

RflySim数据通信（交互）与网络仿真接口学习实验

1. 实验目的

本实验旨在帮助学习者深入理解和掌握RflySim平台中无人机网络通信API的使用方法和原理。通过本实验，学习者将能够：

- 掌握RedisUtils工具类的使用方法，包括数据存储、队列管理、发布订阅机制等核心功能
- 理解NetUavAPI中的心跳机制和无人机状态监控原理，学会构建可靠的无人机网络通信
- 熟练运用NetSimAPIV4进行多无人机仿真，掌握单点通信、广播通信、组播通信等不同网络拓扑结构的实现
- 了解多无人机集群通信系统的整体架构和关键技术要点
- 掌握分布式系统中数据同步、状态管理、延迟监测等核心技术

2. 实验要求

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

3. 实验地址

例程目录：

[\[安装目录\]\RflySimAPIs\9.RflySimComm\0.ApiExps\e1.RflyNetAPIExps\0.APILearn](#)

- **RedisUtils.py**：基于 Redis 数据库的工具类，提供数据存储和共享功能，适用于分布式系统中的数据操作。
- **NetUavAPI.py**：无人机网络通信的核心接口库，主要负责无人机之间的网络通信和状态管理，包含心跳检测机制。

- **NetSimAPIV4.py**：高级网络仿真实接口，专门用于模拟无人机集群的通信和控制，支持多种通信模式。

4. 实验内容或步骤

4.1 步骤1：单点通信模拟

用enNetForward或enUavForward指定数据包发到指定的飞机（端口）列表，然后StartNetRec指定接收自己飞机（端口）数据

例如，1号飞机，指定将数据转发给2 3 4 5，并指定接收发往1号飞机数据

.....

5号飞机，指定将数据发往1 2 3 4，并指定接收5号飞机数据

通过上述配置，可以指定组建一种指定飞机ID的通信方式（对应单播）。

4.2 步骤2：广播(组播)模拟

用enNetForward或enUavForward指定数据包发往默认端口60000（或0号飞机），然后StartNetRec绑定并接收60000（或0号飞机）数据。

通过这种方式，相当于构建了一个广播（组播）网络，每个飞机都能收到其他飞机的数据，通过数据包内的CopterID，可以识别出数据来自哪个飞机。

4.3 步骤3：广播模拟+飞机筛选

在“广播模拟”的基础上，

1. 可以通过netResetSendList或netAddUavSendList来维护一个飞机列表，使得数据包只会被指定的飞机响应。
2. 可以通过StartReqUavData函数，来请求别的飞机，将自己加入飞机列表，以便能收到指定飞机的数据。

通过上述列表控制功能，可以实现网络数据的筛选。

4.4 步骤4：连接网络仿真器

在“单点通信模拟”的基础上，还可以通过enNetForward或enUavForward将数据包统一发给网络仿真器的端口（例如，65000）。

其次，每个飞机订阅自己的数据接口（例如，60000+CopterID）。

网络仿真器，对于每个飞机本应该收到的数据，会经过网络仿真模拟，加入延迟、丢包等环节，再将数据包转发到各个飞机。

例如，1号飞机本身要将数据发给3号飞机，但是它们之间被高墙遮挡，网络仿真器将会把1号飞机发给3号飞机的数据包拦截。

或者，3号飞机被对方网络干扰器锁定，则网络仿真器将不会转发任何数据包给3号飞机。

4.5 步骤5：数据类型

本接口能够转发和接受的数据类型主要有两种：

1. 来自mav的底层飞控数据，包括位置点等（需要将mav传入NetSimAPI构造函数），数据包结构见UAVSendData
2. 使用netForwardBuf直接将某个buf包转发出去，对方需要用同样的方法进行解包

注意：后一种方法，需要在外层python中，写一个死循环，并调用bufEvent.wait()来接收buf包来的信号。然后，从bufData中，获取buf包的值，并进行数据解析。

注意，还可以从bufHead=

[checksum,CopterID,sendMode,StartIdx,SendMask,TimeUnix]中获取数据包源头信息。

5. 关键知识点

本实验的整体思路是构建一个多无人机网络通信系统，通过三个核心API组件实现不同层面的功能。整个系统分为数据存储层(RedisUtils)、网络通信层(NetUavAPI)和仿真管理层(NetSimAPIV4)，形成一个完整的无人机集群通信架构。

关键知识点1: RedisUtils.py 关键知识点

该模块封装了Redis数据库的操作，提供数据存储和消息传递功能：

```

1 class RedisUtils:
2     def __init__(self):
3         # 建立连接
4         self.host = "localhost"
5         self.port=6379
6         self.db=0
7         self.password=""
8         self.rdb = redis.StrictRedis(host = self.host, port=self.port, db=self.db,
9 password=self.password)
        self.pubsub = self.rdb.pubsub()

```

- **JSON序列化存储**: 通过 `json.dumps()` 和 `json.loads()` 实现复杂数据结构的存储
- **发布订阅模式**: 利用Redis的发布订阅机制实现实时消息传递
- **多通道订阅**: 支持同时订阅多个频道的数据

关键知识点2: NetUavAPI.py 关键知识点

该模块实现无人机间的心跳检测和状态监控:

```

1 class CheckInfo:
2     def __init__(self):
3         ##解包/打包消息格式定义
4         #解析格式6i
5         self.Header = 123456789 # 校验码
6         self.cpID = 0 # 源ID
7         self.tgID = -1 # -1表示转发给所有飞机, 目标ID
8         self.msgID = 30 # 消息的类型, 这里是心跳
9         self.Resv = 0 # 保留位
10        ##消息解包打包
11        self.buffformat = '6i'
12        self.buffsize = struct.calcsize(self.buffformat)

```

- **结构化数据打包**: 使用 `struct.pack_into` 和 `struct.unpack_from` 进行二进制数据打包
- **心跳机制**: 定期发送心跳包检测无人机连接状态
- **多线程并发处理**: 使用threading模块实现接收、发送、状态检查的并发处理

关键知识点3: NetSimAPIV4.py 关键知识点

该模块提供高级网络仿真功能, 支持多种通信模式:

```

1 class UAVSendData:
2     # 无人机数据类，从mav中拿到无人机的数据，并转发到网络，给别的飞机
3     # 包含飞机的位置、速度、姿态角等
4     def __init__(self):
5         self.hasUpdate=False
6         self.timeDelay=0
7         self.CopterID=0
8         self.uavTimeStmp=0
9         self.uavAngEular=[0,0,0]
10        self.uavVelNED=[0,0,0]
11        self.uavPosGPSHome=[0,0,0]
12        self.uavPosNED=[0,0,0]
13        self.uavGlobalPos=[0,0,0]

```

- **数据包转发机制**：通过 `netForwardBuf` 方法实现数据包的网络转发
- **组播通信**：支持通过组播IP实现一对多通信
- **64位掩码算法**：高效管理无人机组的发送列表
- **延迟监测**：通过时间戳差值计算网络延迟

关键知识点4：整体架构设计思想

- **模块化设计**：三个模块各司其职，职责分离
- **网络通信优化**：采用UDP协议和多线程提升通信效率
- **容错机制**：具备失联检测和自动恢复能力
- **扩展性**：支持多种通信模式，适应不同场景需求

6.参考资料

1. RflySim官方文档：<https://rflysim.com/doc/zh/>
2. Redis官方文档：<https://redis.io/documentation>
3. Python Socket编程指南：<https://docs.python.org/3/library/socket.html>
4. MAVLink协议规范：<https://mavlink.io/en/>
5. 无人机集群控制技术研究论文

| 7.常见问题

| Q1: Redis连接失败怎么办?

A1: 检查Redis服务器是否启动，确认host、port、db等参数配置正确，确保Redis服务允许Python客户端连接。

| Q2: NetUavAPI心跳检测显示无人机失联?

A2: 检查网络连接是否正常，确认端口未被防火墙阻止，检查发送和接收IP、端口配置是否匹配。

| Q3: NetSimAPIV4无法正常转发数据?

A3: 确认enNetForward或enUavForward已正确启用，检查ForwardPort和ForwardIP配置，确认目标端口处于监听状态。

| Q4: 多无人机通信时出现数据混乱?

A4: 检查CopterID是否唯一，确认消息包中的ID标识正确，验证数据包解析格式是否匹配。

| Q5: 网络延迟过高影响通信效果?

A5: 优化网络配置，减少通信频次，使用64位掩码算法优化发送列表，考虑使用更高性能的网络设备。

1. <https://rflysim.com/> ↩

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