

# 《计算机网络及应用》 实验四 TCP协议实现实验文档

2022年版

## 实验目的

- 通过对TCP协议的实现，加强对TCP协议的掌握和理解，加深对“TCP协议是如何确保可靠数据传输”的问题的理解。
- 通过动手实现代码，了解网络报文的基本特点和传递过程，增强在数据包级别进行网络编程的能力。

## 实验内容及评分标准

本实验要求大家实现一个能够支持外发连接(outgoing, 即向外发出SYN, 即本机作为客户端)的TCP协议。如果你的TCP协议正确实现的话, 你可以借助它访问WWW互联网, 甚至下载几个GB的大文件, 用亲手搭建的TCP协议与世界通联!

本实验需要大家在虚拟机中完成, 这是因为这个实验必须依赖于Linux操作系统。

- 如果有同学物理机使用的就是Linux系统、且不想在虚拟机上完成这个实验, 也没关系, 直接把实验套件clone到机器上, 相信你读了实验套件中的README之后就知道该怎么做了。如有疑问也可随时联系助教。

本实验的具体要求及评分标准如下:

- 功能要求: 80分。要求主机上的程序能够借助实验套件, 通过你实现的TCP协议, 向其他人发起TCP连接并传送数据。正确实现所需功能可获满分, 未能实现指定功能, 可根据实现的完成度给予部分分数。
  - 正确实现TCP的三次握手, 成功建立连接 (20分)
  - 能够双向发送较小规模的数据 (30分)
    - 测试方法: `curl www.baidu.com`, 可通过肉眼查看
  - 正确实现TCP的四次挥手, 正确关闭连接 (10分)
  - 能够收到中等规模的数据 (1~10MB) (5分)
    - 测试方法:

```
curl
https://mirrors.bfsu.edu.cn/ubuntu/pool/main/p/python3.10/libpython3
.10-dev_3.10.9-1_amd64.deb
# 可通过sha256sum命令验证文件的哈希, 方法为: sha256sum <文件名>
# 上述文件的哈希应为:
6b803051e551a708f74e2fc6823797b6076e94637c758fbdfd0488891b7d4551
```

- 能够向外发送中等规模的数据 (1~10MB) (5分)
  - 测试方法:

```
dd if=/dev/urandom of=/tmp/testfile bs=1M count=3 # 生成一个3MB的文件。
只需每次虚拟机开机后执行一次就可以
curl https://starrah.cn/test_upload -F "file=@/tmp/testfile" # 服务器
会返回你所上传的文件的哈希
```

- 能够通过浏览器访问真实的网页 (5分)
  - 测试方法：打开浏览器（Firefox），访问你喜欢的网页。
  - 可以是网页，但希望其界面复杂程度不低于baidu.com。
  - 这是为了确保你访问的网页中有多个引用对象，而不是单一的HTML，以验证你的程序在处理多个连接时的正确性。
- 具有超时重传机制 (5分)
- **不要求实现的内容：凡是上面功能要求中未列出的需求和特性，均不在实验要求实现的范围**  
**内，你可以自愿实现。**举例包括：
  - 流量控制（包括可不遵守对方声明的发送窗口、可以为自己设置一个固定的接收窗口）
  - 拥塞控制
  - 快速恢复（三个重复ACK自动重传缺失段）
  - 对收到的乱序报文进行缓存（指，当你收到乱序报文时，可以简单地丢弃它或回复ACK（相当于快速恢复），直接等对方重传就好）
  - 所有TCP options
- 代码质量和文档分：20分。
  - 代码质量：10分。
    - 编写的代码应该清晰、明确、易于理解
    - 具有适当的注释
    - 符合所使用语言的编程最佳实践
    - 提交的代码中不应有cmake-\*、target等编译中间产物文件夹，或node\_modules之类的巨大的且很容易重复下载的依赖文件
      - 如果你提交的src文件夹超过了1MB（C++除外，因为里面有一个内嵌的1M左右的依赖），大概率就是没满足这个要求。
  - 文档：10分。具体文档中应包含的内容详见后文[提交方法与迟交政策](#)部分。

## 实验环境

请大家使用虚拟机。该虚拟机与实验三中相同，已经使用过虚拟机完成实验三的同学，可不必重复下载。

下载方法：

1. 安装虚拟机软件：我们推荐大家安装 VirtualBox。当然，你也可以使用 VMWare、QEMU/KVM 等其他的虚拟机软件。
  - 为了方便大家，我们在清华云盘中提供了 Windows 和 Mac(Intel)的 VirtualBox 安装包。
  - 如你的计算机不能使用我们上面提供的安装包，请自行到下载地址 <https://www.virtualbox.org/wiki/Downloads> 下载。
2. 下载虚拟机镜像文件：由于虚拟机镜像文件过大、无法上传清华云盘,请尽量使用校园网环境(不是校园网可能会造成下载速度过慢),访问 <https://lab.starrah.cn:21011/> , 下载虚拟机镜像文件 ubuntu22.04-NetworkCourse-v0.1.ova。
3. 在 VirtualBox 中导入上述的虚拟机镜像,然后开启虚拟机：请详见实验三文档的附录2、导入和运行虚拟机，篇幅所限这里不再重复。

4. 虚拟机的用户名为user，密码为thu。

实验三的虚拟机中缺少一些必要或你很可能需要的依赖，请下载安装一下：

```
sudo apt install git curl python3-pip wireshark
sudo usermod -aG wireshark $USER
```

注意，安装wireshark的时候，会弹一个窗，最后的问题是问你 `should non-superusers be able to capture packets?`，选择是。

安装好后，要重启虚拟机一次，wireshark才能用。

然后，请在虚拟机中使用如下命令，下载实验套件：

```
git clone https://gitee.com/Starrah/tcp-impl.git
```

下载好后，`cd tcp-impl`，其中即为本次作业的实验套件。可选地，你可以进一步阅读这个文件夹中的 `README.md`，以增强对实验套件的了解（其中重要的内容也已包含在了本文档里，因此不阅读 `README.md`也是可以的）。

## 实验套件和思路提示

### 关于实验套件

实验套件分为driver和SDK两部分。你应当使用给出的多种语言SDK的**任意一种**，实现指定的函数（实现的过程中会需要调用一些给定的API）。

### 项目结构

- driver-src: driver部分的源代码。
  - 构建方法：`cd driver-src && sh build-driver.sh`
- driver：使用 driver-src 中的 `build-driver.sh` 脚本编译得到的可执行文件。
  - 该文件的目标平台：`amd64`
  - 如果你不能或不想使用该预编译的文件，请自行编译。
- sdk：包含所有语言的SDK。子目录名和对应支持语言如下表：（打勾表示已完成支持，未打钩表示计划但尚未完成支持）
  - ☒ cpp: C++
  - ☒ python: Python
  - ☒ rust: Rust
  - ☐ java: Java/Kotlin，不晚于1月9日24点前完成支持
  - ☐ nodejs: 基于Node.js运行时的JavaScript/Typescript语言，不晚于1月10日24点前完成支持
  - ☐ go: Go，不晚于1月10日24点前完成支持

当完成支持后，你可以更新实验套件，详见[更新实验套件](#)部分所述。

### SDK具体解释

需要你实现的函数都在 `sdk/<对应语言的子目录>/src/outgoing.<对应语言的后缀名>` 中，共五个函数。

- `app_connect`
- `app_send`

- `app_fin`
- `app_rst`
- `tcp_rx`

每个函数函数头均有注释描述其被调用的时机和参数含义，函数体内有TODO注释提示你应该在这里编写代码。

这些函数使得你实现了一个TCP协议（仅支持外发连接）所应有的功能，包括应用层的发起连接、发送数据、FIN半关闭连接、RST重置连接，和接收网络层协议转交给我们的TCP报文。

为了实现这些功能，你将免不了调用一些函数，你会调用的函数均在名为 `api.<对应语言的后缀名>` 的文件中，共有六个函数：

- `app_connected`
- `release_connection`
- `app_recv`
- `app_peer_fin`
- `app_peer_rst`
- `tcp_tx`

每个函数函数头均有注释描述作用。你不应该修改这些函数的实现。

这些函数为你提供了你所需要的功能，包括通知应用层连接已建立、向应用层传递对端发来的数据、当对端FIN半关闭连接或RST重置连接时及时通知应用层，和将你封装好的TCP报文交给网络层，请上层协议栈帮你发出去。

## 运行程序

- 首先运行driver：在项目目录(`tcp_impl` 目录)下，运行 `sudo ./driver`
  - **一定要记得sudo！**
  - 一旦 `driver` 被以默认模式运行起来，你虚拟机中所有的外发网络流量就都会被driver拦截。如果你的TCP协议还没有正确实现完成，那么这是你在虚拟机内就是上不了网的。关掉driver就可以解除拦截。
    - 特殊情况下，driver如果崩溃，会造成driver退出了但仍在拦截网络的状态。此时可使用 `sudo ./driver clean` 解除拦截。
  - 使用 `./driver --help`，可以查看更多使用帮助。
- 然后，运行你自己的代码。根据语言不同，运行的方法不尽相同，可查看SDK对应目录下的 `README.md`。
  - 想要重启你自己的代码时，并不需要重启driver。但如果你发现出现了奇怪的问题，重启driver可能有用。

## 更新实验套件

- 本实验是第一年研发出来，不能保证完全bug-free。当你发现遇到困难，特别是自己看不懂的问题的时候，请及时向助教询问。
  - 本实验的研发助教是赵乙宁，建议优先通过微信联系。
- 不能排除本实验的实现会有bug，而且现在还有几个语言的SDK没有来得及写完。因此，助教可能会更新实验套件，并在群内说明。你也可以自行去<https://gitee.com/Starrah/tcp-impl>，根据git版本记录查看实时的更新情况。
- 在助教更新后，你应该做的事情是：

- **备份你写的所有代码！** 虽然熟练使用Git的人不会出问题，但由于大家可能不太熟悉，**为了防止自己的心血前功尽弃，请在更新实验套件前一定一定一定备份好自己的代码！** 助教不为更新套件造成的代码丢失负责！
- 在项目文件夹（即 `tcp_impl` 文件夹下）执行命令：`git pull`，就更新完了。
- 如果更新失败，发生了你不能理解或无法解决的问题，可以联系助教。**谨慎删除任何文件！**

## 思路提示

为了完成实验，认真阅读并理解 `outgoing` 文件和 `api` 文件是必不可少的。

- 以下是一些大致思路，帮助你捋清：
  - 当有程序想要连接（比如你运行了一个 `curl` 的时候），`app_connect` 函数就会被调用。在这里，你应该为这个新的连接创建恰当的状态记录（TCP状态机），然后组织好一个SYN报文（数据格式是长为20的字节数组或其类似物，取决于语言，请看 `tcp_tx` 函数接受什么类型的参数就知道了），调用 `tcp_tx` 函数，这样你的报文就会被发出去；
  - 如果你实现了正确的SYN报文，那么你通信的对方应该会回复给你SYN-ACK报文，这时 `tcp_rx` 函数就会被调用，你会拿到收到报文的内容。你应该对报文进行解析，更新你关于这个连接状态记录，回给它ACK报文（调用 `tcp_tx`），并调用 `app_connect` 函数通知应用层连接已建立了。
  - 每次应用层想发数据给对方，`app_send` 函数就会被调用，你应该根据用户想发的内容组织成报文，然后调用 `tcp_tx` 函数发出去。
  - 同理，每次收到对面回给你的数据包，无论是数据、ACK还是其他的什么，`tcp_rx` 都会被调用。那么你应该根据包里的内容做适当的处理，比如有数据的包，要调用 `app_recv` 函数把数据递交给应用层，纯ACK包指更新状态机就好，FIN包，进行ACK同时调用 `app_peer_fin` 函数通知应用层对面已经FIN了。
- 关键点在于：
  - TCP是有状态的模型，因此你不可避免会去存储好这个连接的状态。
    - 状态中至少得包括，状态机状态（请参见[TCP状态机](#)英文版，不想看可自行百度该关键词），当前seq号，当前ack号，等等。你还可以搞发送缓存和接收缓存（比如，要是有时重传，可能就不太好不搞发送缓存？倒也不一定，你可以用闭包、消息传递之类的技术，关键看你怎么实现）
  - 每个函数，无论是 `api` 还是 `outgoing`，第一个参数全都是连接的标识符。这个标识符可以用来唯一地确定一个连接。
    - 进一步的小提示：哈希表hashmap是极适合于存储状态量的数据结构，但 `ConnectionIdentifier` 这个类型没法直接哈希，甚至有些语言的SDK中它们没有直接提供比较两个 `ConnectionIdentifier` 是否相等的办法。所以你要想一想，怎么让它变成能哈希的。思路至少有两个，一是把它转换成字符串之类可直接哈希的类型，二是在 `api` 文件中做些修改、实现些该实现的方法。
- wireshark是一个非常强大好用的、带有图形界面的网络分析器。通过wireshark进行抓包，可以极大地辅助完成这个实验。
  - 根据上面的环境配置部分的命令，wireshark已为你装好，命令行运行或开始菜单运行都可以。
  - 可供参考：[Wireshark零基础使用教程（超详细）](#)
  - 思考你抓包的网卡：比如当你想要访问百度时，抓enpxxx那张网卡（即虚拟机上唯一一张能离开虚拟机的网卡）会是比较好的选择。
  - 关掉我们的driver发一个请求，保存下来抓包记录不时回看，**以现实中系统TCP协议栈的发送内容作为你发送的内容的参考**，是一个很有效、比硬着头皮想可能更容易的思路。

- 关于WireShark的使用提示（是助教本人的经验教训）
  - WireShark默认是不开启TCP检验和校验的，这会导致你无法发现自己算错的检验和。
  - 开启方法：编辑——首选项——Protocols——TCP——Validate the TCP checksum if possible
  - 开启之后，你会发现很多包的TCP checksum都是错的，不用惊慌，这是由于TCP checksum offload的存在（感兴趣的同学可自行查阅了解）。你只需关注连接2的、**由你亲手发出的TCP包的检验和是否正确**就可以了，其他的TCP检验和错误（包括连接2中你收到的包的检验和错误）都完全不需要在意。
- 关于如何写代码：
  - 你可以选择在虚拟机内写代码，这种情况下请自行在虚拟机里装IDE；
  - 或者，用宿主机的IDE，然后SSH到虚拟机上面也是可以的。
    - 这种情况下，需要在虚拟机上安装SSH Server：`sudo apt install openssh-server`
    - 查看宿主机的IP：`ip a`
    - 放心，SSH流量永远不会被driver捕获，这是特殊处理了的。感兴趣具体原理，可看 `driver-src/nftables.conf`

## 提交方法、迟交政策和集中答疑

- 提交方法：在网络学堂提交一个压缩包。
  - 首先是代码目录：将你使用的sdk的代码（**注意，是src外层、以编程语言命名的目录，而不是src目录**）提交上来。
    - 例：C++语言，提交cpp目录——正确，提交src目录或sdk目录——错误
    - 提交前，请删除目录中所有的不该被提交的内容，如cmake-\*、target等编译中间产物文件夹，或node\_modules之类的巨大的且很容易重复下载的依赖文件。如果未能这样做，将影响你“代码质量”部分的分数
      - 如果你提交的src文件夹超过了1MB（C++可增加至2M，因为里面有一个内嵌的1M左右的依赖），大概率就是没满足这个要求。
  - 如果你是编译型语言（C++、Rust、Go、Java、Kotlin），请编译出可执行文件，放在bin目录中。（Java、Kotlin编译到class就可以）
  - 文档：要求PDF格式命名为doc.pdf。文档要求如下：
    - 文字内容**不能超过2页A4纸**（不含图片，和最后的“对本作业的建议”部分）
    - 应该至少包含以下内容：
      - 你实现了哪些功能：建议对照[实验内容及评分标准](#)的内容来写。
        - curl访问百度和大文件上传下载，不必贴截图，因为无论你贴不贴截图助教都会亲自运行你的代码亲自测试
        - 浏览器查看网页，如果你做的，可以贴个截图
        - 没给出明确测试方法的内容（三次握手/四次挥手/超时重传之类的），建议提供一些确实实现成功的证明（比如抓包截图）
      - 你编写代码和完成实验的基本思路
      - 你认为你完成的亮点（如有）
      - 你从本次作业中学到的东西
      - 你对本作业的建议，可以从难度、设计合理性、指导充分性等各个方面（自愿，可写可不写，实事求是即可，放心不会影响分数）（这部分不计入文档页数限制）
- 迟交政策：与其他作业不同，本作业的迟交政策按如下规定：

- DDL为1月15日23:59
- 迟交12小时之内的（即1月16日11:59前），分数折扣率为97%
- 12~24小时的（即1月16日23:59前），分数折扣率为90%
- 24~36小时的（即1月17日11:59前），分数折扣率为80%
- 36~48小时的（即1月17日23:59前），分数折扣率为70%
- **1月17日23:59后，不再接受作业补交！**
- 集中答疑
  - 为帮助大家完成作业，本作业提供三次集中答疑。
    - 第一次：1月10日19:00-21:00
    - 第二次：1月12日10:00-12:00
    - 第三次，1月14日，时间待定
  - 每次答疑时间，都将会把腾讯会议号发送在微信群里作为通知。