**概述**

Simulation of Urban Mobility（SUMO）作为一款交通仿真软件，其通过结合OpenStreetMap官网提供的真实道路网可以生成逼真的车辆交通轨迹。OPNET是一款流行的网络仿真软件，它通过建立网络设备、通信链路和协议模型并模拟流量的传输，来评估网络性能并对其进行设计优化，可以被很好应用于车辆网通信仿真。然而，SUMO生成的轨迹文件不直接提供对OPNET仿真的支持。为此，本项目采用JAVA语言开发了一款可以实现SUMO生成轨迹转换为OPNET仿真轨迹的工具，旨在实现二者的联合仿真以为车联网的研究提供更加真实的仿真场景。

**部署环境**

Win 7 + java-1.8.0-openjdk-devel.x86\_64

SUMO-1.6.0

OpenStreetMap

OPNET 14.5

**部署步骤**

整体的部署过程如图1所示，详细的步骤如下所述：



图1.基于SUMO和OPNET联合的车联网仿真

**Step 1**：OSM真实地图文件获取

打开openstreetmap官网（https://www.openstreetmap.org），点击右上角的“导出”，在4个空白框中输入需要选择的区域经纬度范围，在次点击蓝色背景框的“导出” 生成该区域的OSM地图文件***map.osm***，具体如图2所示。



图2.OSM真实地图文件获取

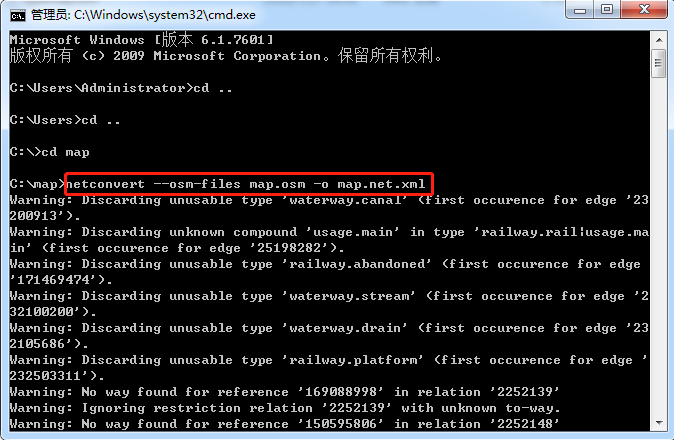
官网下载的osm除了路网信息还包含别的东西，需要做简单处理：手动打开sumo\doc\userdoc\Networks\Import\OpenStreetMap.html文件，然后在打开的网页中间部分 Importing additional Polygons (Buildings, Water, etc.)的下面有一大段xml的代码，复制这些代码，然后在notepad++中级新建文件并把复制的代码粘贴到里面，在本实验中因为不需要，所以删除了下面的id=“power”的那一行代码，然后保存为***typemap.xml***, 把这个文件保存到上一步骤建的map文件中去，和osm地图文件放一起。

**Step2**: 生成***map.net.xml***和***map.poly.xml***文件

在cmd命令框中打开***map***文件夹，然后依次运行如下两条命令（如图3所示）：

(1) netconvert --osm-files map.osm -o map.net.xml

(2) polyconvert --net-file map.net.xml --osm-files map.osm --type-file typemap.xml -o map.poly.xml



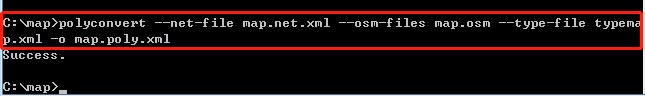


图3.生成map.net.xml和map.poly.xml文件

**Step3：**生成***map.rou.xm****l*文件

在cmd命令框中，继续依次运行如下两个命令，如图4所示.

(1) python C:/sumo/tools/randomTrips.py -n map.net.xml -e 100 –l

(2) python C:/sumo/tools/randomTrips.py -n map.net.xml -r map.rou.xml -e 400 -l

(命令2中400表示场景中设定的车辆数量，使用者可以结合仿真场景自行设定)

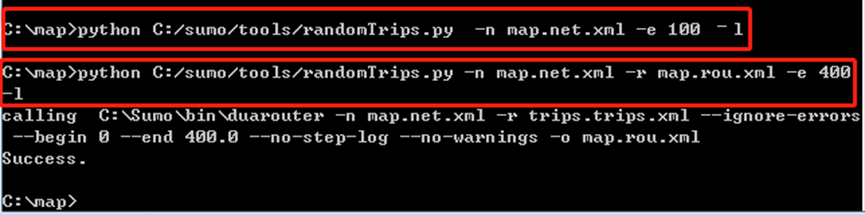


图4.生成map.rou.xml文件

**Step4**：编辑配置文件***myConfig.sumocfg***

可以在sumo文件夹中搜索test.sumocfg然后更改input文件，便可以得到如下图5所示文件，然后把文件命名为***myConfig.sumocfg***并保存到map文件夹中



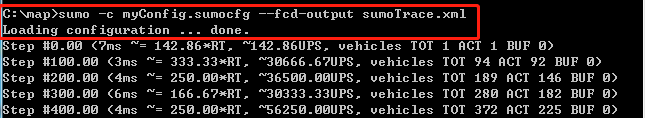
图5. 编辑配置文件myConfig.sumocfg

Step5：生成SUMO轨迹文件***ns2mobility.tcl***

在cmd命令框中，依次运行下面两条命令，如图6所示。

(1) sumo -c myConfig.sumocfg --fcd-output sumoTrace.xml

(2) traceExporter.py --fcd-input sumoTrace.xml --ns2mobility-output ns2mobility.tcl

**

*C:\Users\ADMINI~1\AppData\Local\Temp\WeChat Files\8e090c9b65e20f740dd6184d151d0ca.png*

图6.生成SUMO轨迹文件ns2mobility.tcl

**Step6：**SUMO轨迹文件***ns2mobility.tcl***转换为OPNET仿真软件支持的文件格式 (\*.***trj***)

(1)在Java IDE中新建工程**SUMO\_Pretreatment**，并运行代码***SUMO\_Pretreatment.java***, 将SUMO轨迹预处理的结果按单个车辆分类写入文件夹SUMO\_F。

(2) 在Java IDE中再次新建工程**SUMO\_To\_OPNET\_Trj**，并运行代码***SUMO\_To\_OPNET\_Trj.java***, 将SUMO轨迹预处理的结果集逐一转化为OPNET轨迹文件

**Step7：OPNET轨迹文件导入车联网仿真场景**

将第6步生成的车辆轨迹文件复制到OPNET安装文件夹op\_models中，新建OPNET工程，添加节点并关联移动轨迹。OPNET 的一大优点是提供了方便的图形化建模，但是有时候图形化建模方式并不能满足我们的需求，比如需要部署几千个节点，如果一个个手工部署这些节点，则太过烦琐，用程序随机化部署有可能导致节点位置随机化一部分节点簇拥到一起，而有些地方又无法部署到。我们在随机部署节点时，随机生成下一网络节点的坐标；在生成节点坐标的过程中，要避免让新的节点坐标距离已有的节点坐标太近。针对这个需求采用了文本方式建模(external model access, EMA) , EMA 可以精确定义节点属性，可以用循环语句来刻画多个特定规格的节点，其具体步骤如下：

（1）在图形界面下做一个简单的网络模型，只包含一个移动节点，如下图7(a)所示。.

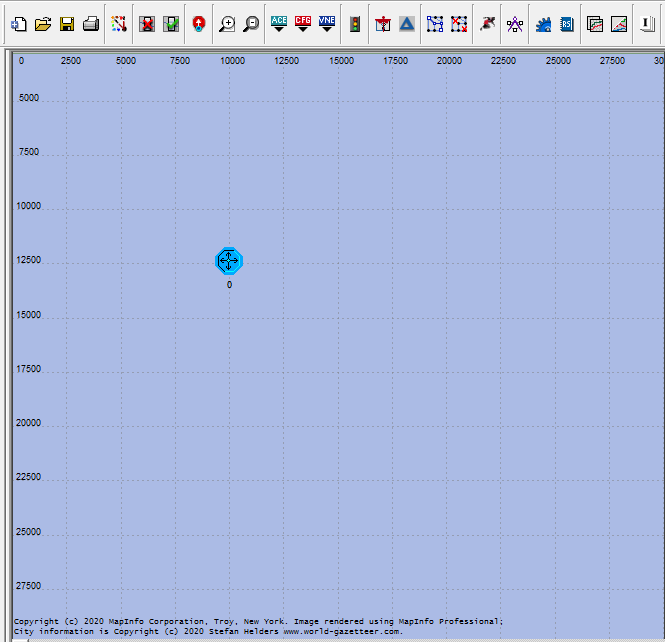


图7(a).单节点网络模型

（2）选择菜单**Topology**->**Export Topology**->**To EMA** 生成EMA文件(***\*.em.c***), 观察EMA文件，查找name属性为"0"的节点，代码如下图7(b)所示。

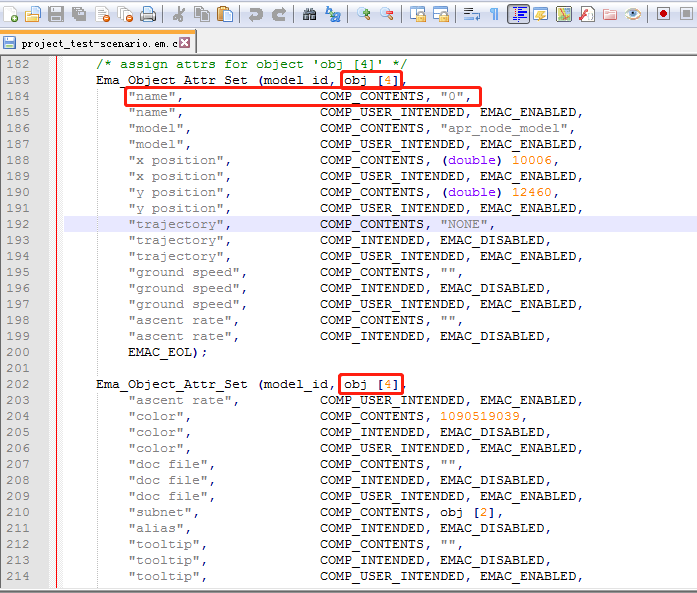


图7(b). EMA

从上面可以看出，对象数组下标4为0号节点（注意在你的场景中不一定是obj[4])。在后面用for循环将这些关于obj[4]的语句括起来。

(3) 添加要用到的头文件。

a.因为要用到rand()函数，因此需要添加头文件stdlib.h，即#include <stdlib. h>。

b.在main()函数前添加需要用到的变量，如下图8所示。

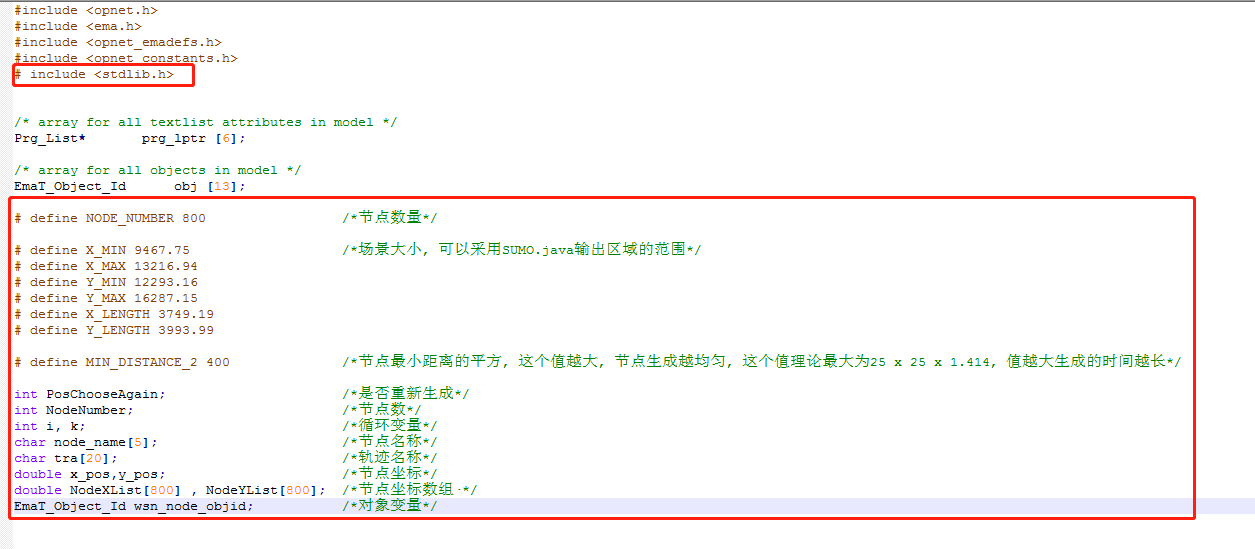


图8.添加头文件

(4) 接下来设置节点的属性（如图9所示）：将obj [4]的代码（主要是***ema\_obj\_attr\_set()***函数）手动放到一个for循环语句里，替换其中的节点名称属性和坐标(x\_postion和y\_position )为变量，通过循环生成其他节点，在for循环中随机生成节点的纵横坐标，将下面代码放到文件的末尾。

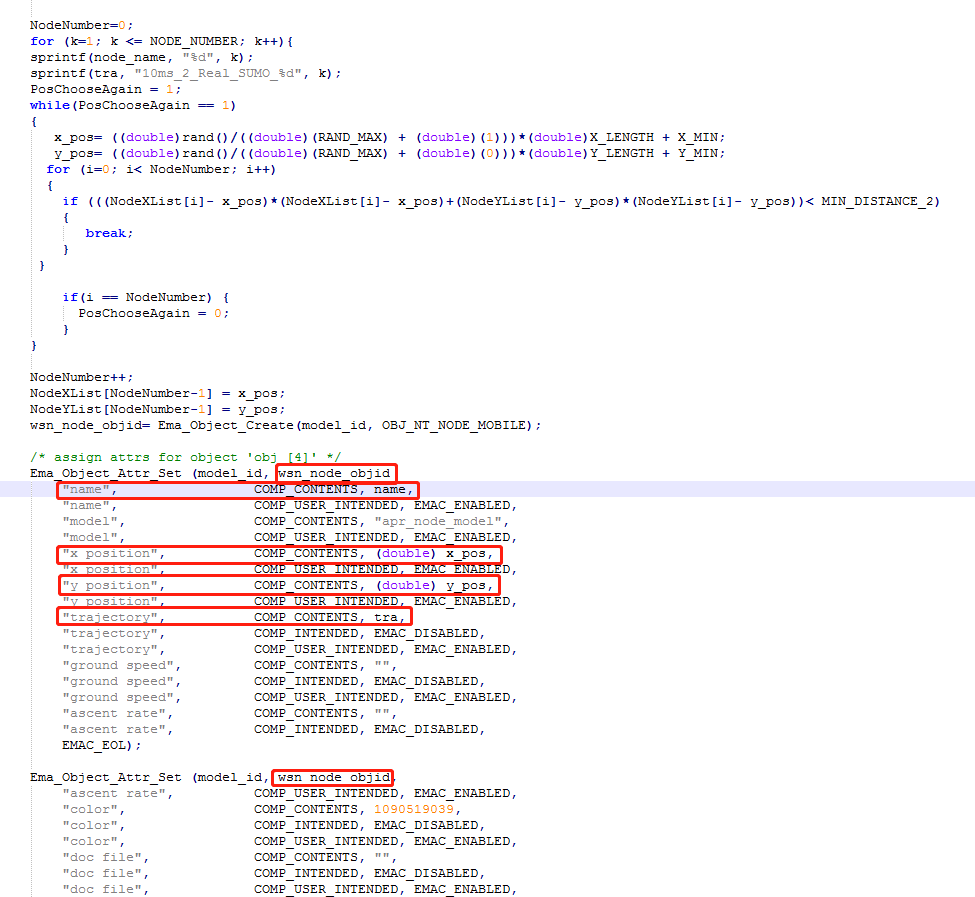


图9.设置节点属性

(5) 打开OPNET 控制台窗口，进入修改过的***\*.em.c***所在目录，分别执行如下命令，如图10所示。

① 输入命令op\_mkema -m 文件名＜不加后缀＞，执行成功后生成 ***\*.dev32.i0.em.x***

② 执行刚创建的可执行文件 <文件名.***i0.em.x***>, 将生成新的模型文件(***\*.m***)

C:\Users\ADMINI~1\AppData\Local\Temp\WeChat Files\538fe8ffad32f326016c20828b23afa.png

C:\Users\ADMINI~1\AppData\Local\Temp\WeChat Files\51d0b2bae909174e346960e20f2bc3e.png



图10.设置节点属性

(6) 在OPNET菜单中单击**File**->**Model Files**->**Refresh Model Directories** 选项刷新模型目录，否则新创建的模型文件在已启动的OPNET中是不可见的。生成的网络模型，需要导入项目场景中才能看到，在项目编辑器中选择**Scenarios**->**Scenario Components**->**import**选项，得到的模型如下图所示，可发现节点大致均匀分布，并且关联了我们采用SUMO生成的真实道路轨迹，如图10所示。



图11.基于真实轨迹的车联网仿真