

Anexo A: Descripción de simuladores de tráfico

En este anexo se detallan los simuladores de tráficos que se presentan en el Capítulo 2 de la memoria principal, los cuales son el principal objeto de estudio de esta tesis de fin de máster.

A.1. INTEGRATION

El modelo del sistema INTEGRATION fue desarrollado por Michel Van Aerde de la Universidad de Waterloo, Canadá, entre los años 1984-1986. El nombre INTEGRATION proviene del hecho de que el modelo integra una serie de capacidades únicas: asignación de tráfico y simulación microscópica. Algunas de sus características son:

- Modelado del control de señal encendida por cada clase de vehículo que permite la modelación de prioridad de la señal del tránsito.
- Modelado en tiempo real de la información del tráfico para los conductores, durante y antes de su viaje.
- Modelado del dinamismo del vehículo utilizando una constante de energía del modelo dinámico del vehículo que considera la fuerza de tracción del vehículo, la aerodinámica, la rodadura y el grado de fuerza de la resistencia. El modelo es capaz de modelar hasta 25 vehículos de diferentes tipos, incluidos vehículos ligeros, camiones pesados y ligeros, autobuses, etc.
- Modelado de 14 tipos de accidentes diferentes con sus correspondientes niveles de gravedad utilizando las tasas medias de choques para los Estados Unidos.
- Estimación de autopistas y señalizaciones utilizando el Manual de Procedimientos de Capacidad de las Autovías (*Highway Capacity Manual Procedures*).

INTEGRATION es un modelo de simulación completamente microscópico, hace un seguimiento de los movimientos tanto lateral como longitudinal de los vehículos de forma individual a una resolución de hasta una décima de segundo.

El enfoque microscópico también permite una flexibilidad considerable en términos de representación de las variaciones espaciales en las condiciones del tráfico. Por ejemplo, mientras que la mayoría de los modelos consideran que las condiciones de tráfico deben ser uniformes a lo largo de un tiempo determinado, INTEGRATION permite que la densidad de tráfico pueda variar de forma continua.

A.2. SmartPath

SmartPath es uno de los proyectos del programa PATH (*Partners for Advanced Transportation Technology*) de la Universidad de California, Berkeley que se inició en 1991.

Una de sus características es que proporciona un entorno de simulación para probar varios diseños de controlador, evaluar el rendimiento y la observación de la interacción entre los controladores de vehículos y la autopista. La salida en este simulador es una descripción completa del estado de cada vehículo en la carretera.

Presenta una interfaz gráfica para visualizar los datos simulados (carreteras y vehículos) de una manera natural y el usuario puede navegar a través de la autopista, seguir a los vehículos y observar su comportamiento de cerca. Es importante señalar que este simulador sólo se ejecuta en estaciones de trabajo SGI (*Silicon Graphics*).

La versión más reciente y última de SmartPath es la versión 3.0 ya que se desarrolló posteriormente su sustituto SmartAHS, basado en el lenguaje de programación SHIFT, un lenguaje para modelar y ejecutar sistemas híbridos complejos.

A.3. VISSIM

La empresa PTV (*Planung Transport Verkehr AG*) de Karlsruhe, Alemania, líder innovador en movilidad urbana, logística e ingeniería de tránsito por más de tres décadas, ha desarrollado tres grandes productos: VISSIM, VISUM y VISTRO, los cuales forman parte del paquete de software propietario *Vision Traffic Suite*, que cubre todo el espectro de la planificación del transporte, desde la planificación estratégica e ingeniería del tráfico hasta la simulación del tráfico y peatonal; funciona sobre la plataforma Windows.

VISSIM es un simulador microscópico para modelado de tránsito multimodal (modela más de un tipo de tránsito y sus interacciones), se dio a conocer en los años 80, pero no fue hasta el año 1992 que apareció como herramienta comercial. Esta herramienta permite modelar escenarios reales con un alto nivel de detalle y complejidad, con un potente diseño gráfico en 3D, por lo que destaca entre otros simuladores similares. Algunas de sus funcionalidades son mencionadas a continuación:

- Gestión de modelos con variedad de rutas: autovías con peajes, control de flujo del tráfico, vías de control de acceso, etc.
- Análisis de viabilidad de autovías con opciones de rutas alternativas asignadas automáticamente.
- Indicaciones de carreteras desviadas, capacidades de plazas de aparcamiento, etc.
- Representación gráfica de los flujos de tráfico.

VISUM es una herramienta de simulación macroscópica y mesoscópica que permite el análisis de flujo tráfico, planificación de servicios, estrategias avanzadas y soluciones de transporte, con una interfaz de usuario bastante intuitiva. Se basa en el algoritmo de los cuatro pasos (*four-step algorithm*) y en modelos de planificación de transporte como generación de viajes, distribución de viajes (según tiempos entre zonas), selección modal y selección de ruta.

VISTRO por su parte, optimiza la frecuencia de la señal, evalúa los impactos del tránsito, gestiona múltiples escenarios y genera informes; al igual que VISUM, es una herramienta de simulación macroscópica y mesoscópica.

A.4. STRAW

El simulador STRAW (*Street Random Waypoint*) fue desarrollado por la Universidad de Illinois, Northwestern Evanston, está basado en el simulador de red SWANS (*Scalable Wireless Ad hoc Network Simulator*).

Es un simulador de licencia gratuita y libre distribución que sigue el modelo *car-following* de Nagel-Schreckenberg; desarrollado con la herramienta JiST (*Java In Simulation Time*), simulador de eventos discretos que funciona directamente sobre la Máquina Virtual de JAVA y que optimiza los tiempos de ejecución y reduce el consumo de memoria.

STRAW forma parte del proyecto C3 (*Car-to-Car Cooperation*) desarrollado por el proyecto AquaLab de la Universidad de Northwestern en los Estados Unidos. Proporciona resultados bastante fieles al comportamiento real de ciudades en los Estados Unidos, donde se centra su análisis e implementación. A diferencia de SWANS, este simulador posee diversas implementaciones de protocolos de transporte, enrutamiento y acceso al medio.

A.5. VanetMobiSim

VanetMobiSim es una extensión para CanuMobiSim (*Canu Mobility Simulation Environment*), incluye analizadores de mapas GDF (*Geographical Data Files*) y proporciona implementaciones de varios modelos de movilidad, así como modelos de la física y la dinámica vehicular; está basado en Java y puede generar trazas de movimiento en distintos formatos, para luego ser utilizadas por diferentes simuladores de red, NS-2, GloMoSim y QualNet.

VanetMobiSim se centra en la movilidad vehicular, y cuenta con nuevos modelos realistas de movimiento de automoción, tanto a nivel macroscópico como microscópico. A nivel macroscópico, puede importar mapas de los Estado Unidos o generarlo aleatoriamente utilizando *grafos de Voronoi*. A nivel microscópico, implementa nuevos modelos de movilidad, proporcionando una interacción realista entre vehículos y vehículos con infraestructura. Según estos modelos, los vehículos regulan su velocidad en función de los coches cercanos, superan a los demás y actúan de acuerdo con las señales de tráfico en presencia de las intersecciones.

A.6. CORSIM

El simulador microscópico CORSIM surgió en 1998 y fue desarrollado por la Universidad de Florida, se encuentra por su versión 6.3 publicada en agosto del 2012 y requiere del sistema operativo Windows e Internet Explorer para su uso.

CORSIM es de licencia comercial, está compuesto por la aplicación NETSIM para simular escenarios de tráfico urbano y FRESIM que se encarga de simular el tráfico en autopistas. Al igual que SUMO incluye un paquete de herramientas para realizar la simulación:

CORSIM: Se encarga de realizar las simulaciones integrando las aplicaciones FRESIM y NETSIM.

TRAFED: Editor de mapas para CORSIM. Nos permite crear una red a nuestra medida, así como manipular modelos ya definidos o cargar modelos reales compatibles con el programa.

TRAFVU: Módulo de animación gráfica. Su función es mostrar ya sea en una visión global o en particular cualquier parte del escenario simulado en tiempo real. Nos permite procesar archivos de animación de más de 2 GB, no obstante, si el usuario lo desea puede hacer caso omiso de cualquier conjunto de enlaces durante la animación, lo que reduce el tamaño del archivo.

TSIS Next (*Traffic Software Integrated System*): Se trata de una nueva interfaz de usuario para CORSIM, presenta la misma funcionalidad que nos brindan TSHELL, TRAFED y TRAFVU pero incluyendo algunos aspectos de todas ellas, con una arquitectura completamente diferente. La herramienta está compuesta por dos partes: la interfaz para la configuración de los proyectos y ejecución de los programas y la otra parte de conexión que permite la especificación, control y ejecución de los datos. Soporta varias funcionalidades: *carries hot* (de peaje automático), plazas de peaje avanzadas con distintas formas de pago, control de crucero adaptativo (mediante radares adaptar las maniobras de cambio a la velocidad del vehículo que va delante), volantes a la izquierda y vehículos de tipo O-D (origen-destino).

Streets Editor y *Freeways Editor*: Estos editores son capaces de utilizar el volumen básico y el tiempo medio de los usuarios para construir en cuestión de segundos simulaciones de redes complejas.

Tshell: Es la interfaz del sistema GUI (*Graphic User Interface*) y administrador de programas, integra las herramientas del sistema y permite gestionar eficazmente los proyectos de análisis de tráfico desde una vista única.

TSIS OUTPUT PROCESSOR: El procesador de salida permite al usuario calcular estadísticas de tráfico mientras CORSIM está simulando. Los datos recogidos pueden ser guardados en distintos formatos como Excel, texto plano o XML.

RTE (*Run Time Extension*): Es un entorno de desarrollo integrado que permite al usuario realizar análisis en operaciones de tráfico. Construido con una arquitectura de componentes, es un conjunto de funciones que aumentan la potencia del simulador, tanto a nivel lógico como en lo que a variables existentes se refiere. A pesar de que viene pre-configurado con un conjunto de herramientas prevé un mecanismo por el cual una aplicación externa puede interactuar directamente con la simulación CORSIM.

A.7. MITSIM

Este simulador forma parte del laboratorio basado en simulación MITSIMLab que se desarrolló para evaluar los impactos de las alternativas de diseño de sistemas de gestión de tráfico a nivel operativo y ayudar a perfeccionar el diseño posterior. Los ejemplos de sistemas que se pueden evaluar con MITSIMLab incluyen sistemas avanzados de gestión del tráfico aéreo ATM (*Air Traffic Management*) y los sistemas de guía de ruta.

Fue desarrollado por el Programa de Sistemas de transportación ITS (*Intelligent transportation Systems*) del Instituto de Tecnología de Massachusetts, MIT (*Massachusetts Institute of Technology*).

Los diversos componentes de MITSIMLab se organizan en tres módulos:

- Simulador de tráfico microscópico MITSIM: representa el “mundo”; los elementos de tráfico y de red están representados en detalle con el fin de capturar la sensibilidad de los flujos de tráfico para el control y las estrategias de enrutamiento.
- Simulador de Gestión de tráfico TMS (*Traffic Management Simulator*): imita el sistema de control de tráfico en evaluación. Se pueden evaluar el control de tráfico y los sistemas de guías de rutas.
- Interfaz gráfica de usuario es utilizada para la depuración y demostración de los impactos del tráfico a través de la animación vehicular.

MITSIM se aplicó en Estocolmo eligiendo una red en anillo con dos autopistas y tramos urbanos alrededor de Brunnsviken, al norte de la ciudad, donde en horarios picos opera bajo una fuerte congestión. El MIT tuvo que mejorar los modelos de simulación y el calibrado de los parámetros del modelo para que pudiera coincidir con las condiciones del sitio; para ello utilizó datos de tráfico de las observaciones del año 1999, simulando entonces la red en el año 2000. Una vez validado el sistema se llegó a la conclusión de que el MITSIM pudo simular con eficacia las medidas reales del tráfico recomendándose su uso en las ciudades suecas.

A.8. SUMO

El simulador SUMO (*Simulation of Urban Mobility*) de software libre y desarrollado en C++ desde el año 2001 por el Instituto de investigación del transporte que pertenece al centro espacial alemán, se utiliza principalmente en proyectos de investigación donde se estudian patrones de comportamiento de conductores, patrones de movilidad, etc.

SUMO permite modelar varios escenarios con bastante detalle como vías con diversos carriles, limitación de velocidad, intersecciones con semáforos, además permite generar mapas de rutas mediante una aplicación denominada "Netgen" o importarlos desde otras herramientas disponibles como TIGER y OSM permitiendo simular escenarios similares a los reales, incluye el estudio de comportamientos generales del entorno así como comportamientos particulares de cada uno de los vehículos que forman parte de la simulación, utiliza el modelo *car-following* para implementar el comportamiento del conductor dando más realismo a la simulación y es capaz de generar trazas de salida directamente utilizables por varios simuladores de red.

SUMO cuenta con una herramienta gráfica para la generación de movilidad denominada MOVE (*Mobility Model Generator for Vehicular Networks*) y utilizando Netconvert se pueden importar redes desde otros simuladores como VISUM, VISSIM, etc.

SUMO no es sólo el simulador de tráfico también tiene incluido un paquete de herramientas compuesto por varias aplicaciones necesarias para realizar una simulación de tráfico microscópica, este paquete se ha probado en plataformas Windows y Linux. A

continuación se citan las principales aplicaciones de este paquete con una pequeña descripción de cada una de ellas:

- SUMO: Es un simulador y se encarga de llevar a cabo la simulación cuando se han definido todos los parámetros y características en las restantes herramientas del paquete.
- SUMO-GUI: Es la interfaz gráfica de *SUMO* que nos permite visualizar de forma más amigable los escenarios hemos generado.
- NETCONVERT: Esta aplicación de línea de comandos. Utiliza redes viales generadas por otras herramientas en diferentes formatos para convertirlas al formato que emplea *SUMO*.

NETCONVERT es capaz de importar mapas desde los siguientes formatos:

- "SUMO native" XML descriptions (*.edg.xml, *.nod.xml, *.con.xml)
- OpenStreetMap (*.osm.xml)
- VISSUM (*.net)
- VISSIM (*.net)
- openDRIVE (*.xodr)
- MATsim (*.xml)
- SUMO (*.net.xml)
- Shapefiles (.shp, .shx, .dbf) ejemplos: *ArcView* y *TIGER*
- Robocup Rescue League
- DlrNavteq

- NETGENERATE: Se encarga de generar redes de carreteras abstractas que pueden ser utilizadas por otras herramientas de *SUMO*. Permite generar 3 tipos de redes: redes con forma de rejilla, redes con forma de tela de araña y redes aleatorias.
- OD2TRIPS Esta aplicación importa matrices O-D y luego las divide en viajes individuales de vehículos.
- DUAROUTER: Otra aplicación para generar rutas de vehículos definiendo algunas características como origen, destino, tipo de vehículo, etc., y que es de gran ayuda para realizar cálculos de trayectoria más corta entre dos puntos entre otras.
- JTRROUTER: Es otro generador de rutas del paquete de SUMO. En este caso calcula las rutas utilizando porcentajes de giro en las intersecciones, para generar los vehículos define un parámetro con valor entre 0 y 1 que significa que incluirá un número de vehículos por unidad de tiempo.
- DFROUTER: Esta aplicación permite definir y realizar cálculos de las rutas que seguirán los vehículos mediante flujos. De este modo, se pueden definir varios grupos de vehículos con un origen y un destino común, con iguales velocidades y tiempos de salidas entre otras características definidas de antemano.

- POLYCONVERT: Se utiliza para importar formas geométricas (polígonos o puntos de interés) desde diversas fuentes de datos como *OSM*, *VISSUM*, *XML (Extensible Markup Language)* y los convierte a una representación que puede ser visualizada por *SUMO-GUI*.
- ACTIVITYGEN: lee la definición de una población que coincide en una red dada y calcula los posibles movimientos de esta.

Para finalizar, SUMO es una herramienta muy potente para el desarrollo y manejo de simulaciones de tráfico y que permite implementar y crear varios entornos muy reales para ser analizados y estudiados y de esta forma obtener conclusiones que puedan ser muy útiles para determinar el funcionamiento y la distribución de las redes ad-hoc vehiculares.

A.8.1 Herramienta MOVE

MOVE es un generador de modelos de micro-tráfico de código abierto implementado en Java que funciona sobre el simulador SUMO.

Consta de dos componentes principales: el Editor de Mapas y el Editor de Tráfico de Vehículos; el Editor de Mapas se utiliza para generar la topología, que puede ser implementada manualmente por el usuario, automáticamente o a partir de mapas reales como los disponibles en TIGER. La generación manual a diferencia de la automática, requiere como parámetros de entrada, los nodos, que son puntos particulares del mapa que puede ser la unión, el final de la carretera o semáforos y los enlaces, son las carreteras que unen dos nodos del mapa cuyos atributos son: velocidad, número de carriles de la carretera, prioridades, entre otros.

Existen tres tipos distintos de mapas generados automáticamente: el de "parrilla", "tela de araña" y "red aleatoria". La generación de movimiento de vehículos se puede realizar de forma manual o automática, de forma manual se utilizaría el editor de tráfico, que permite incluir el número de vehículos que siguen una determinada ruta, el origen y destino, la velocidad, la duración del trayecto, la probabilidad de cambio de dirección, entre otros parámetros, MOVE en cambio, genera como resultado un archivo de trazas que puede ser empleado en simuladores de red como NS-2.

El uso de MOVE tiene muchas ventajas, entre las que se encuentran: la importación de mapas a través de *Google Maps*, la configuración de los movimientos de los vehículos, las velocidades, el número de vehículos en la carretera, etc. La Figura A.1 muestra su arquitectura. Sin embargo, también tiene sus desventajas y es precisamente que al simular grandes cantidades de vehículos, el sistema consume un alto número de recursos, por eso es necesario emplear ordenadores de alta gama, esto se debe a que SUMO antes de realizar la simulación calcula las tutas de cada uno de los vehículos que intervienen en la misma.

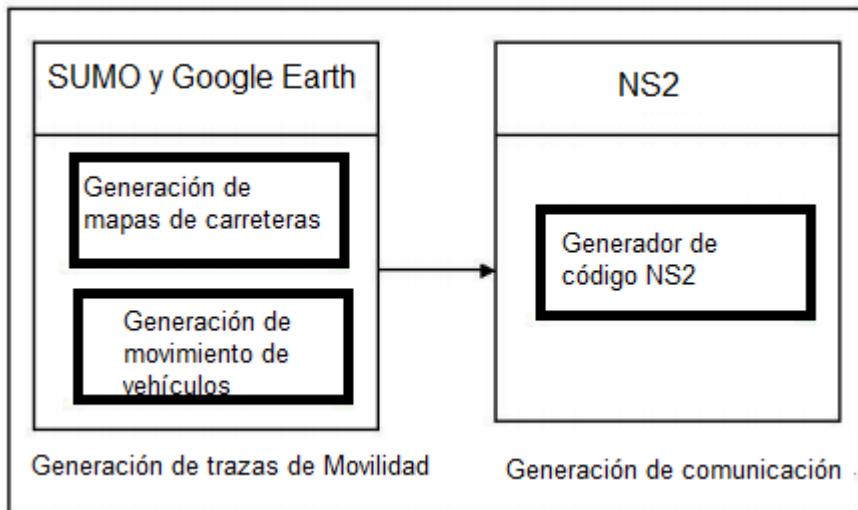


Figura A.1: Arquitectura de MOVE.

A.9. CityMob

CityMob es un generador de patrones de movimiento para redes vehiculares desarrollado por el GRC (Grupo de Redes de Computadores) de la Universidad Politécnica de Valencia. Tiene como finalidad ser utilizado por la comunidad científica por lo que es de licencia gratuita con acceso a su código fuente lo que nos permite adatar sus algoritmos a un proyecto determinado. Está diseñado para que sus trazas se utilicen por el simulador de red NS-2.

Algunas características que presenta CityMob son:

- Eficiencia y facilidad de uso.
- Utilizar mapas reales de todo el mundo basado en OpenStreetMap.
- Comparación entre los diferentes modelos de movilidad.
- Utiliza SUMO para generar el movimiento de los vehículos.

Además se encarga de crear escenarios de movilidad urbana incluyendo los coches parados haciendo uso de la red para intercambiar información entre vehículos, con el fin de evitar accidentes o embotellamientos. Utiliza tres patrones de movilidad: *simple*, *manhattan* y *centro de ciudad*.

Anexo B: Descripción de simuladores de red

En el presente anexo se detallan cada uno de los simuladores de red utilizados en la memoria principal, con el objetivo de brindar más información relevante de los mismos.

B.1. NS-2

El simulador de eventos discretos NS-2 (*Network Simulator 2*) permite realizar investigaciones sobre las redes, simular protocolos TCP (*Transmission Control Protocol*), enrutamientos, protocolos *multi-cast* (entrega información a un grupo de destinatarios simultáneamente) sobre redes cableadas e inalámbricas.

NS nació en 1989 como una variante del simulador de redes REAL. Aproximadamente en 1995-1996 salió la versión NS-2, desarrollo realizado por DARPA (*Defense Advanced Research Projects Agency*), la agencia del Departamento de Defensa del gobierno de los Estados Unidos a través del proyecto VINT (*Virtual InterNetwork Testbed*) en colaboración con LBL (*Lawrence Berkeley National Laboratory*), Xerox PARC (*Palo Alto Research Center Incorporated*), UCB (*University of California, Berkeley*) y USC (*University of Southern California*) / ISI (*Information Sciences Institute*). En el 2001-2004 continuó el desarrollo la agencia DARPA junto a SAMAN (*Simulation Augmented by Measurement and Analysis for Networks*) y otras colaboraciones.

Utiliza herramientas como el compilador de C++, TCL (*Tool Command Language*) / TK (*Tool Kit*) para la ejecución de los scripts, NAM (*Network Animator*) como interfaz gráfica de usuario para la visualización de las simulaciones y otras bibliotecas y paquetes. Funciona en *GNU/Linux*, *FreeBSD*, *Solaris*, *Mac OS X* y *Windows 95/98/NT/2000/XP* y se utiliza la versión NS-2 bajo la licencia GNU.

NS-2 es un proyecto que no se terminó de probar y verificar, más bien es el resultado de continua investigación y desarrollo que se detuvo en el año 2009 para dar paso a su sucesor NS-3, por tanto, la tendencia a partir de ese año ha sido reducir la utilización de ese simulador.

B.2. NS-3

NS-3, al igual que NS-2, es un simulador de redes de eventos discretos, dirigido principalmente a la investigación y el uso educativo. Es software libre, bajo la licencia GNU/GPLv2. Su objetivo es abrir el caudal de investigación de las redes alineado con la simulación, desarrollo y validación de software.

El trabajo en NS-3 se inició entre los años 2004 y 2005 dirigido por *Tom Henderson* y *Sumit Roy* de la Universidad de Washington, *George Riley* de *Georgia Tech* y *Sally Floyd* del ICSI (*International Computer Science Institute*).

Algunos de los principales objetivos de este proyecto fueron la construcción de un mejor soporte para la emulación de red y la reutilización de código de la aplicación, para una mejor integración con las pruebas. Se decidió abandonar la compatibilidad con NS-2, debido al alto coste de mantenimiento que hubiera tenido, por tanto, se comenzó

a desarrollar desde cero utilizando C++ y Python, desarrollos que comenzaron en julio del 2006 y ya en julio del 2008 salió la versión NS-3.1. En diciembre del 2012 salió la versión NS-3.16, caracterizada por su integración con el generador de topologías aleatorias BRITE y muchas otras correcciones de errores y extensiones pequeñas.

NS-3 también es compatible con un planificador de tiempo real que facilita una serie de simulación en ciclos (*simulation-in-the-loop*), casos de uso para interactuar con los sistemas reales. Por ejemplo, los usuarios pueden emitir y recibir paquetes NS-3 generados en los dispositivos de red reales, y NS-3 puede servir como un *framework* de interconexión para añadir efectos de vínculo entre las máquinas virtuales.

Ns-3 tiende a convertirse en un simulador fuerte, lo que se puede observar con los desarrollos constantes que han salido desde su aparición. La desventaja actual es la limitada cantidad de librerías que tienen. No obstante, al pasar el tiempo, ya se van observando cambios sustanciales.

B.3. OMNeT++

OMNeT++ es simulador modular de eventos discretos para redes, es decir, está construido a partir de componentes que se comunican mediante el intercambio de mensajes. Los módulos pueden ser anidados, o sea, varios módulos se pueden agrupar juntos para formar un módulo compuesto. Al crear el modelo, será necesario asignar el sistema en una jerarquía de comunicar los módulos. Es multiplataforma, orientado a objetos y desarrollado en el lenguaje C++ por la empresa *Simulcraft Inc*, es gratuito para uso académico y sin fines de lucro, no obstante, cuenta con una versión comercial llamada *OMNEST*.

Este simulador nos brinda grandes prestaciones y es ampliamente utilizado actualmente en el ámbito científico, su última versión (4.3RC1b) nos permite modelar cualquier entorno de red con una herramienta gráfica de desarrollo basada en eclipse, pero aún limitado por la complejidad de su uso y es anfitrión de otras herramientas como editores, vistas, asistentes y funcionalidades adicionales para la creación y configuración de los modelos *NED* y archivos *.ini*.

Existen extensiones para simulación en tiempo real, emulación de la red, lenguajes alternativos de programación como *Java* y *C #*, la integración de bases de datos y otras funciones. Contiene varios modelos de movilidad como *Random Waypoint* o *mass-based*. No obstante, se sabe que estos modelos no se adaptan a la realidad de las redes vehiculares.

OMNeT++ proporciona un marco de simulación potente y claro, pero carece de apoyo directo y una cadena de modelado concisa para la comunicación inalámbrica por lo que se integra con un *framework* llamado MIXIM.

MIXIM es un *framework* de modelado OMNeT++ para redes inalámbricas móviles y fijas como las redes inalámbricas de sensores, redes ad-hoc, redes vehiculares, etc. Ofrece modelos detallados de la propagación de ondas de radio, la estimación de la interferencia, el consumo de energía de radio transmisor y los protocolos MAC, por ejemplo *Zigbee*, protocolos de alto nivel de comunicación inalámbrica.

B.4. QualNet

QualNet es una herramienta de planificación, pruebas y entrenamiento que "imita" el comportamiento de una red de comunicaciones real, donde los usuarios pueden evaluar el comportamiento básico de una red y realizar pruebas de servicios de red. QualNet proporciona un entorno completo para el diseño de protocolos, creación y animación de escenarios de red, además de análisis de su desempeño.

QualNet contiene los siguientes componentes:

- Arquitecto: herramienta de diseño de escenario gráfico y de visualización. En el modo de diseño se puede crear el terreno, las conexiones de red, subredes, patrones de movilidad de los usuarios de redes inalámbricas, entre otros; personalizando además, la pila de protocolos de cualquiera de los nodos y especificar en la capa de aplicación, el tráfico y los servicios que se ejecutan en la red. En el modo de visualización se puede realizar el análisis de un escenario de red que ha sido diseñado con anterioridad.
- Analizador: herramienta gráfica estadística que muestra cientos de métricas recopiladas durante la simulación de un escenario de red; pudiendo ver los informes pre-diseñados o personalizar los gráficos con estadísticas propias. Todas éstas estadísticas son exportables a hojas de cálculo en formato CSV.
- Paquete de Trazas/Rastreo: herramienta gráfica que proporciona una representación visual de los paquetes de rastreo generados durante la simulación de un escenario de red. Los ficheros obtenidos son de tipo texto en formato XML, los que contienen información acerca de los paquetes a medida que se van moviendo hacia arriba y abajo en la pila de protocolos.
- Fichero de Edición: herramienta de edición de texto.
- Interfaz de líneas de comandos: da acceso a la línea de comandos del simulador.

La herramienta fue desarrollada por SCALABLE, un grupo estadounidense con experiencia demostrada en proyectos como GloMoSim. Su software para redes virtuales con hardware físico permite a los usuarios tener una amplia gama de escenarios totalmente realistas para la realización de sus operaciones y la obtención de resultados eficientes. Este grupo también desarrolla soluciones personalizadas y presta servicios técnicos a contratistas de defensa y aeroespacial, al Departamento de Defensa de Estados Unidos, a operadores de redes móviles, agencias de investigación y universidades.

QualNet y su librería de modelos se ejecutan en una amplia gama de plataformas, incluyendo los sistemas operativos Windows y Linux, con procesadores de 32 o 64 bits, empleando un ordenador o un portátil, pudiendo posteriormente, si se desea, transferir el modelo desarrollado o el diseño de la red a un potente servidor Linux con varios procesadores para ejecutar la capacidad, el rendimiento y la escalabilidad de los análisis.

B.5. JiST

El sistema de simulación de eventos discretos JiST (*Java in Simulation Time*) construido en el 2004 en la Universidad de *Cornell*, en los Estados Unidos, con el propósito de poder ejecutar de forma eficiente simulaciones de eventos discretos de alto rendimiento. Este sistema se ejecuta sobre una máquina virtual de Java y los resultados que se obtienen son realmente alentadores, optimizando los tiempos de ejecución de las simulaciones y el Consumo de memoria.

Su arquitectura consta de cuatro componentes: un compilador, una reescritura de *bytecode*, un *kernel* de simulación y una máquina virtual. Figura B.1.

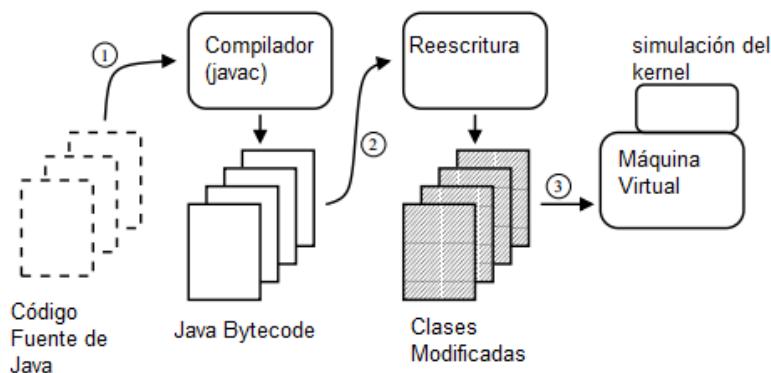


Figura B.1: Arquitectura de JIST.

En una simulación de red con JiST se puede observar cada nodo con sus propios tiempos, estos nodos interactúan entre sí en determinados puntos de sincronización, facilitando la ejecución en paralelo de sus códigos (escritos en Java), logrando así la obtención de un alto rendimiento de la red.

B.6. SWANS

SWANS es un Simulador de Redes Inalámbricas *Ad-Hoc* Escalable construido sobre la plataforma JiST, cuyo objetivo principal es tener un motor de simulación de eventos discretos.

Con este simulador, basado en Java, se pueden simular redes con un mayor número de nodos, obteniendo resultados positivos en el rendimiento, ahorro de memoria y ejecución de las aplicaciones, además implementa una estructura de datos, denominada clasificación jerárquica, para el cálculo eficiente de los modelos de propagación de señal.

Cada componente en SWANS se encapsula como una entidad JiST: se almacena en su propio estado local e interactúa con otros componentes expuestos a través de las interfaces basadas en eventos. SWANS contiene componentes para la construcción de un nodo de una “pila”, los cuales poseen gran variedad de modelos de movilidad y configuraciones de campo lo cual simplifica el desarrollo de la simulación reduciendo el problema a la creación de componentes relativamente pequeños dirigidos por eventos.

SWANS tiene una ventaja única sobre los simuladores de red existentes: se puede ejecutar de manera regular, en la red simulada, las aplicaciones de red en Java sin modificar, lo que permite la inclusión de *software* existentes basado en Java, como servidores web, aplicaciones *peer-to-peer* y los protocolos de multidifusión a nivel de aplicación. Estas aplicaciones no se limitan a enviar paquetes del simulador a otros procesos. Operan en tiempo de simulación dentro del mismo espacio del proceso JiST, lo que permite lograr una mayor escalabilidad.

B.7. GloMoSim

El simulador de red GloMoSim (*Global Mobil Information Systems Simulation Library*) está basado en Parsec, un lenguaje de simulación basado en el lenguaje C, desarrollado por el Laboratorio de Computación Paralela de la Universidad de California, para la ejecución paralela y secuencial de modelos de simulación de eventos discretos por lo que si se desea desarrollar protocolos en GloMoSim o conocer el funcionamiento de su *kernel* es necesario familiarizarse con este lenguaje.

En este simulador cada nodo es tratado como una entidad en *Parsec* independiente, al igual que cada partición de la red (que puede ser definida por el usuario) compuesta por varios nodos con este particionamiento de la red se logra que el Consumo de memoria sea más bajo. Además cada entidad representa un área geográfica de la simulación, por lo tanto los nodos de la red que representan una entidad particular se determinan por la posición física de los nodos.

El sistema está compuesto por varias funciones que se pueden utilizar para enviar mensajes entre las capas. Cuando una capa recibe un mensaje, se invoca automáticamente una función que es proporcionada por el desarrollador de esa capa particular y en base a los contenidos del mensaje, la función a continuación, puede ejecutar las instrucciones apropiadas. Al final de la simulación, para cada capa de cada nodo se llama también una función. Una capa puede utilizar esa función para recopilar las estadísticas relevantes.

GloMoSim tiene dos versiones una gratuita para uso educacional, bastante limitada en cuanto a funcionalidad y número de simulaciones y otra comercial cuyo precio es muy elevado.

B.8. GTNetS

El entorno de simulación de redes GTNetS (*Georgia Tech Network Simulator*) presenta funcionalidades que permiten estudiar el comportamiento moderado de redes de gran escala, bajo una variedad de condiciones. Su filosofía de diseño es crear un entorno de simulación con estructura muy similar a las redes reales. Por ejemplo, en GTNetS, existe una separación clara y distinta de las capas de la pila de protocolos. Existen varias versiones de este simulador, tanto para usuarios de Windows como para Linux, donde la última versión fue desarrollada en el 2008.

Los paquetes en GTNetS consisten en una lista de unidades de datos de protocolos PDU (*Protocol Data Unit*) que se añaden y se eliminan del paquete si se mueven hacia arriba y abajo de la pila de protocolos. Los objetos de la simulación representan los nodos de

la red, tienen una o más interfaces y pueden tener una dirección IP y un vínculo asociado. Los objetos del protocolo de la capa 4 en GTNetS están obligados a ser puertos, de una manera casi idéntica a la unión de los puertos de los protocolos de las redes reales. Las conexiones entre los objetos de un protocolo en la capa de transporte se especifican mediante una IP y un puerto de origen, IP y puerto de destino, *tuplas* como en las conexiones reales TCP. La interfaz entre las aplicaciones y protocolos de transporte utiliza una conexión de escucha, envío y llamadas de envío similares a los sockets API en entornos Unix.

La creación y ejecución de una simulación en GTNetS requiere la creación de un programa *main* en C++ con instancias de los diversos elementos de la red para describir la topología y las diversas aplicaciones y protocolos utilizados para mover los datos simulados a través de dicha topología. El programa *main* en C++ se compila con cualquier compilador que cumpla con el estándar de este lenguaje. Una vez que la compilación haya tenido éxito, el *main* estará vinculado con bibliotecas de objetos GTNetS que crearán el fichero binario ejecutable, el cual será ejecutado como cualquier otra aplicación y cuyo resultado será la simulación de la topología con los datos especificados en el programa.

Anexo C: Descripción de simuladores híbridos e integrados

Se brinda información detallada sobre simuladores más completos estudiados en este trabajo, para modelar y simular redes de vehículos como son los simuladores híbridos e integrados.

C.1. TraNS

El simulador híbrido TraNS (Traffic and Network Simulation Environment) es una herramienta de interfaz gráfica que integra dos simuladores, SUMO, de tráfico y NS-2 de red. TraNS es de código abierto desarrollada en Java en la Escuela Politécnica Federal de Lausana en Suiza.

Esta herramienta está basada en comunicaciones inalámbricas y usa la lógica equivalente a los desarrollos de las redes VANETs, lo que permite actuar sobre el comportamiento de los vehículos en el modelo de movilidad, ya sea ante una situación de accidente en la carretera informando a los demás vehículos, puntos de objetos de tráfico, como semáforos, etc.

TraNS dispone de dos modos de funcionamiento: el primero, "centrado en la red", sirve para evaluar la movilidad real del nodo; los protocolos de comunicación VANET no tienen influencia en tiempo real sobre la movilidad de los nodos. Por ejemplo, el intercambio de contenidos (música o información de viajes) entre los usuarios. El segundo modo, "centrado en la aplicación", se puede utilizar para evaluar las aplicaciones VANET que influyen en la movilidad en tiempo real de los nodos, y por consiguiente, durante el tiempo de ejecución de la simulación de tráfico. Por ejemplo, las aplicaciones de seguridad para evitar colisiones, rupturas, etc.

TraNSLite es la versión reducida de TraNS para generar trazas de movilidad realistas para NS-2 desde SUMO.

TraNS no soporta las últimas versiones de SUMO y NS-2, porque es un proyecto que en este momento se encuentra detenido, por tanto, si se desea generar trazas de movilidad para NS-2 hay que utilizar TraNSLite, que es compatible con SUMO 0.10. Como aconsejan en la propia web de TraNS y cuya bibliografía se cita al inicio de esta sección.

C.2. MOBIREAL

MOBIREAL es un simulador novedoso para el campo de los dispositivos móviles. Desarrollado en C++ por la Universidad de Osaka en Japón. Permite simular la movilidad de los humanos y los vehículos en tiempo real, cambiando su comportamiento en función de un contexto de aplicación dado y así obtener una evaluación detallada de las aplicaciones de red, protocolos de enrutamiento, infraestructuras, etc.

MOBIREAL permite simular redes móviles ad-hoc mediante la adición de modelos de movilidad a un simulador de redes GTNetS.

Para el desarrollo de la herramienta se adoptó un modelo probabilístico basado en reglas con el objetivo de describir el comportamiento de los nodos móviles: cambio de destinos, rutas y velocidades, direcciones en función de sus posiciones, el entorno (obstáculos y nodos vecinos), la información obtenida de las aplicaciones, y así sucesivamente.

La aplicación de animación de MOBIREAL visualiza de forma dinámica el movimiento del nodo, la conectividad, y la transmisión de paquetes, lo que contribuye positivamente a la fácil comprensión de los resultados de la simulación. Con MOBIREAL, es posible simular al mismo tiempo una mezcla de diferentes modelos de movilidad.

Para la movilidad vehicular, los autores de MOBIREAL modificaron un simulador de tráfico llamado NetStream que fue desarrollado por Toyota y como este software es propietario, los usuarios no pueden acceder ni modificar esa parte del simulador, lo que consiste una desventaja del simulador.

C.3. VEINS

VEINS es un *framework* de simulación para Comunicación Inter-Vehicular IVC (*Inter-Vehicular Communication*) de código abierto compuesto por un simulador de redes basado en eventos y un modelo de microsimulación de tráfico. Proyecto construido por Universidad de Erlangen-Nuremberg en Alemania, en específico el departamento de redes de ordenadores y sistemas de comunicación y la Universidad de Innsbruck en Austria, con la colaboración del Centro Aeroespacial Alemán, cuyo comienzo data del año 2008.

Ambos modelos están bidireccionalmente relacionados y las simulaciones se realizan on-line. De esta manera, puede modelarse la influencia del tráfico por carretera en el tráfico de red y viceversa. En particular, las influencias de IVC del tráfico vial pueden ser modeladas, así como examinadas las complejas interacciones entre ambos dominios.

VEINS se caracteriza por:

- Permitir la reconfiguración y reenrutamiento de los vehículos online en reacción a los paquetes de red.
- Se basa en el modelo de confianza de movilidad vehicular, en modelos de IEEE 802.11p y IEEE 1609.4 DSRC / WAVE de las capas de red, incluyendo la operación *multi-canal*, el canal de acceso QoS, el ruido y los efectos de interferencia.
- Puede simular bloques de simulaciones de ciudades en tiempo real en una sola estación de trabajo.
- Puede importar escenarios enteros de OpenStreetMap, incluyendo edificios, límites de velocidad, semáforos, restricciones de acceso, etc.

- Proporciona una fuente de datos para una amplia gama de mediciones, incluyendo el tiempo de viaje y las emisiones.
- Tiene una base de usuarios sólida y diversa de los cinco continentes

Está compuesto por dos simuladores: OMNeT++ para la simulación de la red y SUMO para la simulación de tráfico. Para llevar a cabo evaluaciones de IVC, ambos simuladores corren en paralelo, conectados a través de un socket TCP. El protocolo para esta comunicación se ha estandarizado como la Interfaz de Control de Tráfico TRACI (*Traffic Control Interface*). Esto permite la simulación de acoplamiento bidireccional del tráfico por carretera y el tráfico de red. El movimiento de vehículos en el simulador de tráfico de la carretera SUMO se refleja en el movimiento de los nodos en una simulación OMNeT++. Los nodos pueden interactuar con la simulación en ejecución del tráfico por carretera, por ejemplo, para simular la influencia de IVC en el tráfico por carretera.

C.4. GrooveNet

GrooveNet es un simulador de tráfico híbrido desarrollado por la Universidad Carnegie Mellon en Pittsburgh, Estados Unidos, funciona sobre el sistema operativo Linux y es de libre distribución. Permite la comunicación entre vehículos reales mediante el empleo de elementos de comunicación como GPS (*Global Positioning System*), vehículos simulados y entre vehículos reales y simulados, en esta última, es posible incluso, el intercambio de paquetes. Es posible controlar simulaciones con cientos y cientos de vehículos y agregar modelos de interconexión y seguridad, por ejemplo.

Las principales características de GrooveNet son:

- Simulador basado en distintos eventos con interfaces bien definidas que facilitan la adición de modelos externos sin que generen conflictos con los ya existentes.
- Permite la comunicación entre vehículos, su seguimiento e inclusión de semáforos, cambio de carril y simulación de modelos de GPS.
- Soporta tres tipos de nodos simulados: vehículos que son capaces de transmitir información sobre uno o más canales DSRC, nodos de infraestructura fija y *gateways* móviles capaces de desarrollar comunicaciones *car-to-car* y *car-to-infrastructure*.
- La generación automática de simulaciones es sencilla porque su interfaz gráfica permite crear simulaciones de miles de vehículos, los cuales obedecen límites de velocidad y son mostrados a partir de las coordenadas que devuelven sus GPS.
- Soporta múltiples tipos de mensajes, como los GPS, que se transmiten periódicamente para informar a los nodos vecinos de la posición actual del vehículo, y mensajes con prioridad, como los de los vehículos de emergencia y los mensajes de situación de peligro.

- Soporta varias interfaces de red para las propiedades *car-to-car* y *car-to-infrastructure*, tales como interfaz DSRC a 5.9 GHz e interfaces celulares EVDO (*Evolution-Data Optimized* o *Evolution-Data Only*). La comunicación se puede establecer a través de sockets TCP o UDP.

Este simulador está diseñado para desarrollos rápidos, pruebas de correcciones y pruebas de estrés de protocolos de redes vehiculares, además de la creación de prototipos para la comunicación *multi-hop* y el estudio de estándares de Comunicaciones Dedicadas de Corto Alcance DSRC (*Dedicated Short Range Communication*).

La última versión de este simulador se desarrolló en Ubuntu 1204 y como librería gráfica QT 3.x ya que la versión con QT 4 está en desarrollo.

C.5. AIMSUN

AIMSUN es un simulador híbrido de licencia propietaria, desarrollado por la empresa TSS (*Transport Simulation System*), la Universidad Politécnica de Cataluña y una serie de colaboradores en investigación de la Unión Europea desde 1989.

Este simulador integra tres tipos de modelos de transporte: herramientas de asignación de tráfico estático, simulador mesoscópico y microscópico; esto último permite crear un modelo de grandes superficies, mientras que el zoom sobre todas las áreas requiere un mayor nivel de detalle.

Algunas características principales de AIMSUN son las siguientes:

- Definición de las áreas para la simulación microscópica dentro de una red con características mesoscópicas.
- Recopilación de estadísticas.
- Asignación de tráfico dinámico basado en la elección de ruta estocástica.
- Asignación de tráfico dinámico basado en el equilibrio dinámico de los usuarios.
- Gestión del tráfico.
- Posee buena documentación.

El simulador microscópico en AIMSUN posee modelos muy similares al de la vida real, lo que hace que las simulaciones sean bastante realistas siempre y cuando el proyecto que se realice esté bien estructurado. Y el simulador mesoscópico permite la modelación dinámica de las redes, basando sus modelos en componentes del microsimulador ya utilizados y probados.

AIMSUN permite crear una total infraestructura de tráfico, empleando para ello gráficos en 2D y 3D con componentes como vehículos, semáforos, paradas de transporte público y otros, y una vez creada dicha infraestructura, se pueden insertar los datos de tráfico, mediante una Matriz O/D o una tabla de Estado del Tráfico.

C.6. GrooveSim

GrooveSim es un simulador de tráfico híbrido y de carácter propietario desarrollado por la Universidad de Pensilvania, la Universidad de Carnegie Mellon en Pittsburg y la empresa automovilística General Motors con el objetivo de simular protocolos móviles inalámbricos, empleando modelos de movilidad y comunicación en un escenario topográfico real a través de mapas digitales. Es utilizado además para la realización de pruebas de conducción, análisis y pruebas de estrés de redes vehiculares.

GrooveSim tiene cinco modos distintos de funcionamiento:

- Modo de transmisión: GrooveSim es capaz de formar conexiones UDP (*User Datagram Protocol*) con otros vehículos, que abarcan múltiples saltos con el protocolo de enrutamiento. Un vehículo es capaz de activar y enviar transmisiones de mensajes de seguridad y transferir archivos entre los vehículos que estén accesibles. Además, un vehículo puede establecer una conexión TCP sobre una conexión celular 1xRTT (*1 times Radio Transmission Technology*), por lo que un nodo remoto puede controlar la actividad de conducción en el camino a través de Internet. Las coordenadas GPS y las posiciones del vehículo se actualizan periódicamente con una tasa máxima de 5 Hz. Este modo evalúa la robustez y el protocolo de pruebas del tráfico real y las condiciones del canal.
- Modo Simulación: GrooveSim puede soportar miles de vehículos concurrentemente comunicándose y en movimiento. Cada vehículo puede tener su propia movilidad, viaje y modelo de comunicación. Los usuarios pueden optar por ver todos los vehículos gráficamente, lo que permite evaluar la escalabilidad y rendimiento bajo diversas cargas de tráfico. GrooveSim posibilita la descarga automática de los archivos TIGER / LINE a través de Internet si no están disponibles a nivel local.
- Modo de reproducción: los viajes pueden ser reproducidos utilizando controles como VCR y los usuarios con la interfaz gráfica pueden avanzar y rebobinar un archivo de registro, permitiendo el análisis visual con resultados reproducibles. Además, del análisis de los datos de rendimiento, tales como mensaje de distancia de penetración y demora, tamaño de los grupos de vehículos, transmisión / recepción de paquetes. También se registra la información relacionada con el GPS como el número de satélites visibles, partidas, posición y errores de sincronización.
- Modo de simulación híbrida: en este modo, pueden interactuar los vehículos reales en la carretera y los vehículos virtuales. Cada vehículo real puede alojar múltiples vehículos virtuales o los vehículos virtuales pueden alojar nodos remotos conectados a través de la conexión celular 1xRTT.
- Modo de Generación de Pruebas: este modo proporciona una prueba de generación de escenarios con 1000 vehículos, cada uno con diferentes modelos y parámetros, pudiendo especificarse el modelo de viaje y hora a través de la interfaz gráfica de usuario. Además, de las distribuciones de la velocidad y el tiempo inicial generadas a través de una aplicación externa, por ejemplo, Matlab

y Excel. Esto asegura una fácil repetitividad de los experimentos y permite al usuario modificar la configuración antes de cada ejecución de la simulación.

GrooveSim genera mapas de calles de cualquier lugar en los Estados Unidos mediante la importación de TIGER / LINE, que constituye una base de datos digital de características geográficas, tales como carreteras, ferrocarriles, ríos, lagos y límites legales de ese país. La base de datos contiene información sobre los segmentos de carretera: latitud y longitud, nombre, tipo, rangos de direcciones, límites de velocidad, y otras; siendo sus datos registros de tipo texto. GrooveSim se basa en el código abierto de *roadnav*, que es el navegador disponible en Windows, Linux y Mac que permite localizar la posición de un vehículo utilizando un GPS y realiza indicaciones verbales de una dirección, pero GrooveSim le añade una abstracción gráfica de las calles, redes, modelos de simulación y una plataforma de interfaz gráfica de usuario en QT.

C.7. NCTUns

NCTUns es un simulador y emulador de redes que permite que los dispositivos reales interactúen en tiempo real con una red simulada fiable y extensible, capaz de simular diferentes protocolos tanto en redes cableadas como inalámbricas. Desarrollado en el Laboratorio de Sistemas y Redes de la Universidad Nacional *Chiao Tung*, con el profesor S.Y. Wang al frente.

Algunas características de NCTUns son:

- Se puede utilizar fácilmente como un emulador. Un host externo en el mundo real puede intercambiar paquetes, por ejemplo, establecer una conexión TCP con nodos como *host*, *router* o estación móvil en una red simulada por NCTUns, como también lo pueden hacer dos servidores externos en el mundo real.
- Es compatible con la emulación distribuida de una amplia red con varios equipos. Cuando la red emulada tiene muchos nodos, muchas aplicaciones del mundo real necesitan varios dispositivos para poder ejecutarse sobre esos nodos, y conectarse a la red emulada, por otra parte, cuando en una red emulada se intercambian una gran cantidad de paquetes con los dispositivos del mundo real, una sola máquina puede no tener suficiente potencia de CPU (*Central Processing Unit*) y memoria para ejecutar la emulación en tiempo real. En tal situación, NCTUns puede dividir la red emulada en varias partes más pequeñas y cada parte puede ser emulada por una máquina NCTUns; como la emulación distribuida es totalmente automática, el usuario no se dará cuenta de que la emulación se lleva a cabo en varios equipos.
- Existe una buena integración entre emulación y simulación por muy complicada que sea la red simulada. El tráfico de la red de la vida real puede pasar a través de una red simulada e interactuar con el tráfico de red simulada.
- Se puede utilizar cualquier configuración de red UNIX y herramientas de monitoreo de la vida real. Por ejemplo, *ifconfig*, *netstat*, *tcpdump*, los comandos de *traceroute* se pueden ejecutar en una red simulada para configurarla o supervisarla.

- Se pueden simular varias redes importantes como son las redes LAN, redes móviles ad hoc, redes celulares GPRS, redes ópticas, redes LAN inalámbricas, comunicaciones por satélite DVB-RCS, redes vehiculares inalámbricas WAVE (*Wireless Access Vehicular Environment*), entre otras.
- Presenta una interfaz de usuario muy profesional que ayuda al usuario a trabajar más fácilmente: configurar módulos de protocolos, especificar rutas de movimiento de nodos móviles, dibujar topologías de las redes, hacer animaciones de una traza de paquetes, trazar gráficos de rendimiento de la red, entre otras tantas funcionalidades importantes.

No obstante, se podría decir que sus dos características más importantes son: que las aplicaciones del mundo real se puedan ejecutar directamente en nodos simulados para el intercambio de paquetes y que la configuración y uso de los programas de aplicación y red simulada sean exactamente los mismos que los utilizados en las redes IP de la vida real, por lo que si un usuario está familiarizado con las redes IP de la vida real podrá aprender y operar fácilmente NCTUns.

Tiene siete componentes principales: la interfaz gráfica de usuario, el gestor de trabajos, el coordinador, el motor de simulación, el *kernel* de Linux parcheado, las aplicaciones y los *daemons*. La interfaz gráfica permite al usuario crear, configurar y controlar una simulación/emulación, ejecutándola de forma fácil. El coordinador del programa se ejecuta en todos los equipos participantes para el seguimiento de su actual estado de ocupado/inactivo e informa del estado al programa gestor de trabajos; este último puede monitorizar los estados de varios equipos que participan en la recepción de los informes de estado enviados por los programas coordinadores que se ejecutan en estas máquinas.

La Figura C.1 muestra la arquitectura de las simulaciones y emulaciones distribuidas.

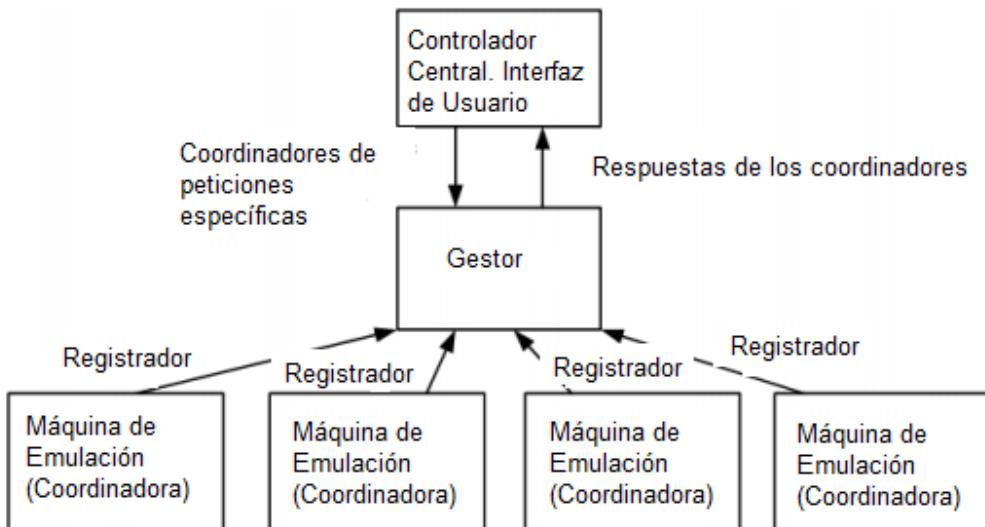


Figura C.1: Arquitectura de las simulaciones y emulaciones distribuidas.

C.8. iTETRIS

El simulador iTETRIS (*Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions*) tiene como objetivo proveer de una plataforma de simulación capaz de evaluar estrategias de tráfico basadas en sistemas cooperativos ITS (*Information Technology System*) a través del intercambio dinámico de mensajes entre vehículos y vehículo e infraestructura.

iTETRIS integra dos simuladores; SUMO y NS-3, el primero, permite la realización de la simulación del tráfico, mientras que el segundo, es utilizado para el intercambio de mensajes de comunicación entre estaciones ITS, no obstante esta herramienta fue desarrollada para adaptarse a cualquier plataforma de simulación que implemente la misma funcionalidad.

SUMO implementa el componente TRACI que es el que permite a un proceso externo interactuar con la simulación de tráfico. SUMO inicia su ejecución con TRACI activado, a la escucha en un puerto determinado, como un servidor y el iCS (*iTETRIS Control System*) como un cliente que controla la sincronización de SUMO con el resto de la plataforma. Por otro lado, está la interfaz INCI (*iTETRIS Network Control Interface*) que es un módulo de NS-3 que sirve como punto de entrada a las peticiones del iCS.

La Figura C.2 muestra la arquitectura de iTETRIS, como se puede observar se encuentran SUMO, NS-3, y iCS, este último permite desacoplar ambos simuladores haciendo que el desarrollo e integración de ambos sea independiente. iCS controla a SUMO mediante mensajes que desencadenan acciones que modifican el estado de los objetos involucrados en el tráfico y es responsable de que la simulación de NS-3 esté sincronizada con la de SUMO.

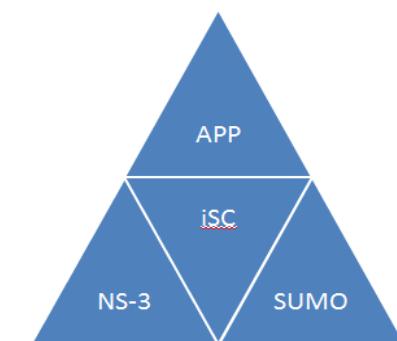


Figura C.2: Arquitectura de iTETRIS.

Los sockets basados en el protocolo IP es la tecnología encargada de la comunicación entre los distintos bloques de simulación (SUMO, iCS y NS-3) permitiendo que dos procesos se puedan comunicar estableciendo una dirección IP y puerto de escucha entre ellos, siendo el iCS el responsable de que la comunicación sea correcta y que la información que se comparta se lleve a cabo de forma eficaz.