**NS3实验报告**

------161520322仲奕杰

1. 环境安装

首先在官网下载ns-allinone-3.29.tar.bz2，解压后使用自带的build.py脚本编译生成完整版ns3.随后在example/下或scratch下编写代码后用waf脚本编译生成可执行文件并运行。由于scratch下文件目录结构简单，并且存在的wscript编译脚本能够遍历目录下所有.cc和.py文件，并且编译模块动态添加，所以我们将源代码放在scratch目录下。

同时我们还需要netanim来将仿真可视化，官网bz2包中已经带有未编译的netanim，我们进入netanim-3.108目录下，首先make clean清理环境，随后使用qt编译文件qmake NetAnim.pro，最后使用Makefile进行编译，执行指令make生成可执行文件NetAnim。

同时之后还需要安装gawk脚本处理数据，gnuplot绘制图像。

1. 星型拓扑网络
2. 代码部分(详情见代买及注释)

通过nWifi设置无线站点个数，创建节点

NodeContainer wifiStaNodes;

wifiStaNodes.Create (nWifi); //创建无线站点STA的节点

建立AP节点

NodeContainer wifiApNode;

wifiApNode.Create(1); //创建一个AP节点

使用默认通信信道，PHY模型，安装默认无线设备，设置STA模型可以移动，在xy轴正负50范围内随机移动

安装协议栈

InternetStackHelper stack;

stack.Install (wifiApNode);

stack.Install (wifiStaNodes); //安装协议栈

设置服务器客户端起止时间，网络最多发包数，发包间隔，包大小等

ApplicationContainer serverApps = echoServer.Install (wifiStaNodes.Get (0));

serverApps.Start (Seconds (1.0));

serverApps.Stop (Seconds (15.0)); //服务器起止时间

UdpEchoClientHelper echoClient (wifiInterfaces.GetAddress (0), 9);

echoClient.SetAttribute ("MaxPackets", UintegerValue (5000));

echoClient.SetAttribute ("Interval", TimeValue (Seconds (0.1))); //发包间隔

echoClient.SetAttribute ("PacketSize", UintegerValue (1024)); //包大小

ApplicationContainer clientApps =

echoClient.Install (wifiStaNodes.Get (nWifi - 1));

clientApps.Start (Seconds (1.0));

clientApps.Stop (Seconds (15.0)); //客户端起止时间

Simulator::Stop (Seconds (15.0)); //仿真结束时间、

* 代码中导出数据部分

1.对数据包进行追踪得到.tr文件用于后续数据处理

AsciiTraceHelper ascii;

phy.EnableAsciiAll (ascii.CreateFileStream ("star.tr"));

2.生成pcap用于查看具体数据包内容

phy.EnablePcap ("starwifi", apDevices.Get (0));

3.生成用于netanim的xml文件

AnimationInterface anim("anim3.xml");

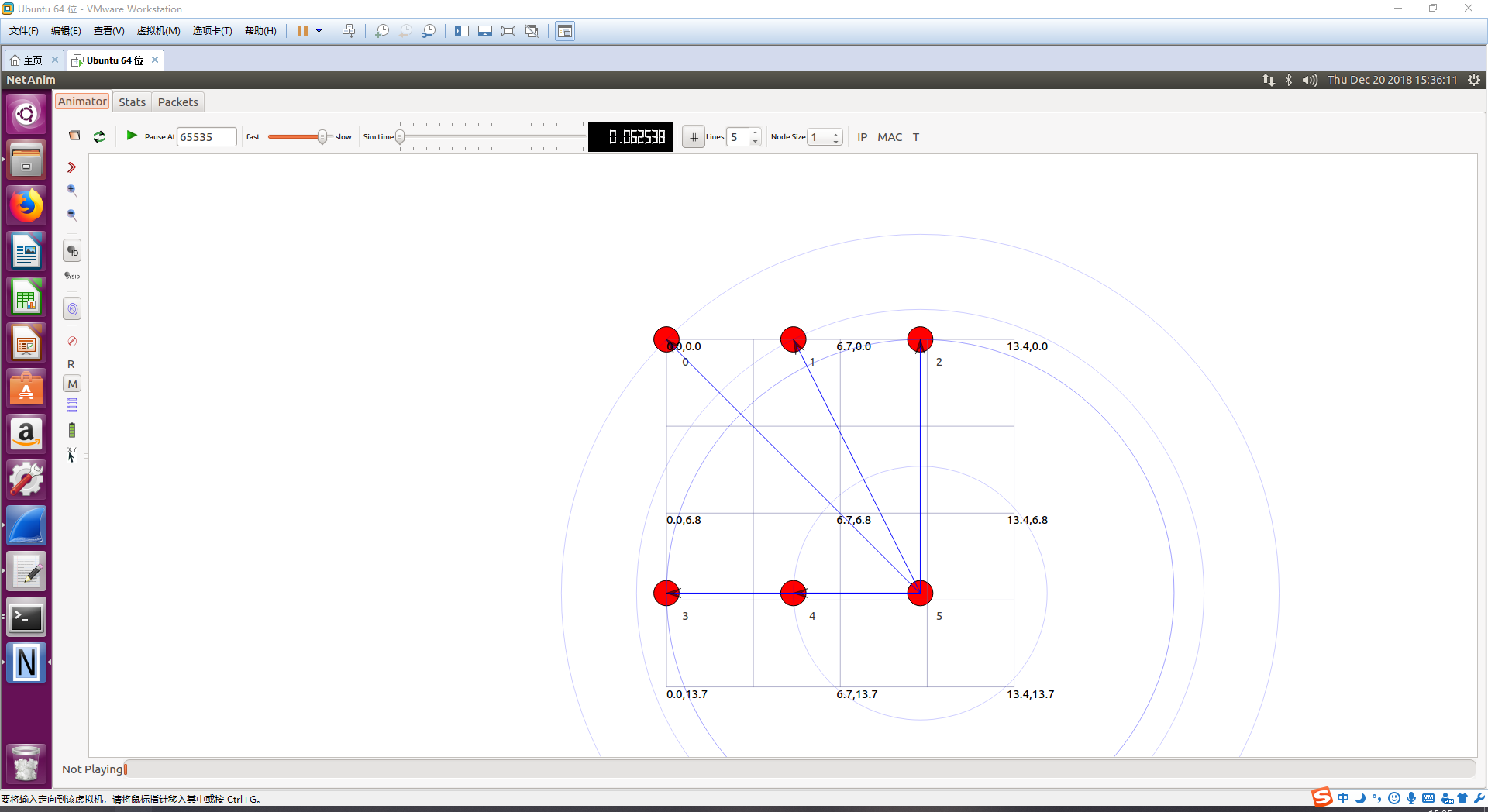
* Awk脚本代码

myScrIP = "10.1.3.5";#This is the link that we pay attention to

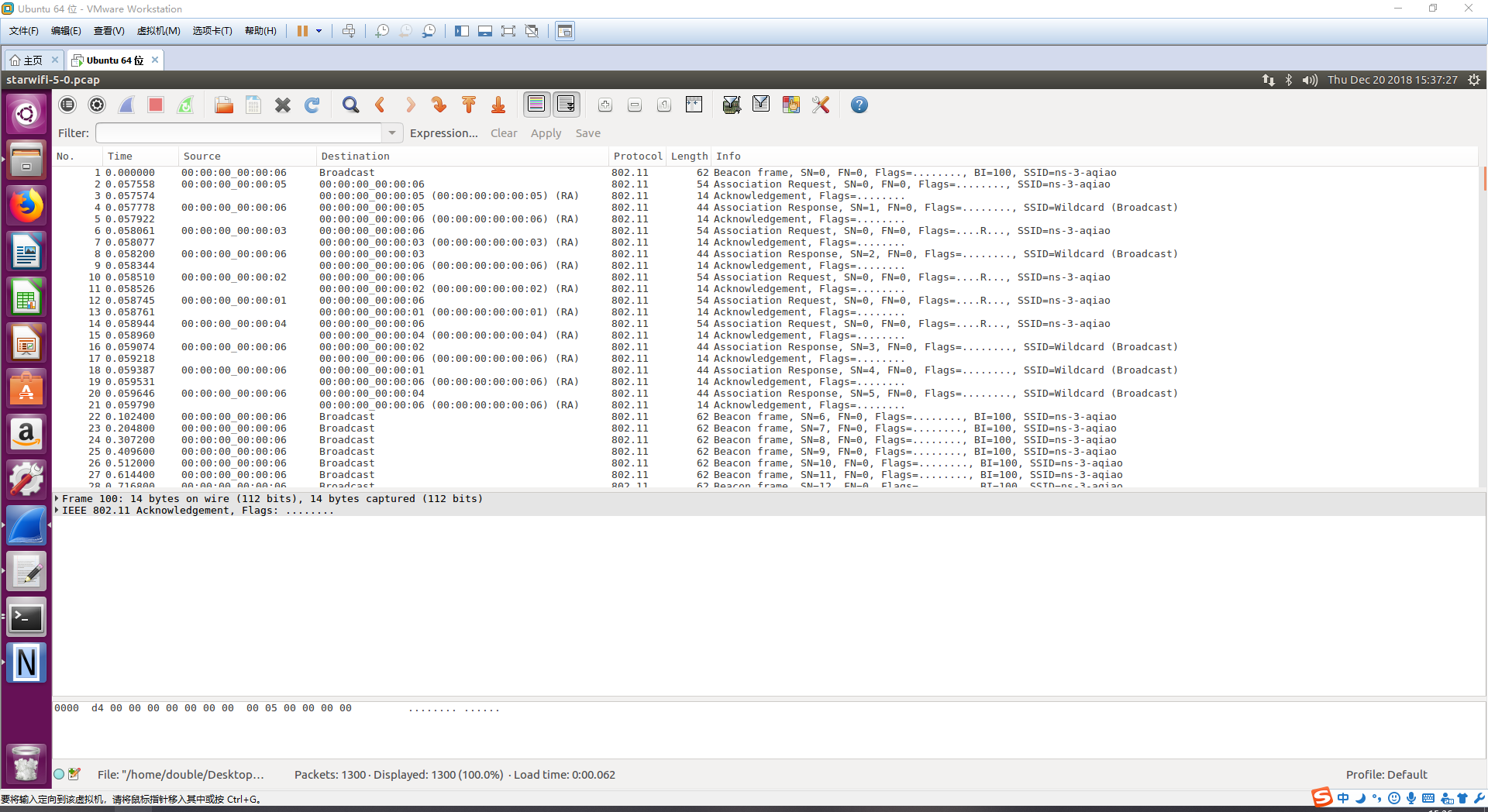
myDstIP = "10.1.3.1";

对于星型无线网络需要设置STA设备的首个和最后一个ip，对于多跳无线网络可以设置为首个和其余任意一个ip

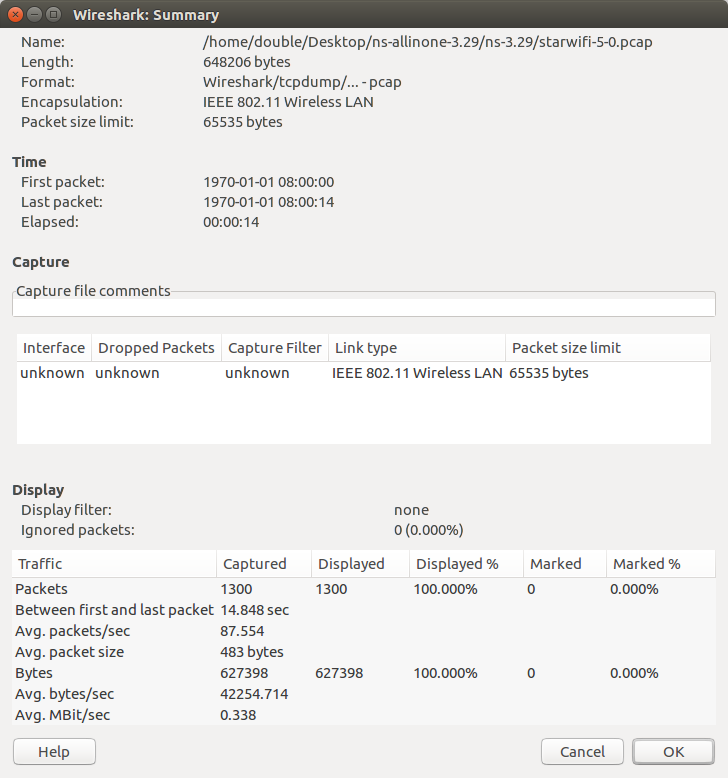
1. 实现结果
2. 在./netanim-3.108目录下执行./NetAnim打开所生成的anim3.xml查看仿真图像



1. 使用wireshark打开生成的starwifi-5-0.pcap,可以查看所有数据包



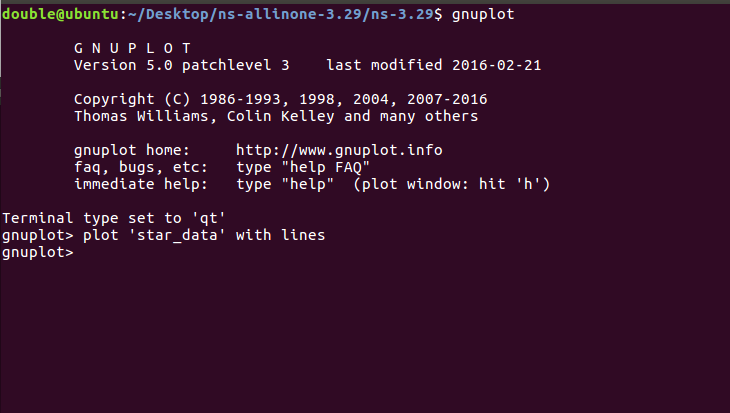
使用统计功能可以查看到总体数据流情况

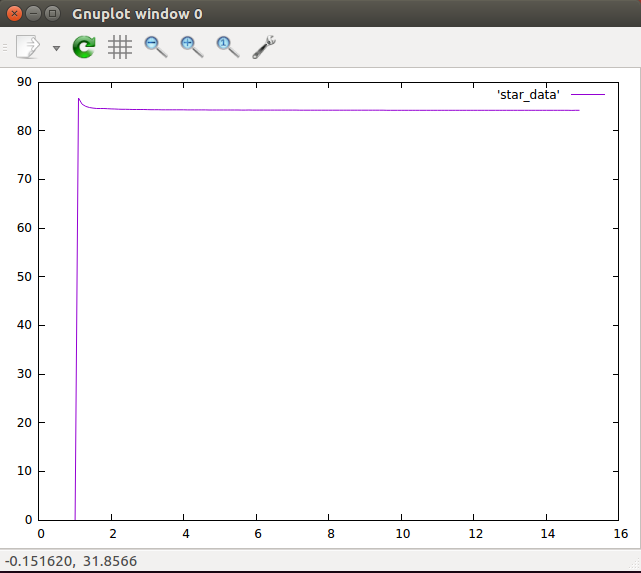


1. 在. /ns-3.29目录下执行

gawk –f ttl.awk star.tr > star\_data

得到star\_data数据组，即各个时刻的数据包数量，用于后续统计吞吐量，随后进入gnuplot，执行plot ‘star\_data’ with lines得到图像





我们可以发现最终趋向的值和wireshark中的每秒平均包数量87个是很接近的。这个值受到代码中具体数值的设定的影响。

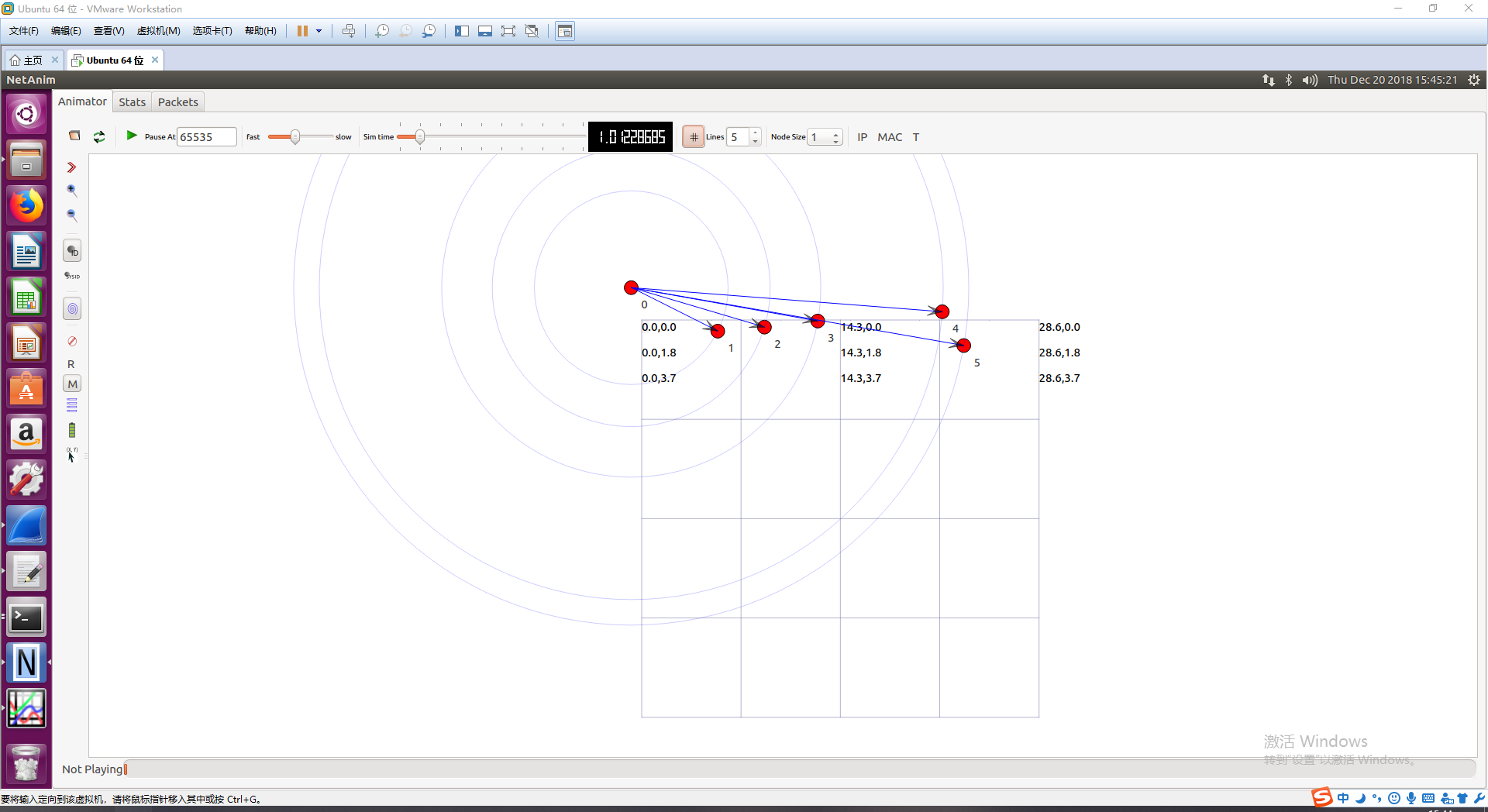
1. 多跳无线网络

多跳无线网络没有一个固定的ap，也没有固定的sta，每一个节点都可以成为ap和sta，所以需要建立一个hocwifi网络。

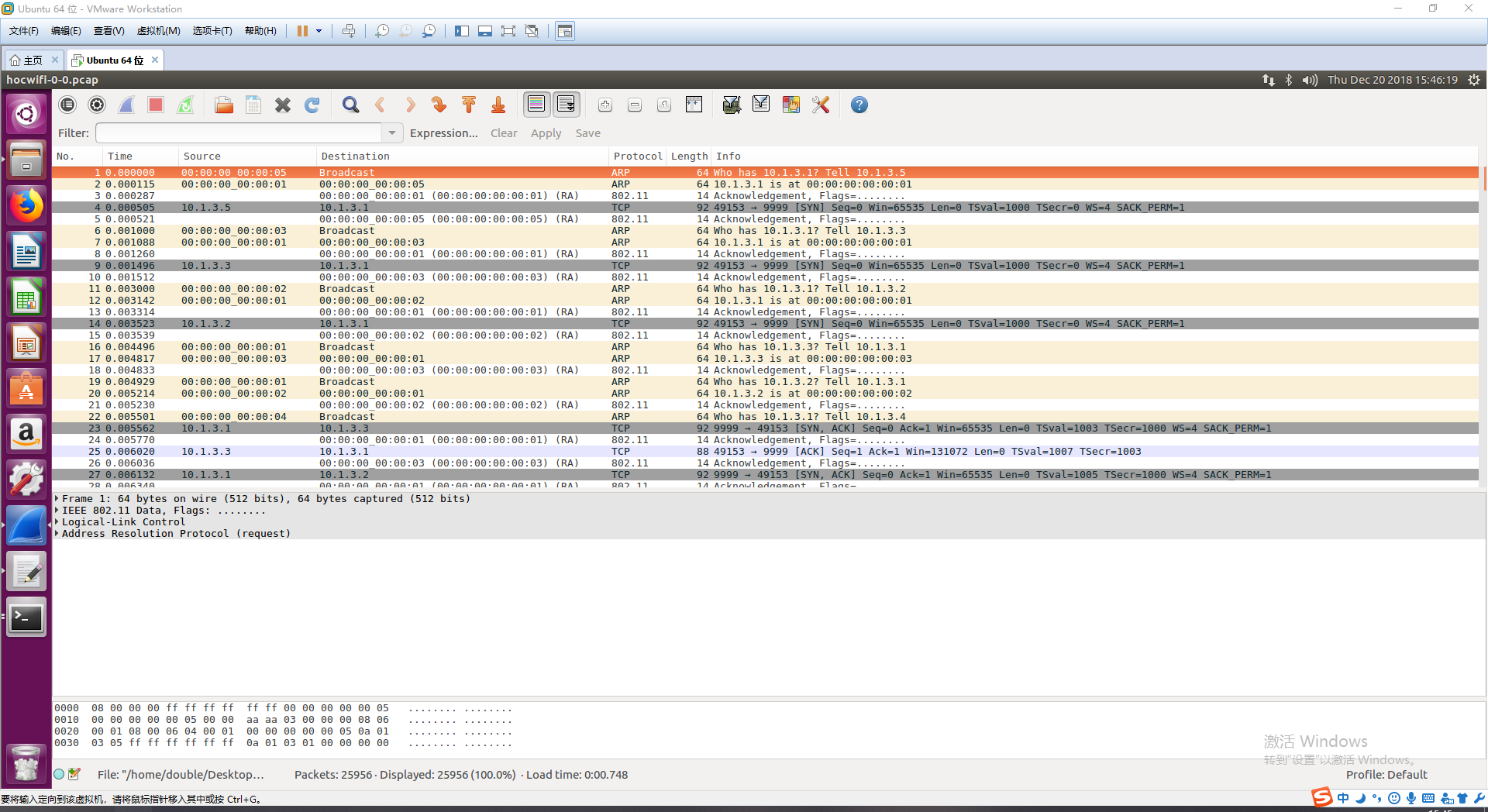
1. 代码部分

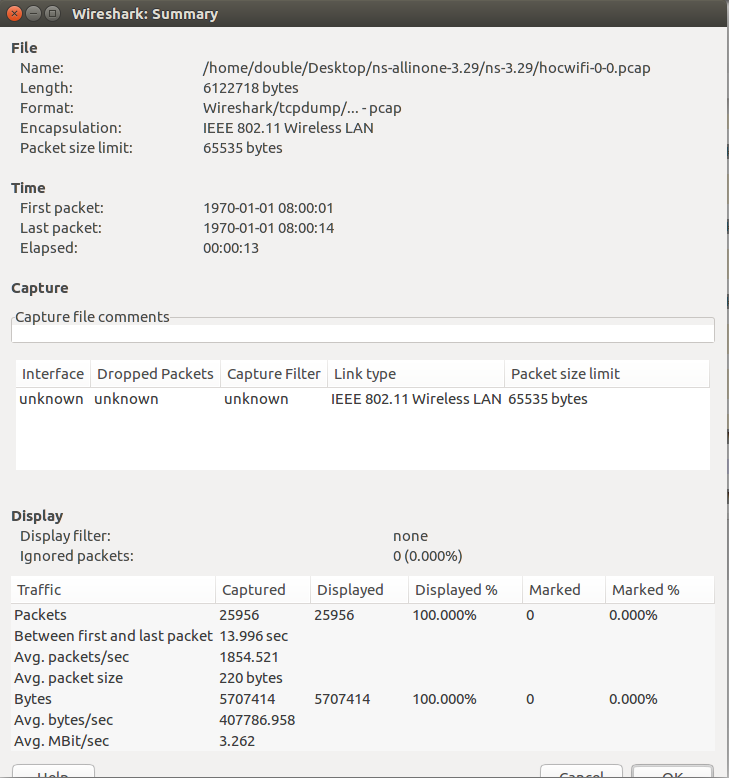
具体详情见代码，一些数值设定和数据生成代码与上述星型网络相似。

1. 实现结果
2. 同样使用netanim打开仿真模拟



1. 使用wireshark打开数据包

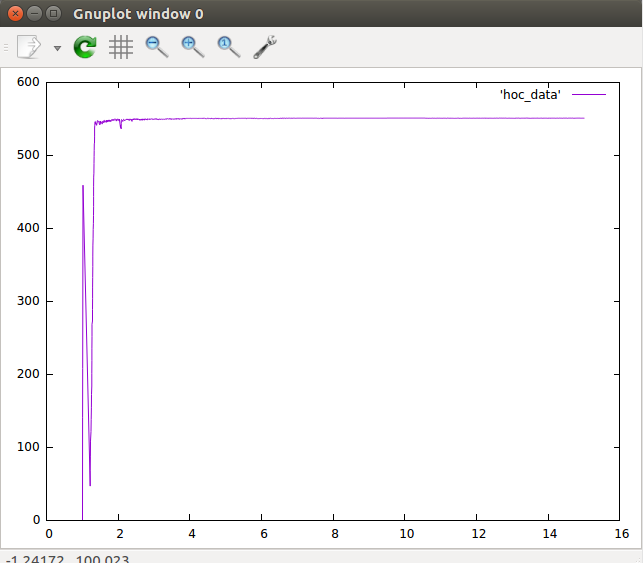




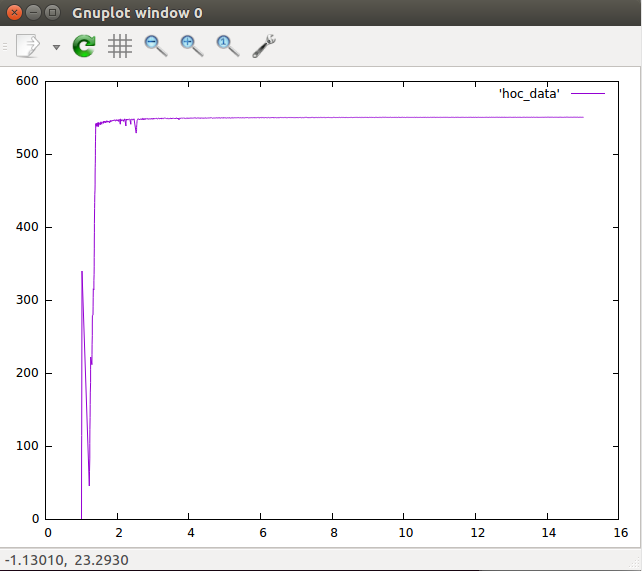
1. 使用awk脚本运算数据

这里我们可以分别计算

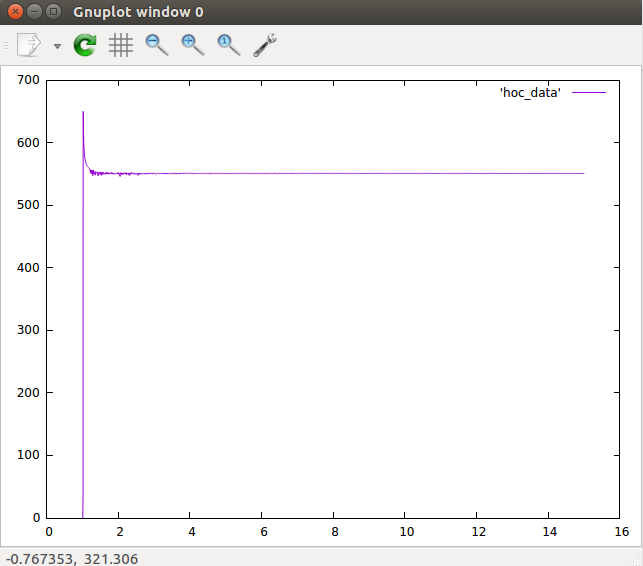
10.1.3.1 -> 10.1.3.2



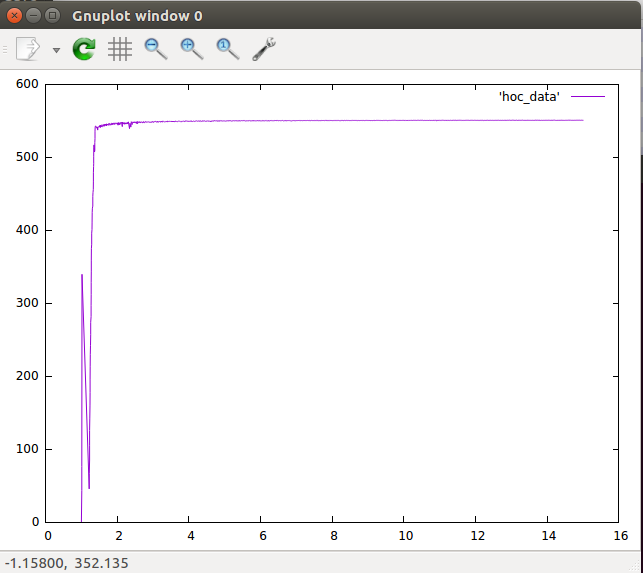
10.1.3.1 -> 10.1.3.3



10.1.3.1 -> 10.1.3.4



10.1.3.1 -> 10.1.3.5



可以发现最后趋向的数值是相同的，但是相比于wireshark做出的统计就少了很多，也是因为它统计的是整个网络的总量，而这只是统计一个节点的量。