



EBook Gratis

APRENDIZAJE omnet++

Free unaffiliated eBook created from
Stack Overflow contributors.

#omnet++

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con omnet ++.....	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Capítulo 2: Simulación de VANETs usando OMNeT ++ y Artery.....	3
Introducción.....	3
Examples.....	3
El marco de simulación de la arteria.....	3
Resumen de características:.....	3
¿Cómo funciona la arteria?.....	3
Instalación.....	4
Instalando OMNeT ++.....	4
Instalando SUMO.....	4
Instalando la arteria.....	5
Ejecución de un ejemplo.....	6
Ejecutando un ejemplo en modo de depuración.....	6
Agregando mi propio servicio ITS-G5: ¿Dónde debo empezar?.....	6
Creditos.....	8

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [omnetplusplus](#)

It is an unofficial and free omnet++ ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official omnet++.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con omnet ++

Observaciones

Esta sección proporciona una descripción general de qué es omnet ++ y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de omnet ++, y vincular a los temas relacionados. Dado que la Documentación para omnet ++ es nueva, es posible que deba crear versiones iniciales de los temas relacionados.

Examples

Instalación o configuración

Instrucciones detalladas sobre cómo configurar o instalar Omnet ++.

Lea [Empezando con omnet ++ en línea](#):

<https://riptutorial.com/es/omnetplusplus/topic/9564/empezando-con-omnet-plusplus>

Capítulo 2: Simulación de VANETs usando OMNeT ++ y Artery

Introducción

Esta documentación le dará una idea de cómo pueden simularse los VANET europeos basados en ETSI ITS-G5 e IEEE 802.11p utilizando el simulador de eventos discretos OMNeT ++.

Examples

El marco de simulación de la arteria

Artery (<https://github.com/riehl/artery>) es un marco de simulación para OMNeT ++ . Está diseñado para simular la pila de protocolos *ETSI ITS-G5* que se utiliza en las VANET europeas. Actualmente, está usando OMNeT ++ en la versión 5.0. Es de **código** completamente **abierto** y en **desarrollo activo** .

Resumen de características:

- Aplicaciones del Día Uno como *Cooperate Awareness Message* (CAM) y *Mensaje de Notificación Ambiental Descentralizado* (DENM) especificados por *ETSI ITS-G5*
- Proporciona *instalaciones* comunes para estas aplicaciones.
- Permite la creación rápida de prototipos para crear nuevas aplicaciones fácilmente.
- Enrutamiento geográfico, que es manejado por *Vanetza* (<https://github.com/riehl/vanetza>)
- *Vanetza* también cubre el *Protocolo Básico de Transporte* (BTP).
- Utiliza *Veins* (<http://veins.car2x.org/>) o *INET* (<https://inet.omnetpp.org/>) como capa MAC IEEE 802.11p; Estos proporcionan también varios modelos de propagación de onda de capa física
- Movimiento realista de vehículos provisto por el simulador de tráfico de código abierto SUMO (<http://www.dlr.de/>)
- Una forma fácil de crear diferentes escenarios de tráfico (como embotellamientos o influencias climáticas)

¿Cómo funciona la arteria?

La arteria se compone básicamente de cuatro partes. Primero, está la simulación de red que maneja el envío y la recepción de paquetes. Esto se basa en las redes OMNeT ++ , donde *Veins* o *INET* proporcionan modelos razonables y realistas del medio de radio. Además, *Veins* e *INET* proporcionan una implementación de las capas física y de enlace IEEE 802.11. Usted es libre de elegir el marco que desea usar al cambiar el archivo de configuración de la red.

El enrutamiento entre los participantes de la red es realizado por *Vanetza* . *Vanetza* incluye

algoritmos topológicos de enrutamiento ITS-G5 como el Single Hop Broadcast (SHB), así como enrutamiento geográfico como GeoBroadcasts (GBC). Cada estación ITS-G5 en la red está representada por un Router *Vanetza*. Actualmente, solo los vehículos se utilizan como estaciones ITS-G5, pero las Unidades del lado de la carretera (RSU) ya están en la hoja de ruta. Basado en el algoritmo de enrutamiento elegido, el enrutador determina el siguiente salto en la red y envía el paquete a la capa física, que es proporcionada por *Veins* o *INET*.

Las aplicaciones son la razón por la que los paquetes se crean de primera mano. La capa de aplicación es proporcionada por *Artery*. Las aplicaciones del *Día Uno* mencionadas por *ETSI* ya están implementadas y generan Mensajes de Concientización Cooperativa (CA) y Notificación Ambiental Descentralizada (DEN) de acuerdo con la definición de la norma. Para activar los mensajes DEN, se necesita la definición de los escenarios de activación adecuados. Para este propósito se puede utilizar el *Storyboard*. Permite definir condiciones de escenario y efectos para provocar diversos escenarios de tráfico como accidentes, condiciones climáticas o embotellamientos. De acuerdo con la situación del tráfico, la aplicación activa el mensaje DEN apropiado.

Por último, pero no menos importante, está la simulación de tráfico proporcionada por *SUMO*. *SUMO* y *Artery* se conectan mediante la interfaz *TraCI*. *TraCI* permite leer información sobre el estado actual de cada vehículo dentro de la simulación, así como cambiar los parámetros del vehículo. El cambio de parámetros del vehículo es utilizado principalmente por el *Storyboard* para lograr las situaciones de tráfico mencionadas anteriormente. Los mapas utilizados por *SUMO* se pueden derivar de mapas reales con flujos de tráfico realistas (como <https://github.com/lcodeca/LuSTScenario/wiki> o Open Street Map) o escenarios sintéticos.

Instalación

Es bastante fácil instalar *Artery* en sistemas basados en Linux. Debe tener un compilador de C++ con soporte de C++ 11, así como las bibliotecas *Boost* y *Vanetza* para construir *Artery*. Además, necesitas uno de *Venas* e *INET*. Por supuesto, también se requieren *OMNeT++* y *SUMO*.

Instalando OMNeT++

1. Descargue el archivo *OMNeT++ 5.0* (<https://omnetpp.org/omnetpp>).
2. Extrae el archivo en una nueva carpeta.
3. Escriba `./configure` y después de este tipo `make`.
4. Agregue su directorio `path/to/OMNeT/build` en su `PATH` entorno `PATH`.
5. Asegúrese de que su instalación funciona escribiendo `omnetpp`. Se espera que el entorno de desarrollo *OMNeT++* comience.
6. Se pueden encontrar más instrucciones de instalación en <https://omnetpp.org/doc/omnetpp/InstallGuide.pdf>.

Instalando SUMO

1. Para instalar *SUMO*, primero descargue la última versión (se puede encontrar en

- <http://www.sumo.dlr.de/userdoc/Downloads.html>) . No se recomienda usar la versión *SUMO* que viene con el sistema Ubuntu o Debian, ya que estas versiones son bastante antiguas.
2. Extrae los archivos descargados.
 3. Después de esto tienes que construir tu versión SUMO. Asegúrese de tener el paquete `libproj-dev` (en Debian o Ubuntu) instalado.
 4. Vaya al directorio, donde ha extraído SUMO.
 5. Escriba: `./configure` . Una vez realizada la configuración, consulte el `Optional features summary` , que se imprimió al final. Debe incluir la entrada `PROJ` en la lista `Enabled:` . Esta función necesita el `libproj-dev` mencionado anteriormente y permite a SUMO proporcionar coordenadas geográficas para cada vehículo. **Sin esta característica, la simulación se detendrá en tiempo de ejecución debido a coordenadas no válidas.**
 6. Escriba: `make` para compilar SUMO en el directorio `./build` .
 7. Agregue la `path/to/your/sumo-version/build` a su `PATH` entorno `PATH` .
 8. Verifique su instalación escribiendo `sumo-gui` .
 9. Puede encontrar más información sobre la instalación de *SUMO* en <http://www.sumo.dlr.de/userdoc/Installing.html>
-

Instalando la *arteria*

Para instalar *Artery* , necesitas tener *Vanetza* , *Veins* e *Inet* . Para facilitar la instalación, el repositorio de *Artery* contiene todos estos marcos como enlaces de sub-repositorio. Esto garantiza que solo se utilicen versiones compatibles de * *INET* , *Veins* y *Vanetza* con *Artery* .

1. Tire del repositorio *Artery* desde <https://github.com/riegl/artery>
2. Para compilar *Vanetza* , asegúrese de estar en el directorio raíz y escriba `make vanetza` . *Vanetza* ahora se construirá en `extern/vanetza/build`
3. Para compilar *Inet* , asegúrese de estar en el directorio raíz de *Artery* y escriba `make inet` . *Inet* ahora será construido.
4. Para crear *Veins* , asegúrese de estar en el directorio raíz de *Artery* y escriba `make veins` . *Las venas* ahora se construirán en `extern/veins/build`
5. Para construir *Artery* , asegúrese de estar nuevamente en el directorio raíz. Escriba `mkdir build` .
6. Escriba `cd build`
7. Escriba `cmake ..`
8. Tipo `cmake --build`

Puede encontrar más información sobre la construcción de las herramientas mencionadas anteriormente en las siguientes fuentes. Esto puede ser útil en caso de un error.

1. *Vanetza* : <https://github.com/riegl/vanetza>
2. *Arteria* : <https://github.com/riegl/artery>
3. *Inet* : <https://github.com/riegl/artery/blob/master/extern/inet/INSTALL>
4. *Venas* : <http://veins.car2x.org/>

Felicitaciones, ahora tiene todos los requisitos previos para avanzar e intentar lanzar el primer ejemplo de *Arteria* . Cómo se hace esto se explica en la siguiente sección!

Ejecución de un ejemplo

Como *Arteria* es bastante compleja, se recomienda comenzar por entender *OMNeT++*. Un buen punto para comenzar es el tutorial de *TicToc*. Se puede encontrar en <https://omnetpp.org/doc/omnetpp/tictoc-tutorial/>. Este tutorial proporciona una descripción general de la funcionalidad básica de *OMNeT++*. Esto incluye, entre otros, el lenguaje *NED* y la definición de *Redes*.

Si ya se *contactó* con *OMNeT++*, puede intentar iniciar el ejemplo enviado con *Artery*. Para hacerlo, sigue estos pasos:

1. Navegue al directorio de `build` de *Artery* (que se creó mientras se construía)
2. Escriba `make run_example`. Este comando verifica los cambios de código (como lo hace `make`) y luego *ejecuta* *OMNeT++* utilizando `omnetpp.ini` en la carpeta `scenarios/artery`.
3. Si desea que `SUMO` abra su GUI mientras simula, agregue esta línea a su `omnetpp.ini`:
`*.traci.launcher.sumo = "sumo-gui"`

Ejecutando un ejemplo en modo de depuración

Si desea investigar el código mientras se ejecuta la simulación, debe crear *Artery* en modo de depuración. Para ello se recomienda utilizar `ccmake`.

1. Navega a tu directorio de `build` de *Arteria*
2. Escriba `ccmake`.
3. Vaya a la entrada `CMAKE_BUILD_TYPE` (generalmente esta entrada está en la primera posición superior) presione `[enter]` y escriba: `Debug`. A continuación, pulse `[enter]`.
4. Presione `[c]` para configurar.
5. Presione `[q]` para salir.
6. De nuevo, escriba `make` en el directorio de `build`.
7. Para ejecutar el ejemplo en modo de depuración, escriba `make debug_example`
8. Una vez que el depurador *GDB* esté listo, escriba `run`
9. Siéntase libre de presionar `[ctrl + c]` para ingresar al depurador y establecer sus puntos de interrupción

Agregando mi propio servicio ITS-G5: ¿Dónde debo empezar?

Para comenzar a crear su propio servicio, eche un vistazo a la clase base de cada servicio, que se llama `ItsG5Service`. También puede consultar `CamService` y `DenmService` ya que ya son servicios implementados. Todos los archivos y clases relacionados con la aplicación se pueden encontrar en la subcarpeta `artery/application`.

La inicialización de un servicio es realizada por `ItsG5Middleware`. Antes de agregar su servicio, proporcione una clase de `C++` adecuada y un archivo `.ned` que pertenezca a su clase. Su clase de servicio debe derivarse de `ItsG5Service`. Entonces, ponga su servicio en `examples/yourExample/services.xml`. Como punto de partida, puede copiar la entrada de `CamService` (`examples/artery/services.xml`) y cambiar el número de puerto.

El `ItsG5Middleware` también invoca los servicios en cada paso de simulación llamando a `ItsG5Service::trigger()` . Esto significa que, si desea hacer algo periódicamente con su servicio, anule este método y coloque su código allí.

Lea [Simulación de VANETs usando OMNeT ++ y Artery en línea](https://riptutorial.com/es/omnetplusplus/topic/9675/simulacion-de-vanets-usando-omnet-plusplus-y-artery):

<https://riptutorial.com/es/omnetplusplus/topic/9675/simulacion-de-vanets-usando-omnet-plusplus-y-artery>

Creditos

S. No	Capítulos	Contributors
1	Empezando con omnet ++	Community
2	Simulación de VANETs usando OMNeT ++ y Artery	Ventu