

## **PLATAFORMA MODULAR PARA LA INTEGRACIÓN DE HERRAMIENTAS DE SIMULACIÓN DE COMUNICACIONES INALÁMBRICAS Y MOVILIDAD URBANA EN LA EVALUACIÓN DE SISTEMAS COOPERATIVOS ITS A GRAN ESCALA**

### **Julen Maneros**

Ingeniero Informático. Analista de Software. CBT Multimedia.

### **Ramón Bauzá**

Ingeniero de Telecomunicación. Investigador contratado. Laboratorio Uwicore, Universidad Miguel Hernández de Elche.

### **Michele Rondinone**

Ingeniero de Telecomunicación. Investigador contratado. Laboratorio Uwicore, Universidad Miguel Hernández de Elche.

### **Javier Gozávez**

Doctor en Comunicaciones Móviles. Ingeniero Electrónico, especialidad Telecomunicaciones. Profesor Contratado Doctor. Director del laboratorio Uwicore. Universidad Miguel Hernández

### **Oscar Lázaro**

Doctor de Telecomunicación. Director General de la Asociación Innovalia. Asociación Innovalia.

### **Ainara González**

Ingeniera de Telecomunicación. Investigadora unidad TIC. Asociación Innovalia.

**RESUMEN:** La plataforma iTETRIS combina e integra dos herramientas, una dedicada a la simulación de tráfico y otra destinada a la simulación de redes inalámbricas, lo cual permite analizar de forma realista y escalable el impacto de la utilización de tecnologías cooperativas de comunicaciones V2V y V2I en la gestión del tráfico. El presente artículo presenta la plataforma iTETRIS, y el proceso que ha llevado a su arquitectura final. Seguidamente discute cómo se realiza la conexión entre los componentes que permiten una simulación eficiente y precisa de las soluciones ITS cooperativas a gran escala. Finalmente, muestra el proceso de simulación que integra los beneficios de ambos simuladores con un alto grado de rendimiento. El diseño de iTETRIS habilita la evaluación de sistemas cooperativos ITS

basando su implementación en la arquitectura europea de comunicaciones definida por ETSI.

## 1 INTRODUCCIÓN

Una de las cuestiones que las autoridades de tráfico se plantean hoy en día es cómo estimar la eficiencia y solvencia de las inversiones en ingeniería de sistemas de comunicaciones cooperativos para mejorar la gestión de tráfico. El proyecto iTETRIS (<http://ict-itetris.eu/>) tiene como objetivo proveer de una plataforma de simulación capaz de evaluar estrategias de tráfico basadas en sistemas cooperativos ITS a través del intercambio dinámico de mensajes entre vehículos (V2V) y vehículo e infraestructura (V2I) [1].

iTETRIS no pretende únicamente obtener estimaciones preliminares sobre el rendimiento y operación de las estrategias de tráfico cooperativas, sino evaluarlas en un contexto realista, en entornos a gran escala, involucrando un número de vehículos significativo y considerando áreas a nivel de ciudad. Todo ello simulando periodos prolongados de actuación de la red de tráfico. En este sentido, la plataforma es capaz de ofrecer resultados fiables cubriendo el vacío existente entre las pruebas de campo (FOT, “Field Operational Tests”) y los modelos basados en análisis teóricos. Así, iTETRIS sirve como herramienta de apoyo con valor añadido para que las autoridades de tráfico puedan estimar la eficiencia de estrategias ITS cooperativas de gestión de tráfico antes de invertir dinero en un despliegue físico de las tecnologías que las soportan.

Con el objetivo de identificar las herramientas software más apropiadas para desarrollar la plataforma iTETRIS, en el marco del proyecto se ha realizado una labor de investigación buscando las mejores alternativas dentro de la ingeniería del software. De este modo, se pretende alcanzar un diseño óptimo en los distintos componentes de la plataforma y una comunicación lo más eficiente posible para reducir el impacto en el rendimiento global de la simulación. En este artículo se presentan las motivaciones que han llevado a la arquitectura final y la interacción entre los bloques que la componen.

## 2 ARQUITECTURA DE LA PLATAFORMA DE SIMULACIÓN iTETRIS

La arquitectura de la plataforma de simulación iTETRIS consiste en una serie de bloques funcionales que trabajando cooperativamente interactúan para simular sistemas noveles de gestión de tráfico en entornos ITS. Con tal de obtener estimaciones realistas del verdadero potencial que pueden ofrecer dichos sistemas, es necesario considerar escenarios de evaluación a gran escala. Escenarios de tal tamaño implican un elevado número de vehículos distribuidos sobre un área extensa. El estudio de estrategias cooperativas de gestión de tráfico en dichos escenarios permitirá ofrecer resultados precisos y fiables, pero por otra parte, requerirá elevados recursos computacionales; tanto en lo que se refiere a la capacidad computacional como en la duración de la simulación. Por tanto, la arquitectura de la plataforma ha de diseñarse de forma que el impacto de estos efectos sea el menor posible. Este diseño afecta desde la elección de las herramientas de simulación a integrar, los modelos de simulación usados en dichos simuladores hasta la forma en la que estas interactúan.

El objetivo primordial de iTETRIS es lograr que cada estación ITS se simule de forma precisa. Este primer objetivo se completa mediante el uso combinado y simultáneo de dos plataformas de simulación dedicadas, una para el intercambio de mensajes de comunicación radio y otra para la simulación del movimiento y la gestión del tráfico.

El intercambio de mensajes de comunicación entre estaciones ITS corre a cargo de la plataforma de código abierto ns-3 (<http://www.nsnam.org>), la cual ofrece la posibilidad de implementar nodos con más de una tecnología de comunicación [2], como por ejemplo WiFi o WiMAX.

Por otro lado, la simulación del tráfico es ejecutada por la plataforma SUMO (<http://sumo.sourceforge.net>) desarrollada por el laboratorio alemán DLR. Este software "open-source" permite ejecutar simulaciones que emulan la movilidad de los vehículos a nivel microscópico.

La elección de estas dos plataformas, así como la forma de integrarlas, ha sido objeto de estudio del proyecto iTETRIS. La arquitectura basada en sub-componentes sobre la que desarrollar iTETRIS debe ofrecer un grado de precisión suficiente, y a la vez emplear modelos simplificados que sean capaces de reducir el costo computacional. La interacción entre estos componentes debe ser diseñada de forma que el intercambio de información sea

óptimo evitando transferencias de datos incontroladas o ineficientes. En este contexto, se han barajado soluciones basadas en un control centralizado, de tiempo regulado y datos agregados, en lugar de comunicaciones inter-bloque autónomas y carentes de sincronía, usando transferencias de datos de tamaños reducidos pero muy frecuentes.

En lo que concierne a la integración de los simuladores, se han analizado varios enfoques. Empezando por la posibilidad de conectarlos directamente, pasando por la posibilidad de usar software específico para conectar simuladores independientes o utilizar un controlador central separado que se encargue de mantener los simuladores sincronizados y transferir la información de un simulador a otro.

En este proceso, se han considerado aspectos como la interoperabilidad y la modularidad de la plataforma. Estas características permiten que potenciales desarrolladores de iTETRIS puedan reemplazar fácilmente cualquiera de las plataformas de simulación por otro simulador que implemente la misma funcionalidad. Esto permite que en el caso de que se desarrollasen nuevos simuladores más potentes y eficientes, iTETRIS pudiera adoptarlos, garantizando así la continuidad de la plataforma. Además permite que el desarrollo y optimización de cada simulador se pueda realizar de forma autónoma. Para cubrir estos objetivos es necesario que la arquitectura de la plataforma iTETRIS se diseñe con la mayor modularidad posible.

## **2.1 Arquitectura estándar distribuida de simulación**

Con la intención de garantizar la modularidad de la plataforma final la arquitectura iTETRIS se divide de tres grandes bloques. Esta se compone de los simuladores de tráfico y comunicaciones inalámbricas, presentados anteriormente, y de un tercer módulo que lleva como nombre “iTETRIS Control System” (iCS). Este módulo permite desacoplar ambos simuladores, haciendo que el desarrollo e integración de ambos sea independiente. Como consecuencia, no es necesario que los simuladores conozcan los detalles y funcionamiento del otro simulador, resultando en un mayor grado de extensibilidad de la plataforma iTETRIS.

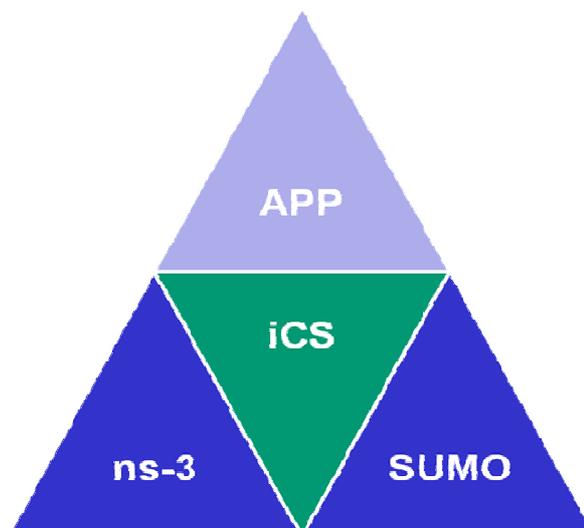


Figura 1. Organización modular de tres bloques y aplicaciones

En términos generales, el iCS es el bloque encargado de asegurar que la ejecución de ambos simuladores esté sincronizada durante el tiempo de simulación pero también tiene como misión el transferir la información de un simulador a otro. Además del papel de orquestación de toda la plataforma, el iCS es la entidad que se encarga de iniciar los simuladores y parametrizar el entorno de simulación con los valores que los expertos en soluciones ITS cooperativas consideren oportunos.

En la arquitectura iTETRIS, definida en la Figura 1, destacan los tres bloques ya presentados (ns-3, SUMO y iCS), los cuales en su conjunto hacen que sea posible la evaluación de las estrategias de gestión de tráfico basadas en sistemas cooperativos ITS. La separación de las aplicaciones del resto de la plataforma permite a los expertos involucrados en el desarrollo de las aplicaciones implementar los algoritmos en el lenguaje de programación que más se adecue a sus necesidades, independientemente del desarrollo de iTETRIS. Esta misma característica se aplica también al iCS que no está condicionado por los detalles de implementación de ns-3 o SUMO.

La estación ITS es el objeto de relevancia dentro de la simulación y es representado tanto en el entorno de SUMO, en su vertiente de movilidad, como en el de ns-3, en sus tareas de comunicación. El iCS reconoce todas las estaciones ITS (nodos) a simular y distingue entre estaciones vehículo y estaciones de infraestructura de comunicación, las cuales están fijas y por tanto no tienen capacidades de movilidad.

El iCS controla SUMO mediante mensajes que desencadenan acciones que modifican el estado de los objetos involucrados en el tráfico. Esto sugiere, entre otras cosas, la posibilidad de imponer a un vehículo la velocidad óptima a la que debe circular o modificar las condiciones y reglas que controlan el flujo del tráfico. No obstante, esta comunicación es bidireccional puesto que durante el periodo de simulación SUMO proporciona al iCS información de tráfico como la posición y velocidad actualizada de las estaciones ITS. También es posible recuperar otros estadísticos como la emisión de gases contaminantes por parte de los vehículos y que podrían servir como “input” para las estrategias de gestión de tráfico o como resultados para sus evaluaciones.

Por otro lado, el iCS es responsable de que la simulación de ns-3 esté sincronizada con la de SUMO. La misión de ns-3 es simular el intercambio de mensajes radio entre los nodos ITS con el fin de alimentar de datos sobre el tráfico a las aplicaciones que estén ejecutándose en los nodos. En iTETRIS, estas aplicaciones podrán solicitar que un nodo comience a transmitir información. El iCS es el responsable de captar esta petición, procesarla y planificar en ns-3 la comunicación que las aplicaciones demanden. Por el contrario, ns-3 informa al iCS del resultado de las simulaciones de comunicación y notifica a las aplicaciones sobre los mensajes radio correctamente recibidos. Durante todo el periodo de simulación, el iCS redirigirá a ns-3 las posiciones de los vehículos obtenidos en SUMO para que las posiciones de los nodos se actualicen. Esta información es de gran trascendencia para efectuar simulaciones realistas ya que la posición de los nodos tiene un impacto enorme en las capacidades de comunicación de los mismos. Por ejemplo, un vehículo podría entrar en una zona donde la tecnología radio que lleva instalada no tendría cobertura.

## **2.2 Arquitectura europea de comunicaciones definida por ETSI**

iTETRIS adopta la arquitectura de comunicaciones definida por ETSI para los Sistemas Inteligentes de Transporte (ITS). La arquitectura se muestra en la Figura 2. Dicha arquitectura especifica el diseño de un sistema de comunicaciones para diversos tipos de aplicaciones de tráfico que usan distintas tecnologías de comunicaciones radio coexistentes. La arquitectura asume tres tipos diferentes de roles desde el punto de vista de las comunicaciones en un escenario ITS, cada uno de ellos representando un subsistema: vehículo, infraestructura de comunicación y estación central. Las diferentes capas que se

muestran en la Figura 2 representan las diferentes funcionalidades que una estación ITS necesita para poder ejecutar aplicaciones cooperativas basadas en sesiones de comunicaciones radio en un escenario ITS. Desde el punto de vista de la implementación, iTETRIS desarrolla un subconjunto de dichas funcionalidades permitiendo a futuros desarrolladores modificar la estructura inicial añadiendo nuevas extensiones y funcionalidades. A continuación, se describirán junto con cada una de las capas de la arquitectura ITS de comunicaciones de ETSI detalles de la implementación de los mismos en iTETRIS.

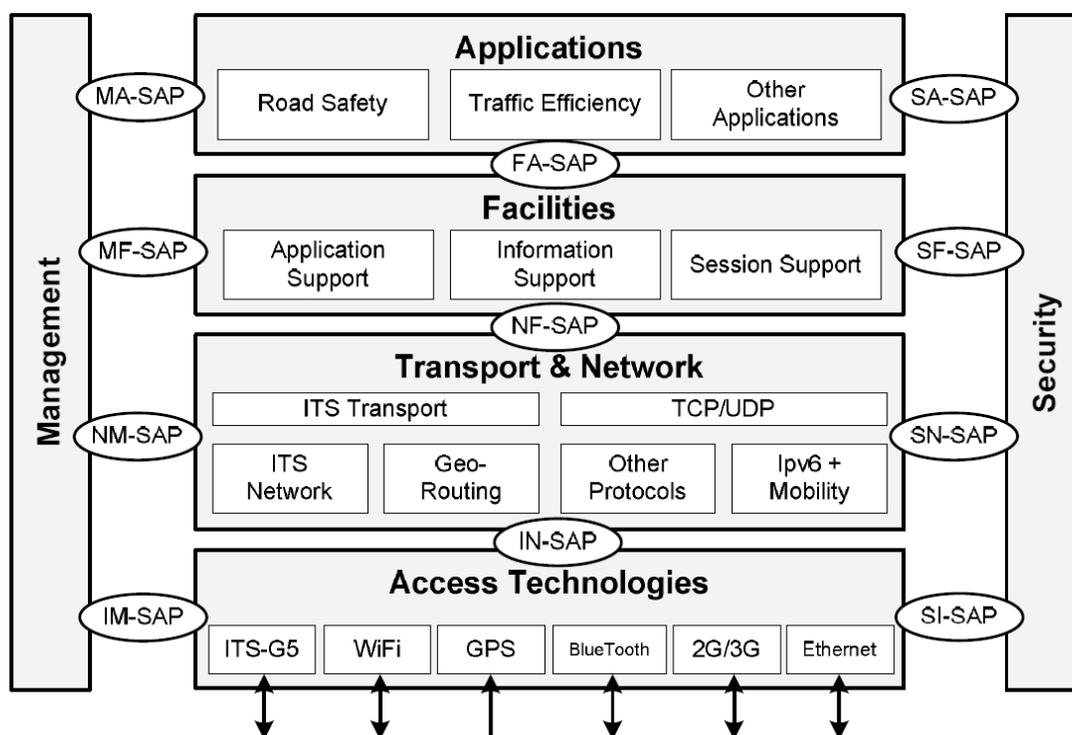


Figura 2 - Arquitectura ITS Europea de Comunicaciones

La capa “Access Technologies” representa el grupo de tecnologías de comunicaciones inalámbricas que permiten el intercambio de información del tráfico entre las distintas estaciones ITS. La arquitectura incluye mecanismos de selección dinámica de la tecnología radio más apropiada en cada caso. iTETRIS implementa y modela cuatro tecnologías radio: IEEE 802.11p (también conocido como WAVE), IEEE 802.16 (WiMAX), UMTS y DVB-H. Este conjunto de tecnologías permite asegurar diferentes tipos de comunicaciones radio: con y sin infraestructura, de corto y largo alcance, unidireccionales y bidireccionales, unicast/broadcast/multicast, one hop-/multi hop, etc. La capa “Networking & Transport” engloba los protocolos de red y transporte necesarios para una comunicación completa en un escenario ITS. A nivel de red y transporte, iTETRIS diferencia entre pila de protocolos

ITS y pila TCP/IP. Mientras que la pila ITS incluye protocolos de comunicación que están siendo desarrollados específicamente para comunicaciones entre vehículos para permitir aplicaciones de seguridad vial y eficiencia de tráfico, la pila TCP/IP contempla protocolos que habitualmente son empleados en otro tipo de redes, pero que también pueden utilizarse en sistemas ITS. La capa ITS con el nombre “Facilities” engloba un conjunto de funcionalidades que son compartidas por las aplicaciones para que puedan ejecutar diferentes tareas. Además, dicha capa provee de un conjunto de estructuras de datos que permite el almacenamiento y posterior proceso de la información de tráfico. Como se define en [7], las funcionalidades se clasifican en tres bloques: “Application Support”, “Information Support” y “Communication Support”. Si bien las capas “Access Technologies” y “Networking & Transport” de la arquitectura ITS de ETSI se implementan íntegramente en ns-3, la implementación de la capa “Facilities” se divide entre ns-3 y el iCS. Algunas de las funcionalidades de las “Facilities” son demandadas frecuentemente por las aplicaciones y por tanto se implementan dentro del iCS. Otras tienen mayor relación con las comunicaciones y se desarrollan como bloques de ns-3. De acuerdo con esta separación, ns-3 y el iCS tan solo tienen acceso directo a las funcionalidades de la capa “Facilities” que estrictamente necesitan para su funcionamiento. Así se evita tanto un consumo innecesario de recursos de simulación como el intercambio ineficiente de mensajes entre el iCS y ns-3. La capa “Applications” se agrupa en tres clases de aplicaciones: “Road Safety”, “Traffic Efficiency”, y “Other Applications”. El interés del proyecto iTETRIS se centra en las aplicaciones que están relacionadas con mejorar la eficiencia del tráfico, permitiendo el desarrollo e implementación de otro tipo de aplicaciones externas a la propia plataforma. La capa “Management” es una capa transversal que gestiona el intercambio de mensajes entre las distintas capas. Las principales funcionalidades implementadas en este bloque incluyen la selección dinámica de las tecnologías de acceso para una aplicación, la monitorización de los parámetros de comunicación de las interfaces, la gestión de los permisos y prioridades de transmisión, la gestión de los servicios y el soporte a los mecanismos de control de la congestión. Únicamente se han considerado estas funcionalidades y todas están implementadas dentro de ns-3. Por último, el bloque “Security” implementa los servicios de seguridad ofrecidos a las pilas de protocolos de comunicaciones y la capa “Management”. Aunque por el momento iTETRIS no contempla las funcionalidades de este último bloque, la plataforma sí asegura la conformidad con las especificaciones requeridas por el mismo. Por ejemplo, el número de bytes adicionales que introducirían los campos de seguridad está incluido en el formato de los mensajes radio. Con lo presentado en este capítulo, la arquitectura distribuida de simulación diseñada para iTETRIS se define en la Figura 3.

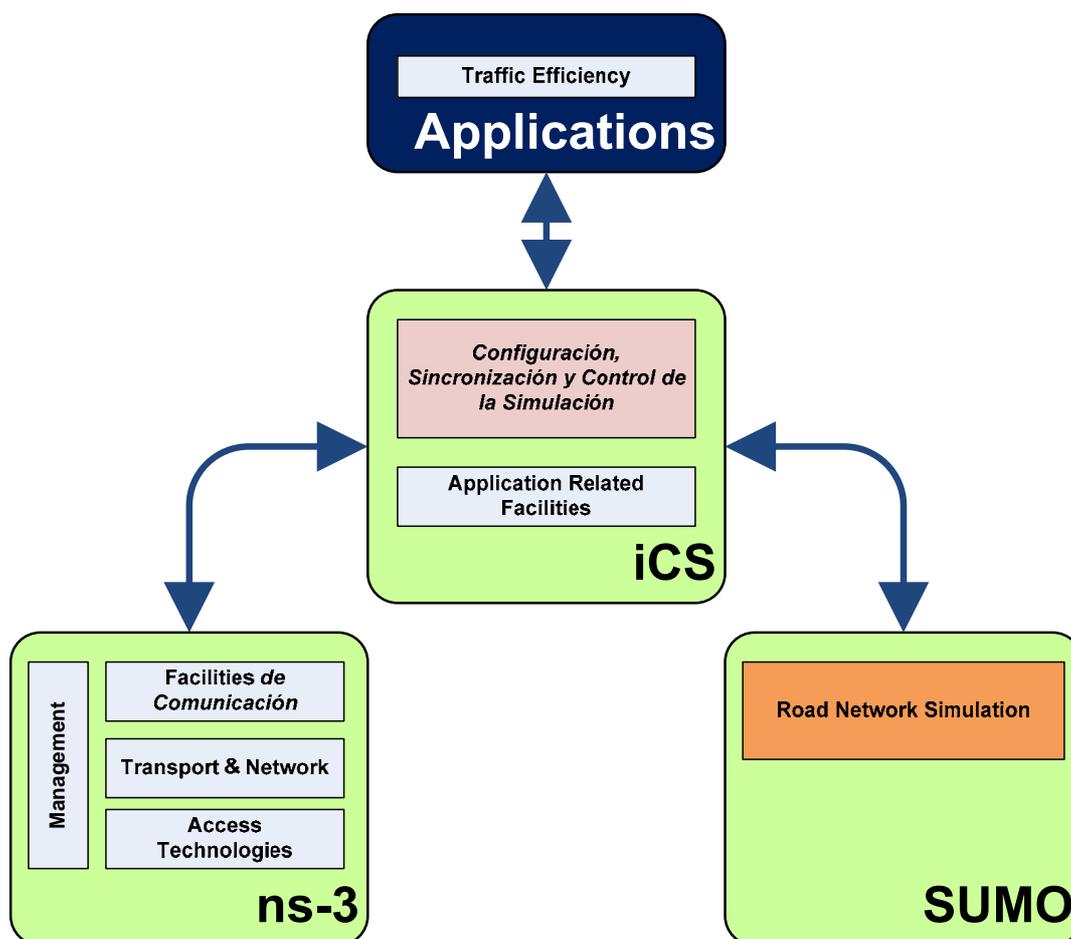


Figura 3 - Arquitectura de iTETRIS alineada con la arquitectura de comunicaciones ITS

### 3 CONECTIVIDAD

En lo que a la comunicación entre los distintos bloques de simulación (SUMO, iCS y ns-3) se refiere, se han considerado varias tecnologías como posible solución para la conectividad.

“Common Object Request Architecture” (CORBA) es un estándar definido por el OMG que posibilita que software escrito en lenguajes de programación distintos puedan trabajar conjuntamente. Esta característica es importante dentro de la arquitectura de iTETRIS puesto que cada uno de los bloques presentados arriba podrían estar escritos en diferentes lenguajes de programación y la arquitectura pretende aislar los detalles de cada simulador. No obstante, esta tecnología ha sido descartada porque penaliza de la misma manera las funciones de procesos que se ejecutan en la misma máquina y los que lo hacen de forma remota. Puesto que iTETRIS contempla la ejecución de todas sus partes bajo la misma

máquina el coste computacional de esta tecnología y la complejidad que impone a los desarrolladores no justifican los beneficios que se obtienen. Por este motivo, se descartó esta tecnología como una alternativa válida. [3] [4]

En la misma línea que CORBA, otra tecnología denominada “Simple Object Access Protocol” (SOAP) también posibilita la conexión de herramientas software, independientemente de su lenguaje y plataforma de ejecución. No obstante, los mensajes que SOAP utiliza se basan en un formato XML lo cual aumenta de forma considerable el tamaño de los mensajes repercutiendo así en la velocidad de transmisión y afectando por tanto al tiempo de simulación total. Esta desventaja hizo que SOAP fuera eliminada como opción. [5]

La especificación con el nombre “High Level Architecture” (HLA) parece la opción idónea para un marco de trabajo como el de iTETRIS. La misión de esta tecnología es la de crear un software que permite integrar simuladores desarrollados independientemente. Socios del proyecto llevaron a cabo experimentos con esta tecnología cuyos rendimientos en simulaciones con un gran número de objetos involucrados han dado resultados poco satisfactorios lo que terminó descartando esta tecnología. [6]

Como resultado iTETRIS ha optado por utilizar como tecnología de comunicación el uso de Sockets basados en el protocolo IP. Esta tecnología tiene un mecanismo simple que permite a dos procesos comunicarse estableciendo una dirección IP y puerto de escucha entre ellos. Esta tecnología requiere que el iCS se encargue de que la comunicación sea correcta y que la información que se comparta entre los diferentes simuladores y las aplicaciones se lleve a cabo de manera adecuada. A su vez, ofrece un rendimiento aceptable y cumple con las exigencias de interoperabilidad requeridas en la plataforma.

### **3.1 Conexión iCS y SUMO**

SUMO implementa un componente que permite a un proceso externo, como puede ser otra herramienta software, interactuar con la simulación de tráfico. Este componente tiene como nombre “Traffic Control Interface” (TraCI) y comprende una serie de comandos que pueden controlar la simulación así como modificar los objetos que la componen. Es posible que un proceso externo inicie, pare y finalice la simulación. Además, también tiene métodos que

pueden modificar el comportamiento de un vehículo asignándole un límite de velocidad máxima al nodo.

En iTETRIS, la relación entre el iCS y SUMO es la de tipo cliente/servidor. SUMO inicia su ejecución con TraCI activado, a la escucha en un puerto determinado y será quien actúe como servidor. Por otro lado, el iCS es quien ejecuta la función de cliente en esta relación y quien controla en todo momento la sincronía de SUMO con el resto de la plataforma. El iCS se configurara con el conocimiento del puerto en el que TraCI acepta órdenes.

El iCS recupera la información usando un método de suscripción. Con este método, el iCS obtiene la información con una sola llamada (la suscripción) por objeto (vehículo) cada vez que la información se necesite. De esta forma, en lugar de solicitar información a SUMO continuamente, el iCS tan solo necesita indicarle una vez la información que desea recoger, logrando así un proceso más eficiente. Además de recoger información, el iCS actúa sobre la simulación de SUMO en función de los resultados que generen las aplicaciones.

### **3.2 Conexión iCS y ns-3**

La conexión entre el iCS y ns-3 se acomete también utilizando el paradigma cliente/servidor que se ha expuesto en la sección anterior con SUMO. El proyecto ns-3 no contempla en este momento la implementación de una interfaz que ofrezca los recursos y posibilidades del simulador a un proceso externo. Es objeto del proyecto iTETRIS desarrollar un módulo dentro de ns-3 que sirva como punto de entrada a las peticiones del iCS. El nombre de esta interfaz es "iTETRIS Network Control Interface" (iNCI).

En la relación del iCS y ns-3, el primero ejecuta el papel de cliente y el segundo el de servidor. El módulo iNCI se inicia junto con el resto del simulador quedando a la escucha de peticiones en una dirección IP y puerto predeterminado. La tecnología de sockets IP es la que hace posible que se pueda acceder a los comandos que iNCI soporta.

Las funcionalidades que iNCI ofrece se implementan en forma de primitivas. Para empezar, iNCI soporta primitivas que permiten al usuario configurar todos los nodos asignándoles las tecnologías de radio y los parámetros de transmisión adecuados para ser utilizados durante la simulación.

Además, a través de del uso de primitivas definidas en iNCI, es posible ordenar a ns-3 la ejecución de la simulación paso por paso, planificar el envío de mensajes radio y recoger los resultados de las comunicaciones. Es importante mencionar que ns-3 no simula la transmisión real de datos. Debido a que esta información solamente es procesada a nivel de aplicación, no es necesario que ns-3 transmita el “payload” o los datos generados a nivel de aplicación. En su lugar, ns-3 simplemente tiene en cuenta el número de bytes de datos para calcular el tamaño total del paquete que será transmitido en el canal radio. Corresponde al iCS llevar un registro de los mensajes que se están simulando en ns-3 y la información que contienen.

Si se considera un vehículo ITS informando a los vehículos de su alrededor sobre su posición, velocidad etc. mediante el uso de mensajes de difusión periódicos, puesto que en todo momento el iCS obtiene esta información de SUMO, el iCS es capaz de almacenar estos datos en una tabla de mensajes donde se guardan todos los “payload” de las transmisiones que se han planificado para simular en ns-3. Después de la simulación de la transmisión, ns-3 comunica al iCS los nodos que recibieron correctamente los mensajes. De esta forma el iCS puede recuperar el contenido del mismo y utilizar la información que contiene para provocar, por ejemplo, la ejecución de una determinada aplicación en todos los receptores del mensaje.

### **3.3 Conexión iCS y Aplicaciones**

En iTETRIS, el término aplicaciones se refiere a los programas que los ingenieros de tráfico han desarrollado para dar una solución a un problema de tráfico en concreto. Estas aplicaciones ITS cooperativas se basan en datos que son recogidos de los simuladores que iTETRIS integra. Por tanto, es necesario que el iCS se configure de forma que sea capaz de acumular la información relevante y transmitirla a las aplicaciones.

La primera barrera que supera la arquitectura de iTETRIS es la de la conectividad. De forma análoga al procedimiento empleado en la interconexión de los simuladores, el iCS acopla las aplicaciones al resto de la plataforma mediante el uso de un protocolo basando en sockets IP. El uso de esta tecnología permite al ingeniero de tráfico abstraerse de las particularidades de la implementación del iCS como es el lenguaje de programación en el

que ha sido desarrollado este último. Por otro lado, para que la aplicación pueda comunicarse correctamente con iTETRIS es necesario incorporar al resto de código un sencillo mecanismo de escucha por el que pueda recibir la información que el iCS le transmitirá.

En segundo término, el proyecto iTETRIS ha propuesto un modelo y transferencia de datos para que una aplicación reciba solamente aquella información que necesita para alimentar el algoritmo que implementa. Así, se pretende optimizar el envío de mensajes desde el iCS a las aplicaciones reduciendo el coste de los mismos, tanto en el tiempo efectivo de envío como en sobrecarga de la red.

Para recibir únicamente la información relevante, el iCS incorpora una serie de datos o acciones a los que las aplicaciones pueden subscribirse. El iCS, además de llevar el control de las estaciones ITS que se están simulando, gestiona las subscripciones de cada una de ellas, así como cada una de las aplicaciones que las estaciones están ejecutando. Para ello, en el iCS se define un objeto con el nombre "Application Handler" que actúa como un representante de la aplicación en el iCS. Este objeto tiene como misión gestionar el procesado de las peticiones que llegan de las aplicaciones, dar de alta y de baja subscripciones para una aplicación, recoger la información de la subscripción y transferirla a la aplicación, así como aplicar el resultado de la ejecución de una aplicación.

## **4 EL FLUJO DE SIMULACIÓN**

Este apartado pretende ilustrar de forma general cómo se lleva a cabo una simulación en iTETRIS y cómo cada uno de los bloques e interfaces presentados hasta ahora actúan para que esta se pueda llevar a cabo de forma correcta, precisa y eficiente. La simulación en iTETRIS se puede separar en dos partes diferenciadas: la configuración de la simulación y el tiempo de ejecución.

### **4.1 Configuración de la simulación**

La configuración de la simulación se establece mediante la lectura de unos ficheros de configuración. Estos archivos, formateados mediante las reglas XML, son definidos por los

usuarios finales de la plataforma para conformar el escenario tanto en términos de tráfico como en parámetros de comunicación.

Para garantizar la simplicidad de los ficheros de configuración, iTETRIS propone un modelo jerárquico de ficheros, ver Figura 4, donde existe un archivo maestro que además de información básica apunta a ficheros de configuración específicos de cada plataforma de simulación.

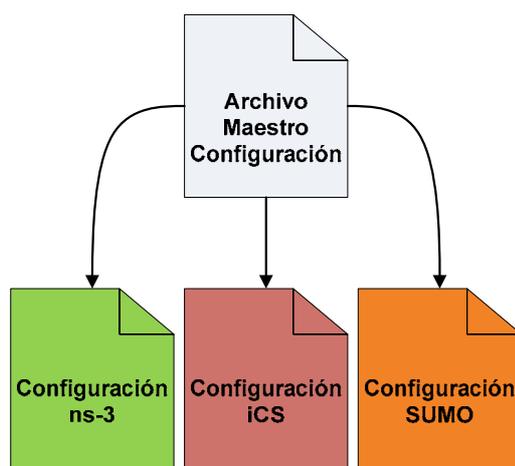


Figura 4 - Jerarquía de archivos de configuración de la plataforma iTETRIS

En el archivo Maestro de Configuración se pueden encontrar los puertos donde SUMO y ns-3 atienden las peticiones, y el tiempo de inicio y fin en los cuales el iCS comenzará y terminará de procesar las ejecuciones de las aplicaciones. Además, también se definen los ratios de penetración de vehículos equipados con capacidad de comunicación; es más, se definirán distintos perfiles de comunicación con una o varias tecnologías de radio y distintas configuraciones. Asimismo, las diferentes aplicaciones tendrán asignados ratios de penetración para que el iCS posea un criterio de asignación.

En primer término, el iCS procesa el fichero maestro de configuración y recupera la ruta donde se encuentran los ficheros de configuración de SUMO y ns-3. El siguiente paso es utilizar la tecnología de proceso de hilos para crear dos ejecuciones separadas dentro del sistema operativo donde se ejecuten ns-3 y SUMO. En este instante, tanto ns-3 como SUMO se encuentran a la espera de peticiones por parte del iCS.

En segundo lugar, el iCS procesa su fichero de configuración en las que reconoce qué tecnologías de acceso a radio para vehículos e infraestructuras ITS están involucradas en la

simulación. También procesa la información de las aplicaciones, como el nombre de su ejecutable, y lanza dichos programas en diferentes hilos.

Para finalizar el proceso de configuración, el iCS hace efectiva la conexión mediante sockets con ns-3, SUMO y las aplicaciones.

#### **4.2 Ejecución de la simulación**

La ejecución de la simulación se caracteriza por dar paso a cada bloque de iTETRIS en un bucle que terminará cuando se alcance el final del tiempo de simulación establecido en el fichero de configuración maestro. El iCS será el módulo encargado de dar paso a cada bloque. El número de eventos que se suceden en una misma ventana de tiempo en el dominio de las comunicaciones inalámbricas es por lo general más alto que el del entorno del tráfico vehicular. Actualmente, SUMO se ejecuta en intervalos de tiempo de un segundo mientras que ns-3 simula eventos con una precisión diferente, desde el uso de segundos hasta el femtosegundo ( $10^{-15}$  segundos). El iCS tiene la responsabilidad de sincronizar ambos simuladores haciendo coincidir la ventana de tiempo en la que trabajarán durante el periodo de simulación. En este caso, la ventana de simulación mínima viene restringida por aquella que ejecute el intervalo más amplio, en este caso SUMO. No obstante, el proyecto iTETRIS contempla en su marco de trabajo implementar en SUMO un intervalo de tiempo inferior a un segundo.

Para obtener resultados fiables será posible establecer un periodo de tiempo en el que las simulaciones de SUMO y ns-3 se ejecuten hasta estabilizarse. Esta característica es necesaria debido a que durante los primeros instantes no existirá un suficiente número de vehículos en la simulación y podrían alterarse los datos de los resultados finales.

La primera acción en el tiempo de ejecución es notificarle a ns-3 usando el interfaz iNCI que debe ejecutar los eventos de comunicación planificados en el próximo segundo. Una vez terminado este periodo el iCS recuperará los resultados del intercambio de mensajes y los transferirá al módulo de "Facilities" en el iCS para su proceso.

La segunda operación consiste en ordenar a SUMO mediante la interfaz TraCI que ejecute acciones en el entorno del tráfico rodado para el siguiente periodo de simulación. Como respuesta SUMO devolverá las posiciones actualizadas de los vehículos. Algunos de estos vehículos se moverán en la simulación por primera vez por lo que el iCS les asignará un

perfil de comunicación y una aplicación a ejecutar en función de los ratios de penetración que han sido explicados anteriormente.

El siguiente paso es ejecutar el proceso de las aplicaciones. En este instante, el iCS recorrerá todos los nodos y tratará cada una de las aplicaciones instaladas en ellas. Entre los distintos cometidos, cabe destacar: dar de alta nuevas subscripciones, dar de baja a las existentes si fuera necesario, enviar a la aplicación la información que la subscripción solicita y recoger el resultado de la aplicación si esta se ejecutó. El resultado de la aplicación podría implicar disparar acciones en SUMO o ns-3 que el iCS sería encargado de provocar mediante la interfaz oportuna.

Para terminar una vuelta completa del bucle y antes de volver a ejecutar ns-3 es indispensable actualizar el estado de la simulación. Esto implica actualizar la posición de los nodos, crear nuevos y configurarlos según lo asignado por el iCS en pasos anteriores. También se deberán planificar las transmisiones de mensajes.

Finalmente el iCS actualiza el contador interno que gestiona la sincronización de la plataforma y si este no ha llegado al valor asignado para terminar la simulación se vuelve a ejecutar el bucle comenzando por ordenar a ns-3 que ejecute el siguiente intervalo de eventos de la simulación. El esquema resumido de esta fase de simulación se puede describir mediante la Figura 5.

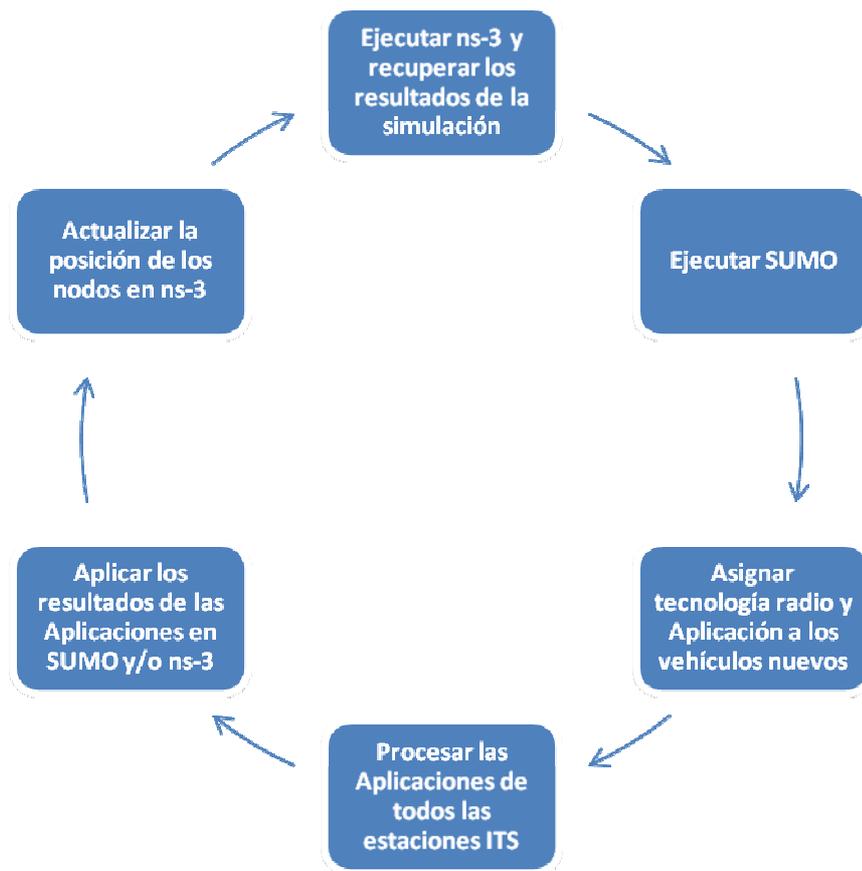


Figura 5 - Flujo de Simulación de la fase de ejecución de iTETRIS

## 5 CONCLUSIONES

Este artículo ha presentado la arquitectura de iTETRIS y algunas características de su implementación. Se han destacado los esfuerzos para conseguir una plataforma lo más modular posible que garantice la interoperabilidad de sus componentes. Así, se ha propuesto una arquitectura estándar distribuida de simulación con un enfoque de tres bloques donde el iCS asegura la sincronización de los simuladores involucrados y hace accesible los recursos de la plataforma a las aplicaciones de gestión de tráfico. Este diseño modular permite un desarrollo flexible y autónomo de los subsistemas ofreciendo la posibilidad de reemplazarlos completamente asegurando así la vigencia de la plataforma después de finalizar el proyecto. También se han ofrecido algunas pautas que optimizan la eficiencia de la conectividad y las comunicaciones entre los bloques que componen iTETRIS. En este ámbito, los sockets IP han mostrado ser la solución que mejor relación ofrece entre rendimiento y complejidad de desarrollo. El artículo también ha mostrado los

pasos completados para garantizar que la arquitectura de iTETRIS está alineada con la arquitectura ITS de comunicaciones definida por ETSI.

## AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por la Comisión Europea a través del proyecto FP7 TIC iTETRIS: An Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions (No. FP7 224644). Los autores desean agradecer a la Comisión el apoyo recibido.

## BIBLIOGRAFÍA

- [1] RAMÓN BAUZÁ, JULEN MANEROS, MIGUEL SEPULCRE, JAVIER GOZÁLVEZ, OSCAR LÁZARO, AINARA GONZÁLEZ. (2009). "iTETRIS: Plataforma Europea para el Análisis del Impacto de Sistemas ITS Cooperativos en la Gestión de Tráfico"; ITS España. Andorra.
- [2] GOZÁLVEZ J., TURKSMA S., LAN L., LAZARO O., CARTOLANO F., ROBERT E., KRAJZEWICZ D., BAUZA R., FILALI F., RÖCKL M., LEGUAY J., MICHELACCI C., VREESWIJK J., MANEROS J., GONZALEZ A. AND LENARDI M. (2009). "iTETRIS: the Framework for Large-Scale Research on the Impact of Cooperative Wireless Vehicular Communications Systems in Traffic Efficiency"; Information and Communications Technologies (ICT-MobileSubmit). Santander.
- [3] [http://www.davidchappell.com/articles/article\\_Trouble\\_CORBA.html](http://www.davidchappell.com/articles/article_Trouble_CORBA.html). Último acceso Marzo 2009.
- [4] <http://queue.acm.org/detail.cfm?id=1142044>. Último acceso Marzo 2009.
- [5] <https://www.ibm.com/developerworks/library/ws-pyth9/>. Último acceso Marzo 2009.
- [6] <http://www.pitch.se/technology/about-hla>. Último acceso Marzo 2009.

[7] ETSI TR 102 638 v1.1.1, (2009), "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions".