本章目录

# 本章实验案例与综合验证

**第九章 多机通信与智能组网**

9.1 组网通信基础与背景需求

9.2 硬件选型指标与通信机制

9.3 软件分层与通信协议

9.4 通信建模与干扰仿真

9.5 RflySim组网框架与通信仿真

9.6 关键通信协议与开发接口

**9.7 本章实验案例与综合验证**

9.8 习题与课堂讨论

**本节 4 个教学单元**

**9.7.1 例程体系与最小接口实验**

教材对应：第9.3.1节，接口学习实验

**9.7.2 多机通信与粗粒度仿真**

教材对应：第9.3.1节，基本使用与进阶开发

**9.7.3 联合仿真与高阶案例**

教材对应：第9.3.1-9.3.2节，图9.10

**9.7.4 结果解释与综合验证**

教材对应：第9.3节，案例综合验证

## 9.7.1 例程体系与最小接口实验：教材实验列表

# 实验分类与学习任务

共同父目录：[安装目录]/RflySimAPIs/9.RflySimComm/

| 教材分类与目录 | 要回答的问题 | 代表资源 |
| --- | --- | --- |
| **接口学习**<br>**0.ApiExps** | 数据如何发出、寻址和解析? | RflySim接口、DDS、MAVLink、MQTT、Redis、NS-3、组播 |
| **基本使用**<br>**1.BasicExps** | 通信怎样嵌入多机控制? | RflySim接口以及DDS、MQTT、Redis多机案例 |
| **进阶开发**<br>**2.AdvExps** | 约束网络后任务如何变化? | 粗粒度仿真、资源分配、路由仿真专题 |

选题查 Index.pdf，步骤查例程内 Readme.pdf，接口说明查 API.pdf。

## 9.7.1 例程体系与最小接口实验：同一组状态数据

# 四阶段默认实验路线

**P2P**
**两机状态收发** → **组播**
**三机状态共享** → **环形跟随**
**四机状态依赖** → **NetSim4Demo**
**粗粒度网络约束**

| 阶段 | 输入与操作 | 判断证据 |
| --- | --- | --- |
| **两机 / 三机** | 固定载具ID、端口和状态字段 | 源ID集合正确，时间戳连续更新 |
| **四机 / 网络约束** | 固定轨迹，再改变距离或模型参数 | 目标包确实经过模型，核对交付与时效 |

## 9.7.1 例程体系与最小接口实验：版本与副作用

# 环境准备与启动脚本的影响

**准备：**使用已安装工具链与例程自带环境，确认 PSP_PATH 指向实际安装目录；脚本缺省盘符可能仍为E盘。

**版本：**本资料核对基于RflySim 5.01安装版；Python38只是目录名。用python --version查版本，动态库还须匹配解释器版本与32/64位。

**副作用：**SITLRun*.bat 可能提权，并按名称结束 CopterSim、QGC、RflySim3D、python 等进程，影响其他会话。

**网络与依赖：**保留防护，检查具体应用、UDP端口、组播网卡和局域网范围。NetSim界面依赖 PyQt5、Matplotlib、NetworkX。

## 9.7.1 例程体系与最小接口实验：最小收发操作

# 两机点对点的启动顺序

0.ApiExps/e1.RflyNetAPIExps/
1.P2PSendData/

**准备：**打开 Readme.pdf，核对载具编号1、2和工具链环境；本例仅验证状态交换，无需起飞。

**启动：**运行 SITLRun2MavlinkFull.bat；等待两个 CopterSim/PX4 实例就绪。

**运行：**用 Python38Run.bat 分别打开两窗口，先执行 python UAV2Ctrl.py，再执行 python UAV1Ctrl.py。

## 9.7.1 例程体系与最小接口实验：点对点观察与判断

# 目标ID、UDP端口与源ID字典




发送目的：2　接收端口：60002　源字典：UavDict['1']
显式等价接收：net.StartNetRec(60002)

预期提示：Got UAV1 Data。核对 CopterID 为1，位置单位为米；本地NED位置与全局地图位置不能互换。

判断：出现目标源状态支持“接口已接通”；一次打印不代表持续包率或时延已达标。

### 9.7.1 例程体系与最小接口实验：一对多分发

## 三机组播的操作与源集合

0.ApiExps/e1.RflyNetAPIExps/
2.BroadSendData/

**准备与启动：**打开 Readme.pdf，运行 SITLRun3MavlinkFull.bat，再运行 UavPythonRunALL.bat。

**观察：**各窗口每秒打印所收到的载具ID和全局位置；默认发送与接收使用 224.0.0.10:60000。

**判断：**列出每个接收方的源ID集合，检查包含另两机且状态持续更新。本机回环另记，不能冒充另一节点。

## 9.7.1 例程体系与最小接口实验：接通与持续更新

# 组播不通时的逐层定位

| 观察现象 | 优先检查 | 复查证据 |
| --- | --- | --- |
| **完全无源ID** | CopterSim状态、Python环境、地址与端口、组播成员加入 | 发送侧状态变化，接收侧源ID出现 |
| **只收部分节点** | ID重复、源脚本未启动、目标网卡与网络范围 | 每个接收方的源ID集合 |
| **位置不变** | 飞机静止还是消息停止更新? | 时间戳与接收计数是否继续增长 |
| **本机通、跨机不通** | 所需程序/UDP规则、网卡、交换机组播策略 | 在同样发包条件下逐主机复查 |

本章目录

# 本章实验案例与综合验证

**第九章 多机通信与智能组网**

9.1 组网通信基础与背景需求

9.2 硬件选型指标与通信机制

9.3 软件分层与通信协议

9.4 通信建模与干扰仿真

9.5 RflySim组网框架与通信仿真

9.6 关键通信协议与开发接口

**9.7 本章实验案例与综合验证**

9.8 习题与课堂讨论

**本节 4 个教学单元**

**9.7.1 例程体系与最小接口实验**

教材对应：第9.3.1节，接口学习实验

**9.7.2 多机通信与粗粒度仿真**

教材对应：第9.3.1节，基本使用与进阶开发

**9.7.3 联合仿真与高阶案例**

教材对应：第9.3.1-9.3.2节，图9.10

**9.7.4 结果解释与综合验证**

教材对应：第9.3节，案例综合验证

## 9.7.2 多机通信与粗粒度仿真：基本使用实验

# 四机环形案例的准备与入口

1.BasicExps/e1.RflyNetSimExps/
3.PythonNetSimAPI-SimpPack/

**准备：**打开 Readme.pdf，确认4个实例编号1至4，仿真场景与初始位置一致。

**操作：**运行 SITLRun4MavlinkFull.bat，待全部3D Fixed后运行 UavPythonRunALL.bat。

**运行影响：**initOffboard修改仿真保护参数，随后解锁、起飞与跟随。
**坐标含义：**NED下向坐标−10 m表示参考原点上方10 m。

### 9.7.2 多机通信与粗粒度仿真：消息拓扑

## 环形通信与跟随控制的区别

| 源机 | 发送目的端口 | 接收者如何使用 |
| --- | --- | --- |
| 1 | 60002，发给2号 | 2号跟随1号，期望北向位置后移1 m |
| 2 | 60003，发给3号 | 3号读取2号状态并跟随 |
| 3 | 60004，发给4号 | 4号读取3号状态并跟随 |
| 4 | 60001，发给1号 | 1号等待4号就绪后执行预定运动 |

消息流为 1→2→3→4→1；持续跟随链为 1→2→3→4。

## 9.7.2 多机通信与粗粒度仿真：更新事件与控制

# 降采样和跟随误差的一步计算

接口：enNetForward([60002], '224.0.0.10', 2)；Interval=2 表示每3个已处理事件发送一次。

$$f_{\text{send}} \approx \frac{f_{\text{event}}}{I+1} = \frac{90}{3} = 30\ \text{Hz}$$

假设已处理状态事件为90 Hz；实际发送频率还受事件合并与调度影响。

教学输入：1号北向位置10 m，2号北向位置8.4 m；跟随后移1 m，比例增益为1 / s。

$$v_{2,N} = 1\ \text{s}^{-1}(10 - 1 - 8.4)\ \text{m} = 0.6\ \text{m/s}$$

代码在各轴速度绝对值小于0.2 m/s时置零，循环约每0.03 s执行；这是给定状态的计算示例，非实测响应。

## 9.7.2 多机通信与粗粒度仿真：粗粒度约束实验

# NetSim4Demo 的启动顺序

2.AdvExps/e1.CoaGraNetSimExps/
1.NetSim4Demo/

**准备：**打开 Readme.pdf 与 readme.txt；核对 PSP_PATH、4机环境和 PyQt5 / Matplotlib / NetworkX 依赖。

**默认操作：**SITLRun4MavlinkFull.bat → 全部3D Fixed → PythonRunALL.bat。

**分步排错：**在例程环境中先 python NetworkSimulation.py，再 UavPythonRunALL.bat，最后 python PyViaGui.py；勿重复启动。

## 9.7.2 多机通信与粗粒度仿真：转发数据路径

# 状态、业务包与界面使用不同端口

| 来源 → 去向 | 端口与地址 | 数据的作用 |
| --- | --- | --- |
| CopterSim → 网络模型 | 224.0.0.11:20008 | 168字节状态包，位置用于距离计算 |
| UAVCtrl → 网络模型 | 组播 224.0.0.11:30000 | 业务包，24字节头部含源/目标ID |
| 网络模型 → 接收程序 | 目标主机 UDP 22000+ID | 经概率丢弃/延迟处理的业务包；默认仅打印，未接入运动控制 |
| 网络模型 → PyViaGui | 224.0.0.11:32000 | 位置、直接链路质量与逐源统计 |

### 9.7.2 多机通信与粗粒度仿真：教学模型

## 距离怎样变成链路质量

默认 max_range=0.3 km，可靠区至120 m。令 d 为节点距离（m），q 为模型的单链路交付质量。

$$q(d)=\begin{cases}1, & d\le 120\\ 1-\dfrac{\log_{10}(d/120)}{\log_{10}(300/120)}, & 120<d<300\\ 0, & d\ge 300\end{cases}$$

| 距离 / m | 120 | 180 | 240 | 300 |
|---|---|---|---|---|
| 质量 q | 1.000 | 0.557 | 0.244 | 0.000 |

以上为确定性教学计算，非实测；模型未含射频标定与完整无线接入竞争。

## 9.7.2 多机通信与粗粒度仿真：模型一步计算

# 路径可靠性与计划延迟

教学网络：A到C直连质量0.25；A到B为0.8，B到C为0.7。各边质量仅作为本例模型输入。

$$Q(A \to B \to C) = 0.8 \times 0.7 = 0.56 > 0.25$$

$$\tau_{\text{plan}} = \frac{0.001}{Q}\ \text{s} = \frac{0.001}{0.56}\ \text{s} \approx 1.79\ \text{ms}$$

执行规则：随机数小于路径质量才入队；到期后转发至目标22000+ID。质量为0时丢弃。

该最大乘积模型与教材ETX最小和路径不同，也不等同于OLSR路由协议。

## 9.7.2 多机通信与粗粒度仿真：读取模型结果

# 界面统计的口径与实现边界

| 实现项 | 当前规则 | 解释结果时的边界 |
| --- | --- | --- |
| **丢包计数** | 按源统计每次目标转发尝试，<br>并含部分格式错误 | 广播一条消息可计多次；<br>未注册目标可能在计数前跳过 |
| **时间窗口** | 约1 s更新和清零统计 | 不同窗口不能直接相加成<br>统一的端到端丢包率 |
| **带宽门限** | data_rate 累加 len(buf)/128；<br>默认 bandwidth=100.0 | 是软件门限口径，<br>不是已标定的无线吞吐能力 |
| **延迟队列** | 按入队顺序检查到期队首 | 计划延迟不等于实际到包延迟，<br>后包可能被前包阻挡 |

## 9.7.2 多机通信与粗粒度仿真：对照操作

# 三种距离工况的实验判据

| 工况 | 操作与观察 | 判断 |
| --- | --- | --- |
| **近距离 ≤120 m** | 保持同一业务频率，<br>记录距离、质量、到包序号 | 直接质量为1；仍检查队列、<br>带宽门限和真实接收计数 |
| **边缘 120-300 m** | 按距离分段比较，<br>记录中继路径与到包间隔 | 模型直接质量下降；<br>随机损失需重复种子统计 |
| **直接距离 ≥300 m** | 同时查看其他链路和接收端 | 直接质量为0；存在中继时<br>路径质量可能仍大于0 |

固定：轨迹、业务大小、发包频率、控制器、统计区间；每次只改变一个网络因素。

本章目录

# 本章实验案例与综合验证

**第九章 多机通信与智能组网**

9.1 组网通信基础与背景需求

9.2 硬件选型指标与通信机制

9.3 软件分层与通信协议

9.4 通信建模与干扰仿真

9.5 RflySim组网框架与通信仿真

9.6 关键通信协议与开发接口

**9.7 本章实验案例与综合验证**

9.8 习题与课堂讨论

**本节 4 个教学单元**

**9.7.1 例程体系与最小接口实验**

教材对应：第9.3.1节，接口学习实验

**9.7.2 多机通信与粗粒度仿真**

教材对应：第9.3.1节，基本使用与进阶开发

**9.7.3 联合仿真与高阶案例**

教材对应：第9.3.1-9.3.2节，图9.10

**9.7.4 结果解释与综合验证**

教材对应：第9.3节，案例综合验证

### 9.7.3 联合仿真与高阶案例：细粒度网络仿真

# NS-3 联合仿真的准备条件

**一般入口：** 0.ApiExps/e6.NS3DepsTestExps/ 内有 RflyNet.cc、TestSend.py 与 PythonAllUavDemo/。

**定制环境：** 3.CustExps 下相关案例仅限完整版及以上。现有脚本使用 WSL 发行版 RflySim-20.04 和 ns-3.33 定制源码树。

**依赖条件：** 核对 gpsr、aodvKmeans、parrot 等自定义模块，编译器、网络接口、实时调度与UDP端口映射。

**源码核对：** 该入口RflyNet.cc有位置/速度字段互换，先修正映射再验证。普通NS-3发行包不能直接替代定制树。

### 9.7.3 联合仿真与高阶案例：条件性选学

# 定制入口的功能与目录依赖

共同目录：3.CustExps/e2.NS3_UAVCommExps/
仅限完整版及以上版本；各目录入口均为 OneScript.bat。

| 子目录 | 目标与观察对象 | 依赖 |
| --- | --- | --- |
| **1.NS3** | 基础联动；文档明确飞机不飞 | 自身 ns-3.33.zip 与 netsim/ |
| **2.NS3-Udp-PyQtViz** | UDP网络状态与PyQt可视化 | 复用相邻 1.NS3 源码树 |
| **3.NS3-PyViz** | NS-3中的PyViz拓扑显示 | 复用相邻 1.NS3 源码树 |
| **4.NS3-Ctrl** | 网络与控制联动目标 | 自身源码树与独立压缩包 |

## 9.7.3 联合仿真与高阶案例：安装与启动副作用

# OneScript 的操作顺序与写入影响

适用：3.CustExps/e2.NS3_UAVCommExps/，仅限完整版及以上。

**核对 Readme.pdf**
**备份实验副本**

→

**进入选中子目录**
**运行 OneScript.bat**

→

**检查构建与进程**
**再核对端口收发**

脚本行为：启动uav仿真，WSL内解压 ns-3.33.zip，将 netsim/RflyNet.cc 复制到 scratch，再 sudo ./waf configure / build / --run。

影响：源码覆盖、构建文件写入、提权和多进程启动。网络不通时检查应用与端口规则，保持防护开启。

### 9.7.3 联合仿真与高阶案例：准备、操作与判断

# 联合仿真的分阶段验收

| 阶段 | 操作与观察 | 通过判据 |
| --- | --- | --- |
| **构建与配置** | 记录源码版本、模块、构建输出 | 构建退出成功，目标程序存在 |
| **最小包转发** | 用 TestSend.py 或案例uav入口，跟踪源/目标ID和UDP端口 | 同一序号消息可在目标端对应 |
| **运动与网络** | 输入已知轨迹，核对坐标和单位 | 运动变化对应到NS-3节点位置 |
| **控制与对照** | 在控制专用案例里切换网络工况 | 控制输入来自模型出口，共同窗口统计通信与任务指标 |

## 9.7.3 联合仿真与高阶案例：虚实联动案例

# 真机与虚拟场景的通信联动

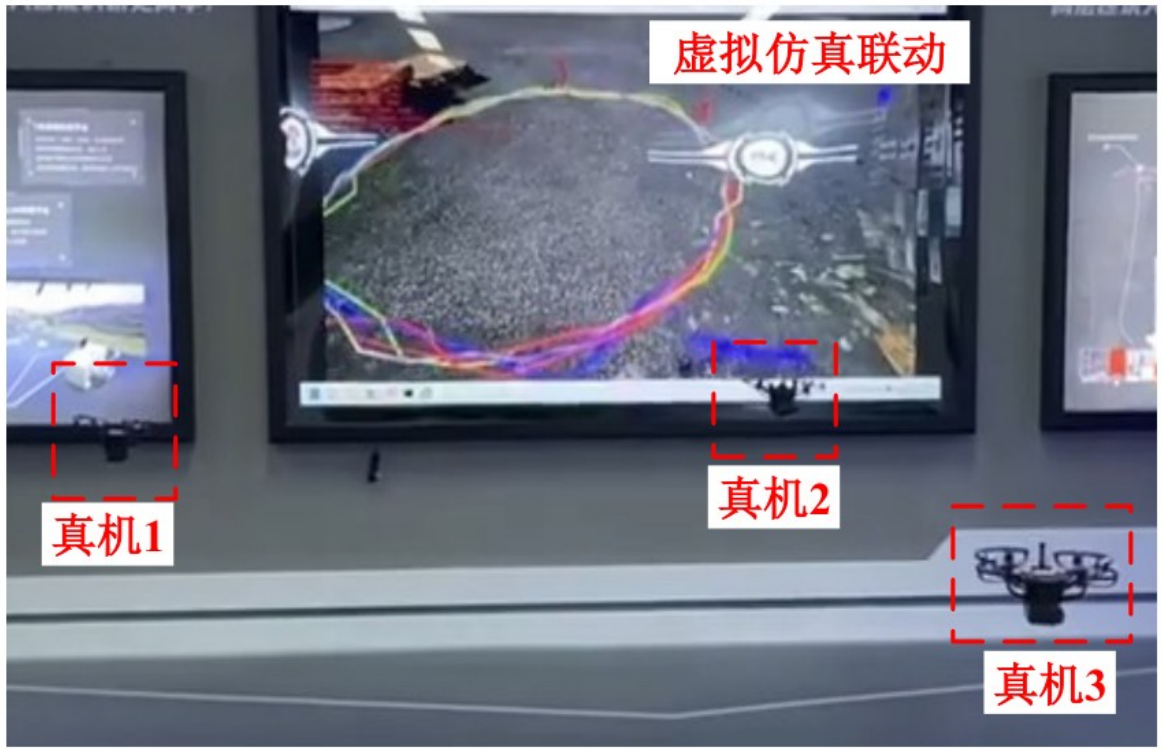


图9.10(a) 虚实联动通信仿真

**对象：**真机节点经局域网与仿真节点连接，虚拟场景提供任务环境。

**机制：**节点消息经过网络仿真器中转，再进入控制与显示。

**判断：**记录哪些消息受网络约束，确认没有理想直连旁路。

## 9.7.3 联合仿真与高阶案例：数字孪生可视化

# 三维通信特效承载哪些信息

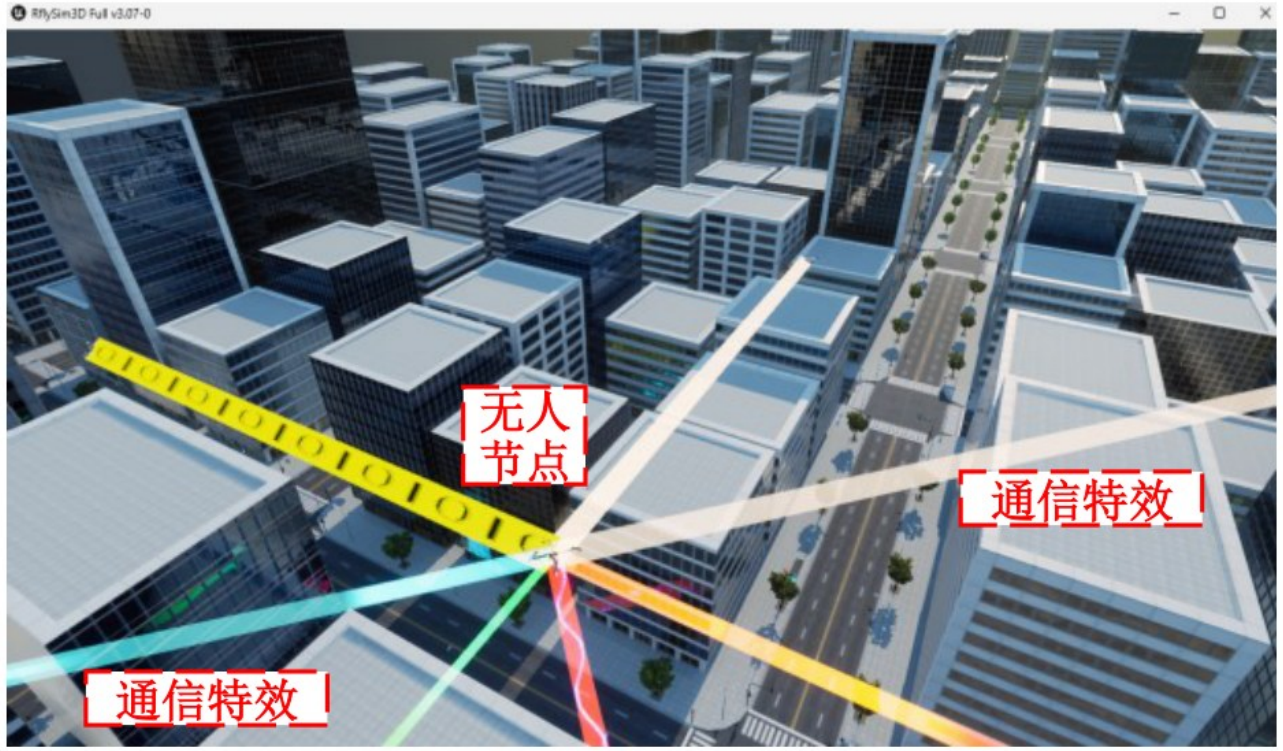


图9.10(b) 数字孪生通信可视化

**对象：**场景中的无人节点、连线和区域显示通信状态。

**机制：**UE蓝图接口把连通性、速率、时延或丢包统计映射为特效。

**判断：**先明确颜色和线型的数据来源，再与同一时刻日志对齐。

## 9.7.3 联合仿真与高阶案例：粗粒度结果示意

# 节点位置与直接链路质量

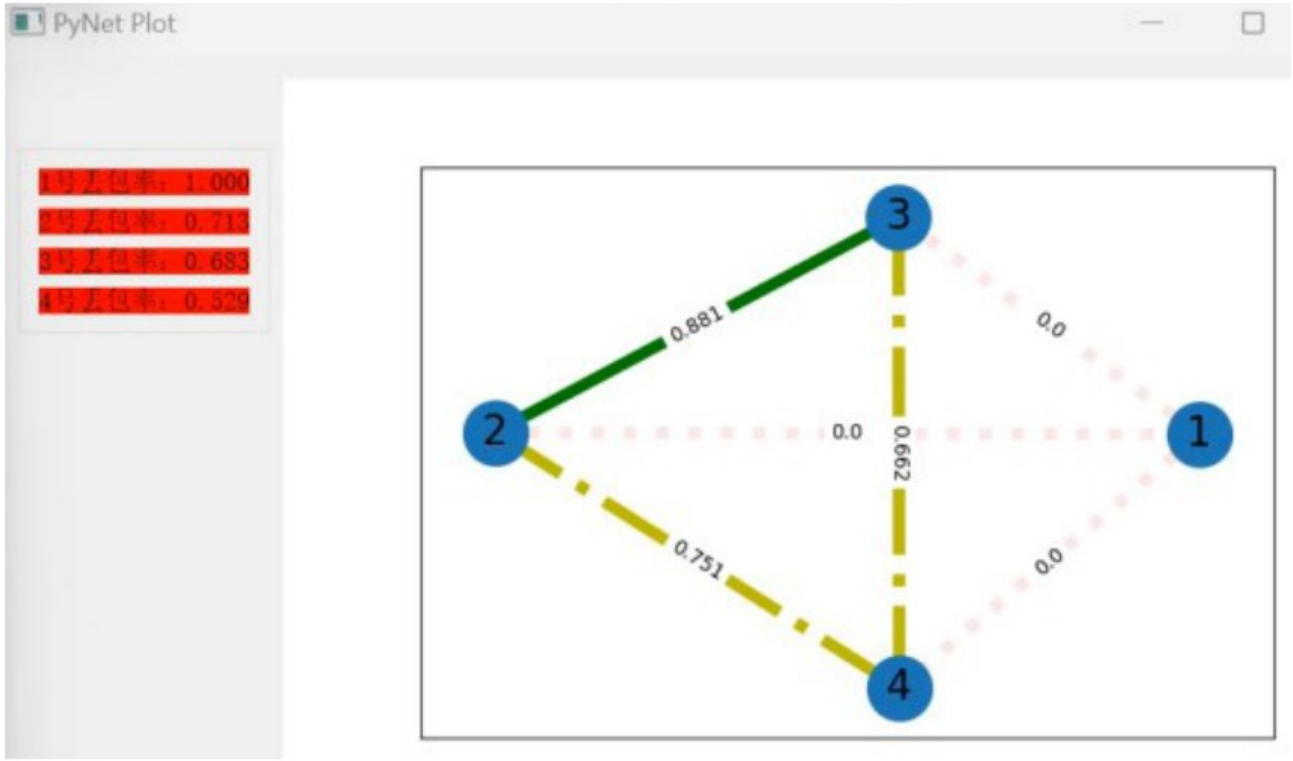


图9.10(c) Python粗粒度网络仿真

**对象：**节点位置和节点对之间的连接质量。

**对应：**距离进入质量映射，质量再影响选路和交付概率。

**判断：**将某条直接链路与端到端路径区分，按同一时间窗核对。

## 9.7.3 联合仿真与高阶案例：NS-3结果示意

# 细粒度仿真中的拓扑与分组事件

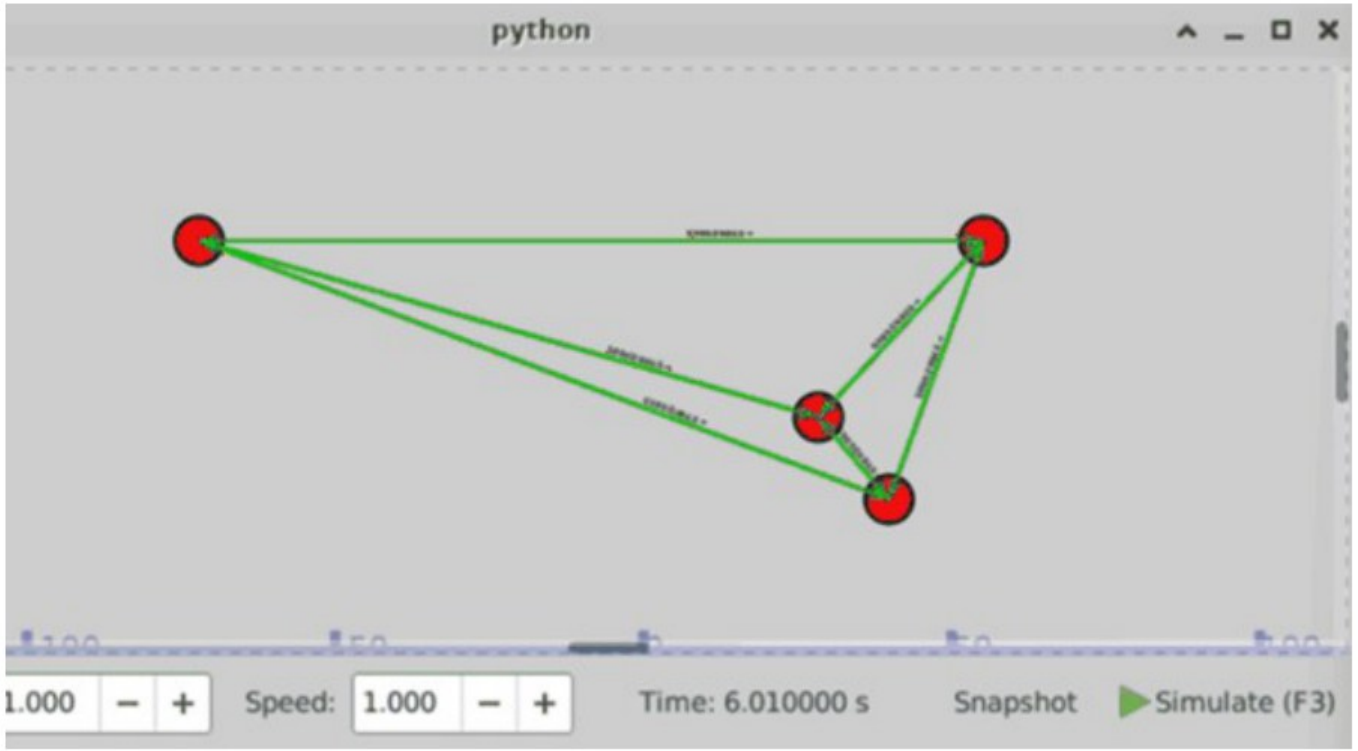


图9.10(d) NS-3细粒度通信仿真

**对象：**网络节点、连接关系及通信可视化。

**机制：**设备、信道、应用与路由在离散事件中共同作用。

**判断：**对应包事件、路由表与接收统计，不能仅凭连线判断时延达标。

## 9.7.3 联合仿真与高阶案例：资源分配与中间件

# 其他方案的选学边界

| 目录或入口 | 可以研究什么 | 条件与边界 |
| --- | --- | --- |
| 0.ApiExps/e2.DDSDepsTestExps/<br>Demo/HelloWorldExample.py | 发布 / 订阅匹配，参数<br>-p subscriber / -p publisher | 先核对DDS与Python绑定版本，<br>以及动态库的32/64位 |
| 0.ApiExps/e4.MQTTDepsTestExps/<br>0.ApiExps/e5.RedisDepsTestExps/ | Broker分发、Pub/Sub<br>或键值轮询 | 额外服务与版本需匹配；<br>不作为默认实验必装项 |
| 2.AdvExps/e2.NetResAlloc/<br>main.py | 随机策略、环境与界面<br>之间的交互流程 | 主入口固定random，<br>其他算法分支为TODO |

本章目录

# 本章实验案例与综合验证

**第九章 多机通信与智能组网**

9.1 组网通信基础与背景需求

9.2 硬件选型指标与通信机制

9.3 软件分层与通信协议

9.4 通信建模与干扰仿真

9.5 RflySim组网框架与通信仿真

9.6 关键通信协议与开发接口

**9.7 本章实验案例与综合验证**

9.8 习题与课堂讨论

**本节 4 个教学单元**

**9.7.1 例程体系与最小接口实验**

教材对应：第9.3.1节，接口学习实验

**9.7.2 多机通信与粗粒度仿真**

教材对应：第9.3.1节，基本使用与进阶开发

**9.7.3 联合仿真与高阶案例**

教材对应：第9.3.1-9.3.2节，图9.10

**9.7.4 结果解释与综合验证**

教材对应：第9.3节，案例综合验证

### 9.7.4 结果解释与综合验证：指标口径

## 通信指标的定义与统计对象

统计约定：固定源-目标业务流，发送窗口长度T；去重、剔除预热，停止发送后等待约定排空时间。

$$P_{\text{loss}} = 1 - \frac{N_{\text{rx,unique}}}{N_{\text{tx}}}$$

$N_{\text{tx}}$：发送条数；$N_{\text{rx,unique}}$：去重后收到的条数。

$$G = \frac{8\sum L_{\text{payload,rx}}}{T} \quad [\text{bit/s}]$$

$L_{\text{payload,rx}}$：每条成功消息的负载字节数；$T$以秒计。

时延：每条成功消息的接收时刻减发送时刻，报告中位数、95分位数和最大值，并注明时钟误差。

任务：同步记录跟随误差、失联持续时间和降级次数。通信达标与任务达标分别给出判据。

### 9.7.4 结果解释与综合验证：教学记录，非实测

# 六条消息的可复算记录示例

| 序号 | 发送 / ms | 接收 / ms | 时延 / ms |
|---|---|---|---|
| 1 | 0 | 10 | 10 |
| 2 | 20 | 35 | 15 |
| 3 | 40 | 65 | 25 |
| 4 | 60 | 75 | 15 |
| 5 | 80 | 丢失 | 不计入时延 |
| 6 | 100 | 140 | 40 |

本例生成即发送，使用同一时钟；窗口1 s，每包100 B。6发5收：丢包率16.7%，有效吞吐4,000 bit/s。

成功包均值21 ms、中位数15 ms；最近秩P95取排序后第5项，即40 ms。

## 9.7.4 结果解释与综合验证：时钟与状态新鲜度

# 时间戳与信息年龄

$$\hat{d} = t_{\mathrm{rx}}^{(B)} - t_{\mathrm{tx}}^{(A)} = d + \Delta_{BA}$$

$d$为真实时延，$\Delta_{BA}$为接收主机B相对发送主机A的钟差。跨机相减前须统一时间基准。

$$A(t) = t - t_{\mathrm{gen,latest}};\qquad A(100\ \mathrm{ms}) = 100 - 60 = 40\ \mathrm{ms}$$

上页100 ms时，已收状态中生成时刻最新的是序号4（60 ms生成）。若以10 m/s匀速运动，这40 ms内位移为0.4 m。

0.4 m是状态陈旧期间的位移示例，实际跟随误差还取决于预测、反馈和控制器。

### 9.7.4 结果解释与综合验证：综合验证设计

# 从理想通路到受限网络的对照

| 对照阶段 | 保持一致 | 增加的机制与归因 |
| --- | --- | --- |
| **理想直连** | 参考轨迹、初值、业务大小/频率、控制器、窗口与时钟口径 | 基线接口和任务表现 |
| **粗粒度模型** | 同一业务流及相同参考轨迹 | 距离质量、概率损失、计划延迟和软件门限对结果的影响 |
| **NS-3细粒度模型** | 同一节点身份与业务输入 | 设备、信道、接入与路由事件解释排队和交付差异 |

输出：参数表、逐包日志、实际轨迹与任务误差；记录工况和重复次数。

### 9.7.4 结果解释与综合验证：失败定位与回退

# 结束实验与保留可追溯记录

**记录：**保存版本、脚本SHA、参数、轨迹、随机种子、统计窗口、时钟基准，以及每条流的发送/接收序号。

**结束：**在各实验窗口先 Ctrl+C 停控制和发送，再停网络模型与界面，最后关闭本次仿真器。

**阻塞处理：**endNetLoop() 存在等待接收线程的风险；必要时按窗口与PID只结束本次进程。

**恢复：**停止后确认端口释放，保留失败日志；重试前恢复实验副本参数，不用全局终止python作为常规退出。

练习与讨论

# 课堂讨论：结果能够支持什么结论

**问题1：**直接链路质量为0，目标仍收到包。怎样区分有效中继、旧缓存与绕过模型的通路？

**问题2：**方案甲平均时延15 ms、丢包30%；方案乙25 ms、丢包1%。能否只凭平均时延选甲？

**问题3：**跨机时间戳相减得到负时延，应先改网络模型，还是检查时钟、单位和消息身份？

参考要点

# 讨论参考：沿同一业务流寻找证据

| 问题 | 参考判断 | 需要的证据 |
| --- | --- | --- |
| 1 直接断链仍到包 | 中继可能有效，也可能是旧样本或理想通路绕过模型 | 同序号新包、路径质量、模型出口与目标到包日志 |
| 2 低平均时延 | 更高丢包可能只留下快速样本；按业务目标综合评价 | 去重丢包率、P95、信息年龄、跟随误差与失联时长 |
| 3 负时延 | 优先定位时钟和数据配对 | 单位、时钟偏差界、源/目标ID与消息序号 |

本节小结

# 本节小结：从接通到有边界的验证

**方法：**沿默认四阶段路线逐步增加约束，保留消息身份和可复算的数据记录。

**判断：**通信指标解释数据交付，任务指标解释使用这些数据后的控制效果；结论绑定版本、工况与窗口。

**易错点：**ID不是UDP端口；组播不是全互联控制；模型质量和界面颜色不是实测链路性能。

**衔接：**课件9.8复算链路与路由，并用明确的容量、时延和失效条件设计综合验证方案。