

华东师范大学软件工程学院实验报告

实验课程：无线网络安全	姓名：	学号：
实验名称： VanetMobiSim模拟车辆轨迹	实验日期：2023.11.21	指导老师：张磊

实验目的

- 利用NS 3，基于LTE-V协议模拟简单车联网环境进行车辆之间的通信。
- 使用VanetMobiSim模拟车辆轨迹。

实验内容与实验步骤

- VanetMobiSim安装
- 随机生成50个对等节点（趋向于均匀分布）
- 通信协议任选（如使用Zigbee 协议框架，802.11a）
- 最大传输速率200kb/s，最大通信距离150m，源节点与目标节点间距1KM以上。

实验环境

- NS3网络模拟套件
- Ubuntu 18.04

实验过程与分析

0.安装VanetMobiSim

官网提供的安装方式有两种：二进制文件或源码编译，其中源码编译需要CanuMobiSim这款软件的源码，两者整合后编译，而这款软件官网已经失效。因此我们只能考虑二进制文件直接运行。

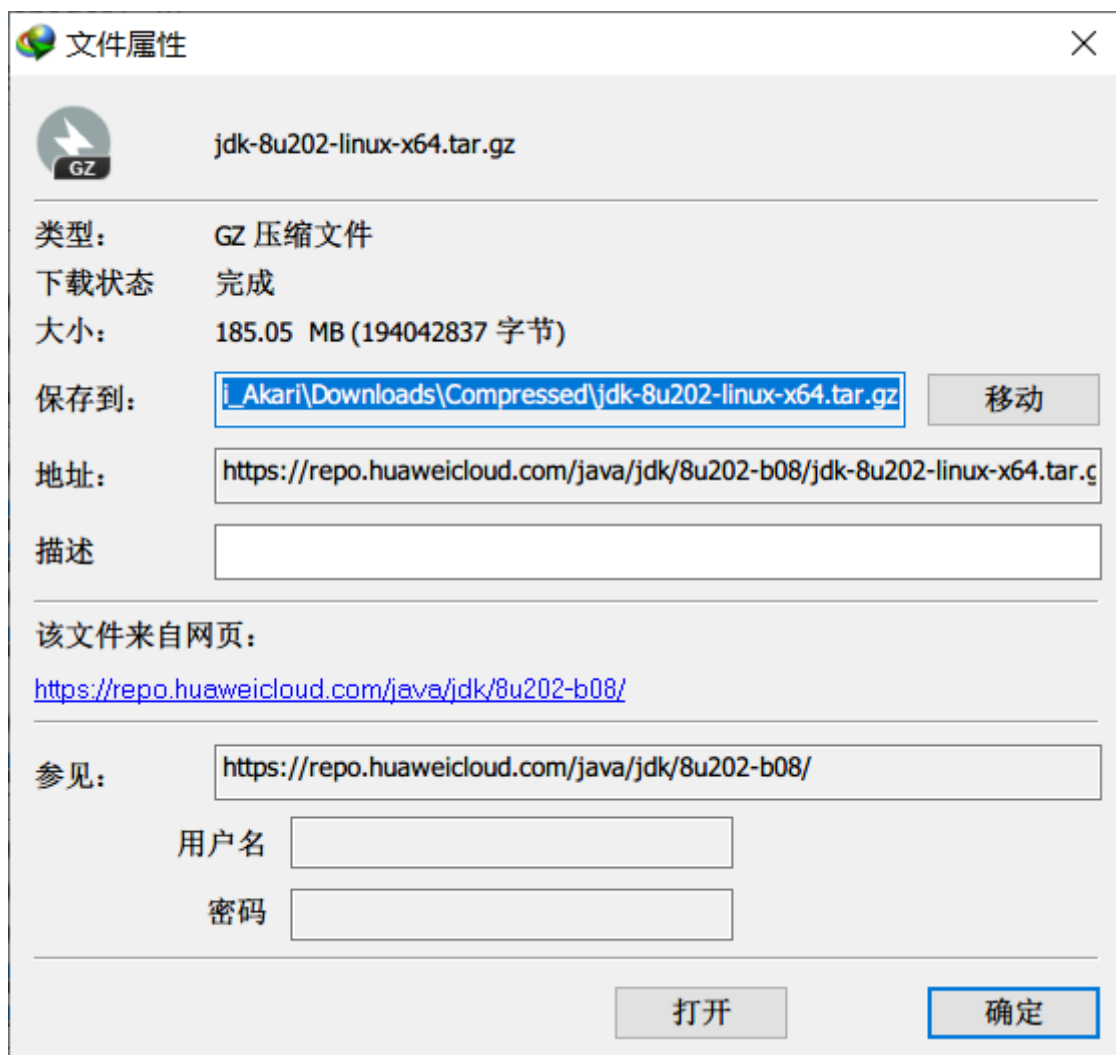
由于VanetMobiSim由Java编写，故我们需要配置jre环境，根据官方要求JDK必须是Sun/Oracle版本，不能是OpenJDK。由于Oracle官网需要登录下载，我们选择从华为云镜像站下载。

Index of java-local/jdk

Name	Last modified	Size
../		
10.0.1+10/	22-Aug-2021 00:21	-
10.0.2+13/	22-Aug-2021 00:22	-
11+28/	22-Aug-2021 00:22	-
11.0.1+13/	22-Aug-2021 00:23	-
11.0.2+7/	22-Aug-2021 00:23	-
11.0.2+9/	22-Aug-2021 00:24	-
12+33/	22-Aug-2021 00:24	-
12.0.1+12/	22-Aug-2021 00:24	-
12.0.2+10/	22-Aug-2021 00:25	-
13+33/	22-Aug-2021 00:25	-
6u45-b06/	22-Aug-2021 00:26	-
7u80-b15/	22-Aug-2021 00:26	-
8u151-b12/	22-Aug-2021 00:26	-
8u151-b12-demos/	22-Aug-2021 00:27	-
8u152-b16/	22-Aug-2021 00:27	-
8u152-b16-demos/	22-Aug-2021 00:27	-
8u171-b11/	22-Aug-2021 00:27	-
8u171-b11-demos/	22-Aug-2021 00:28	-
8u172-b11/	22-Aug-2021 00:28	-
8u172-b11-demos/	22-Aug-2021 00:29	-
8u181-b13/	22-Aug-2021 00:29	-
8u181-b13-demos/	22-Aug-2021 00:30	-
8u191-b12/	22-Aug-2021 00:30	-
8u191-b12-demos/	22-Aug-2021 00:30	-
8u192-b12/	22-Aug-2021 00:31	-
8u192-b12-demos/	22-Aug-2021 00:31	-
8u201-b09/	22-Aug-2021 00:31	-
8u201-b09-demos/	22-Aug-2021 00:32	-
8u202-b08/	22-Aug-2021 00:32	-
8u202-b08-demos/	22-Aug-2021 00:33	-
9.0.1+11/	22-Aug-2021 00:33	-

ArtifactRepo/ Server at repo.huaweicloud.com Port 443

下载后如下，我们将其复制入虚拟机并解压。



解压jdk安装包后，我们在环境变量中设置相关路径，并重启终端，这样Java安装完成。

同样我们将VanetMobiSim的jar文件也复制进虚拟机，至此软件安装完成。

1. 编写VanetMobiSim配置文件

整个XML配置文件的根元素必须为universe,几个重要的子元素如下：

```
[<dimx>1000</dimx>]
[<dimy>1000</dimy>]
[<extension>extension_parameters</extension>]
[<node>node_parameter</node>]
[<nodegroup>nodegroup_parameters</nodegroup>]
```

扩展SpatialModel：指定空间环境模型。

```
<extension name="SpatialModel"
class="de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialModel"
traffic_light="UserTrafficLight"
min_x="0" max_x="1000" min_y="0" max_y="1000"><number_lane full="false"
max="4">2</number_lane>
<reflect_directions>true</reflect_directions>
<max_traffic_lights>0</max_traffic_lights>
```

子元素number_lane配置车道，属性full控制是否所有车道都是多车道，属性max表示多车道道路的数量。

子元素reflect_directions表示是否在物理上区分两个交通流。

子元素max_traffic_lights表示交通灯管理的交叉口数量。

```
<extension name="UserTrafficLight"
class="eurecom.spatialmodel.extensions.TrafficLight"
step="10000"
spatial_model="SpatialModel"/>
```

name表示扩展的名字，用户可以自定义。class表示该扩展使用到的类的名称，不要更改。

step表示交通灯多长时间改变一次（单位：毫秒）。

spatial_model表示此交通灯策略应用到的空间环境模型。

扩展TIGERReader：读取地图数据文件（GDF或TIGER格式）。

```
<extension class="eurecom.tigerreader.TIGERReader"
source="file:///home/akari/TGR11001/TGR11001"
shapeCoord="true"
center_lat="+38905050" center_long="-77016160"
size_x="1000" size_y="1000">
/home/akari/TGR11001/speedLimits.txt
```

source为读取的地图数据文件的路径（不要加文件的后缀名）。

这里地图文件在美国有关部门官网下载，并从官网提供的源码压缩包内找到speed_list.txt。

shapeCoord表示是否加载路段中间点，路段中间点用于表示弯曲路段。

center_lat/center_long表示地图中心点的纬/经度。

size_x/size_y表示截取的地图尺寸（单位：米）。

speed文件指定道路类型与其对应的速度限制。

完整xml代码如下。

```

<universe>
  [<dimx>1000</dimx>]
  [<dimy>1000</dimy>]
  <extension name="SpatialModel"

  class="de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialModel"
    traffic_light="UserTrafficLight"
    min_x="0" max_x="1000" min_y="0" max_y="1000">
    <number_lane full="false" max="4">2</number_lane>
    <reflect_directions>true</reflect_directions>
    <max_traffic_lights>0</max_traffic_lights>
  </extension>

  <extension name="UserTrafficLight"
    class="eurecom.spatialmodel.extensions.TrafficLight"
    step="10000"
    spatial_model="SpatialModel"/>

  <extension
    class="de.uni_stuttgart.informatik.canu.mobisim.extensions.NSOutput"
    output="ns3_trace.txt"/>

  <extension

  class="de.uni_stuttgart.informatik.canu.mobisim.simulations.TimeSimulation"
    param="60.0"/>

  <extension name="PosGen"

  class="de.uni_stuttgart.informatik.canu.tripmodel.generators.RandomInitialP
ositionGenerator"/>

  <extension name="TripGen"

  class="de.uni_stuttgart.informatik.canu.tripmodel.generators.RandomTripGene
rator">
    <reflect_directions>true</reflect_directions>
    <minstay>50.0</minstay>
    <maxstay>100.0</maxstay>
  </extension>

  <nodegroup n="300">
    <extension class="polito.uomm.IDM_IM"
      spatial_model="SpatialModel"
      initposgenerator="PosGen"
      tripgenerator="TripGen">
      <minspeed>8.3</minspeed>
      <maxspeed>22.2</maxspeed>
      <b>0.9</b>
      <step>1.0</step>
    </extension>
  </nodegroup>

  <extension class="eurecom.tigerreader.TIGERReader"
    source="file:///home/akari/TGR11001/TGR11001"
    shapeCoord="true"
    center_lat="+38905050" center_long="-77016160"
    size_x="1000" size_y="1000">

```

```
<speed>/home/akari/TGR11001/speedLimits.txt</speed>
</extension>

<extension
  class="de.uni_stuttgart.informatik.canu.mobisimadd.extensions.GUI">
  <width>640</width>
  <height>480</height>
  <step>1</step>
</extension>
</universe>
```

2. 使用VanetMobiSim模拟车辆轨迹

编写完成xml文件后，我们便以此为参数，开始模拟车辆运行，由于模型较大，运行时间可能较长，等待一段时间后，我们得到了输出的trace.txt。

```
akari@ubuntu:~/Desktop/tarballs/ns-allinone-3.27/ns-3.27$ sudo ./waf --run LTE-V
Waf: Entering directory `/home/akari/Desktop/tarballs/ns-allinone-3.27/ns-3.27/build'
Waf: Leaving directory `/home/akari/Desktop/tarballs/ns-allinone-3.27/ns-3.27/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.449s)
```

```
ns3_trace1.txt
#
# nodes: 300, pause: 3.4028235E38, max speed: 3.4028235E38 max x = 12052.658, max y: 13143.453
#
$node_(0) set X_ 11826.830131303333
$node_(0) set Y_ 12850.920575277893
$node_(0) set Z_ 0.0
$node_(1) set X_ 11993.872608803333
$node_(1) set Y_ 12772.967261834034
$node_(1) set Z_ 0.0
$node_(2) set X_ 11225.477212303334
$node_(2) set Y_ 13140.461470684522
$node_(2) set Z_ 0.0
$node_(3) set X_ 11706.981414639595
$node_(3) set Y_ 12143.453126
$node_(3) set Z_ 0.0
$node_(4) set X_ 11125.251725803333
$node_(4) set Y_ 13029.09958466364
$node_(4) set Z_ 0.0
$node_(5) set X_ 11136.38789097
$node_(5) set Y_ 12850.920575277893
$node_(5) set Z_ 0.0
$node_(6) set X_ 11648.651488636668
$node_(6) set Y_ 12349.792164703127
$node_(6) set Z_ 0.0
$node_(7) set X_ 11225.477212303334
$node_(7) set Y_ 13143.453126
$node_(7) set Z_ 0.0
$node_(8) set X_ 11826.830131303333
$node_(8) set Y_ 12594.788266823285
$node_(8) set Z_ 0.0
$node_(9) set X_ 11715.468479636667
$node_(9) set Y_ 12338.65597867936
$node_(9) set Z_ 0.0
$node_(10) set X_ 11648.651488636668
$node_(10) set Y_ 12349.792164703127
$node_(10) set Z_ 0.0
$node_(11) set X_ 11392.519689803334
$node_(11) set Y_ 12282.975049121516
$node_(11) set Z_ 0.0
$node_(12) set X_ 11548.426002136666
$node_(12) set Y_ 13107.052904458773
$node_(12) set Z_ 0.0
$node_(13) set X_ 11225.477212303334
$node_(13) set Y_ 12695.013950298338
$node_(13) set Z_ 0.0
$node_(14) set X_ 11548.426002136666
$node_(14) set Y_ 12772.967261834034
$node_(14) set Z_ 0.0
$node_(15) set X_ 11826.888240993187
$node_(15) set Y_ 12143.453126
$node_(15) set Z_ 0.0
$node_(16) set X_ 11760.013140303334
$node_(16) set Y_ 12873.192950900284
$node_(16) set Z_ 0.0
$node_(17) set X_ 11595.643342443334
$node_(17) set Y_ 12642.339785149092
$node_(17) set Z_ 0.0
$node_(18) set X_ 11826.830131303333
$node_(18) set Y_ 12338.65597867936
$node_(18) set Z_ 0.0
$node_(19) set X_ 11537.289836970001
$node_(19) set Y_ 12216.15793488748
$node_(19) set Z_ 0.0
$node_(20) set X_ 11371.312702338288
```

3. 在NS-3中使用VanetMobiSim生成的轨迹文件

```

//使用LTE-V简单模拟车联网
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/wifi-module.h"
#include "ns3/mobility-module.h"
#include "ns3/applications-module.h"
#include "ns3/internet-module.h"
#include "ns3/wave-module.h"
#include "ns3/netanim-module.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("LTE-V");

int main (int argc, char *argv[])
{
    NodeContainer wifiNodes;
    wifiNodes.Create(300);

    YanswifiChannelHelper channel =YanswifiChannelHelper::Default();
    channel.AddPropagationLoss("ns3::RangePropagationLossModel",
        "MaxRange", DoubleValue(300));//创建300个结点，结点数量必须与VanetMobiSim轨迹
文件的结点数
量一致。
    YanswifiPhyHelper phy = YanswifiPhyHelper::Default();
    phy.SetChannel(channel.Create());
    wifi80211pHelper wifi;
    NqosWaveMacHelper mac;
    Ssid ssid = Ssid("ns-3-wifi");
    mac.SetType("ns3::OcbWifiMac",
        "Ssid", ssidValue(ssid),
        "QosSupported", BooleanValue(false));
    NetDeviceContainer wifiDevices;
    wifiDevices = wifi.Install(phy, mac, wifiNodes);
    //导入VanetMobiSim轨迹文件
    Ns2MobilityHelper mobility =
    Ns2MobilityHelper("/home/akari/Desktop/ns3_trace.txt");
    mobility.Install();

    //配置TCP/IP协议族（网络层/传输层），分配IP地址
    InternetStackHelper stack;
    stack.Install(wifiNodes);
    Ipv4AddressHelper address;
    address.SetBase("10.1.0.0", "255.255.0.0");
    Ipv4InterfaceContainer wifiInterfaces;
    wifiInterfaces = address.Assign(wifiDevices);
    //配置应用层
    UdpEchoServerHelper echoServer(9);
    ApplicationContainer serverApps = echoServer.Install(wifiNodes);
    serverApps.Start(Seconds(0));
    serverApps.Stop(Seconds(60.0));
    UdpEchoClientHelper echoClient(Ipv4Address("255.255.255.255"), 9);
    echoClient.SetAttribute("MaxPackets", UintegerValue(600));
    echoClient.SetAttribute("Interval", TimeValue(Seconds(0.1)));
    echoClient.SetAttribute("PacketSize", UintegerValue(150));
    ApplicationContainer clientApps = echoClient.Install(wifiNodes);
    clientApps.Start(Seconds(0));
    clientApps.Stop(Seconds(60.0));
}

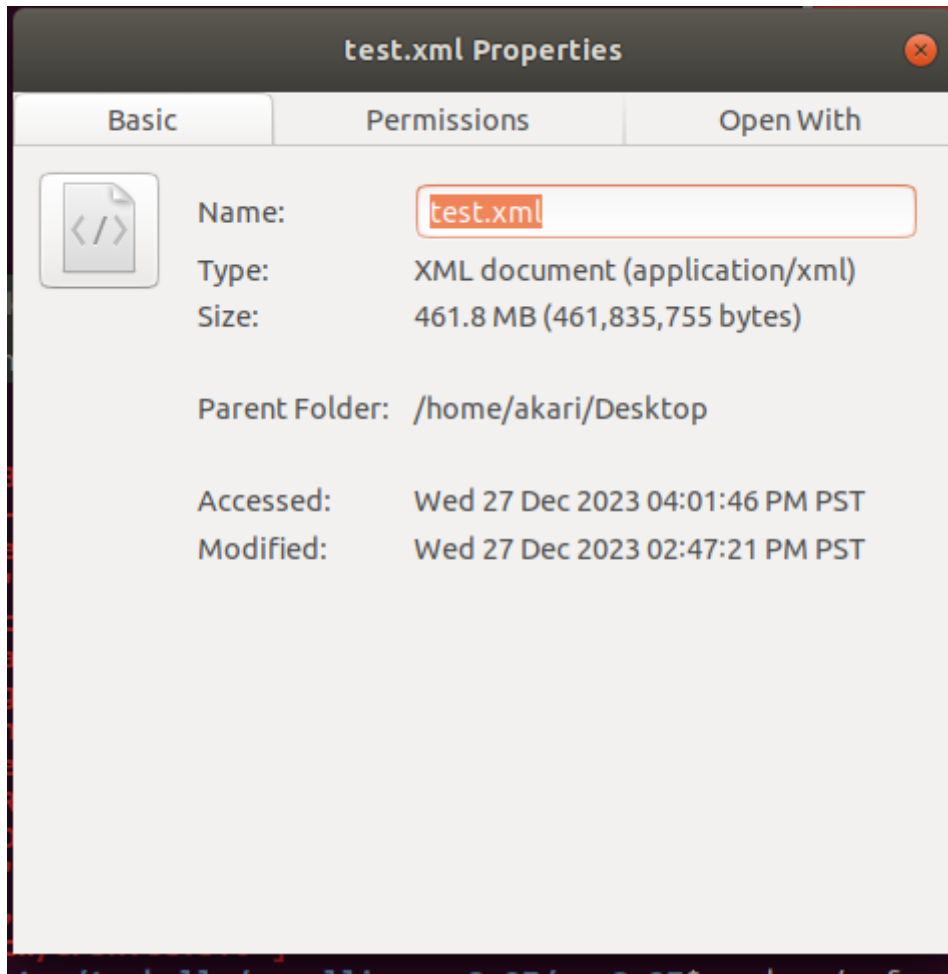
```

```
//生成轨迹文件，用于NetAnim动画分析
AnimationInterface anim("test.xml");
anim.SetMaxPktsPerTraceFile(10000000);

Simulator::Stop(Seconds(60.0));
Simulator::Run ();
Simulator::Destroy ();
}
```

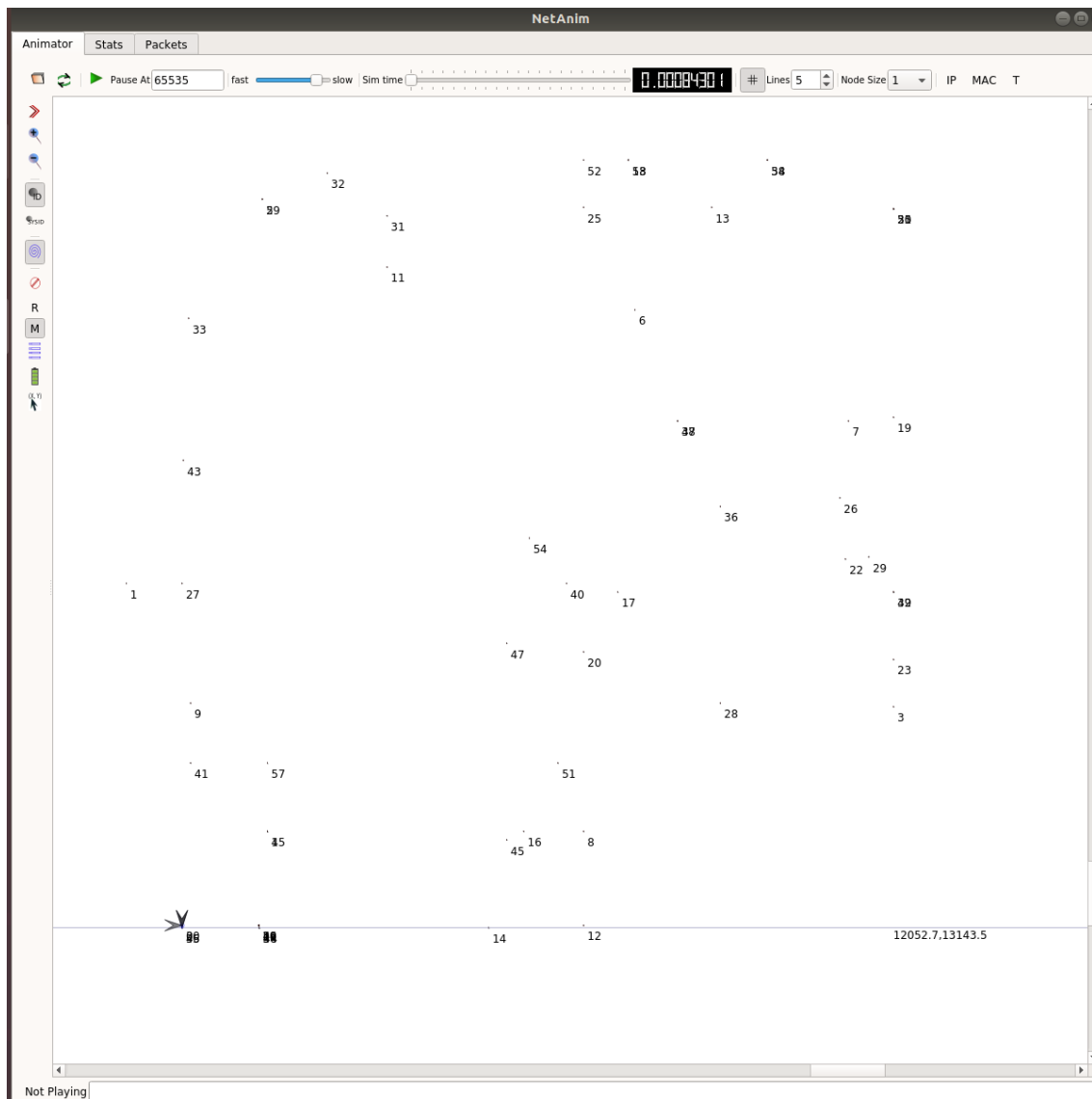
至此，NS-3代码编写结束。需要注意的是，由于模型较大，因此我们添加了SetMaxPktsPerTraceFile参数，确保数据包发送了不会超出最大值。

运行完成后，我们会在ns-3目录下发现test.xml输出结果，该文件体积十分庞大。



4. NetAnim动画分析

我们将上一步得到的test.xml在NetAnim中打开，由于上一步实际得到的xml十分庞大，我们选用缩小参数后的轨迹文件进行演示。可以看到画面中车辆聚集在边长1km的矩形中，数据包在车辆之间跳跃。



实验结果总结&心得体会

- 成功安装配置VanetMobiSim，并生成车辆轨迹文件。
- NS-3代码成功导入轨迹文件，模拟LTE-V车联网环境。
- NetAnim动画展示了车辆在1km范围内的聚集和数据包的传输情况。

结论

通过该实验，成功搭建了车联网仿真环境，使用NS-3和VanetMobiSim工具实现了车辆之间的通信模拟。实验结果表明所设计的系统在模拟环境中能够正常运行，为进一步研究车联网通信提供了基础。

附录(可选)