数据通信 NS3 作业-4

姓名：费扬 学号：519021910917 日期：2022.05.16

一、 实验名称及内容

Lab4：

Background：

The ns-3 tracing system is built on the concepts of independent **tracing sources** and **tracing sinks**, along with a uniform mechanism for connecting sources to sinks.

Trace sources are entities that can signal events that happen in a simulation and provide access to interesting underlying data. Trace sources are not useful by themselves; they must be connected to other pieces of code that actually do something useful with the information provided by the source. The entities that consume trace information are called trace sinks.

Demo 1:

Modify myfourth.cc and Use AddTraceSource:

- 第一个参数是此跟踪源的名称，使其在 Config 系统中可见。
- 第二个参数是帮助字符串。
- 第三个参数，重点是第三个参数的参数：&MyObject::m_myInt。 这是被添加到类中的 TracedValue； 它始终是类数据成员。
- 最后一个参数是 TracedValue 类型的 typedef 的名称，作为字符串。

$./waf --run scratch/myfourth

```
fy@ubuntu:~/tarballs/ns-3-allinone/ns-3.30$ ./waf --run scratch/myfourth
Waf: Entering directory `/home/fy/tarballs/ns-3-allinone/ns-3.30/build'
Waf: Leaving directory `/home/fy/tarballs/ns-3-allinone/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.975s)
Traced 0 to 1
Traced 1 to 4
Traced 4 to 9
Traced 9 to 16
Traced 16 to 25
Traced 25 to 36
Traced 36 to 49
Traced 49 to 64
Traced 64 to 81
```

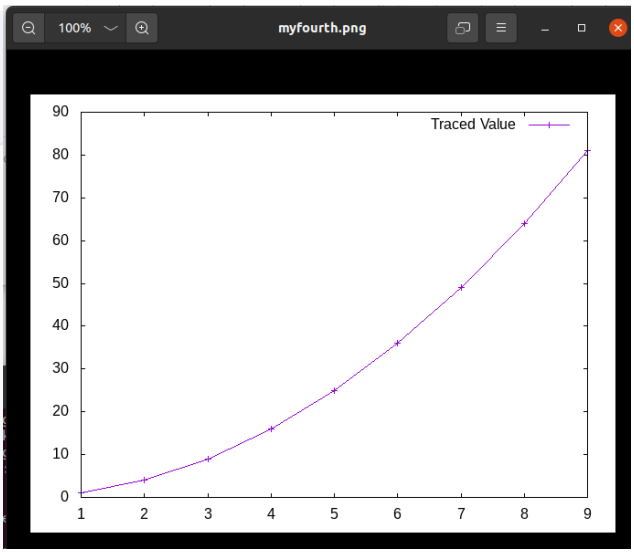
$ls

Check the .dat file, and use gnuplot to draw a figure.

myfourth.dat - 记事本

文件 编辑 查看

1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81



Demo 2:

$./waf --run scratch/myfirst_4

```
fy@ubuntu:~/tarballs/ns-3-allinone/ns-3.30$ ./waf --run scratch/myfirst_4
Waf: Entering directory `/home/fy/tarballs/ns-3-allinone/ns-3.30/build'
Waf: Leaving directory `/home/fy/tarballs/ns-3-allinone/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.947s)
Welcome! Ns3 Simulator!
My traced packet size:1054

/NodeList/1/DeviceList/0/$ns3::PointToPointNetDevice/MacTxMy traced packet size:
 1054
```

Exercise:

在示例代码的基础上使用 Config 函数将 Station 节点的位置写入记录文件并绘图。

- Trace the mobility of wifi station nodes in examples/tutorial/third.cc,

  name your source code lab4.cc.

- Print out the position (x,y) of wifi station nodes as they move

- Use gnuplot to draw the traces of wifi station nodes

Hints:

1) Use trace source of ns3::MobilityModel/CourseChange
   First find **ns3::MobilityModel** on
   https://www.nsnam.org/docs/release/3.28/doxygen/_trace_source_list.html
2) Find example codes:
3) For more information on gnuplot: http://people.duke.edu/~hpgavin/gnuplot.html

二、 实验过程和结果

程序见压缩包内 trace exercise 目录下。本次 ns3 的版本为 3.30。

```
#include "ns3/core-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/mobility-module.h"
#include "ns3/csma-module.h"
#include "ns3/internet-module.h"
#include "ns3/yans-wifi-helper.h"
#include "ns3/ssid.h"
#include "ns3/vector.h"
```

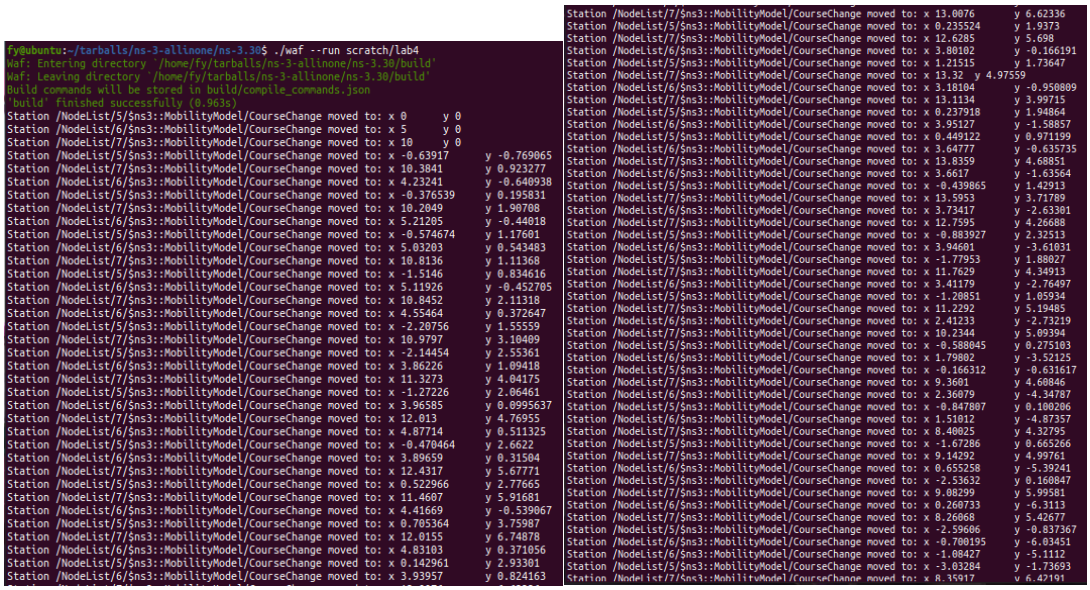
头文件包含内容如上。

本次实验的主要内容是在示例代码 fourth.cc 的基础上添加将记录信息写入文件并绘图的功能，在示例代码的基础上使用 Config 函数将 Station 节点的位置写入记录文件并绘图。

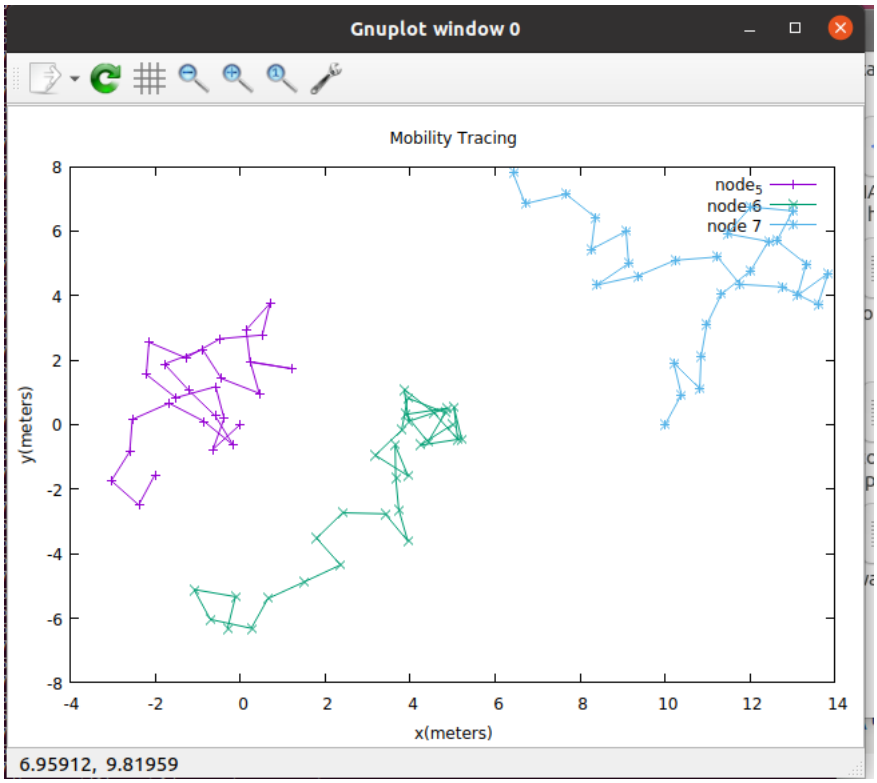
- 首先建立两个 p2p 模型和三个节点 n0，n1，n2，设置 p2p 的参数；
- 然后配置 wifi 网络，配置物理层，MAC 层等；
- 安装 mobility 模型，配置协议，设置 IP 地址；
- 建立 Server 和 Client，传输 UDP 数据包；
- 构建路由，Trace with context and file；
- 使用 Config 函数记录到文件内。

Simulation：

1) *./waf –run scratch/lab4：*



2) *Use gnuplot to draw the traces of wifi station nodes*



三、 实验思考:

- **分别对节点产生文件并合并作图：**

  Gnuplot 支持将多个.dat 文件合并绘制，在使用 Config 定义文件输出时可以考虑分开存储进不同文件。

- **关于 Connect 和 ConnectWithoutContext 相异的问题：**

  会导致对于标准结果的输出格式不同，在.dat 文件中显示空白，因此需要注意使用 ConnectWithoutContext 保证输出格式正确。