tiempo de llegada para un reparto 5-95 %.	37
Figura 36	Distribución de probabilidad para el tiempo de llegada para un reparto 10-90 %.	37

Figura 37	Distribución de probabilidad para el tiempo de llegada para un reparto 20-80 %. 38
Figura 38	Distribución de probabilidad para el tiempo de llegada para un reparto 30-70 %. 38
Figura 39	Distribución de probabilidad para el tiempo de llegada para un reparto 99-1 %. 39
Figura 40	Distribución de probabilidad para el tiempo de llegada para un reparto 95-5 %. 39
Figura 41	Distribución de probabilidad para el tiempo de llegada para un reparto 90-10 %. 40
Figura 42	Distribución de probabilidad para el tiempo de llegada para un reparto 80-20 %. 40
Figura 43	Distribución de probabilidad para el tiempo de llegada para un reparto 70-30 %. 41
Figura 44	Media de tiempo de llegada de los Vehículos ABATIS y DFROUTER. 41

## ÍNDICE DE CUADROS

---

Cuadro 1	Parámetros para la ciudad de Valencia. 17
----------	---

## ACRONYMS

---

XML	eXtensible Markup Language
MMTS	Multi-agent Microscopic Traffic Simulator
OSM	Open Street Map
SUMO	Simulation of Urban MObility
TAZ	Traffic Assignment Zone
TraCI	Traffic Control Interface
TCP	Transmission Control Protocol
NED	NEtwork Description



## INTRODUCCIÓN Y OBJETIVOS

### 1.1 PLANTEAMIENTO DEL PROBLEMA

En áreas urbanas la densidad de la población está relacionada con problemas de tráfico tales como emisiones de CO<sub>2</sub>, accidentes, contaminación ambiental, etc. que son problemas críticos para las autoridades de las ciudades. En términos de soluciones para reducir el tráfico, se han desarrollado varias alternativas como el uso de transporte público o alternar los días de acceso a las vías acorde a número de matrícula. Para ello se analiza el flujo del tráfico de las ciudades de cara a su optimización.

Para este trabajo se seleccionó a la ciudad de Valencia, cuya área metropolitana es la tercera más grande de España con una población de 17.7 millones de habitantes. El Ayuntamiento de Valencia proporcionó archivos detallados sobre el flujo del tráfico en cada calle y avenida, a cada hora, para todo el año de 2012.

Nuestro análisis se centrará en el mes de Noviembre, el cual contiene un volumen de tráfico promedio y no incluye periodos vacacionales [15]. Además, nos centraremos en una hora pico para un día de congestión promedio, como se ilustra en la *figura 1*. Estos datos corresponden al Lunes 5 de noviembre del 2012, y nos centraremos en la franja horaria entre las 8h00 a 9h00.

### 1.2 OBJETIVOS

El principal objetivo de este Trabajo Fin de Máster es analizar y desarrollar un sistema centralizado para optimizar el tráfico usando un servidor de rutas propio al que denominamos ABATIS, y cuyo objetivo último es reducir los tiempos de trayecto para los vehículos que circulan en una ciudad. Para alcanzar este objetivo es necesario cumplir con estos objetivos específicos:

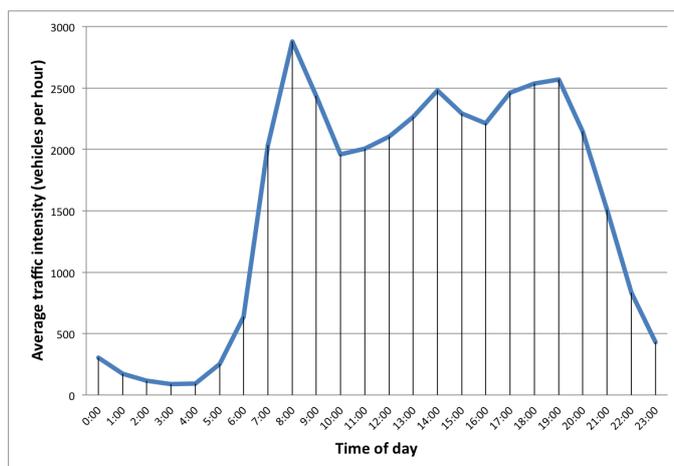


Figura 1: Comportamiento promedio del Lunes en Valencia [15].

- Analizar qué módulo de generación de rutas se adecua mejor a nuestro problema.
- Obtener el tiempo de recorrido de los vehículos.
- Proponer una heurística que permita ajustar el número de vehículos a los detectados en la ciudad de Valencia en el período designado.
- Vincular las herramientas ABATIS, OMNeT y SUMO para ofrecer un entorno de simulación completo.
- Desarrollar un método para computar los tiempos correspondiente a cada vehículo para un trayecto de simulación.
- Calcular las mejoras obtenidas en términos de tiempo de recorrido de cada vehículo.

### 1.3 ESTRUCTURA DE LA MEMORIA

La presente memoria está compuesta por los siguientes capítulos:

Capítulo 2: Se introducen los simuladores para redes vehiculares en los que se comentan los modelos de tráfico y movilidad. También se ofrecen detalles sobre los simuladores de redes y otras herramientas existentes.

Capítulo 3: Se realiza una breve introducción al simulador SUMO y sus herramientas adicionales, así como al simulador de red OMNeT++ agregando ejemplos explícitos con respecto al trabajo de esta memoria. Además, se ofrecen detalles respecto al funcionamiento del servidor de rutas ABATIS con su respectivo proceso de petición y recepción de la información proporcionada. Por último se presenta la interfaz linkABATIS que vincula ambos simuladores (OMNeT++ y SUMO) con el servidor de rutas.

Capítulo 4: Se presenta el proceso de ajuste que permite determinar la cantidad de vehículos que se van a inyectar en la red a través de la información relativa a densidad de tráfico proporcionada por el Ayuntamiento de Valencia.

Capítulo 5: Se realizan simulaciones con respecto al tiempo que se va a simular, y con respecto al porcentaje de vehículos que van a consultar sus rutas hacia ABATIS, y presentar sus respectivos resultados.

## 2.1 INTRODUCCIÓN

En los simuladores para redes vehiculares existen varias herramientas para llevar a cabo simulaciones con sus características particulares, para lo que poseen dependencias significativas de integración con otros programas para mejorar su precisión. Actualmente dichas herramientas tienen una gran aceptación en la comunidad científica, y se clasifican en dos categorías: simuladores de tráfico y simuladores de red. (ver figura 2).

A continuación se presentarán los modelos de tráfico existentes ya que éstos representan el nivel de detalle con el que se desea realizar las simulaciones. Así, los modelos de movilidad son representaciones de los elementos del escenario a simular, y su objetivo primordial es imitar el comportamiento de los diversos elementos en la simulación de la red, para lo cual se observan aspectos relacionados con el comportamiento de los conductores, las múltiples señales de tráfico, los cambios de aceleración y velocidad asociados a los vehículos, etc.

### 2.1.1 Modelos de tráfico vehicular

Como se mencionó anteriormente, los modelos de tráfico vehicular la simulación, y habitualmente se encuentran integrados en los simuladores de redes vehiculares. Las simulaciones dependen de factores como la velocidad del vehículo, la densidad de vehículos, el entorno, y el flujo de tráfico. Estos se clasifican en cuatro tipos de modelos: macroscópico, microscópico, sub-microscópico y mesoscópico [22].

Los simuladores que usan modelos macroscópicos poseen la particularidad de dividir el espacio de la simulación en secciones, por lo que a nivel computacional el coste es bajo pero no logran modelar el entorno real con precisión. Los modelos mesoscópicos representan los sistemas de transporte analizando el comportamiento de los conductores. Para poder lograrlo, se requiere definir funciones que expresen la probabilidad de que un vehículo con una determinada velocidad se ubique en una determinada posición en un tiempo aproximado, para lo que usa métodos de física estadística. Los modelos que se ajustan a

Simuladores de Tráfico	Simuladores de Red
CityMob	NS-3
VISSIM	OMNeT++
SmartPath	GLOMOSIM
SUMO	GTNETS
Integration	QualNet
MITSIM	
VanetMobiSim	

Figura 2: Simuladores por categoría.

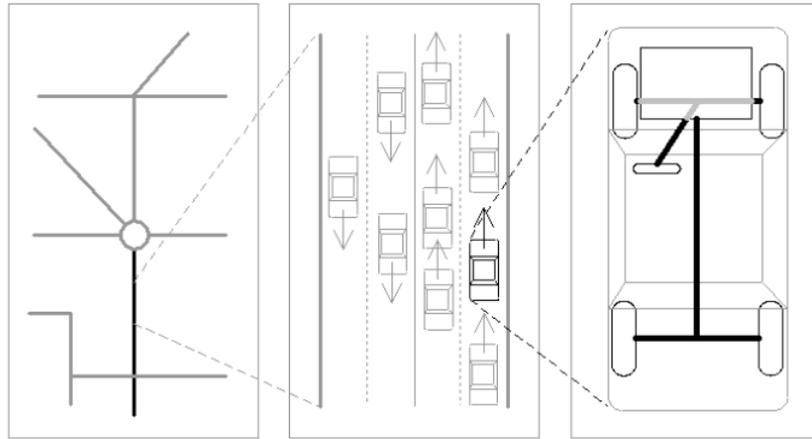


Figura 3: Modelos Vehicular(desde la izquierda macroscópico, microscópico, sub-microscópico).

la realidad con mayor nivel de precisión son los modelos microscópicos, por el hecho de que cada vehículo es un nodo independiente. El nodo contiene información relativa a las vías asociadas a su recorrido, dando como resultado un elevado coste computacional. Los modelos sub-microscópicos están basados en los modelos microscópicos, los cuales denotan el funcionamiento de distintas partes de los vehículos en forma detallada, y los dividen en sub-unidades, lo que conlleva un alto coste computacional, aunque no son de utilidad para nuestro objetivo. Al igual que los modelos microscópicos, realizan seguimientos de los vehículos, su interacción con las calles y los procesos relativos a la conducción. En la *figura 3* muestra los modelos de tráfico vehicular.

#### 2.1.2 Modelos de movilidad

Para poder simular de manera realista las redes vehiculares, es necesario utilizar un modelo de movilidad en que se pueda representar la topología del mapa, los nodos, los movimientos de los vehículos, así como la velocidad, dirección, y aceleración de cada vehículo. El desarrollo de los modelos de movilidad pueden ser clasificados en cuatro clases: modelos sintéticos, modelos basados en datos, modelos basados en trazas, y modelos basados en simuladores de tráfico [19], como podemos observar en la *figura 4*.

##### *Modelos Sintéticos*

Desarrollan un modelo matemático basados en el comportamiento del conductor, el cual es modelado acorde a los vehículos para así poder simular sus diversas acciones de manera virtualizada. El problema de estos modelos es la complejidad para modelar comportamientos humanos, ya que no pueden ser programados para seguir un comportamiento específico en todos los casos.

*Obstacle Mobility Model* [20]: Modela los movimientos de los vehículos de forma similar a un entorno real. Esto permite al usuario poder definir el escenario en base a las dimensiones que posee el terreno, y agregarle diferentes obstáculos como vehículos, edificaciones, etc. Para realizar los cálculos de las rutas potenciales es necesario proporcionar los obstáculos previamente. En base a esa información utiliza el Diagra-

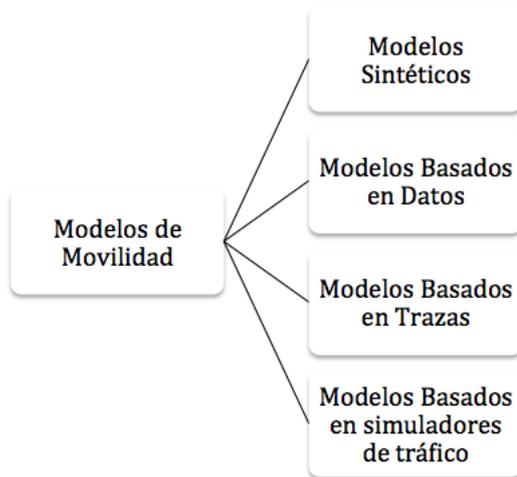


Figura 4: Clasificación de Modelos de Movilidad

ma de Voronoi de vértice de los obstáculos, y se determinan las rutas que se encuentran en medio de dos obstáculos adyacentes. Después se comienzan a ubicar los vehículos seleccionando tanto un destino como una velocidad aleatoriamente, siguiendo la ruta de mínima distancia del Diagrama de Voronoi donde el coste de cada segmento de la ruta es su propia distancia euclidiana. Una vez llega a su destino, el vehículo se para por un periodo predeterminado, selecciona un nuevo punto como destino, y calcula la nueva ruta.

#### *Modelos Basados en Datos*

Para la información de movilidad macroscópica es importante tener datos sobre la circulación urbana, obtenidos de estudios anteriores, y desarrollar una movilidad genérica reproduciendo de manera determinista las pautas correspondientes a tráfico urbano real. Como ejemplo podemos citar:

*Realistic Vehicular Traces* [8]: Las trazas se obtienen de un simulador de tráfico microscópico denominado MMTS (Multi-agent Microscopic Traffic Simulator) desarrollado en la Universidad Técnica de Berlín de Alemania, por K.Nagel. por lo que es capaz de simular de manera real el tráfico público y privado de Suiza.

#### *Modelos Basados en Trazas*

Son aquellos que generan patrones genéricos de movilidad a partir de movimientos de las trazas. La dificultad de estos modelos es extrapolar los patrones no observados directamente por dichas trazas, para lo cual se realiza modelos matemáticos complejos que permite predecir en cierto grado los patrones de movilidad. Otra limitación que posee es la creación de modelos de movilidad vehicular basados en trazas, ya que no se tiene a disposición las trazas vehiculares.

*Reality Mining* [17] es un buen ejemplo de este modelo. Su objetivo principal es estudiar las características de los dispositivos móviles para conocer las interacciones humanas en mayor profundidad comparativamente con la metodología basada en encuestas o simulaciones. La razón por la que no se realizaba antes, es que los teléfonos móviles

no eran lo suficientemente potentes como para poder rastrear a sus usuarios.

#### *Modelos Basados en Simuladores de Tráfico*

Son modelos basados en una intensa validación de procesos en trazas reales o estudio de comportamiento. Las características que se resaltan para este tipo de modelo son tráfico microscópico urbano, el consumo de energía, y la monitorización de los niveles de ruido y de contaminación. Estos modelos son una evolución de sus predecesoras, y las más utilizadas actualmente.

En el caso de SUMO, este utiliza una extensión del modelo de Gipps, inventado por Krauß [21]. En este modelo, por cada paso de tiempo, la velocidad del vehículo es adaptada a la velocidad del vehículo delantero de manera a evitar colisiones. Siguiendo con el modelo de Gipps, este nos ofrece una ecuación en la cual la velocidad es nombrada velocidad segura ( $v_{safe}$ ), tal y como se ilustra en el algoritmo 2.1:

---

**Algoritmo 2.1** Ecuación de Velocidad Segura.

---

$$V_{safe}(t) = v_l(t) + \frac{g(t) - v_l \tau}{\frac{\bar{v}}{b(\bar{v})} + \tau}$$

donde:

- $v_l(t)$  : Velocidad del vehículo delantero en el instante  $t$ .
  - $g(t)$ : Distancia hacia el vehículo delantero en el instante  $t$ .
  - $\tau$  : Tiempo de reacción del conductor, normalmente 1s.
  - $b$ : Función de desaceleración.
- 

## 2.2 SIMULADORES DE TRÁFICO VEHICULAR

Son simuladores que se fundamentan en el tráfico de los vehículos, dando la posibilidad de modelar varios escenarios reales sin olvidarse de las características de los elementos que componen dicho entorno, como los conductores, las características de los vehículos, trenes, peatones, etc. que estén continuamente intercalando con los elementos de las vías como semáforos, obstáculos, cruces, etc.

Lo único que es posible configurar de antemano, es el movimiento de los vehículos en las vías con determinados escenarios y su respectivo modelo de movilidad, en el que se incluyen factores como velocidad, aceleración y tiempo. Permiten además generar trazas de movimiento que, en muchos casos, pueden llegar a ser usadas por simuladores de red.

Para estos simuladores, es posible importar mapas reales de varias fuentes como OSM (Open Street Map) [7]. En este trabajo se importará el mapa de la ciudad de Valencia utilizando el simulador de tráfico denominado SUMO [9].

## 2.3 SIMULADORES DE RED

Estos simuladores se caracterizan por modelar la interconexión entre máquinas, enrutadores y otros dispositivos, ofreciendo la posibilidad de realizar múltiples tipos de pruebas y obtener resultados generalmente

fiables sin necesidad de desplegar una red real, o intervenir una red que esté operativa.

Algunos simuladores de red permiten patrones de movilidad generados por simuladores de tráfico incluyendo la API, para poder establecer una comunicación entre ellos. El simulador que se va a utilizar es OMNeT++[6], el cual posee una API denominada INET [4], que permite la comunicación entre el simulador de tráfico (SUMO) y el simulador de red (OMNeT++).



En el presente capítulo se detalla el simulador de red y el simulador de tráfico, ambos de código abierto, denominados OMNeT++ y SUMO, respectivamente. Ambos pueden comunicarse entre sí por medio de un API o módulo de comunicación, el cual trabaja como una extensión de SUMO denominada TraCI. Además, SUMO posee herramientas adicionales que pueden potenciar en gran medida este simulador. Otro punto que se detallará es el servidor de rutas, ABATIS

### 3.1 SIMULADOR DE TRÁFICO SUMO

#### 3.1.1 *Introducción*

Normalmente se pensaría que el tráfico estaría compuesto de unas pocas variables como hora de partida, la ruta que van a seguir los distintos vehículos, y en algunos casos la duración de una ruta, aunque esta suposición no es correcta ya que dependerá obviamente del vehículo, del conductor, y del estado de congestión del tráfico a lo largo de la ruta. Esto provoca a varios inconvenientes al modelar el tráfico, por lo que su solución, la descripción del tráfico, se consigue mediante el uso de ecuaciones matemáticas complejas que miden estos dos factores: (i) el movimiento del vehículo sobre una vía y (ii) el comportamiento humano, correlacionando ambos entre sí. Otros parámetros que influyen en el tráfico son las condiciones meteorológicas, la infraestructura y las incidencias en las vías..

En esta área destacar el Instituto de Investigación en el Transporte (Centro Aeroespacial Alemán), y el Centro de Informática Aplicada de Colonia (Alemania), los cuales obtienen importantes conclusiones en este ámbito a través de un proyecto desarrollado por diferentes organismos y cofinanciado por la Comisión Europea llamado SMARTTEST [14]. En dicho proyecto se propone desarrollar una herramienta de código abierto capaz de satisfacer las necesidades anteriormente descritas.

Así aparece SUMO [16], el cual posee paquetes de código abierto para la simulación del tráfico, y que ha evolucionando hacia una completa gama de utilidades de modelos de tráfico en el que se incluyen la capacidad de leer redes de vías en múltiples formatos, la generación de la demanda, y utilidades de enrutamiento desde varios orígenes de entrada como matrices de origen-destino, conteo de tráfico, etc. Ofrece además un alto rendimiento en simulación a través de la interfaz TraCI, permitiendo realizar simulaciones en línea.

#### 3.1.2 *Paradigmas Básicos*

SUMO posee el potencial para simular el tráfico en una red vial de una ciudad con la ventaja de que en su simulación es multimodal, por lo que comprende no solo el movimiento de los vehículos dentro de la ciudad, pero también aspectos como el sistema de transporte público, redes de trenes, e incluso trayectos peatonales. Este puede ser descrito

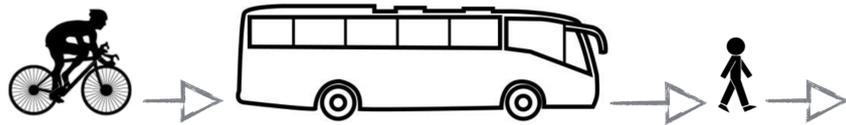


Figura 5: Simulación de tráfico multimodal.

mediante múltiples rutas, las cuales pueden ser compuestas en subrutas, y así poder describir una sola modalidad de tráfico.

Al simular una persona, esta puede tomar un vehículo hasta la estación de transporte público más cercana y poder continuar su viaje, así como también puede ir caminando. El proceso de caminar dentro de la simulación no está estipulado, y lo que se hace es un modelado estimado del tiempo que la persona necesita para buscar su destino. En la *figura 5* se muestra un compendio de ejemplo de este tipo de rutas.

El flujo de tráfico es simulado microscópicamente, lo que significa que cada vehículo que se mueve dentro de la red simulada es modelado de manera individual, y está ubicado en cierto lugar y con su respectiva velocidad asociada. Cada lapso de tiempo tiene una duración de un segundo, pasado el cual dichos valores son actualizados. De esta manera se logra simular de forma discreta un movimiento continuo en el espacio. La simulación de tráfico, debe respetar las normas de velocidad máxima y reglas de prioridad hacia el lado derecho, ya que son atributos de las calles.

### 3.1.3 Características

Al ser un simulador de código abierto, está en constante desarrollo ya que ha tenido una muy buena aceptación en la comunidad científica. Para este proyecto se ha usado la versión 0.18, la cual incluye funcionalidades como [9, 16]:

- Simulación Microscópica.
- Vehículos, peatones y transporte público son modelados explícitamente.
- Interacción en línea.
- Control de la simulación con TraCI
- Simulación de tráfico multimodal, como vehículos, transporte público y peatones.
- Horarios de luces de tráfico pueden ser importada o generados de manera automática por SUMO.
- Soporte de formatos para importar como OpenStreetMap, VISUM, VISSIM, NavTeq.

### 3.1.4 Herramientas Adicionales

*Netconverter*[13]

Debido a su alta complejidad, la descripción de la red que usa SUMO no puede ser generada por un usuario, siendo necesario el uso de una herramienta para convertir información sobre elementos críticos, como

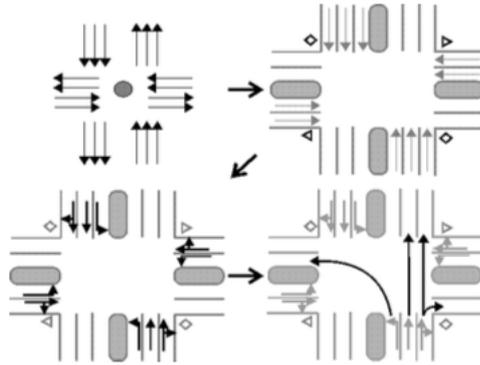


Figura 6: Conversión de Red plana a una descripción completa.

una lista de edges (bordes) y nodos opcionales, para completar la red que requiere SUMO. Durante este proceso, esta herramienta lee los datos, computa las entradas que necesita para SUMO, y escribe los resultados en archivos XML. Los formatos de entrada que usa son:

- "SUMO native" descripciones XML como \*.edg.xml, \*.nod.xml, \*.con.xml, \*.tll.xml
- OpenStreetMap (\*.osm.xml)
- VISUM, que incluye formas y demandas
- Vissim, incluye demandas
- OpenDRIVE
- MATsim • SUMO (\*.net.xml)
- Shapefiles (.shp, .shx, .dbf), de ArcView y el nuevo Tiger networks
- Robocup Rescue League, incluye las formas
- DLR variantes internas de Navteq's GDF

Para el caso específico de la ciudad de Valencia, España, se ha importado un mapa con el formato de OpenStreetMap al formato de red de SUMO.

En la *figura 6* podemos ver como los datos pueden ser computados de una lista simple de nodos y edges, que además agrega el flujo del tráfico de izquierda a derecha, y de arriba hacia abajo. El primer paso es determinar las distintas prioridades en las uniones, el segundo paso es realizar cómputo de relaciones entre los carriles (lanes) y los límites (edges) que pueden ser alcanzados, y el último paso consiste en dividir los destinos entre los carriles entrantes. El cómputo es flexible, dependiendo principalmente del número de entrantes y salientes de los edges, sus respectivos tamaños, así como las prioridades que poseen dentro de la red.

Debido a que existe mucha información dentro del mapa de Valencia, como son vías para bicicleta, trenes, servicios públicos, peatones, etc, y como no nos concierne incluirlos, se procede a excluir esos edges para así tener un plano correspondiente únicamente a vías que usen los vehículos, para lo cual se realiza una conversión con el siguiente comando:

```
netconvert -osm valencia.osm.xml --remove-edges.by-vclass rail_slow, rail_fast,
bicycle, pedestrian, lightrail, cityrail, delivery --output-file valencia.net.xml
```

*Duarouter*[12]

Es una herramienta que importa diferentes definiciones de demanda, y que realiza el cómputo de las rutas seguidas por los vehículos para que puedan ser usadas por SUMO como la ruta más corta. El principal propósito de esta herramienta es construir rutas para los vehículos de acuerdo a la definiciones de demanda correspondientes, computar rutas durante la asignación del usuario, y reparar los distintos problemas de conectividad existentes en los archivos de rutas.

Los algoritmos de enrutamiento que usa esta herramienta son:

- Dijkstra: dependiente del tiempo.
- Astar: dependiente del tiempo.
- CH: no tiene dependencias temporales, más bien asociado a las condiciones de la salida durante un período de tiempo específico.
- CHWrapper: es una variante del CH en la que se construye una jerarquía separada para cada tipo de vehículo.
- Bulkstar: Una variante de Astar donde la velocidad aumenta a medida que surgen las rutas que poseen diferentes orígenes pero, destinos iguales.

*OD2Trips*[5, 3]

Esta herramienta calcula tablas de viajes (trips) de un O/D (Origen/-Destino) en forma de matrices. Para lograrlo asume que las matrices deben ser codificadas de acuerdo al volumen de los vehículos de un distrito o zona asignada al tráfico (TAZ) hacia otra zona.

Normalmente esta herramienta solo puede importar datos almacenados en formato de VISUM/VISION/VISSIM[11]. En el caso que no se pueda contar con matrices almacenadas en estos formatos, hay tres posibilidades:

- Convertirlos a un formato soportado.
- Escribir un lector propio de OD2TRIPS.
- Convertirlos a definiciones de flujo y pasarlas a DUAROUTER.

Esta herramienta lee todas las matrices y genera definiciones de viaje, las definiciones que son numeradas comenzando desde el cero. Cada célula de la matriz O/D describe la cantidad de vehículos que van a ser insertados en la red durante un cierto período, y computa los tiempos de salidas de los vehículos. Esto normalmente lo realiza de forma aleatoria dentro de intervalo, describiendo las células y otorgando uniformemente el tiempo entre inserciones.

*ActivityGen* [1]

Esta herramienta permite generar demanda asociada a la descripción de la población en la red, usando un modelo de tráfico basado en actividades, ya sean de trabajo, escuela o tiempo libre, así mismo para los peatones mientras caminan, y para medios de transporte como vehículos, bicicletas y buses. En otras palabras, el principal propósito es generar demanda para una población sintética y estimar la movilidad con respecto a los deseos de la población. Esta herramienta usa un

```

<city>
  <general inhabitants="10000" households="5000" childrenAgeLimit="18" retirementAgeLimit="65" carRate="0.8"
  unemploymentRate="0.05" footDistanceLimit="500" incomingTraffic="1500" outgoingTraffic="3000" />
  <parameters carPreference="0.5" meanTimePerKmInCity="360" freeTimeActivityRate="0.15"
  uniformRandomTraffic="0.2" departureVariation="120" />
  <population>
    <bracket beginAge="0" endAge="18" peopleNbr="1200" />
    <bracket beginAge="18" endAge="75" peopleNbr="1990" />
  </population>
  <workHours>
    <opening hour="30600" proportion="0.3" />
    <opening hour="32400" proportion="0.7" />
    <closing hour="45000" proportion="0.2" />
    <closing hour="63000" proportion="0.2" />
    <closing hour="64800" proportion="0.6" />
  </workHours>
  <streets>
    <street edge="10305544#0" population="7.95" workPosition="1.59"/>
    <street edge="10305544#1" population="6.7" workPosition="1.34"/>
    <street edge="10305544#2" population="8.9" workPosition="1.78"/>
    <street edge="10305544#3" population="6.25" workPosition="1.25"/>
    ....
  </streets>
</city>

```

Figura 7: Ejemplo de Archivo de Estadística.

```

<a>
  <detectorDefinition id="dd_4693573#0_0" lane="4693573#0_0" pos="23.9" type="source"/>
  <detectorDefinition id="dd_9281513#0_0" lane="9281513#0_0" pos="27.0" type="source"/>
  <detectorDefinition id="dd_98711532#0_0" lane="98711532#0_0" pos="14.3" type="source"/>
  ...
</a>

```

Figura 8: Archivo de detectores.

archivo de estadísticas, en el cual incluye información general de la ciudad como son los habitantes, la tasa de desempleo, número de viviendas, y el tráfico entrante y saliente, etc. como podemos ver en la *figura 7*.

#### *Dfrouter [2]*

Este módulo de enrutamiento se fundamenta en la idea de medir, para cada carretera equipada con anillos de inducción (induction loops), el flujo de entrada y salida de tráfico. Además, usa la información recolectada por los anillos de inducción para reconstruir las rutas seguidas por los vehículos y siguiendo estos pasos:

- Computa los tipos de detectores , los cuales pueden ser: detectores de origen (source detectors), detectores sumidero (sink detectors) o entre detectores (in-between detectors).
- Calcula las rutas entre los detectores.
- Calcula el flujo que existe entre los detectores.
- Almacena el flujo y algunas estructuras de control que va a requerir.

#### *Cómputo del Tipo de Detectores*

Como se mencionó previamente, la idea fundamental de esta herramienta es asumir que la red está completamente cubierta por detectores, y solo contiene la información de los anillos de inducción dados definidos originalmente para así poder computarlos. Para realizar esto, DFROUTER requiere una red que contenga una lista de los detectores, incluyendo su descripción y la posición del anillo de inducción. Estos datos deberán estar en un archivo XML similar al que se presenta en la *figura 8*:

```

<routes>
  <route id="4693573#0_to_9281469#0" edges="4693573#0 41197570#6 41197564 9281469#0"/>
  <route id="4693573#0_to_10380715#0" edges="4693573#0 41197570#6 10380715#0"/>
  <route id="4693573#0_to_41199033#0" edges="4693573#0 41197570#6 41197564 27130663 41199033#0"/>
  ...
</routes>

```

Figura 9: Archivo de rutas.

```

Detector;Time;qPKW;qLKW;vPKW;vLKW
dd_4693573#0_0;0;158;0;25.000;0
dd_9281513#0_0;0;158;0;25.000;0
dd_98711532#0_0;0;158;0;25.000;0

```

donde:

*Detector*: Identificador del detector que debe ser igual al del Archivo de detectores.  
*Time*: Período de tiempo en que comienza la descripción de entrada en minutos.  
*qPKW*: Número vehículos que van a conducir sobre los detectores.  
*qLKW*: Número de vehículos públicos que conducen sobre los detectores.  
*vPKW*: Promedio de velocidad de los vehículos que pasan sobre los detectores en km/h.  
*vLKW*: Promedio de velocidad de vehículos públicos que pasan por los detectores en km/h.

Figura 10: Archivo de flujos.

### Cómputo de Rutas

Una vez obtenida la información relativa a los puntos de entrada y salida de la red por parte de los vehículos, lo siguiente es determinar las rutas para cada uno de estos orígenes y destinos. El archivo, una vez generado, contiene las rutas pero no la información de los vehículos ni de sus tipos.

Comúnmente, las rutas comienzan en los detectores origen y finalizan en los detectores sumidero correspondientes.. Además, existe una opción adicional para que esta herramienta obligue a comenzar las rutas construidas en medio de los detectores, como podemos apreciar en la *figura 9*.

### Cómputo de Flujo

A continuación esta herramienta usa las rutas calculadas y los flujos tomados de los detectores del mundo real para que coincidan con el modelo de red. Esta información debe estar en un archivo con formato CSV usando separadores ";" para que dividan los caracteres, tal y como se ilustra en la *figura 10*.

### Almacenaje de Flujo y otras estructuras

Como última etapa del proceso, esta herramienta guarda los vehículos computados conjuntamente con sus rutas. Con respecto a los vehículos, se generarán a partir de los detectores origen, por lo que son ubicados en ciertas posiciones de los carriles de la red. DFROUTER puede forzar la inserción de vehículos en la red, situación en la que el archivo contendrá la información relativa a cada detector origen, así como los vehículos respectivos. Dichos vehículos tendrán la salida acorde al archivo de flujos y las rutas computadas previamente, como se aprecia en la *figura 11*.

```

<additional>
  <routeDistribution id="dd_98711537#0_0">
    <route refId="98711537#0_to_8082686#2" probability="1.00"/>
  </routeDistribution>
  <vehicle id="emitter_dd_98711537#0_0_8880" depart="8.88" departLane="0" departPos="0.00" departSpeed="max"
    route="98711537#0_to_8082686#2"/>
</additional>

```

Figura 11: Archivo de emisores.

```

<input>
  <net-file value="valencia.net.xml"/>
  <detector-files value="valencia.det.xml"/>
  <measure-files value="valencia.flow.txt"/>
</input>

<output>
  <routes-output value="valencia.rou.xml"/>
  <emitters-output value="valencia.add.xml"/>
</output>

```

Figura 12: Archivo de Configuración DFROUTER.

Para poder incluir los archivos, y que estos realicen el respectivo cómputo de salida (que son las rutas, vehículos e información adicional de los vehículos) hay que disponibilizar dicha información en un archivo de configuración DFROUTER. El archivo correspondiente a la ciudad de Valencia se muestra en la *figura 12*.

## 3.2 SIMULADOR DE RED OMNET++

### 3.2.1 Introducción

El OMNet++ es un framework extensible y modular que incluye una librería de simulaciones con componentes basadas en C++, orientado a objetos. Al tener una arquitectura genérica, es capaz de realizar simulaciones de eventos, siendo capaz de resolver varios problemas como [6]:

- Modelado de comunicación de redes, tanto cableadas como inalámbricas.
- Modelado de protocolos.
- Modelado de colas de redes.
- Modelado de multiprocesadores y otras distribuciones de sistemas hardware.
- Validación de arquitecturas de hardware.
- Evaluación de rendimientos en sistemas de software complejos.
- Modelado y simulación de cualquier sistema basado en eventos que pueda ser mapeado en entidades que intercambian mensajes.

Este simulador provee módulos o componentes escritos en C++, así como un lenguaje de alto nivel (NED) para el uso de estos modelos. Una de las ventajas que se puede tener es la reusabilidad de los modelos ya que son de código abierto. Su nacimiento data de 1997 en el Departamento de Telecomunicaciones de la Universidad de Tecnología y Economía de Budapest, en Hungría [23].

### 3.2.2 Estructura Modular

Los modelos que usa OMNeT++ está compuestos principalmente de modelos jerárquicos, los cuales están divididos en módulos básicos o simples que a su vez pueden reagruparse en otros modelos compuestos, estando todos dentro de un nivel superior al cual lo denominan módulo del sistema (ver *figura 13*).

Existe un módulo básico llamado *type*, el cual ofrece funcionalidad básica para poder implementar los modelos que compondrán el modelo.

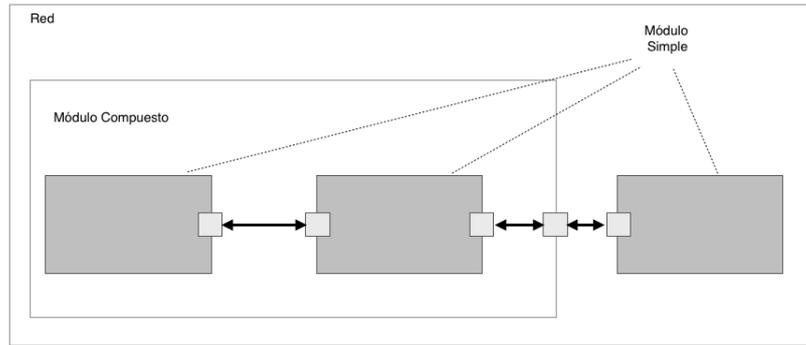


Figura 13: Módulos Simples y Compuestos.

*NED: Es el lenguaje de descripción de la topología de OMNeT++, ya que tiene una sintaxis muy simple pero a su vez poderosa cuando se trata de definir topologías regulares.*

Esto permite que el usuario, de manera transparente, pueda dividir un módulo simple en varios modelos simples dentro de un módulo compuesto.

Para la comunicación se utiliza mensajes con los tradicionales atributos como "timestamp" y datos arbitrarios. Típicamente, el envío de mensajes entre modelos simples se hace por medio de puertas (gates). Estas puertas son entradas y salidas de las interfaces de los modelos, por lo que los mensajes son enviados hacia la salida de las puertas e ingresan por otras puertas de entrada. A estas entradas y salidas se las conoce como conexión. En la estructura jerárquica del modelo, los mensajes viajan por una cadena de conexiones que comienzan y llegan en un módulo simple. Así mismo, los modelos compuestos son como una caja de un modelo que transparentemente se intercambian mensajes, tanto internamente como hacia el exterior.

Los parámetros que pueden tener los modelos son principalmente usados para pasar los datos de la configuración a los modelos simples, y de esta forma poder ayudar a definir el modelo de topología, ya sea mediante un archivo NED o un archivo de configuración *omnetpp.ini*. Los parámetros pueden ser de varios tipos, tales como cadena (string), valores numéricos o punteros. Esto es debido a que los parámetros son representados como objetos en el programa. Los módulos compuestos pueden pasar los parámetros o expresiones a sus submódulos mediante valores o referencias. Usando como referencia, dicho valor puede ser propagado de manera global en el modelo, y así ir cambiando durante la ejecución de la simulación. Esta técnica es útil en el caso de la simulación de escenarios para la optimización de parámetros.

Para nuestro caso específico, en la *tabla 1* se presentan los parámetros de simulación más relevantes.

### 3.2.3 Paradigmas de Simulación

Esta herramienta contiene un gran conjunto de librerías para la simulación debido a que debe suplir la necesidades de las tareas comunes de simulación. Además ofrece a los desarrolladores rutinas para poder tener un control y una interfaz gráfica para la creación de modelos, así como para la ejecución y depuración de las simulaciones. Una ventaja adicional es su capacidad para el dar soporte a la ruta de tráfico en la red, ya que provee la habilidad de explorar la topología de red actual, de la cual extrae una estructura de datos gráfica.

Cuadro 1: Parámetros para la ciudad de Valencia.

Parámetros	Valor	Significado
sim-time-limit	900s	Es el tiempo limite de la simulacion
manager.updateInterval	1s	Intervalo de actualizacion en simulacion
manager.host	localhost	Maquina donde se ejecuta el servidor
manager.port	9999	Puerto de comunicacion
manager.moduleType	valencia.Car	módulo base para la simulacion
manager.moduleName	host	Nombre del módulo
manager.lauchConfig	xmlDoc("lauch.sumo.xml")	Nombre del archivo para simulacion con SUMO
manager.pathStadisticalFile	\${resultdir}/valencia.sta.txt	Ruta de archivo de estadistica de tiempo
manager.pathLinkABATIS	~/inet/ABATIS/linkABATIS.py	Ruta de la interfaz para coneccion con ABATIS
manager.pathAdditionalFile	~/valencia/valencia.add.xml	Ruta del Archivo Emisores de Valencia
manager.pathEdgeFiles	~/valencia/valencia.edges.xml	Ruta del Archivo de Edges de Valencia
manager.pathNodeFiles	~/valencia/valencia.nodes.xml	Ruta del Archivo de Nodos de Valencia
manager.pathOSM	~/valencia/ATAS-Valencia.osm	Ruta del Archivo del mapa de Valencia
manager.pathRoutesFiles	~/valencia/valencia.rou.xml	Ruta del Archivo de rutas de Valencia
manager.carNumbersLinkABATIS	10	Porcentaje de vehiculos a Consultar a ABATIS

OMNeT++ permite al usuario definir la estructura del modelo que va a simular con un lenguaje llamado NED. Lo fundamental de una descripción NED son las declaraciones de los modelos simples, las definiciones de los modelos compuestos, y las definiciones de la red propiamente dichas. En concreto, los modelos simples describen la interfaz del módulo, los modelos compuestos se centran en la declaración de las interfaces de los modelos externos, submódulos e interconexiones, y por último la definición de la red define un modelo como una instancia de un tipo de módulo.

Un modelo consiste de las siguientes partes:

- Mensajes de definiciones: los usuarios pueden definir varios tipos de mensajes, y éstos a su vez pueden agregar los datos que son producidos; además estos mensajes serán traducidos a clases en C++
- Módulos simples: están escritos en C++ con sufijo .h y .cc
- La topología de la descripción del lenguaje NED: son descritos mediante una estructura modular con sus parámetros correspondientes. La ventaja de usar OMNeT++ para abrir estos archivos es ofrecer tanto un entorno gráfico como un editor de texto al mismo tiempo.

#### 3.2.4 Paquete Inet

INET [4] es una librería de código abierto para el entorno de simulación de OMNeT++. Este framework provee protocolos, agentes y otros modelos para el trabajo de los investigadores en área como las redes de comunicación. INET es mayormente utilizada cuando se diseñan y validan nuevos protocolos, o se exploran nuevos escenarios.

Esta librería contiene implementaciones para los protocolos de Internet más relevantes: TCP, UDP, IPv4, IPv6, OSPF, etc. Ofrece además implementaciones para diferentes protocolos en las capas inferiores (niveles físico y de enlace) como Ethernet, PPP, IEEE 802.11, etc. Incluye también protocolos que dan soporte a movilidad, como protocolos de encaminamiento en MANETs, funcionalidades avanzadas para QoS como DiffServ, MPLS con LDP y RSVP-TE, y otros componentes y protocolos.

TraCI [24]; es una arquitectura de código abierto basada en una conexión TCP entre OMNeT++, que actúa como cliente, y el SUMO, que actúa como servidor, usando mensajes TraCI para poder así intercambiar comandos, tal y como se ilustra en la *figura 14*.

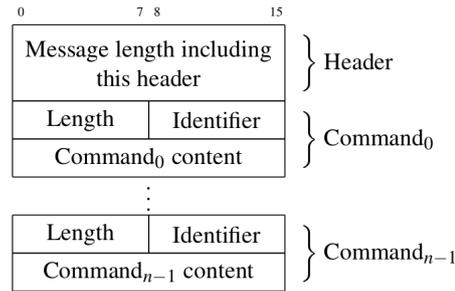


Figura 14: Formato del Mensaje TraCI.

La gestión de la movilidad se hace planificando los movimientos de los nodos en cada intervalo de tiempo por parte de OMNeT++. Esto encaja perfectamente con el funcionamiento de SUMO, ya que este simulador también maneja instantes de simulación en tiempos discreto. Para entender el funcionamiento, se presenta la máquina de estados (*ver figura 15*) que incluye los modelos están presentes tanto en OMNeT++ como en SUMO, y que tienen la capacidad de guardar cualquier comando para su ejecución en cierto intervalos de tiempos y así poder sincronizarse de manera correcta. OMNeT++ envía los comandos almacenados en cada instante de tiempo hacia SUMO, y este comenzará la correspondiente iteración de la simulación del tráfico.

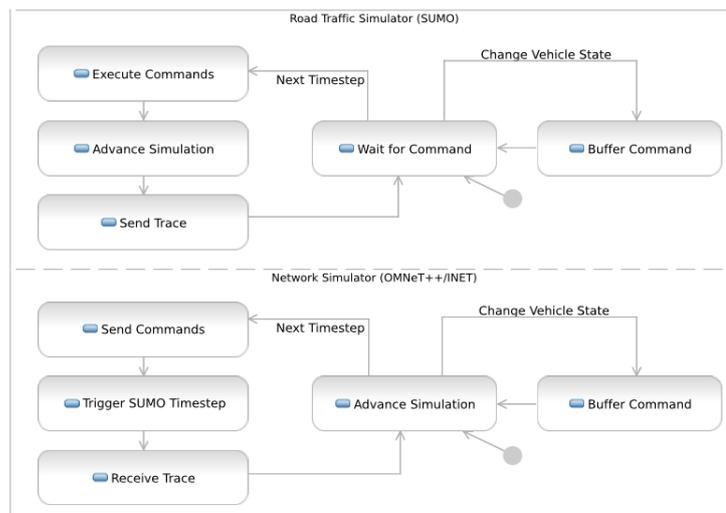


Figura 15: Máquina de Estados OMNeT++ y SUMO.

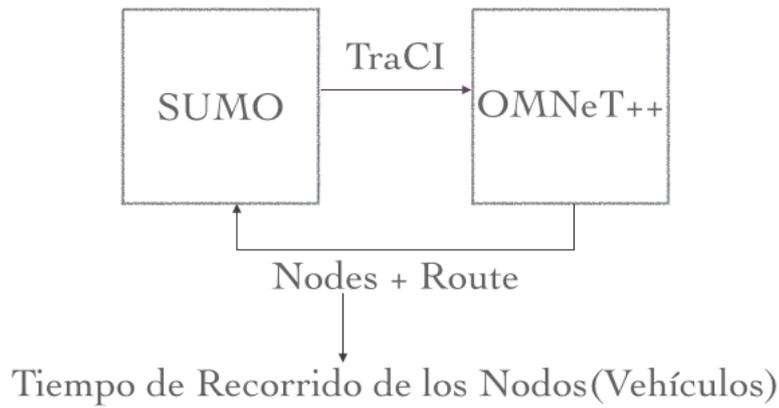


Figura 16: Esquema del tiempo de llegada de Vehículos.

Cuando finaliza la simulación del tráfico para determinado instante de tiempo, SUMO devuelve a OMNeT++ un conjunto de comandos, incluida la información de la posición de los nodos o vehículos. OMNeT++, en base a las trazas de movilidad que recibe, introduce nuevos nodos, y elimina aquellos vehículos que hayan llegado a su destino, y moviendo al resto de nodos de acuerdo a lo que le indica SUMO.

#### *Estadística del tiempo de trayecto*

En nuestro estudio es muy importante poder determinar el tiempo de trayecto de cada vehículo para proceder a su optimización. Para poder obtener una estadística del tiempo de recorrido de los vehículos en una simulación de la ciudad de Valencia se tuvo que modificar ciertos módulos dentro de esta arquitectura. En concreto, se utilizó el esquema representado en la *figura 16*, centrándonos en el módulo TraCIScenarioManagerLaunchd [10], el cual permite crear y mover los nodos controlados por un servidor TraCI.

Esta nueva implementación extrae dos datos mientras está simulado, el primero es el id del vehículo que ha llegado a su destino, y el segundo es el tiempo en realizar dicho recorrido (en milisegundos), ya que nos hemos validado en cierta medida del código ya implementado. No obstante, esta información extraída debe ser suministrada a un archivo, por lo cual se insertó un nuevo método para que pueda escribirlo con una ruta previamente ingresada como parámetro en el archivo de configuración de OMNeT++. Esto lo realiza cada vez que un vehículo termina su recorrido. La información de tiempo dada por la simulación siempre parte de cero, por lo que tenemos que tener en cuenta el tiempo de salida del vehículo, y realizar una diferencia entre los dos para saber el tiempo que le tomó al vehículo dicho recorrido.

### 3.3 SERVIDOR DE RUTAS ABATIS

#### 3.3.1 *Introducción*

ABATIS [15] es un servidor de planificación de rutas que provee recomendaciones de rutas, mediante interacción cliente/servidor. El principal objetivo es ofrecer las mejores rutas posibles, atendiendo al coste en recorrer cada segmento de calle. Como esta varía de forma dinámica, se introduce una mejora a ABATIS que consiste en tener en cuenta las

`http://158.42.55.136:5000/viaroute?loc=la1,lon1&loc=lat2,lon2&atas=true`

Figura 17: Dirección de petición al Servidor ABATIS.

dependencias temporales en el coste del recorrido por las diferentes calles y avenidas de la ciudad de Valencia.

Normalmente los clientes contribuirían para mejorar la información de la base de datos de rutas ya que proveen datos en tiempo real sobre las condiciones de la congestión del tráfico, permitiendo así conocer el comportamiento de una ciudad de manera más exacta. Esto se puede aprovechar también para realizar el balance de carga de tráfico de manera global, y gestionar ciertas necesidades específicas de manera más eficiente, como por ejemplo la reducción de la congestión del tráfico en zonas donde circulen vehículos prioritarios tales como ambulancias..

### 3.3.2 API del Servidor

El propio servidor implementa una conexión por HTTP, la cual sirve la información requerida como si se tratara de un servidor web normal. Para lograr esta funcionalidad se necesita de una dirección IP con su respectivo puerto para poder escuchar las peticiones, así como varios parámetros adicionales. Para el caso de la ciudad de Valencia, consultaremos las rutas agregando a la URL `http` las coordenadas geográficas de la localización del punto de partida y del destino, así como un flag que permite activar los códigos de a diferentes calles de Valencia, denominados ATA, como se muestra en la *figura 17*.

### 3.3.3 Formato de Salida

La respuesta que devuelve ABATIS sigue el formato JSON, el cual contiene información sobre la ruta consultada como se aprecia en la *figura 18* tomada de una consulta para la ciudad de Valencia, y que posee los siguientes parámetros:

- Status: Indica el estado en dos posibilidades, "o" para una transacción existosa o "207" para cuando no existe una ruta.
- Status message: Indica el estado por medio de una cadena de texto relacionada con el parámetro anterior, que puede ser "Found route between points" o "Cannot find route between points".
- Route geometry: por defecto este entrega las rutas de forma codificada con un algoritmo Polilínea codificado, pero este servidor ofrece la capacidad de entregar estas rutas sin ser codificadas agregando a la URL un parametro denominado "compression" y con valor "false".
- Route summary: Es el resumen de la ruta calculada, tanto en el total de la distancia, como el tiempo de trayectoria en segundos, y como forma adicional los nombres de los dos puntos de origen y destino.
- Via points: Son arreglos de los puntos de la via, donde cada uno de ellos es un arreglo de coordenadas geográficas.

*Polyline Algorithm  
Format: Algoritmo  
que codificada las  
coordenadas  
geográficas que  
convierte un valor  
binario en una serie  
de códigos ASCII  
utilizando un  
esquema de  
codificación base 64.  
[18]*

```

{"version": 0.3,
 "status": 0,
 "status_message": "Found route between points",
 "route_geometry": [[39.46900, -0.36417], [39.46900, -0.36445], [39.46874, -0.36427], [39.46805, -0.36384],
 [39.46790, -0.36375], [39.46781, -0.36371], [39.46768, -0.36362], [39.46682, -0.36308], [39.46667, -0.36299],
 [39.46654, -0.36291], [39.46558, -0.36230], [39.46544, -0.36221], [39.46530, -0.36211], [39.46432, -0.36150],
 [39.46414, -0.36142], [39.46408, -0.36161], [39.46371, -0.36250], [39.46323, -0.36369]],
 "route_summary": {
 "total_distance": 844,
 "total_time": 50,
 "start_point": "Gran Via del Marqués del Túria",
 "end_point": "Carrer Mestre Racional",
 "via_points": [[39.46900, -0.36417 ], [39.46323, -0.36369 ]],
 "transactionId": "ABATIS Routing Engine JSON Descriptor (v0.3)"
 }

```

Figura 18: Respuesta de ABATIS.

### 3.3.4 Herramientas ABATIS

#### *osrm\_extract*

Esta herramienta del servidor ABATIS, obtiene información de la ciudad de Valencia en un archivo con formato OpenStreetMap para poder realizar los cálculos de las rutas, y los convierte en archivos en un formato normalizado descrito a continuación:

- Un archivo que contiene los datos para el cálculo de rutas con sufijo `.osrm`.
- El segundo archivo contiene las restricciones de sentido de las calles con sufijo `osrm.restrictions`.
- Un tercer y 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 que tomará como referencia para el cálculo de las rutas como `car.lua`, `foot.lua`, `bicycle.lua`.

#### *osrm\_prepare*

Esta herramienta tiene como parámetro los dos archivos creados mediante la herramienta anterior: `.osrm` y `.osrm.restrictions`, y crea los archivos con cálculos previos como `.osrm.edge`, `.osrm.fileIndex`, `.osrm.hsgr`, `osrm.nodes`, `.osrm.ramIndex`, y `.osrm.timestamp`.

#### *osrm\_routed*

Esta herramienta es la que ejecuta el servidor ABATIS, y su configuración se hace incluyendo en un archivo los parámetros necesarios para inicializarlo, incluyendo la cantidad de hilos que va a usar, la IP del servidor, el puerto de escucha y las rutas de los archivos que contienen información de la ciudad de Valencia, como se puede ver en la [figura 19](#).

## 3.4 INTERFAZ LINKABATIS

### 3.4.1 Situación Actual

Actualmente existe una única interfaz de conexión que permite la comunicación entre el simulador de rutas de tráfico (SUMO) y el simulador de red (OMNeT++). Esto nos permite investigar cuales son

```

Threads = 8
IP = 158.42.55.136
Port = 5000

hsgrData=/home/usuario-grc/ABATIS/ATAS-Valencia.osrm.hsgr
nodesData=/home/usuario-grc/ABATIS/ATAS-Valencia.osrm.nodes
edgesData=/home/usuario-grc/ABATIS/ATAS-Valencia.osrm.edges
ramIndex=/home/usuario-grc/ABATIS/ATAS-Valencia.osrm.ramIndex
fileIndex=/home/usuario-grc/ABATIS/ATAS-Valencia.osrm.fileIndex
namesData=/home/usuario-grc/ABATIS/ATAS-Valencia.osrm.names
timestamp=/home/usuario-grc/ABATIS/ATAS-Valencia.osrm.timestamp

```

Figura 19: Archivo de Configuración de ABATIS.

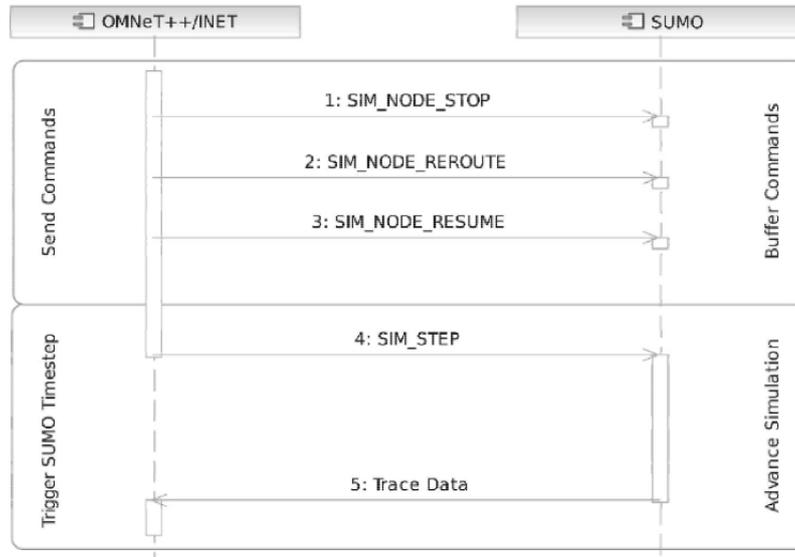


Figura 20: Arquitectura actual.

los diferentes efectos en los movimientos de los vehículos, así como la densidad de población en áreas urbanas. Esto se consigue a través de varios patrones de tráfico que usan como modelos a conductores de la vida real. Esta interfaz es de código abierto, y soluciona la unión entre los dos simuladores: uno de la red de tráfico y un simulador de red, ambos conectados en tiempo real por medio de la interfaz de control de tráfico mencionado con anterioridad (TraCI) la cual da control sobre la movilidad y atributos de cada vehículo simulado. Esta arquitectura, disponible por defecto, se muestra en la figura 20.

### 3.4.2 Objetivos

Uno de los objetivos que se quiere alcanzar al desarrollar esta interfaz es solucionar de la mejor manera posible la congestión presente en algunas zonas de la ciudad, y así poder usar rutas alternativas para balancear el tráfico y lograr una mayor fluidez en las horas pico, ya que se consultaría la ruta en tiempo real al servidor de rutas ABATIS, y éste entregaría las rutas más cortas para su uso tanto en simuladores como en entornos reales.

La estrategia propuesta permitirá mejorar el tiempo de recorrido de los vehículos en sus trayectorias en la ciudad de Valencia, con una reducción significativa tanto en el consumo como en las emisiones de los vehículos.

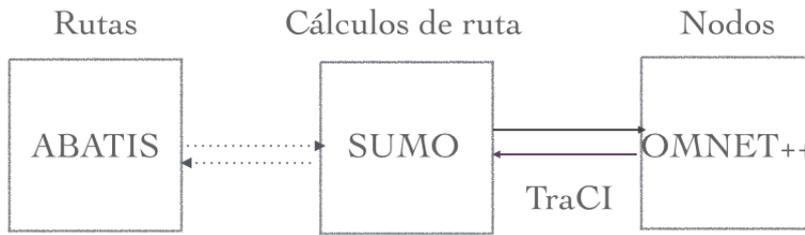


Figura 21: Primera Hipótesis Propuesta.

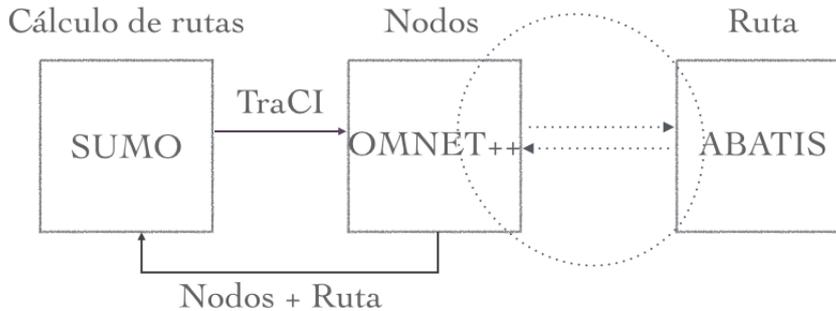


Figura 22: Segunda hipótesis propuesta.

### 3.4.3 Propuestas de arquitecturas

Para poder unificar SUMO y OMNeT++ con ABATIS se propone crear una interfaz la cual proveerá de manera dinámica las diferentes rutas en diferentes condiciones. Esta arquitectura es ilustrada con la propuesta de dos hipótesis que serán presentadas a continuación:

#### Primera hipótesis propuesta

Para esta propuesta OMNeT++ provee los diferentes nodos vehiculares, SUMO gestiona la movilidad de los nodos microscópicos, y ABATIS entrega las rutas a SUMO. ABATIS tendría una conexión directa con SUMO para proveer la demanda de rutas de una manera dinámica, como se puede observar en la *figura 21*. La ventaja de esta propuesta es la comunicación directa entre ABATIS y el simulador de rutas de tráfico, pero el inconveniente que se presenta es la necesidad de modificar el código fuente de SUMO, por lo que, al cambiar algo, se podrían ocasionar algunos errores en los algoritmos de cálculo de rutas.

#### Segunda hipótesis propuesta

La segunda propuesta sigue un planteamiento similar, es decir, TraCI es usado para poder conectar el simulador de tráfico con el simulador de la red. La principal novedad es que ahora la conexión con ABATIS se realiza desde OMNeT++, el cual será responsable de la comunicación entre los sistemas. La ventaja de esta arquitectura es la no manipulación del código de SUMO, el cual solo proveerá información hacia los nodos y las rutas manteniendo OMNeT++ un control completo del sistema, como se puede observar en la *figura 22*.

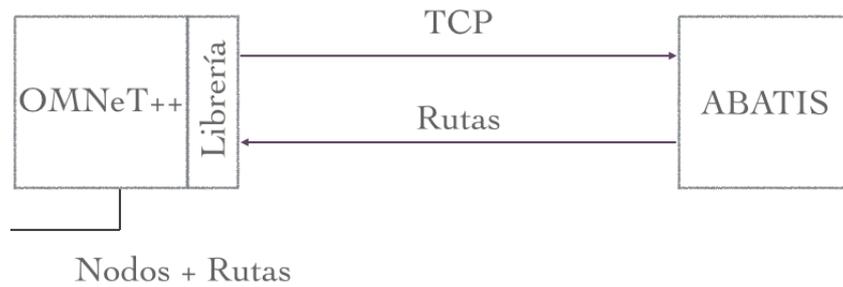


Figura 23: Arquitectura linkABATIS.

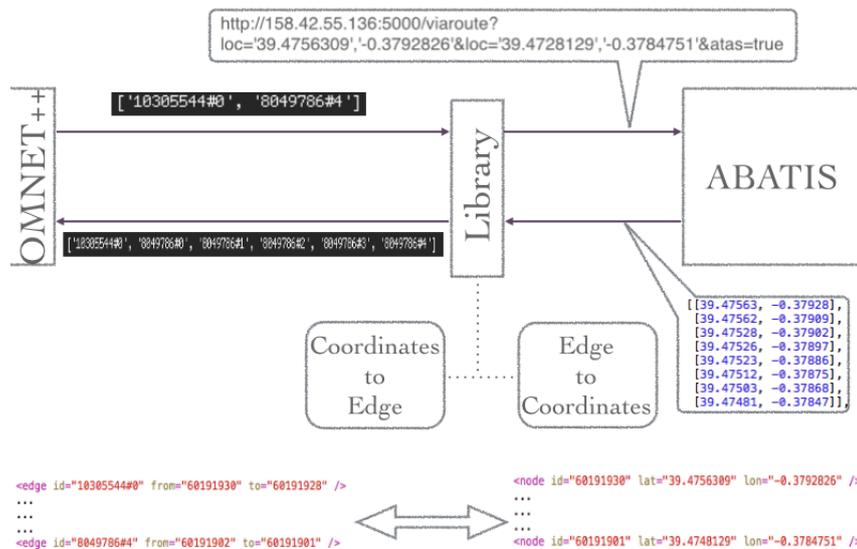


Figura 24: Funcionamiento linkABATIS

### 3.4.4 Estructura de la Interfaz

La figura 23 ilustra la conexión de la nueva interfaz denominada "linkABATIS", la cual debe contener algunos elementos para que sea posible su lectura por parte del sistema. Esta interfaz realiza una petición http del tipo GET, la cual consiste en información básica como la dirección del servidor y el puerto de escucha de ABATIS. Los parámetros que normalmente se incluyen son las coordenadas geográficas del origen y del destino de la ruta. Además, hay que activar el parámetro ATAS para que la plataforma tenga en cuenta el comportamiento específico relativo al estado de circulación del tráfico por las diferentes calles de Valencia.

Como ya es conocido, el servidor ABATIS entrega los puntos que constituyen la ruta propuesta pero en coordenadas geográficas, formato que no es soportado por OMNeT++ o SUMO. Para solucionar esta incompatibilidad es necesario crear un traductor que convierta coordenadas geográficas en códigos compatibles con el simulador de red y de tráfico, así como la conversión inversa para realizar una comunicación bidireccional.

El inconveniente que se presenta es el no tener un archivo en el que consten todos los nodos y edges que están dentro del mapa de Valencia. Otra adversidad que se presenta es la corrección de las rutas que llegan desde ABATIS.

```
<route id="41397537#0 41397537#3" replace="41397537#0 41397537#1 41397537#2 41397537#3"/>
```

Figura 25: Corrección de rutas de ABATIS.

### Creación de edges y nodos

Para solucionar el inconveniente anteriormente presentado, se procede a crear dos archivos en los que el primero solo contendrá información relevante de los nodos, como el identificador que usa OMNeT++ y SUMO para reconocerlos, así como de sus coordenadas geográficas. El segundo archivo contendrá el identificador de los edges conjuntamente con dos parámetros que tendrán estrecha relación con el primer archivo y sus identificadores. Estos archivos nos ayudan a reducir el tiempo de la búsqueda de estos códigos para la ciudad de Valencia.

La interfaz linkABATIS, al no encontrar estos archivos, tendrá la potestad de crearlos empleando como componente principal el mapa de Valencia, el cual será convertido en un archivo que pueda leer SUMO mediante la herramienta NETCONVERTER.

### Conversiones, funcionamiento y correcciones

Al tener los archivos edge y nodos, lo que se procede hacer a continuación es crear un traductor; que convierta las identificaciones de los edges en coordenadas geográficas y viceversa, mediante un análisis de coincidencias de patrones de identificación de los códigos y las coordenadas enviadas por ABATIS. Esto nos garantiza una comunicación bilateral entre los simuladores OMNeT++ y SUMO con ABATIS.

Sin embargo, quien controla este sistema es propiamente OMNeT++, como se demostró anteriormente, ya que desde este simulador de red se procede a ingresar los parámetros de funcionamiento para la interfaz linkABATIS. Para ello se debe agregar los parámetros que se mencionan en la *tabla 1*. Como funcionalidad adicional para permitir diferentes casos de estudio, hay un parámetro extra que da la potestad al usuario de ingresar un porcentaje de vehículos para que estos sean consultados a ABATIS, e incluirlos en el tráfico de Valencia, devolviendo un archivo con las nuevas consultas de rutas.

El funcionamiento del linkABATIS se muestra en la *figura 24* tanto como la petición como la recepción de la información proporcionada por ABATIS.

Al ser diferentes sistemas, ABATIS, como proveedor de rutas, no llega a ser el 100% detallado, ya que en el caso de una vía larga dentro de la ruta, este servidor de rutas optaría por entregar las coordenadas geográficas del inicio de la vía y del final, pero no los puntos intermedios, por lo que es necesario hacer ciertas correcciones para que la información devuelta sea legible desde OMNeT++ y SUMO. Para llevarlo a cabo es necesario tener un archivo que contenga los fragmentos de rutas que se puedan encontrar y los datos complementarios,, como podremos observar en la *figura 25*, y así corregir en el archivo de rutas para su uso por parte de OMNeT++ y SUMO.



## HEURÍSTICA DE AJUSTE DE FLUJOS DE TRÁFICO

---

En este capítulo se describe el procedimiento que se ha seguido para obtener la cantidad de vehículos que van a ser introducidos a la red a partir de la cantidad de vehículos que pasan por los detectores colocados en la ciudad de Valencia, datos que han sido facilitados por el Ayuntamiento de Valencia.

### 4.1 ANÁLISIS DEL TRÁFICO PARA LA CIUDAD DE VALENCIA

Las soluciones de planificación de rutas usan costes fijos para los diferentes segmentos del camino, por lo que seleccionar un origen y destino en particular es un proceso determinista, lo que significa que hay una tendencia a usar las mismas calles o avenidas repetidamente. Sin embargo, el análisis del comportamiento del tráfico en un entorno urbano muestra que usar las principales calles o avenidas en horas pico no suele ser una buena elección ya que el tiempo de recorrido es muy superior al de una situación en que no está congestionadas. Por ello, el cálculo del coste de un trayecto en diferentes momentos del día requiere una granularidad por horas.

Los principales aspectos que afectan al coste de recorrido para un determinado segmento de calles en un entorno urbano son:

- Las condiciones del tráfico serán diferentes para cada calle o avenida, requiriendo modelos de costes distintos para diferentes horas del día.
- Los días laborables son caracterizados por la movilidad de los patrones de congestión de tráfico que difieren del comportamiento en fines de semana y vacaciones.
- Diferencias de temporadas: las personas usan más los vehículos en épocas de frío que en épocas de calor.

Atendiendo a estos factores, y centrándonos en una ciudad europea de tamaño medio como Valencia en la que se miden diariamente comportamientos del tráfico en más de 500 calles-avenidas, esto daría lugar a más de 180.000 funciones de interpolación para describir la variación del tráfico en cada día. Esta información es proporcionada por el Ayuntamiento de Valencia.

Para tener una idea del flujo de tráfico, comenzaremos analizando el tráfico mensual como se muestra en la *figura 26*.

Para nuestro estudio se elige el mes de Noviembre porque no tiene períodos de vacaciones, y tiene valores relativamente similares a los otros meses. Centrándonos en la variación del patrón del tráfico durante la semana, la *figura 27* muestra que existen significativas diferencias entre los días laborales y los fines de semana. El patrón de tráfico obtenido de Lunes a Viernes es de 37.000 vehículos, y en los fines de semana es alrededor de 26.000 vehículos.

Debido a estas diferencias, en cuanto a volumen de tráfico, se escogerá un Lunes para nuestro análisis, el cual, como se muestra en la *figura 28*, tiene la hora pico entre las 8h00 y las 9h00 cuando las personas van

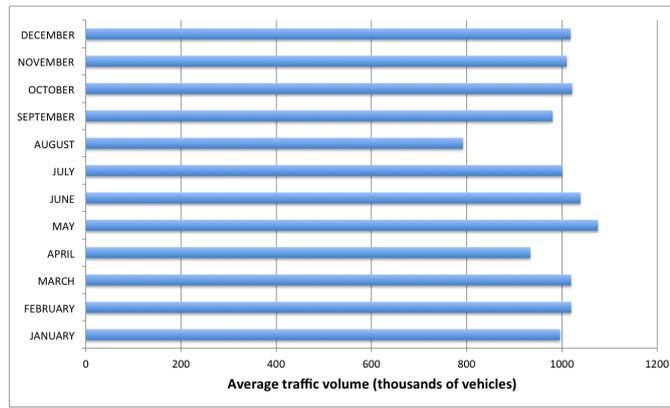


Figura 26: Promedio de volumen de tráfico en Valencia por mes[15].

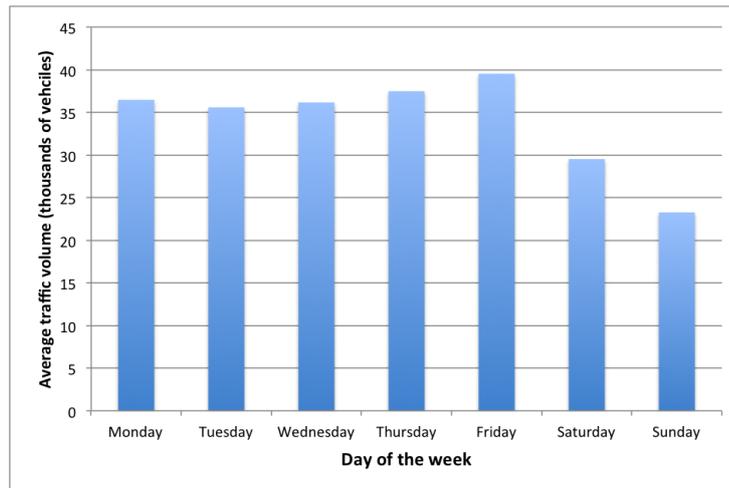


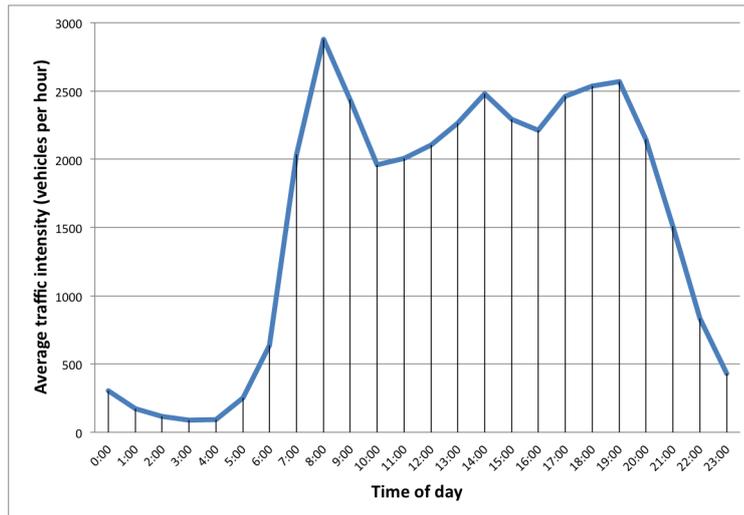
Figura 27: Promedio del volumen del tráfico en Valencia en una semana.

al trabajo. Otro pico está entre las 14h00 y 15h00 cuando las personas salen a comer y vuelven a su trabajo por la tarde, y la última hora pico es entre las 18h00 y 20h00 cuando los trabajadores vuelven a sus casas. Para nuestro caso escogeremos el pico más alto del Lunes.

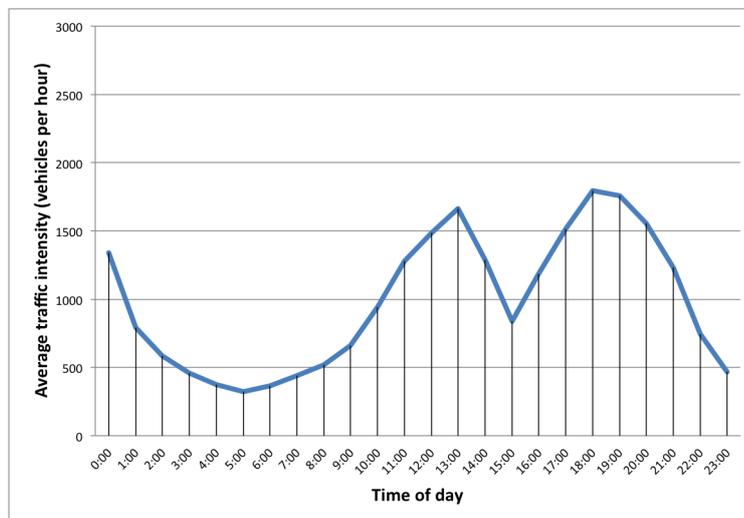
#### 4.2 PROCESOS DE AJUSTE

El dato de entrada para la simulación debe ser la cantidad de vehículos que se van a inyectar a la red. Al no tener este dato fundamental, se desarrolló una técnica de ajuste para intentar que nuestro entorno de simulación incluya la cantidad de vehículos que sea lo más realista posible. La información facilitada por el Ayuntamiento de Valencia, respecto al tráfico vehicular para la hora pico de un determinado día, corresponde a datos obtenidos mediante detectores que han sido ubicados en la mayoría de vías de la ciudad (las más relevantes en cuanto a tráfico), y los mismos nos servirán para los cálculos de los datos de entrada.

Para realizar el ajuste deseado se han probado dos técnicas alternativas: la primera es un ajuste global para todos los ATAs, y la segunda técnica es un ajuste por cada grupo de códigos ATA.



(a) Lunes



(b) Domingo

Figura 28: Comportamiento promedio diario para diferentes días de la semana[15].

En el proceso se debe tener un archivo que tendrá de forma resumida la información relativa a los códigos ATA los cuales, como se ilustra en la figura 29, están compuestos por:

- Códigos ATA.
- Códigos de las vías que incluyen el código ATA.
- Cantidad de vehículos que pasan por ese código ATA.
- Cantidad menor de vehículos que pasan por los códigos ATA.
- Volumen de vehículos a insertar en la simulación.

```

<ATA id="A120/A121/A321/A322">
  <way id="4693573"/>
  <way id="9281513"/>
  <way id="98711532"/>
  <way id="98711534"/>
  <way id="98711537"/>
  <way id="98711543"/>
  <way id="98711572"/>
  <way id="98711574"/>
  <ATAsub id="A120" vehicle="6464"/>
  <ATAsub id="A121" vehicle="10898"/>
  <ATAsub id="A321" vehicle="4221"/>
  <ATAsub id="A322" vehicle="6677"/>
  <ataMenor quantity="4221"/>
  <volume number="237"/>
</ATA>

```

Figura 29: Extracto de información de los Códigos ATA en Valencia.

#### 4.2.1 Cálculo del Ajuste

Para realizar el ajuste, es necesario hay que hacer diversos cálculos; tomaremos como guía el ejemplo de la figura 29, y aplicaremos la siguiente fórmula:

$$\text{reduction} \approx \sum_{i=1}^n \frac{(\text{ataMenor}/\#\text{way})}{\#\text{vehicleCount}_i}$$

donde:

ataMenor: menor cantidad de vehículos de los códigos ATA.

#way: Número de vías que contienen el código ATA.

#vehicleCount: cantidad de vehículos contadas en las vías.

El porcentaje obtenido es un valor de referencia para el factor de ajuste, y así no tener un desfase alto.

Por último aplicaremos la siguiente fórmula para saber la cantidad de vehículos que van a ser inyectados en la red:

$$n\text{Car} = \left( \frac{\text{ataMenor}}{\#\text{way}} \right) \cdot \text{reduction}$$

donde:

ataMenor: menor cantidad de vehículos de los códigos ATA.

#way: Número de vías que contienen ese código ATA.

reduction: Porcentaje de ajuste a la cantidad de vehículos.

El resultado de este cálculo es colocado en la cláusula "volumen" asociado a un determinado ATA, y que se incluye en el fichero de entrada a la herramienta DFROUTER, como se ilustra en la figura 29. Esta operación es realizada para cada detector que va a consumir SUMO. El cálculo se lo realiza reiteradamente hasta conseguir flujos de vehículos similares a los datos reales.

### 4.3 RESULTADOS

La figura 30, muestran los resultados obtenidos mediante la primera técnica utilizada para el Ajuste de Vehículos en Valencia. Para el análisis se han utilizado varios tipos de ajustes con el fin de ver cual es el que más se aproxima a los datos reales otorgados por el Ayuntamiento de Valencia. Como se puede observar, el ajuste del 45% logra un comportamiento similar al deseado en cuanto a la cantidad de vehículos que pasan por los detectores.

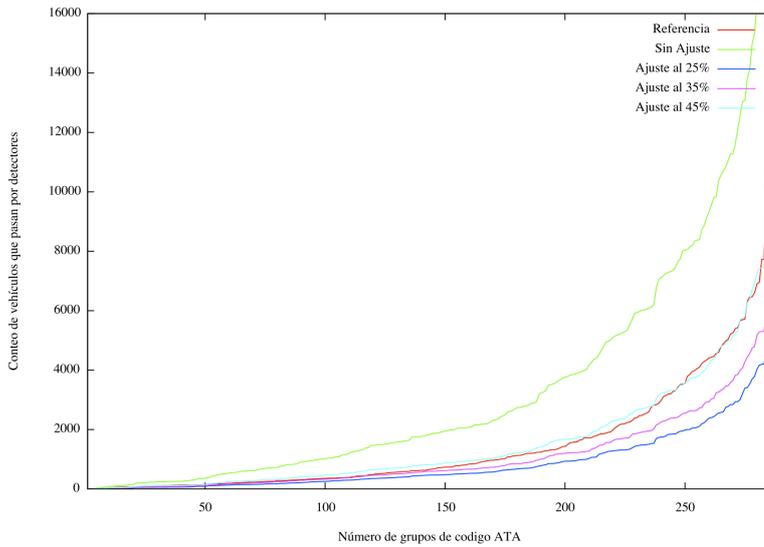


Figura 30: Ajustes de Vehículos en Valencia.

La segunda técnica utilizada se basa en ajustes que discriminan los códigos ATA. En la figura 31 se muestra que el recuento de vehículos es superior a los datos reales otorgados por el Ayuntamiento de Valencia, demostrando estar alejado de la realidad en cuanto a la cantidad de vehículos medidos, por lo que se ha descartado esta opción.

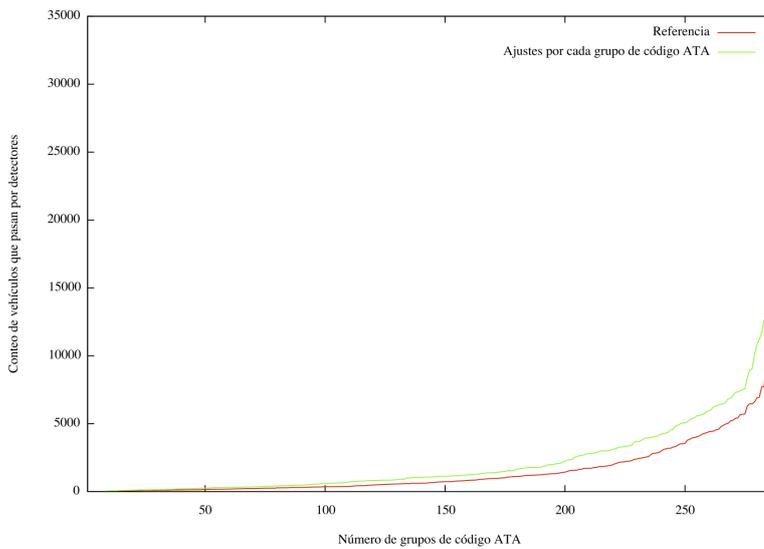


Figura 31: Diferentes ajuste de vehículos en Valencia.



En este capítulo se detallan las pruebas y simulaciones que se realizaron para la ciudad de Valencia. El principal propósito es averiguar el impacto de usar un sistema de asignación de rutas que altere el recorrido que sigue la mayor parte de vehículos, determinando las variaciones experimentadas por los vehículos que mantienen la ruta original, y aquellos que siguen la nueva ruta, en términos de tiempo de recorrido. Con este propósito se han desarrollado diversas herramientas que permiten variar el porcentaje de vehículos que realiza una consulta de ruta a ABATIS. Los resultados se dividieron en cuatro grupos:

- Distribución de probabilidad ABATIS vs DFROUTER usada como referencia, la cual corresponde al uso de cada una de estas alternativas por el 100 % de los vehículos en el experimento.
- Estrategia estándar, en la que un porcentaje variable de los vehículos (entre el 1 y el 30 %) obtiene una ruta alternativa desde ABATIS, en contraste con los demás vehículos, los cuales obtienen su ruta desde DFROUTER.
- Estrategia inversa, en la que un porcentaje variable de los vehículos (entre el 1 y el 30 %) obtiene una ruta alternativa desde DFROUTER, en contraste con los demás vehículos, los cuales obtienen su ruta desde ABATIS.
- Comparativa ABATIS vs DFROUTER en términos de tiempo medio de recorrido por parte de los vehículos implicados.

### 5.1 DESCRIPCIÓN DE LOS PARÁMETROS DE SIMULACIÓN

Las simulaciones hacen uso de la interfaz linkABATIS, comunicando el servidor de rutas, con el simulador de red, y éste a su vez con el servidor de tráfico. A continuación se detallan los parámetros más relevantes de los experimentos realizados:

- Tiempo de 15 minutos con 14237 vehículos.
- Elección aleatoria de los vehículos a los que se asigna una ruta alternativa.
- Definición del porcentaje de vehículos a consultar a ABATIS.
- Porcentaje restante con la herramienta DFROUTER.

Con respecto a los porcentajes, estos fueron de 1 %, 5 %, 10 %, 20 % y 30 % de los vehículos escogidos aleatoriamente, y para los cuales ABATIS les concede una nueva ruta teniendo el origen y destino originales.

Como se puede apreciar en la *figura 32*, el simulador OMNeT++ es el elemento central de la simulación, y registra el porcentaje de vehículos definido por el usuario que realiza la consulta de las nuevas rutas a ABATIS. El proceso es idéntico al realizado anteriormente, es decir, se comunica con ABATIS vía TCP, y este servidor de rutas responde a dichas peticiones con rutas en forma de coordenadas, las

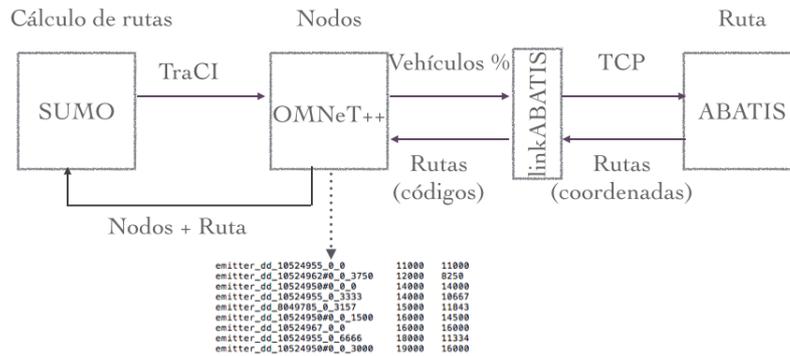


Figura 32: Funcionamiento de la Simulación con Porcentaje variables de vehículos que consulta su ruta a ABATIS.

cuales linkABATIS, convierte en un formato que OMNeT++ es capaz de entender, y continuar así su proceso. El resultado de esta simulación es un archivo de texto plano que contiene tres columnas: el identificador del vehículo que acabó su recorrido, el tiempo de la simulación, y el instante en que el vehículo terminó su ruta.

## 5.2 RESULTADOS

### 5.2.1 Distribución de probabilidad ABATIS vs DFROUTER

Como punto de partida se compara la situación de obtener el 100 % de las rutas mediante a ABATIS contra la alternativa de obtener el 100 % de las rutas mediante DFROUTER (ver figura 33). Nos podemos percatar que los resultados son en general muy igualados como sería deseable, siendo que ABATIS supera con muy poca diferencia a DFROUTER en los tiempos de llegada de los vehículos.

### 5.2.2 Estrategia estándar

Los resultados obtenidos a continuación corresponden a simulaciones con un tiempo de 15 minutos y con un 1 %, 5 %, 10 %, 20 %, o 30 % de los vehículos consultando sus rutas hacia ABATIS, siendo que los porcentajes complementarios corresponden a vehículos cuyas rutas se obtienen previamente con la herramienta DFROUTER.

En el caso correspondiente al 1 % (ver figura 34) podemos observar que la distribución de probabilidad de ABATIS es algo errática ya que la distribución de resultados no guarda similitud con los resultados obtenidos mediante DFROUTER relativos a los demás 99 %. Para estos últimos, los tiempos mejoran ligeramente en algunos casos, como sería deseable.

Si aumentamos el porcentaje de nodos que obtienen rutas alternativas mediante ABATIS al 5 % (ver figura 35), se puede apreciar claramente que este servidor entrega rutas sin poder mejorar a las ya obtenidas por DFROUTER (demás 95 % de los casos). Estos últimos sí mejoran ligeramente sus prestaciones en general.

En la figura 36 se pueden apreciar los tiempos de recorrido de los vehículos para un escenario donde el 10 % de los vehículos obtiene sus rutas mediante ABATIS. Se puede apreciar que ocurre algo similar a la situación anterior, donde los tiempos asociados a las rutas obtenidas

mediante ABATIS son superiores, y de nuevo los demás 90 % de los vehículos logran mejorar sus tiempos de trayecto.

Para porcentajes de uso de ABATIS superiores (20 % y 30 %), las *figuras 37 y 38* muestran que la tendencia general se acentúa. De hecho, para un 30 % de vehículos que usan rutas alternativas, se ve que hay una mejora clara de los tiempos de recorrido para los vehículos que siguen la ruta principal.

### 5.2.3 Estrategia opuesta

Se analizó una estrategia inversa donde se asigna un porcentaje de rutas entre el 1 y el 30 % a DFROUTER, siendo las demás rutas (referencia) dadas por ABATIS, y se ha analizado como anteriormente los comportamientos en las distribuciones de probabilidad.

Los resultados más destacables de este conjunto de experimentos se obtienen cuando únicamente un porcentaje muy reducido de vehículos (el 1 %) usa una ruta alternativa usando DFROUTER, y la mayoría sigue una ruta estándar que provee ABATIS. En este caso, como podemos ver en la *figura 39*, los tiempos relativos a este 1 % de los vehículos es bastante mejor que la referencia en la mayor parte de casos, y los restantes 99 % no sufren alteraciones apreciables en sus tiempos de recorrido.

Cuando el 95 % de los vehículos consulta su ruta a ABATIS, y el restante 5 % lo hacen usando DFROUTER, se observa un comportamiento similar a la frecuencia normal del 100 %, como se aprecia en la *figura 40*.

Si un 90 % de los vehículos consulta su ruta a ABATIS, tienden a mejorar y a superar en cierto grado al 10 % que utiliza DFROUTER, pero no mejorando en gran medida a la curva correspondiente al 100 % (referencia), como se demuestra en la *figura 41*.

Si el 80 % de los vehículos consulta a ABATIS su ruta, éstos tienden a igualar en ciertos momentos a la curva del 100 % (Referencia) pero con respecto al porcentaje restante con DFROUTER el resultado es ligeramente inferior su curva, como podemos ver en la *figura 42*.

Por último, si un 70 % de vehículos consulta a ABATIS, se verifica una mejora en los tiempos de recorrido comparativamente con el 30 % que obtiene la ruta mediante DFROUTER, pero ninguno de los dos supera la curva correspondiente a tener el 100 % del tráfico usando un único servidor de rutas, como vemos en la *figura 43*.

### 5.2.4 Media de Tiempos ABATIS vs DFROUTER

La *figura 44* muestra el tiempo medio de llegada de los vehículos para cada uno de los porcentajes que fueron consultados a ABATIS. Se observa que, a medida que se incrementa el porcentaje de vehículos que consultan su ruta a ABATIS, hay una tendencia a disminuir su tiempo de llegada al destino hasta un cierto límite (para 30 % incrementa el tiempo ligeramente), lo que muestra que ABATIS es capaz de reducir el tiempo de llegada al destino de los vehículos frente a DFROUTER. Además, se verifica que los tiempos de trayecto logran ser reducidos de forma más sustancial cuando el porcentaje de vehículos que sigue rutas alternativas es bajo (alrededor del 1 %), siendo menos destacable para porcentajes superiores al 5 %.

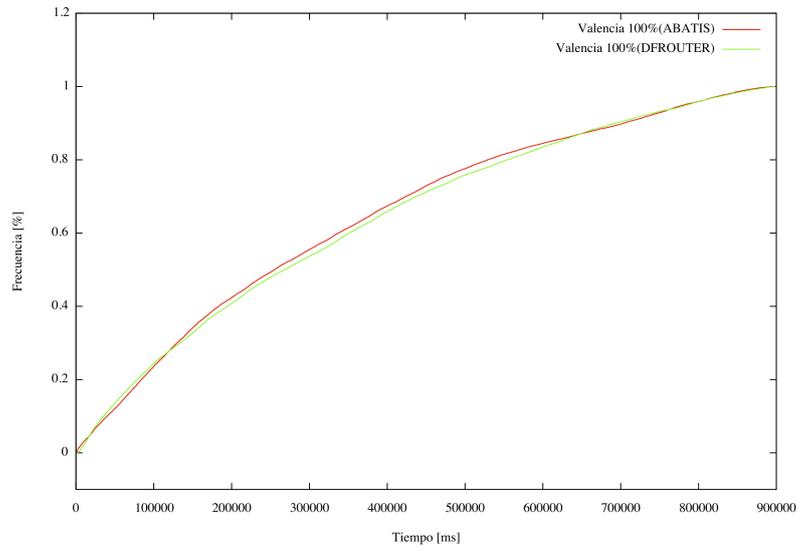


Figura 33: Distribución de probabilidad para el tiempo de llegada ABATIS y DFROUTER

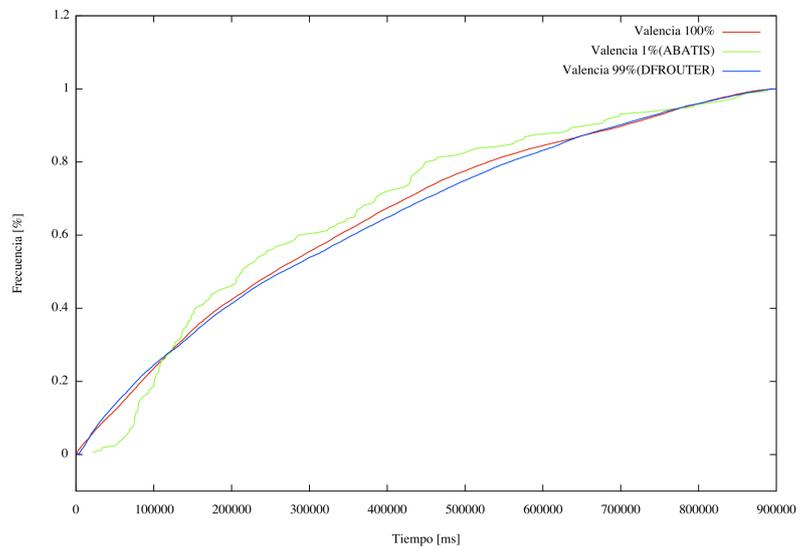


Figura 34: Distribución de probabilidad para el tiempo de llegada para un reparto 1-99 %.

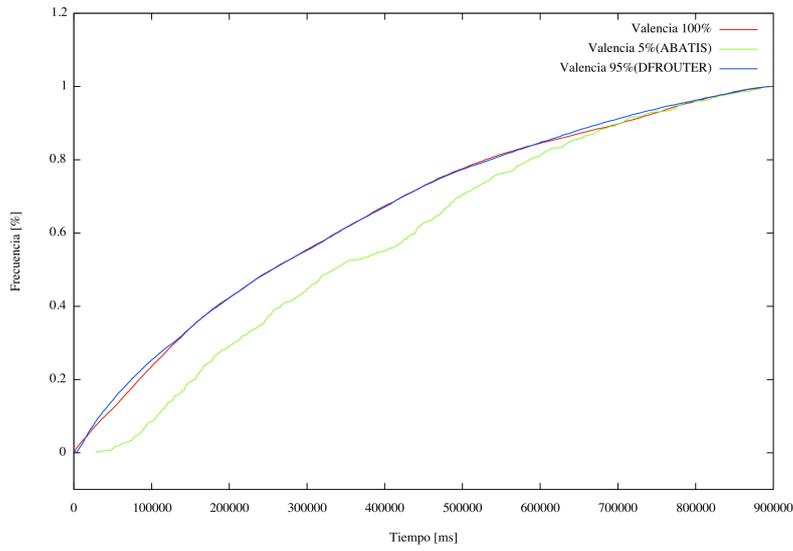


Figura 35: Distribución de probabilidad para el tiempo de llegada para un reparto 5-95 %.

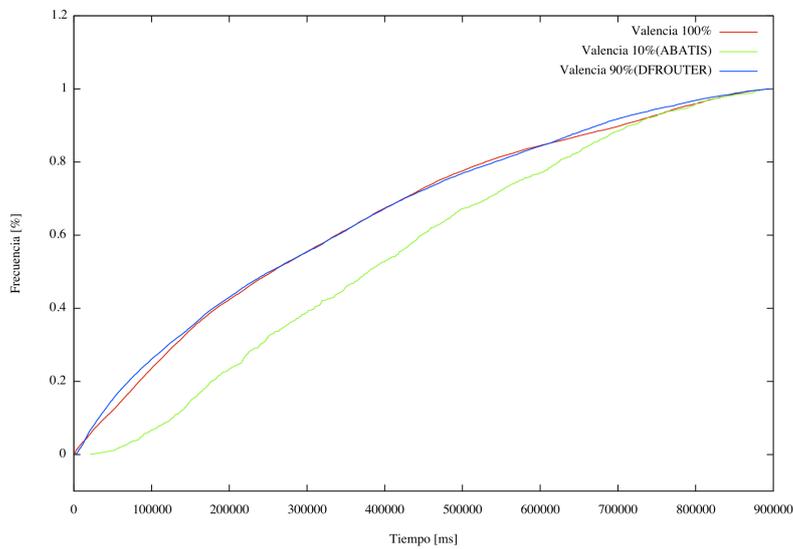


Figura 36: Distribución de probabilidad para el tiempo de llegada para un reparto 10-90 %.

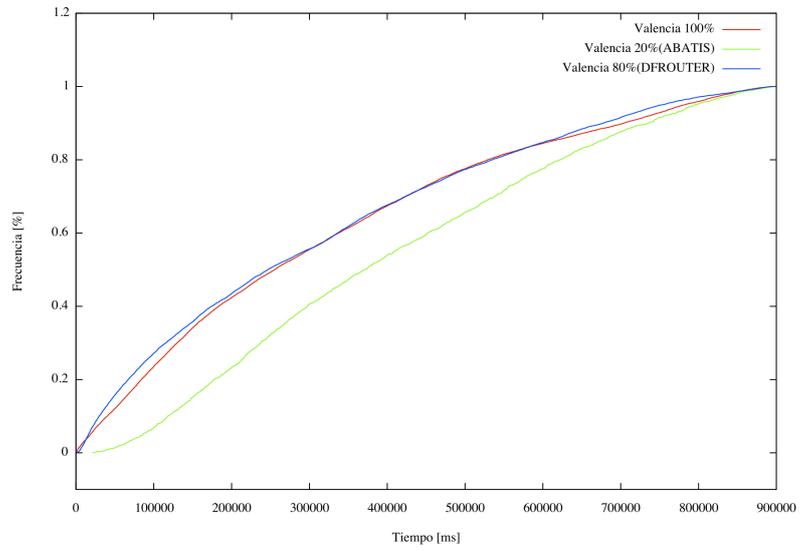


Figura 37: Distribución de probabilidad para el tiempo de llegada para un reparto 20-80 %.

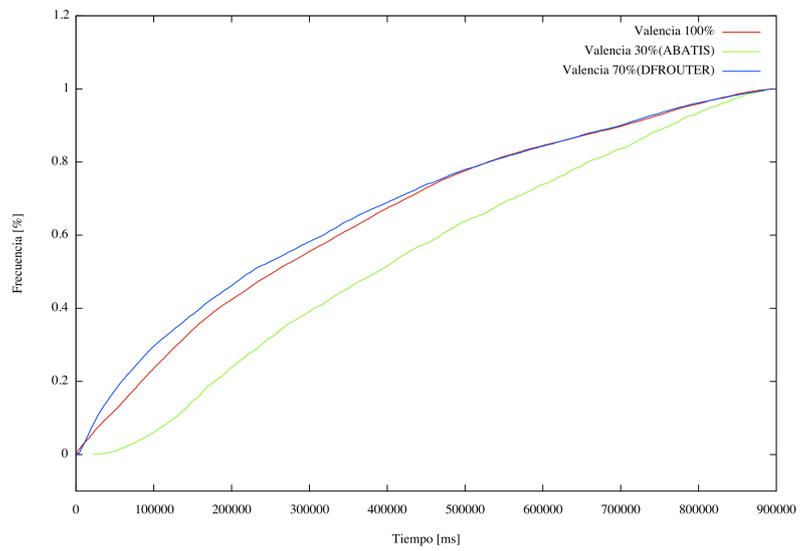


Figura 38: Distribución de probabilidad para el tiempo de llegada para un reparto 30-70 %.

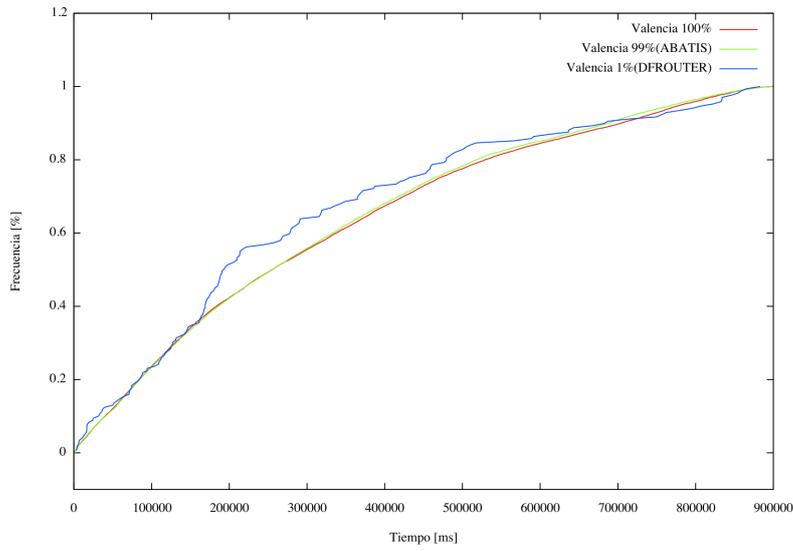


Figura 39: Distribución de probabilidad para el tiempo de llegada para un reparto 99-1 %.

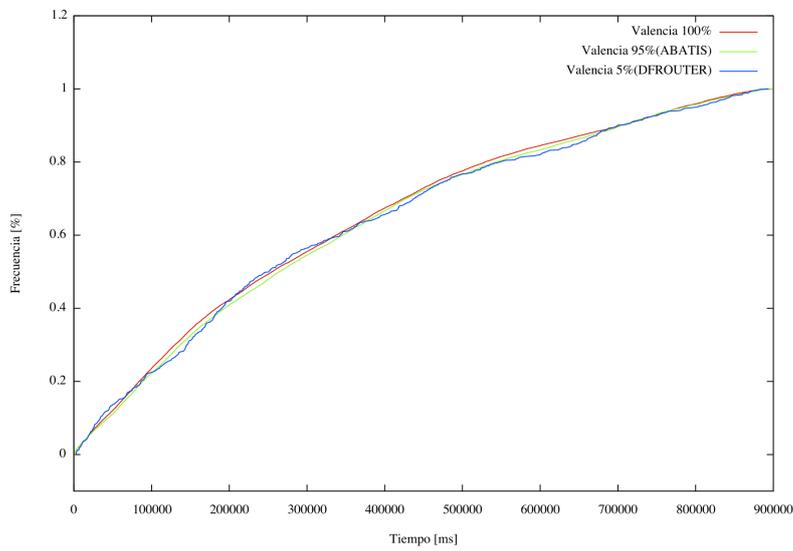


Figura 40: Distribución de probabilidad para el tiempo de llegada para un reparto 95-5 %.

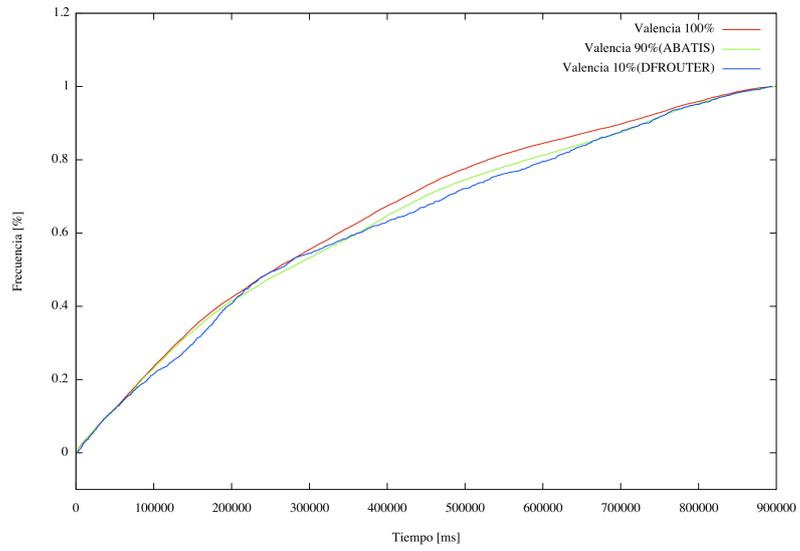


Figura 41: Distribución de probabilidad para el tiempo de llegada para un reparto 90-10 %.

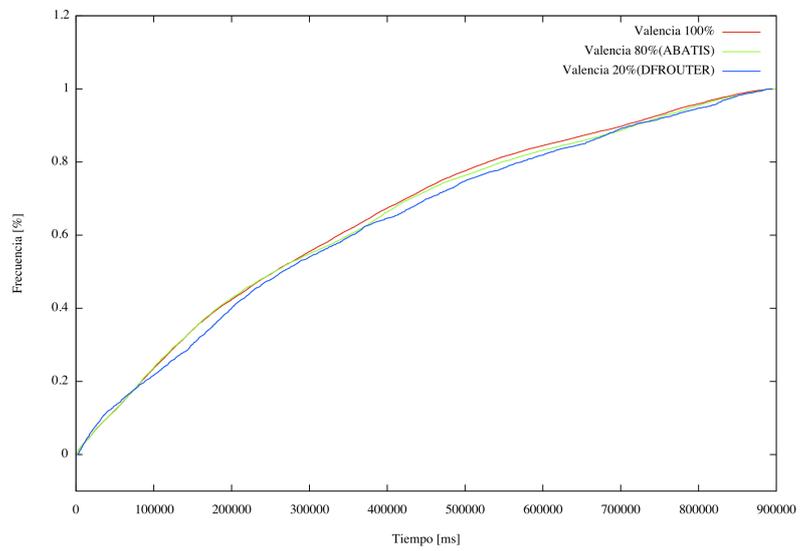


Figura 42: Distribución de probabilidad para el tiempo de llegada para un reparto 80-20 %.

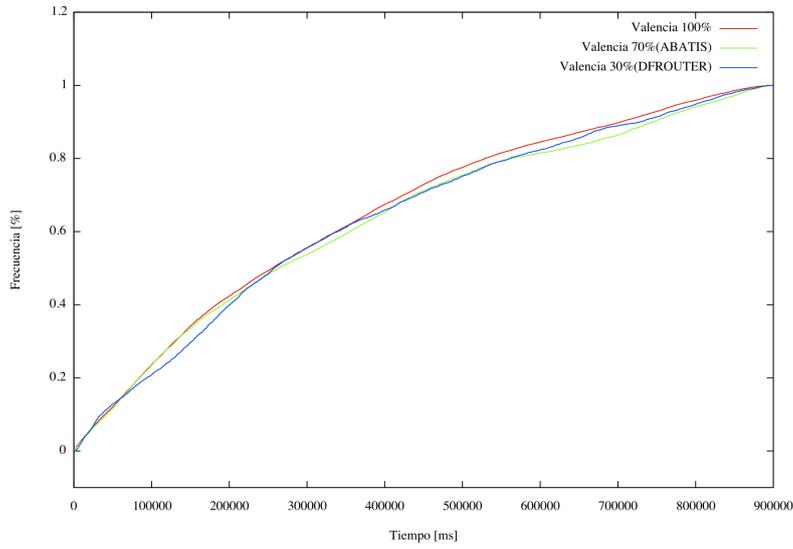


Figura 43: Distribución de probabilidad para el tiempo de llegada para un reparto 70-30 %.

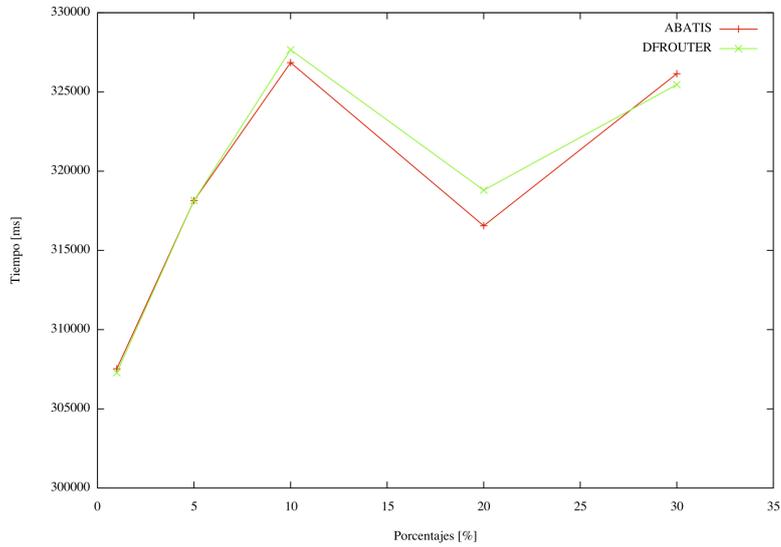


Figura 44: Media de tiempo de llegada de los Vehículos ABATIS y DFROUTER.



## CONCLUSIONES

---

En este trabajo, se empezó por analizar los diferentes módulos generadores de rutas del simulador de tráfico SUMO, determinando el que mejor se ha podido adaptar a la información sobre flujos de tráfico que nos proporcionó el Ayuntamiento de Valencia. Se verificó que la herramienta DFROUTER, la cual utiliza información basada en recuento de vehículos en diferentes puntos de las vías mediante anillos de inducción, era la más adecuada para nuestro propósito.

Una vez que la herramienta DFROUTER no es capaz de ajustar el volumen de tráfico a los datos de entrada, se han propuesto dos heurísticas para ajustar la cantidad de vehículos inyectados al sistema, para lo cual se han propuesto se elaboró dos ecuaciones que permiten ajustar mejor la cantidad de vehículos al flujo de vehículos en Valencia. Aplicando estas ecuaciones, es posible ajustar el volumen de tráfico para cualquier ciudad teniendo en cuenta el tráfico medido en anillos de inducción en las calles de la ciudad como dato principal.

Por otra parte, para obtener el tiempo de recorrido que realizan los vehículos al culminar un trayecto en la simulación, se agregaron ciertas funciones al framework INET, que es parte del simulador de red OMNeT++, lo que permite averiguar el impacto de diferentes sistemas de recomendación de rutas.

La vinculación de OMNeT++ y SUMO con ABATIS se ha logrado mediante una nueva interfaz llamada linkABATIS, la cual posee funciones que permiten comunicación bilateral, teniendo como eje central el simulador de red.

Resultados experimentales preliminares muestran que la interfaz propuesta permiten variar fácilmente el porcentaje de vehículos que utiliza un sistema de recomendación de rutas u otra. Además, se ha verificado que, cuanto más alto es porcentaje de consulta al servidor de rutas ABATIS, más igualdad existe entre los vehículos que siguen las rutas de un proveedor de rutas u otro. Analizaremos la media de los tiempos de recorrido comparando ABATIS vs DFROUTER, verificamos que realizando las consultas mediante ABATIS es posible reducir significativamente los tiempos de recorrido.

Como trabajo futuro se realizarán más pruebas y simulaciones para distintas horas del día, en diferentes épocas del año, para poder comprobar la efectividad de la interfaz y poder analizar el comportamiento del tráfico en Valencia con un mayor grado de detalle, así como proponer soluciones que permitan distribuir el tráfico de manera más eficiente.



## BIBLIOGRAFÍA

---

- [1] Activitygen, 03 2015. URL <http://sumo.dlr.de/wiki/ACTIVITYGEN>. (Citado en la pagina 12.)
- [2] Dfrouter, 03 2015. URL <http://sumo.dlr.de/wiki/DFROUTER>. (Citado en la pagina 13.)
- [3] Demand/importing o/d matrices, 03 2015. URL [http://sumo.dlr.de/wiki/Demand/Importing\\_0/D\\_Matrices](http://sumo.dlr.de/wiki/Demand/Importing_0/D_Matrices). (Citado en la pagina 12.)
- [4] Inet, 03 2015. URL <http://inet.omnetpp.org>. (Cited en las paginas 7 y 17.)
- [5] Od2trips, 03 2015. URL <http://sumo.dlr.de/wiki/OD2TRIPS>. (Citado en la pagina 12.)
- [6] Omnet++, 03 2015. URL <http://omnetpp.org>. (Cited en las paginas 7 y 15.)
- [7] Open street map, 03 2015. URL <http://www.openstreetmap.org/>. (Citado en la pagina 6.)
- [8] Realistic vehicular traces, 03 2015. URL <http://www.lst.inf.ethz.ch/research/ad-hoc/car-traces/>. (Citado en la pagina 5.)
- [9] Simulation of urban mobility, 03 2015. URL [http://sumo.dlr.de/wiki/Main\\_Page](http://sumo.dlr.de/wiki/Main_Page). (Cited en las paginas 6 y 10.)
- [10] Traciscenariomanagerlaunchd, 03 2015. URL [http://omnetpp.org/doc/inet/api-current/doxy/class\\_tra\\_c\\_i\\_scenario\\_manager\\_launchd.html](http://omnetpp.org/doc/inet/api-current/doxy/class_tra_c_i_scenario_manager_launchd.html). (Citado en la pagina 19.)
- [11] Visum-vision-vissim, 03 2015. URL <http://vision-traffic.ptvgroup.com/es/productos/ptv-vissim/>. (Citado en la pagina 12.)
- [12] Duarouter, 03 2015. URL <http://sumo.dlr.de/wiki/DUAROUTER>. (Citado en la pagina 12.)
- [13] Netconvert, 03 2015. URL <http://sumo.dlr.de/wiki/NETCONVERT>. (Citado en la pagina 10.)
- [14] Simulation modelling applied to road transport, 03 2015. URL <http://www.its.leeds.ac.uk/projects/smartest/>. (Citado en la pagina 9.)
- [15] Carlos T Calafate, David Soler, Juan-Carlos Cano, and Pietro Manzoni. Towards a centralized route planning solution supporting time dependencies. In *The International Industrial Information Systems Conference*, pages 222–227, 2014. (Cited en las paginas vi, 1, 19, 28, , 29.)
- [16] Peter Wagner Daniel Krajzewicz, Georg Hertkorn and Christian Rösel. Sumo (simulation of urban mobility), an open-source traffic simulation. *German Aerospace Centre (Berlin, Alemania) y Centre of Applied Informatics (Colonia, Alemania)*, 2002. (Cited en las paginas 9 y 10.)

- [17] Nathan Eagle and Alex (Sandy) Pentland. Reality mining: Sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268, mar 2006. (Citado en la pagina 5.)
- [18] Google. Encoded polyline algorithm format, 03 2015. URL <https://developers.google.com/maps/documentation/utilities/polylinealgorithm>. (Citado en la pagina 20.)
- [19] J. Harri, F. Filali, and C. Bonnet. Mobility models for vehicular ad hoc networks: a survey and taxonomy. *Communications Surveys Tutorials, IEEE*, 11(4):19–41, Fourth 2009. (Citado en la pagina 4.)
- [20] Amit Jardosh, Elizabeth M. Belding-Royer, Kevin C. Almeroth, and Subhash Suri. Towards realistic mobility models for mobile ad hoc networks. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, number 13 in *MobiCom '03*, pages 217–229, New York, NY, USA, 2003. ACM. (Citado en la pagina 4.)
- [21] S Krauss. *Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics*. PhD thesis, Koeln Univ. (Germany). Mathematisch-Naturwissenschaftliche Fakultät, 1988. (Citado en la pagina 6.)
- [22] Piet H. L. Bovy Serge P. Hoogendoorn. State-of-the-art of vehicular traffic flow modelling. *Systems and Control Engineering*, 215(4):1–46, 2001. (Citado en la pagina 3.)
- [23] Andras Varga. The omnet++ discrete event simulation system. Master's thesis, Department of Telecommunications Budapest University of Technology and Economics, 2001. (Citado en la pagina 15.)
- [24] Axel Wegener, Michal Piorkowski, Maxim Raya, Horst Hellbruck, Stefan Fischer, and Jean-Pierre Hubaux. Traci: An interface for coupling road traffic and network simulators. *Communications and Networking Simulation Symposium*, (ISBN 1-56555-318-7):9, 2008. (Citado en la pagina 18.)