## 配置安装过程（在ubuntu12.04-32完成配置安装）：

### 下载SUMO

[http://sourceforge.net/projects/sumo/files/sumo/version%200.23.0/sumo-src-0.23.0.tar.gz/download](http://sourceforge.net/projects/sumo/files/sumo/version0.23.0/sumo-src-0.23.0.tar.gz/download)

### 二、安装依赖包

依赖包：fox、proj、xerces等

sudo apt-get install libgdal1-dev proj libxerces-c2-dev

sudo apt-get install libfox-1.6-dev libgl1-mesa-dev libglu1-mesa-dev

sudo apt-get install g++

如果出现如下错误：

![A description...](data:None;base64,)

输入：

sudo apt-get update

或者更换源。

### 三、建立链接

sudo ln -s /usr/lib/libgdal1.7.0.so /usr/lib/libgdal.so

### 四、解压sumo源码包并移动

tar -xzvf sumo-src-0.19.0.tar.gz

sudo mv -v sumo-0.19.0 /usr/local/src

### 五、正式安装SUMO

cd /usr/local/src/sumo-0.19.0

./configure --with-fox-includes=/usr/include/fox-1.6 --with-gdal-includes=/usr/include/gdal --with-proj-libraries=/usr --with-gdal-libraries=/usr --with-proj-gdal

make

sudo make install

### 六、安装完成启动SUMO

sumo

sumo-gui

![A description...](data:None;base64,)

### 补充：

1、安装sumo make时提示：exec: g++: not found

解决方案：安装gcc-c++  
  
2、安装sumo make时报错

libtool: link: unsupported hardcode properties  
libtool: link: See the libtool documentation for more information.  
libtool: link: Fatal configuration error.

## 使用说明：

首先SUMO模拟器需要根据配置节点和道路的文件来生成一个交通网络；其次，根据编写的车辆的信息和交通网络可以获得所需要的车辆的流文件；最后将车辆的流文件转换为我们所需要的GPS数据。

1. 先生成路网

编写node文件：

<nodes>

<node id="node1" x="0.0" y="0.0" type="priority"/>

<node id="node2" x="1000.0" y="1000.0" type="traffic\_light"/>

<node id="node3" x="2000.0" y="2000.0" type="priority"/>

</nodes>

这里的node就是道路中的每个路口，x，y就是相应的坐标。Type中的traffic\_light就是交通等的类型。通常后缀名为.nod.xml。

将node文件用高斯投影反算公式文件处理，将node文件中node的直角坐标系转换成大地坐标系

(即将文件名改为input.nod.xml，并将其放到~/dtn-experiment/CustomMap/utm2gps/下

执行./ utm2gps，生成文件output.nod.xml，即为所得)

编写公路文件：

<edges>

<edge id="edge1-2" from="node1" to="node2" priority="75"

numLanes="2" speed="40" />

<edge id="edge2-3" from="node2" to="node3" priority="75"

numLanes="2" speed="10" />

</edges>

而边的设定就是从一个点到另一个点，numLanes是车道数目，speed是这条道上的最大速度。通常后缀名为.edg.xml。

使用如下命令，将node文件和edge文件转化为公路网，一般后缀名为.net.xml。

**netconvert --node-files=test.nod.xml --edge-files=test.edg.xml --proj.utm true --output-file=test.net.xml**

生成车辆路径：

<flows>

<flow id="flow0" from="edge1-2" to="edge2-3" begin="0" end="250" number="1" />

</flows>

这里的生成方式是采用流的形式。从一个其实边，到目标边。Begin和end是开始时间和结束时间。后缀名一般为.flow.xml。结合公路网和车辆的流可以生成想要的车辆路径文件。

(number是将在end-start时间内循环的圈数

通常想要得到合适的圈数时，需要进行计算。

首先计算一个flow的总长度，然后将总长度除以aodv.edg.xml中该flow所在edge的 speed，就可得到该车辆在该flow走一圈的时间，再将(end-start)/一圈的时间，就 可得到如果车辆一直循环的情况下的次数，可以用这个值作为flow中number参数的 参考。如果number大于这个值，那么上一辆车还没达到终点，第二辆车就已经出发， 如果小于，则第一辆车已经到达终点后再停止一会才会发出第二辆车。具体车辆的行 为可用sumo-gui观察)

**duarouter -n test.net.xml -f test.flow.xml -o test.rou.xml**

**duarouter -n test.net.xml -f test.flow.xml -o test.rou.xml**

1. 生成车辆的流数据

编写配置文件：

<configuration>

<input>

<net-file value="test.net.xml"/>

<route-files value="test.rou.xml"/>

<additional-files value=""/>

<junction-files value=""/>

</input>

<time>

<begin value="0"/>

<end value="250"/>

<time-to-teleport value="-1"/>

<srand value="23423"/>

<route-steps value="-1"/>

</time>

</configuration>

配置文件中的input中引用相应的公路网文件和相应的车辆路径文件，然后采用如下命令可以得到想要的trace文件。

**sumo -c test.sumo.cfg --fcd-output trace.xml(一般不执行该命令,生成后的文件的坐标不对)**

**sumo -c test.sumo.cfg --fcd-output.geo --fcd-output trace.xml(执行该 命令)**

或者使用**sumo-gui** **-c test.sumo.cfg**来查看车辆的实际效果

(打开sumo-gui后，点击菜单栏的file->open simulator，打开test.sumo.cfg.然后工具栏右方有一个delays，调整该数值，数值越大，车辆移动速度越慢，越小，车辆移动速度越快，推荐80。调整后，按绿色三角，即可观察

)

1. 生成GPS数据

根据上一步生成的Trace文件，利用sumo的tools里面的工具，将其转成想要的GPS文件。

**sudo ./traceExporter.py --fcd-input <trace.xml> --gpsdat-output <GPSTrace.xml>(一般不执行该命令)**

**/usr/local/src/sumo-0.23.0/tools/traceExporter.py --fcd-input=trace.xml --gpsdat-output=gps.txt(执行该 命令，但是traceExporter.py 的目录可能不一样)**

4.处理gps.txt文件，变换格式

将格式从

flowA.0 2016-12-08 21:44:11.664441 808.08 698.11 70 0.0

变换为

1 1481204651664 808.08 698.11 70 0.0

具体做法为：将gps.txt复制到~/dtn-experiment/CustomMap/gpsTImeConvert/trace\_manage中,执行trace\_manage即可，运行后便可生成track.txt

(该程序的flowA 变为1,flowB 变为2 ,是写死的，如果需要更多的转换，自己去改程序)

## 参考资料：

Sourceforge网址：<http://sourceforge.net/p/sumo/wiki/Home/>

Wiki网址：<http://sumo.dlr.de/wiki/Main_Page>

中文版文档：<http://blog.csdn.net/sxsj333/article/details/20408579>

Sumo导出GPS数据：[http://www.sumo.dlr.de/wiki/Tools/TraceExporter#GPSDAT](#GPSDAT)

如何由OSM地图生成道路文件，并生成trace文件。

1. 从OSM官网导出.osm地图。（可以先采取josm对地图进行相关的编辑，<http://josm.openstreetmap.de/>）
2. 利用命令：netconvert --osm-files map.osm.xml -o map.net.xml 将osm地图转变为sumo的net.xml。
3. 编写相关的车辆路径，配置文件，生成想要的GPS-trace文件。

其他方法：

在原有生成的net.xml文件中，参照由osm生成的net.xml，然后在里面加入相应的实际系统的GPS信息。

加入其实位置的的经纬度坐标

|  |  |  |
| --- | --- | --- |
| netOffset | offset ([*<2D-POSITION>*](#Referenced_Data_Types)) | The offset applied for moving the network to (0,0) |

，参考链接：<http://sumo.dlr.de/wiki/Networks/SUMO_Road_Networks>

sumo -c test.sumo.cfg --fcd-output.geo --fcd-output trace.xml

~/Desktop/sumo/sumo-0.23.0/tools/traceExporter.py --fcd-input=trace.xml --gpsdat-output=gps.txt

/\*netconvert --offset.x=116.3120 --offset.y=39.9581 --node-files=test.nod.xml --edge-files=test.edg.xml –output-file=test.net.xml\*/

-422353.93 -4400629.27

参考链接：<http://blog.csdn.net/sxsj333/article/details/22864569>