

前期准备

在开始实现各协议版本之前，需要完成以下环境准备：

1. **Git 环境配置**
2. **IDEA 环境配置**
3. **PPT 学习与实验规划**

由于大多数同学使用 Windows，本手册主要以 Windows 配置为例；Linux / macOS 用户的环境配置相对简单，建议自行查阅或借助 AI 完成。

Git 环境配置

为了体现**迭代开发（Iterative Development）**的过程，本实验要求使用 Git 管理版本，不同协议版本对应不同分支。

Windows 用户可参考以下文章进行 Git 安装（重点关注 PATH 配置部分，使得在 PowerShell 中能够直接使用 `git` 命令）：

👉 **【2025年最新版】Git安装及环境配置超详细教程（以win11为例）**

https://blog.csdn.net/Little_Carter/article/details/155110165

💡 无代理的同学可使用镜像站 clone 仓库，无需配置 SSH。可以参考目录中解压项目处

本次实验涉及的 Git 命令较少，掌握以下几个即可：

Git 命令	用途
<code>git init</code>	初始化仓库
<code>git add</code>	将修改加入暂存区
<code>git commit</code>	提交修改
<code>git checkout</code>	切换分支或版本
<code>git switch</code>	更安全地切换分支
<code>git reset</code>	回退到某版本

如需更方便的操作方式，也可以使用可视化 Git 工具，例如

👉 **Fork - a fast and friendly Git client**

<https://git-fork.com/>

或者 IDEA 自己的 Git GUI 工具。

IDEA 环境配置

本实验使用 IntelliJ IDEA 进行代码编写。社区版（Community Edition）功能已完全足够，无需寻找付费破解版本。

下载链接：

<https://www.jetbrains.com/zh-cn/idea/download/?section=windows>

安装步骤如下：

即可完成安装。

PPT 学习与实验规划

本实验采用 **从简单到复杂的迭代实现方式**，逐步实现：

1. UDP 本体（即 RDT 1.0 的原始程序）
2. RDT_2.0
3. RDT_2.X
4. RDT_3.0
5. TCP
6. Tahoe
7. Reno

实验内容和评分标准如下：

实验内容

- 评分：参见实验报告和评分表
- 实验报告要求：从实验数据文件中自己说明实现了相关功能
- 调试：设置TCP首部中的eFlag指明支持处理的错误类型，包括：
0.信道无差错
1.只出错
2.只丢包
3.只延迟
4.出错 / 丢包
5.出错 / 延迟
6.丢包 / 延迟
7.出错 / 丢包 / 延迟
- 服务器端Log日志：记录所有数据包的传输（DATA和ACK）
- 成绩评定：实验报告 + Log文件 + recvData文件 + 代码包
- 评定准则：报告通过 代码及Log 文件证明实现了每项要求的功能

标准	得分
RDT 2.0	3
RDT 2.2	6
RDT 3.0	8
GB/SR/TCP	13
TCP Tahoe	15
TCP Reno	16
未完成关键困难	2
迭代开发	1
已解决主要问题	1
实验系统建议	1
合计	

你们本地的 Git 分支视图可能类似如下（示例中为远程分支，因此有 `--remote`）：

```
PS C:\Users\24124\Desktop\TCP\OUC_TCP> git branch --remote
origin/HEAD -> origin/main
origin/RDT_1.0
origin/RDT_2.0
origin/RDT_2.X
origin/RDT_3.0
origin/Reno
origin/TCP
origin/Tahoe
origin/main
PS C:\Users\24124\Desktop\TCP\OUC_TCP> |
```

在 IDEA 中查看效果如下：



在此不再赘述 PPT 中说明的开发内容。请参考 PPT 和《自顶向下》进行后续开发。

RDT 1.0

1. 克隆项目

访问洪老师的仓库：

👉 <https://github.com/Alan-Hong/TCP-Lab>

使用 Git 克隆仓库：

若网络不稳定，可使用镜像站：



```
git clone https://ghfast.top/https://github.com/Alan-Hong/TCP-Lab.git
```

示例：

The lab materials for undergraduates to realize TCP protocol on the simulation platform.

Readme
Apache-2.0 license
Activity
0 stars
0 watching
1 fork
Report repository

2. 解压项目

解压 `TCP_Test_1.zip`，得到实验项目文件夹，其结构如下：

```
.
├── lib
├── src
│   └── com
│       └── ouc
│           └── tcp
│               └── test
└── target
    └── ...

```

将 `TCP_TestSys_Linux.jar` 放到 `TCP_Test_1/lib` 目录下，即可在 IDEA 中运行程序。

3. 使用 IDEA 打开项目

打开 IDEA，在项目中找到 `TestRun` 文件（程序入口），即可运行：

```
1 package com.ouc.tcp.test;
2
3 import com.ouc.tcp.app.SystemStart;
4
5 public class TestRun {
6
7     public static void main(String[] args) throws InterruptedException {
8         SystemStart.main(args);
9     }
10
11 }
```

```
Packet data:  
PACKET_TYPE: ACK_99801  
Receive ACK Number: 99801  
  
Clear: 99801  
-> 2025-12-06 08:06:41:210 CST  
** TCP_Receiver  
    Receive packet from: [10.140.246.225:19001]  
    Packet data: 1298173 1298191 1298197 1298221 1298261 1298279 1298291 1298309 1298317 1298329 1298333 1298351 1298357 1298371 1298387 1298467 1298489 1298491 1298537 1298551 1298573 1298581 12986  
PACKET_TYPE: DATA_SEQ_99901  
  
-> 2025-12-06 08:06:41:210 CST  
** TCP_Sender  
    Receive packet from: [10.140.246.225:19002]  
    Packet data:  
    PACKET_TYPE: ACK_99901  
Receive ACK Number: 99901  
  
Clear: 99901  
  
***** TCP_Sender: Data sending ends. *****
```

工具支持

Log 可视化工具 (TCP Lab Visualizer)

若需要可视化分析 Log，可使用仓库中提供的可视化工具。

1. 安装 uv (推荐)

- Windows:



```
powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"
```

- macOS / Linux:



```
curl -LsSf https://astral.sh/uv/install.sh | sh
```

2. 运行可视化工具

进入 `TCP-Lab-Visualizer` 目录:



```
uv sync  
uv run main.py
```

终端会显示：



Dash is running on <http://127.0.0.1:8050/>

打开浏览器 → 上传 **Log.txt** → 完成可视化。

