

# 粗粒度组网模拟实验

## 1. 实验目的

使用平台的无人机通信接口实现通信。

## 2. 实验要求

- 软件要求：Windows 10及以上版本；RflySim工具链<sup>[1]</sup>。
- 硬件要求：笔记本/台式电脑1台<sup>[2]</sup>。

## 3. 实验地址

例程目录：

[\[安装目录\]\RflySimAPIs\9.RflySimComm\2.AdvExps\1.CoaGraNetSimExps\1.NetSim4 Demo](#)

- [PyNetView](#)：网络质量可视化相关的程序，运行前需要安装相关的Python依赖。
- [PyNetWork](#)：粗粒度组网模拟程序的Demo文件，通过文件夹中的readme.txt启动即可运行。
- [PyNetView\PyViaGui.py](#)：网络质量可视化界面程序。
- [PyNetView\UdpService.py](#)：UDP服务程序，用于接收和处理网络数据。
- [PyNetWork\NetUavAPI.py](#)：最新开发的无人机通信接口程序。
- [PyNetWork\NetworkSimulation.py](#)：网络仿真器程序，用于模拟无人机之间的网络通信。
- [PyNetWork\UAV1Ctrl.py](#)：第一架无人机的控制程序。
- [PyNetWork\UAV2Ctrl.py](#)：第二架无人机的控制程序。
- [PyNetWork\UAV3Ctrl.py](#)：第三架无人机的控制程序。
- [PyNetWork\UAV4Ctrl.py](#)：第四架无人机的控制程序。

## 4. 实验内容或步骤

本实验包含一个完整的组网模拟实验。

### 4.1 步骤1：启动四架飞机仿真

进入PyNetWork文件夹下。双击运行SITLRun4MavlinkFull.bat来自动创建四架飞机。等待所有CopterSim的左下角打印"PX4: GPS 3D fixed & EKF initialization finished."。



### 4.2 步骤2：启动网络仿真器

在该目录中，双击打开PyNetWork\Python38Run.bat，输入"python NetworkSimulation.py"，来执行模拟器。

```
C:\windows\system32\cmd.exe x + v
Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.
You can use pip or pip3 command to install other libraries
Put Python38Run.bat into your code folder
Use the command: 'python XXX.py' to run the script with Python

E:\summerwork\work1\9.RflySimComm\2.AdvExps\2-NetSim4Demo\PyNetWork>python NetworkSimulation.py
[0.0, 1.0, 1.0, 1.0, 1.0, 0.0, 1.0, 1.0, 1.0, 1.0, 0.0, 1.0, 1.0, 1.0, 1.0, 0.0]
[0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0]
100.0
[0.0, 1.0, 1.0, 1.0, 1.0, 0.0, 1.0, 1.0, 1.0, 1.0, 0.0, 1.0, 1.0, 1.0, 1.0, 0.0]
[0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0]
100.0
|
```

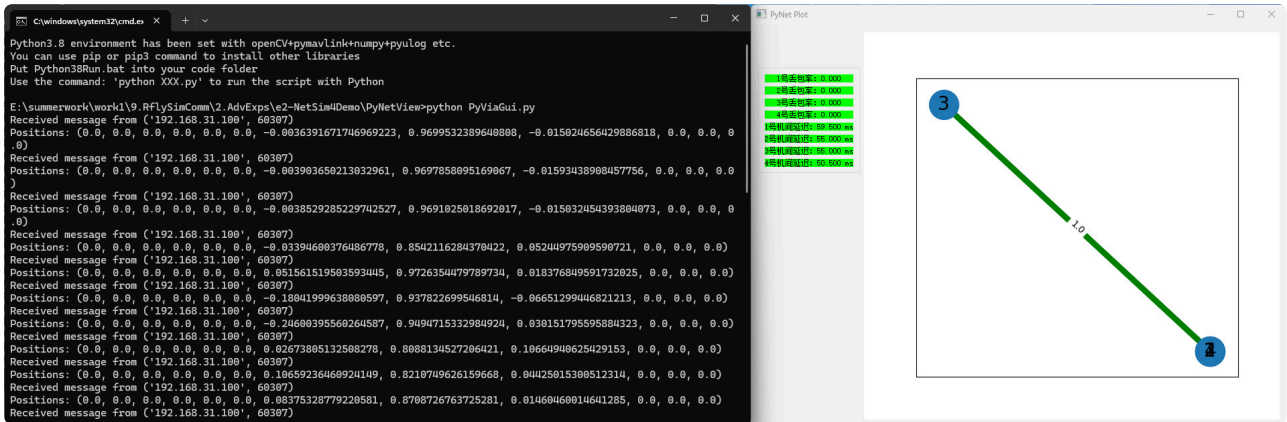
### 4.3 步骤3：启动所有飞机控制程序

在该目录中，双击打开 `PyNetWork\UavPythonRunALL.bat`，来一键启动所有飞机的控制程序。

```
C:\PX4PSP\Python38\python. x + v
Go 4.
Go 28
Go 88
Go 88 5.85353439098]
Go 88 Got msg from Copter #2 Gps pos: (40.1546135, 116.2592382, 99.241) PosE: [1293951.1881304588, 278930.36257353995, 1393
Go 2. Go 25.7136898688]
Go 64 Go 4. Got msg from Copter #1 Gps pos: (40.1526283, 116.2586759, 71.541) PosE: [1293728.5098586145, 278889.44253431435, 1393
5. Go 05.86250750918]
Go 88 Go 28 Got msg from Copter #2 Gps pos: (40.1546135, 116.2592382, 99.243) PosE: [1293951.1885376214, 278930.3626608809, 13932
Go 88 Go 4. Got msg from Copter #1 Gps pos: (40.15263, 116.2586759, 71.541) PosE: [1293728.694601432, 278889.435580649, 139305.90
Go 0. Go 064326173]
Go 65 Go 88 Got msg from Copter #2 Gps pos: (40.1546135, 116.2592382, 99.243) PosE: [1293951.1885376214, 278930.3626608809, 13932
Go 0. Go 5. 5.71173370193]
Go 1. Go 0. Got msg from Copter #1 Gps pos: (40.1526381, 116.2586759, 71.542) PosE: [1293729.5750501538, 278889.40249214077, 1393
5. Go 06.0813708916]
Go 91 Go 65 Got msg from Copter #3 Gps pos: (40.153136, 116.2610954, 97.831) PosE: [1293794.645438383, 279094.32815904805, 139301
Go 2. Go .89713118254]
Go 5. Go 0. Got msg from Copter #1 Gps pos: (40.1526385, 116.2586759, 71.542) PosE: [1293729.6185190503, 278889.40085598396, 1393
Go 91 Go 5. Go 06.09034401714]
Go 61 Go 5. Got msg from Copter #2 Gps pos: (40.1546135, 116.2592382, 99.243) PosE: [1293951.1885376214, 278930.3626608809, 13932
Go 5. Go 5. 5.71173370193]
Go 41 Go 91 Got msg from Copter #3 Gps pos: (40.1531337, 116.2610977, 97.666) PosE: [1293794.3672410478, 279094.52601304057, 1393
Go 5. Go 02.01676933677]
Go 1. Go 5. Got msg from Copter #1 Gps pos: (40.1526393, 116.2586759, 71.542) PosE: [1293729.7054568436, 278889.3975836702, 13930
Go 85 Go 6.10829026805]
Go 1. Go 85 Got msg from Copter #2 Gps pos: (40.1546135, 116.2592381, 99.244) PosE: [1293951.1885093357, 278930.35419824434, 1393
Go 1. Go 5. 25.710327557]
Go 91 Go 2.08957597136]
Go 91 Go 0. Got msg from Copter #3 Gps pos: (40.1531324, 116.2610989, 97.567) PosE: [1293794.2085988428, 279094.6290863069, 13930
Go 25.710327557]
Go 91 Go 0. Got msg from Copter #2 Gps pos: (40.1546135, 116.2592381, 99.244) PosE: [1293951.1885093357, 278930.35419824434, 1393
```

## 4.4 步骤4：启动网络质量可视化界面

回到上级目录，进入PyNetView文件夹下。双击运行PyNetView\Python38Run.bat，在其中输入"python PyViaGui.py"，运行可视化界面。



在本例仿真中，每个飞机都是用组播的方式发给所有其他飞机。因此，每架飞机的Python窗口中可以打印其他三架飞机的数据。

## 5. 关键知识点

### 关键知识点1：NetUavAPI.py

NetUavAPI.py 是最新开发的无人机通信接口程序。

### 关键知识点2：独立控制程序

NetSim4Demo1.py~NetSim4Demo4.py是四个飞机的单独控制程序。

### 关键知识点3：网络传输节点

net = NetUavAPI.NetTransNode(mav)创建了一个网络传输节点，负责通过网络与其他无人机或控制端进行数据通信。

### 关键知识点4：网络服务初始化

net.startNetServ(-1,30000,'224.0.0.11')初始化了网络服务，设置了多播地址和端口，用于在局域网中广播数据，224.0.0.11是一个多播IP地址，30000是端口号。

## 关键知识点5：重要网络命令

`startNetServ(RecPort=-1,netSimPort=20030,netSimIP='224.0.0.10')`是重要的命令，`RecPort`表示本飞机接收数据的接口，`netSimPort`表示数据发送端口，`netSimIP`是数据发送IP。在与粗粒度仿真器仿真时，`netSimPort`和`netSimIP`应该设置成仿真器的IP和端口，建议用组播地址和端口。本程序会将数据发送到(`netSimPort`,`netSimIP`)，经过网络仿真器中转，再根据目标ID，发往对应飞机的IP和端口。网络仿真器需要从接收到的数据中，读取飞机的IP，同时根据`22000+i`的端口规则，向对应飞机发送数据。

## 关键知识点6：心跳服务

`hearServer = NetUavAPI.HeartServer()`创建了一个心跳服务实例，用于监控无人机的连接状态。`hearServer.startHeartSer()`启动了心跳服务，使得无人机定期向指定端口发送心跳包，以确认连接的正常。

## 关键知识点7：数据接收处理

以下代码实现了从其他无人机接收位置数据的功能：

```
1 | for i in range(len(net.posGPSVect)):
2 |     if net.posGPSVect[i].hasUpdate:
3 |         net.posGPSVect[i].hasUpdate=False
4 |         print('Got msg from Copter #' + str(net.posGPSVect[i].CopterID) + ' Gps
    | pos: ', net.posGPSVect[i].latlonAlt, ' PosE: ',net.posGPSVect[i].UePosE)
```

## 6.参考资料

1. [RflySim官方文档](#)
2. 无
- 3.

## 7.常见问题

### Q1: \*\*\*

A1: \*\*\*

**Q2: \*\*\***

A2: \*\*\*

**Q3: .....**

A3: .....

---

1. <https://rflysim.com/> ↩

2. 推荐配置请见: <https://rflysim.com/doc/zh/HowToInstall.pdf> ↩