Summary

SUMO, as a traffic simulation software, can generate realistic traffic trajectory of vehicle by combining with the real road network provided by OpenStreetMap official website. OPNET, a popular network simulation software, establishes network equipment, communication link and protocol model and simulating traffic transmission to evaluates network performance and optimizes its design by, which has been applied to vehicle network communication simulation well.

However, the trajectory files generated by SUMO do not directly support OPNET simulation. Therefore, this project uses Java language to develop a tool that can convert trajectory generated by SUMO into OPNET simulation trajectory, aiming to realize the joint simulation of them, and provide a more real simulation scene for the research of VANET.

Deployment environment

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

SUMO-1.6.0

OpenStreetMap

OPNET 14.5

Deployment steps

The overall deployment process is shown in Fig.1, and the detailed steps are as follows:



Fig.1 The VANETs simulation based on co-simulation of SUMO and OPNET

**Step 1**: OSM real map file acquisition (shown as Fig 2.)

Open the OpenStreetMap website (*https://www.openstreetmap.org* ) , click "*export*" in the upper right corner, input the longitude and latitude range of the area to be selected in the four blank boxes, and then click "*export*" in the blue background box to generate the OSM map file *map.osm.*



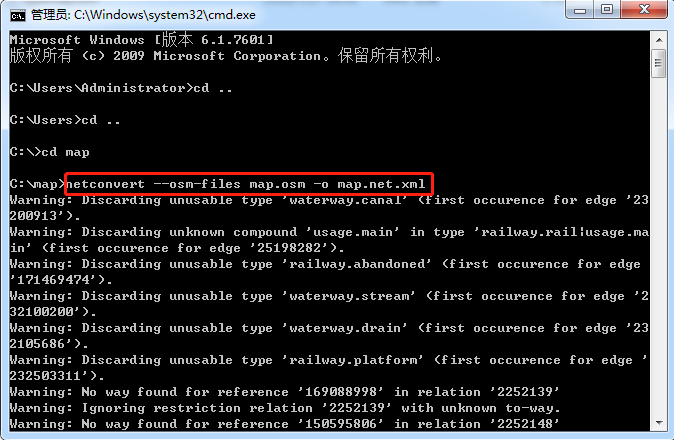
Fig.2.OSM real map file acquisition

**Step 2**: Generate the ***map.net.xml*** and ***map.poly.xml*** files (shown as Fig.3.)

Put the files ***map.osm***and ***typemap.xml***into the new folder named***map***, and then open the ***map*** folder in cmd command box to run the following two commands in turn:

(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



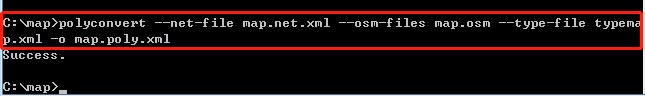


Fig.3. Generate the ***map.net.xml*** and ***map.poly.xml*** files

**Step3**: Generate ***map.rou.xml*** file (shown as Fig 4.)

Open the ***map*** folder in cmd command box to run the following two commands in turn:

(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

(400 indicates the number of vehicles set in the scene, which can be set by the user in combination with the simulation scenario)

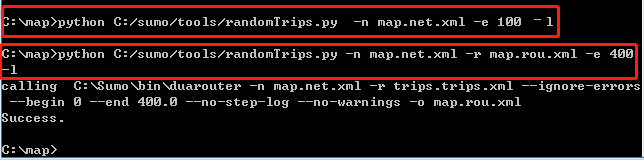


Fig.4. Generate map.rou.xml file

**Step4**: Edit configuration file *myConfig.sumocfg.* (shown as Fig 5.)

You can search the file *test.sumocfg* in the *sumo* folder and then change the input codes “<input> ……</input> ” to get the file and save it into the *map* folder named as *myConfig.sumocfg*.



Fig.5. Edit configuration file *myConfig.sumocfg*

**Step5**: Generate SUMO trace file ***ns2 mobility.tcl****.*(shown as Fig 6.)

Open the ***map*** folder in cmd command box to run the following two commands in turn:

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

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

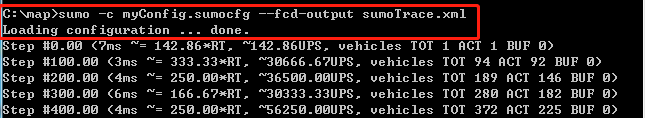
*C:\Users\ADMINI~1\AppData\Local\Temp\WeChat Files\8e090c9b65e20f740dd6184d151d0ca.png*

Fig.6.Generate SUMO trace file *ns2 mobility.tcl*

**Step6:** SUMO trace file ***ns2Mobility.tcl*** converts to the file format (***\*.trj*** ) supported by OPNET simulation.

(1) Create a new project named ***SUMO\_Pretreatment*** in Java IDE and run code***SUMO\_Pretreatment.java***.The result of SUMO trajectory preprocessing will be written to subfolder ***SUMO\_F*** by individual vehicle classification.

(2) Create a new project named ***SUMO\_To\_OPNET\_Trj*** in the Java IDE and run the code ***SUMO\_To\_OPNET\_Trj.java*.** The result set of SUMO trajectory preprocessing will be converted into OPNET trajectory file one by one and stored in the subfolders ***OPNET\_Trj.***

**Step7:** Import OPNET trajectory files into VANETs simulation scenario

Copy the vehicle trajectory files generated in step 6 to the OPNET installation folder ***op\_models*** and create a new OPNET project, add nodes and associate trajectories.

One of the advantages of OPNET is that it provides convenient graphical modeling, but sometimes graphical modeling methods do not meet our needs, such as the need to deploy thousands of nodes. It is too cumbersome to deploy these nodes manually one by one, while randomized deployment with a program may result in randomized node locations where a part of the node cluster is clustered together and in some places cannot be deployed to. Therefore, when we randomly deploy nodes, we can randomly generate the coordinates of the next network node; at the same time, in the process of generating node coordinates, we should avoid making the new node coordinates too close to the existing node coordinates.

In response to this requirement, we adopt a text-based model access (EMA), which can precisely define node attributes and can use circular statements to characterize nodes with multiple specific specifications, as follows:

(1) Make a simple network model with only one mobile node under the graphical interface, shown as Fig.7(a).

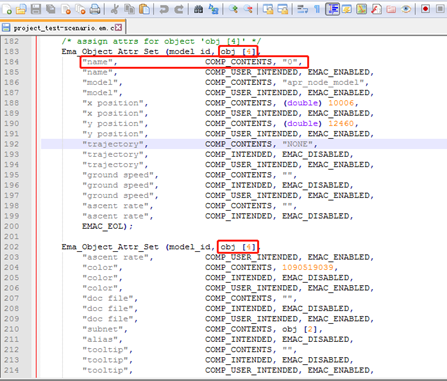
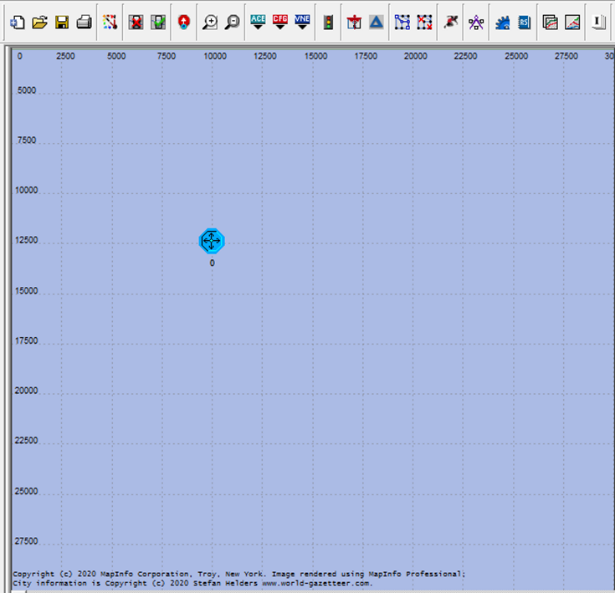
(2) Select the menu **Topology-->Export Topology-->To EMA** to generate the EMA file (***\*. em. c***), and find the node whose name attribute is "0" in the EMA file. The code is shown as Fig.7(b)

Fig.7(a). A simple network model. Fig.7(b). EMA file

（3）Add header file as follows Fig.8

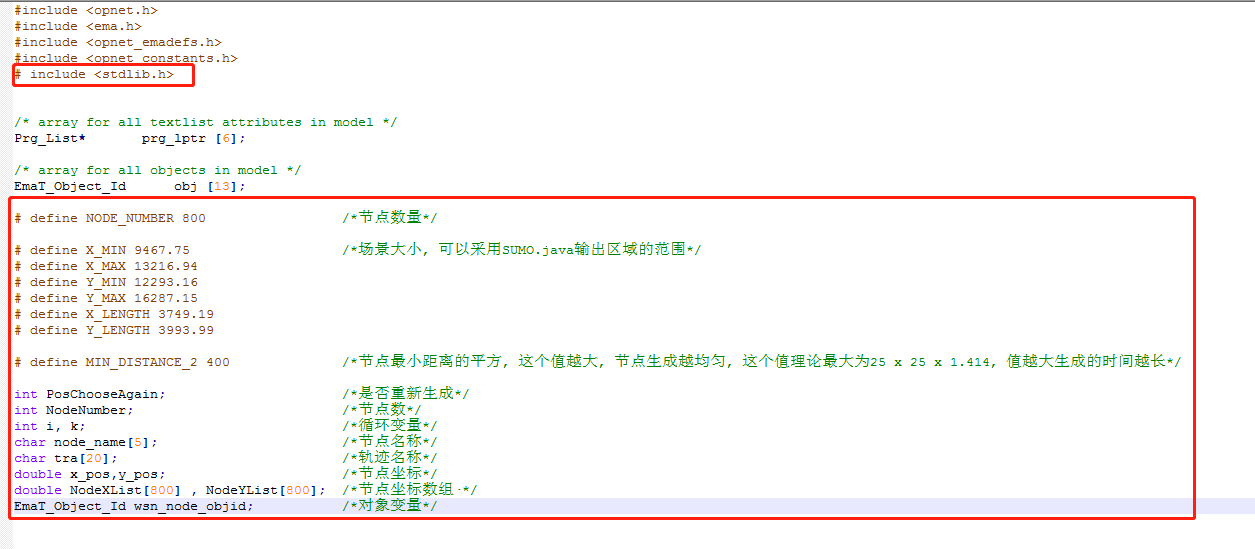


Fig.8. Add header file

（4）Next set the properties of the node (show as Fig.9)：we place the code of obj [4] (***ema\_obj\_attr\_set()***) in a for-loop statement, replacing the node name attribute and coordinates (x\_postion and y\_position) as variables, which makes the other nodes are generated by looping.

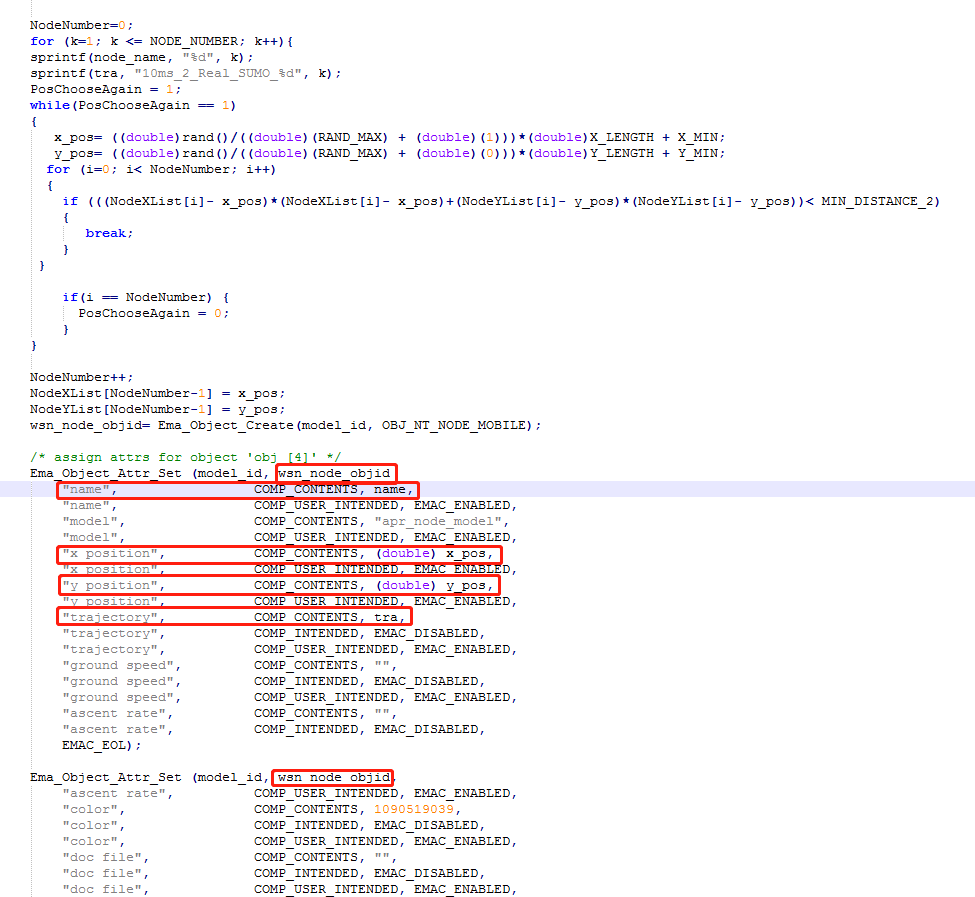


Fig.9.Set the properties of the node

(5) Open the OPNET console and enter the directory where the modified file ***\*. em.c***is located and execute the following commands in turn (shown as Fig.10):

① Enter the command ***op\_mkema - m file name*** < without suffix >, after successful execution generates ***\*.dev32.i0.em.x.***

② Executing the newly created executable file < ***file name. i0. em. x***> to generates a new model file (***\*. m***)

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

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



Fig.10. Generate the file (***\*.m* )**

(6) Click the **File->Model Files->Refresh Model Directories** option in the OPNET menu to refresh the model directory, otherwise the newly created model file (*\*. m*) is not visible in the OPNET. The generated network model, which needs to be imported into the project scenario. And then select **Scenarios->Scenario Components->import** in the project editor, the resulting model is shown in the following figure, and it can be found that the nodes are roughly uniformly distributed and associated with the real road trajectory we generated using SUMO (shown as Fig.11).



Fig 11. VANETs simulation under real-road network