

计算机网络-实践课

Computer Networks

王春阳

2025. 10



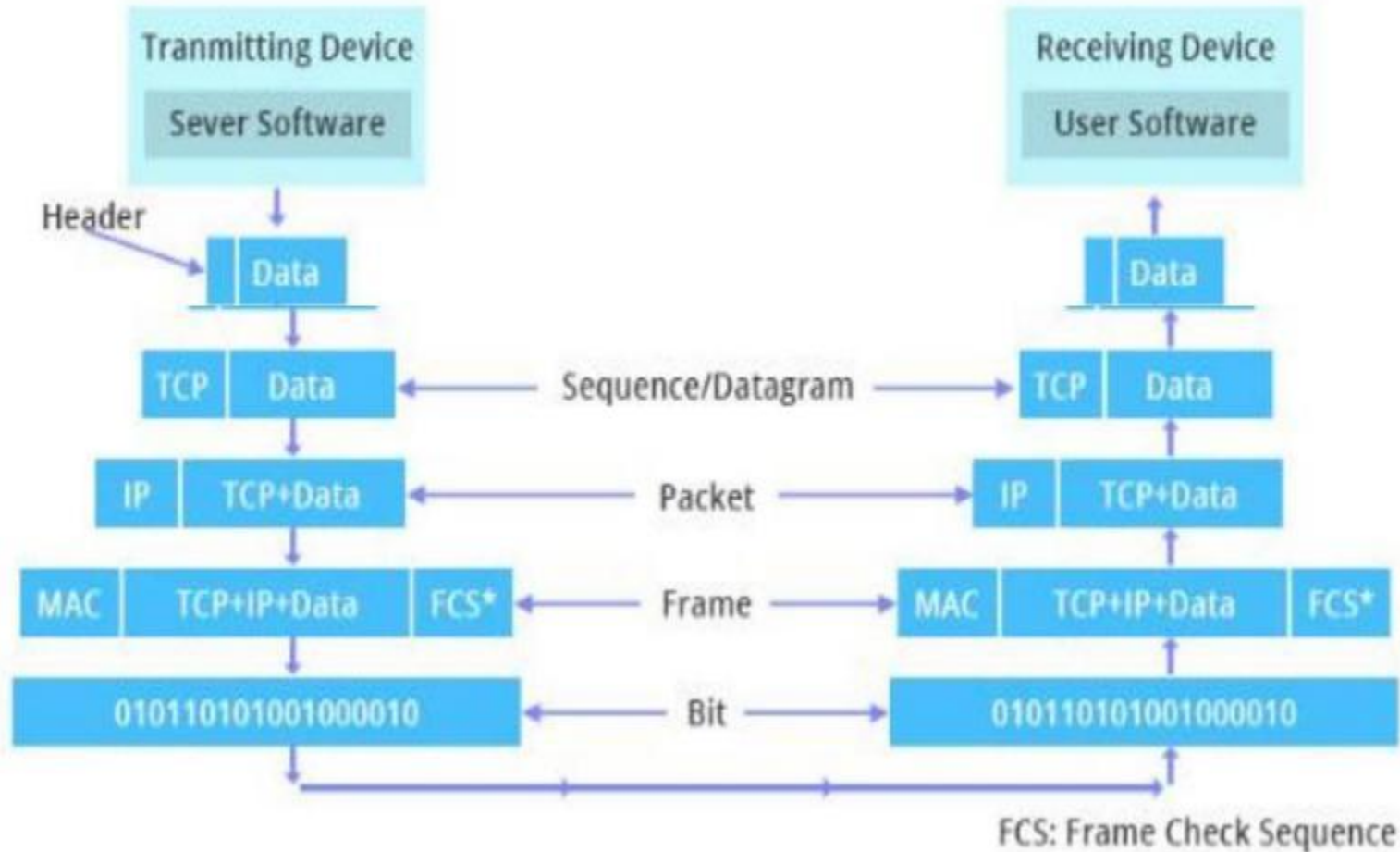
实践课教务

- 主讲老师: 王春阳
- 办公室: 地理馆315
- Email: cywang[@dase.ecnu.edu.cn](mailto:cywang@dase.ecnu.edu.cn)
- 助教TA: 刘蔚美 (51285903074@stu.ecnu.edu.cn)
杨嘉莉 (10234804407@stu.ecnu.edu.cn)

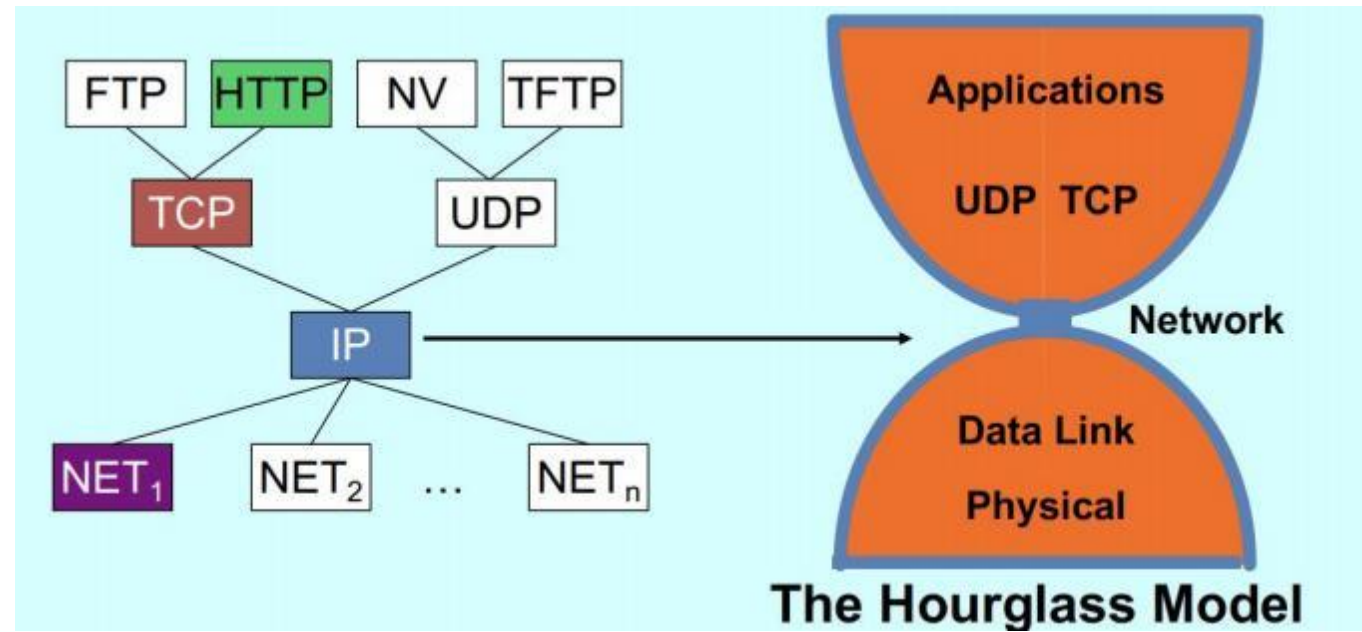
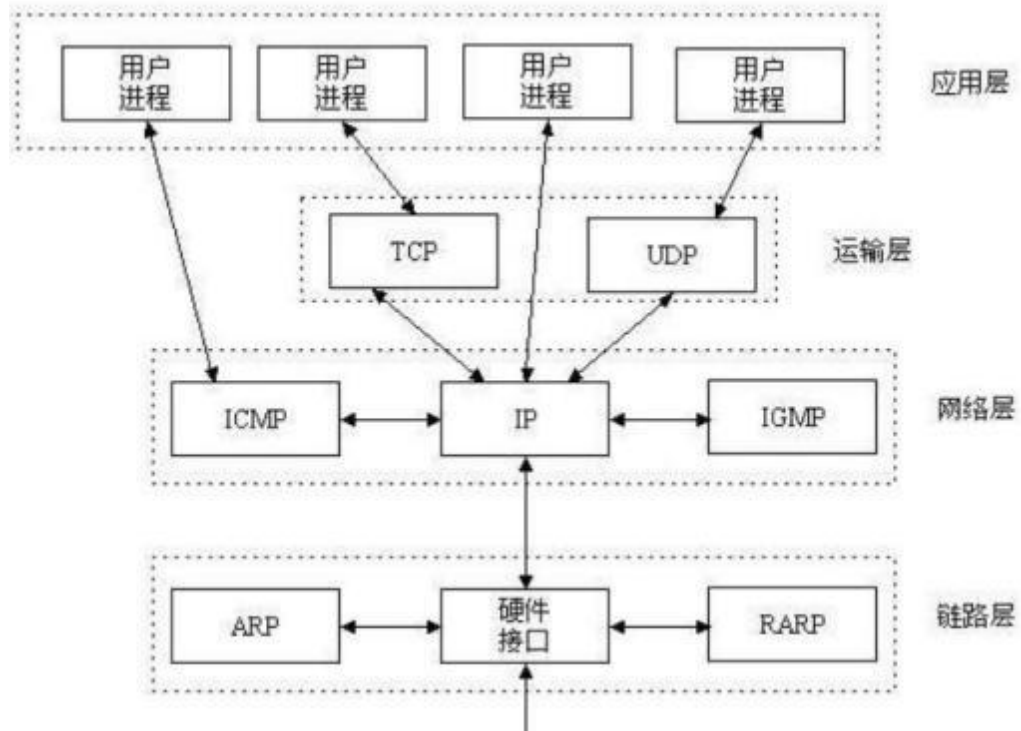
关于本课程实践：

- 实践作业：（4% x 5 = 20%分数）
 - ✓ Assignment 1: Virtual Machine Setup, Sockets.
 - ✓ Assignment 2: TCP Congestion Control and Bufferbloat.
 - ✓ Assignment 3: Passive Network Measurement.
 - ✓ Assignment 4: HTTP Proxy.
 - ✓ Assignment 5: Network Security (Port scan).
- 大作业：（20%分数）
 - Software Defined Networking(SDN) : Mininet

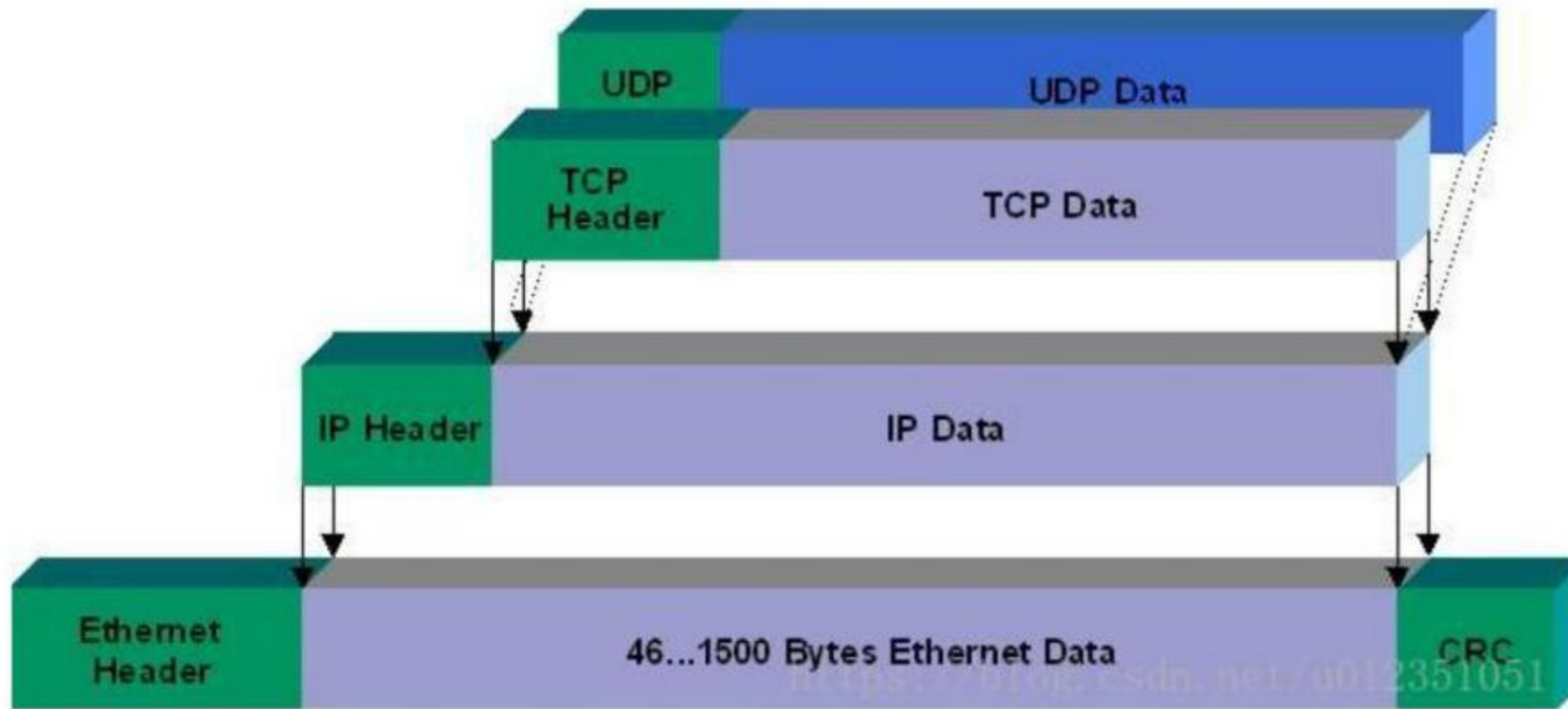
五层网络模型(TCP/UDP IP协议体系)



TCP/IP协议族

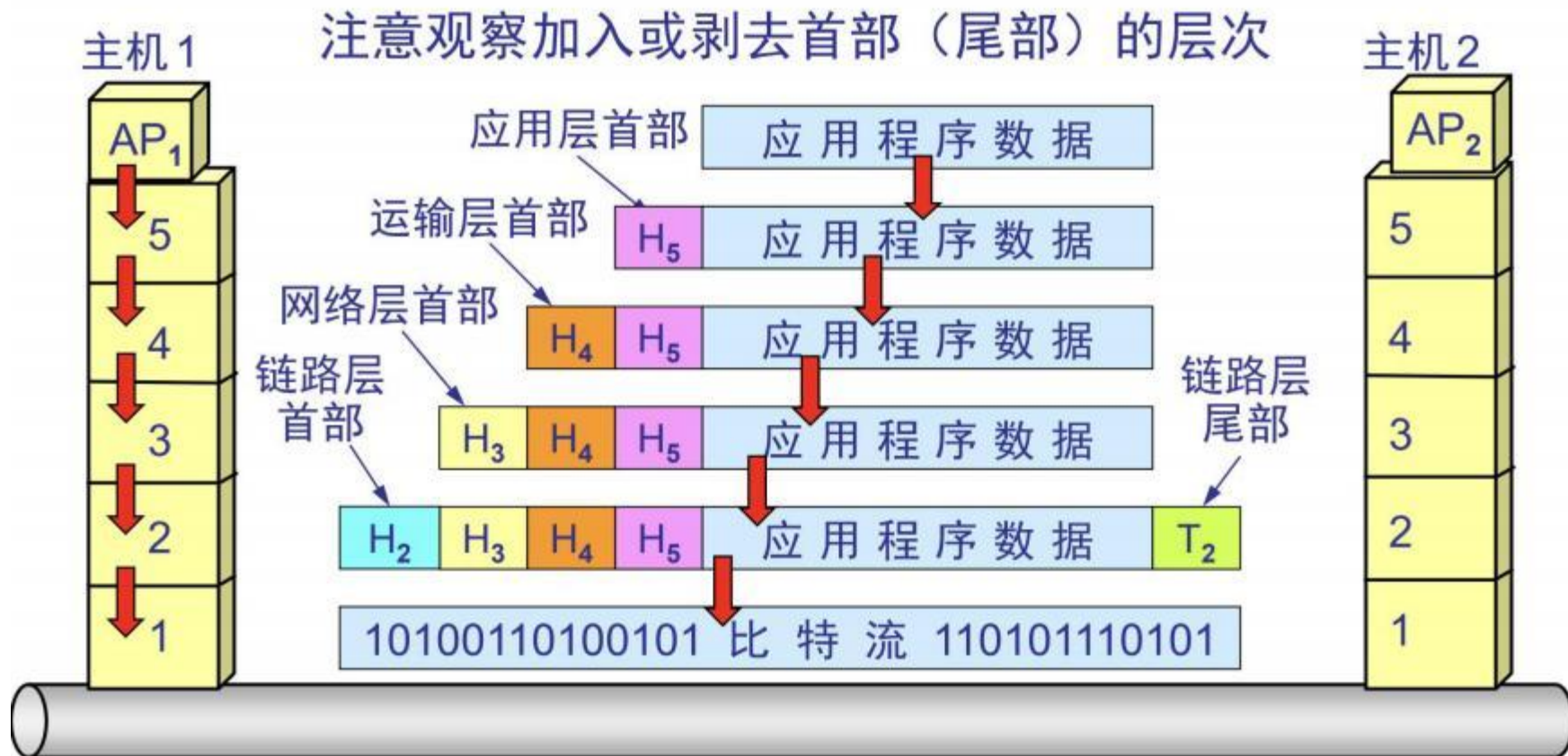


五层网络模型(TCP/UDP IP协议体系)

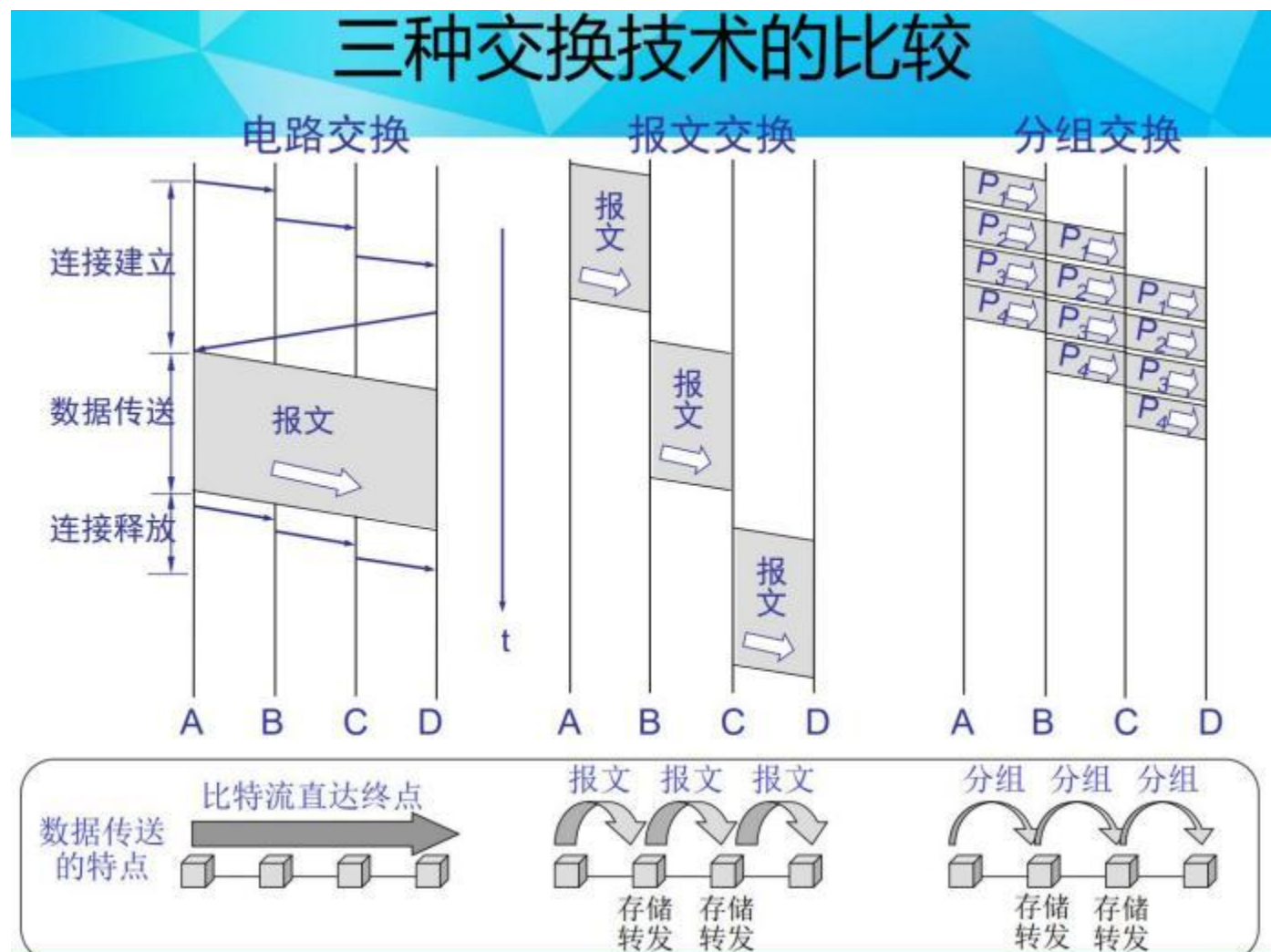


五层网络模型中的数据流实例

主机 1 向主机 2 发送数据



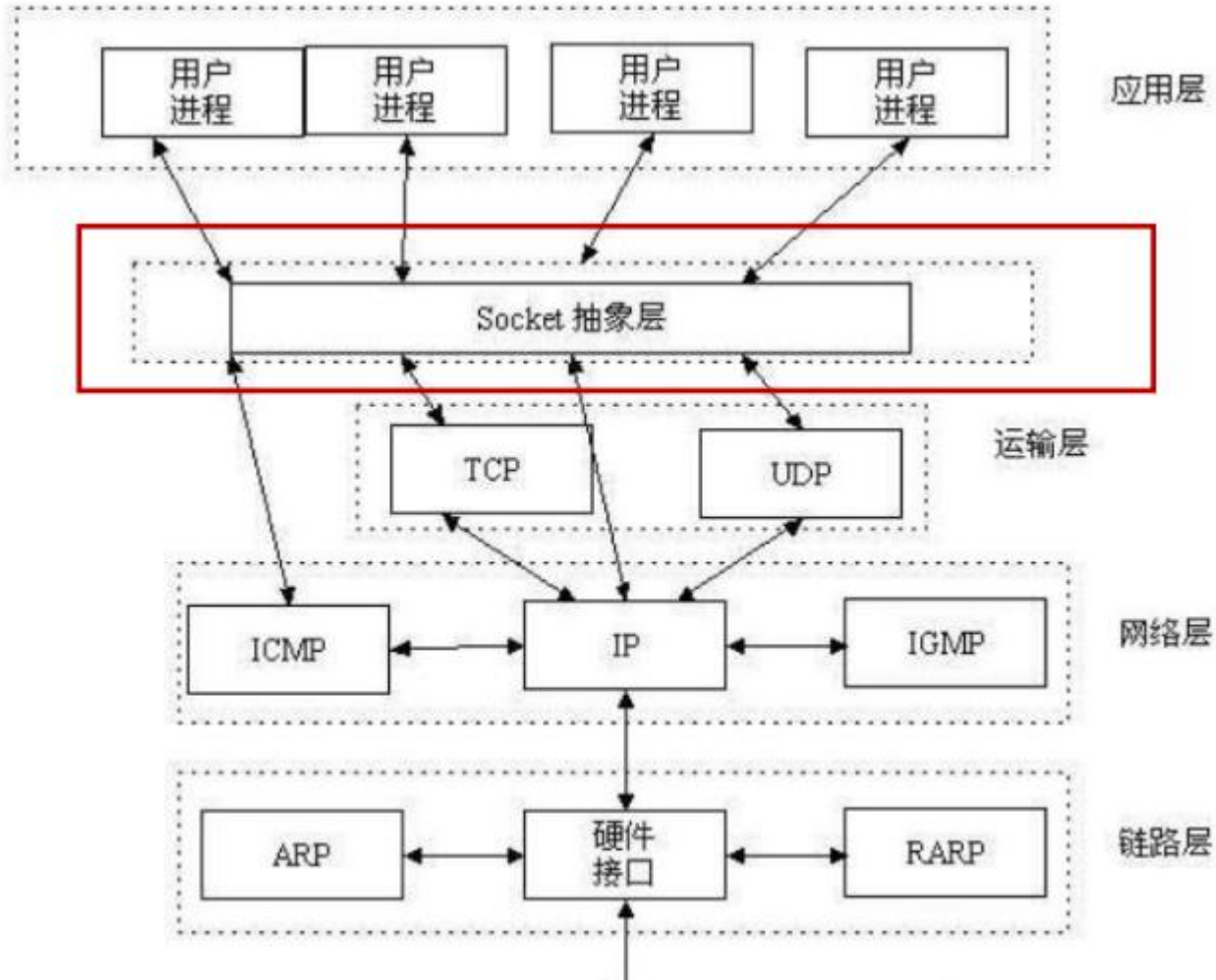
分组交换



Assignment 1.

Virtual Machine Setup and Sockets

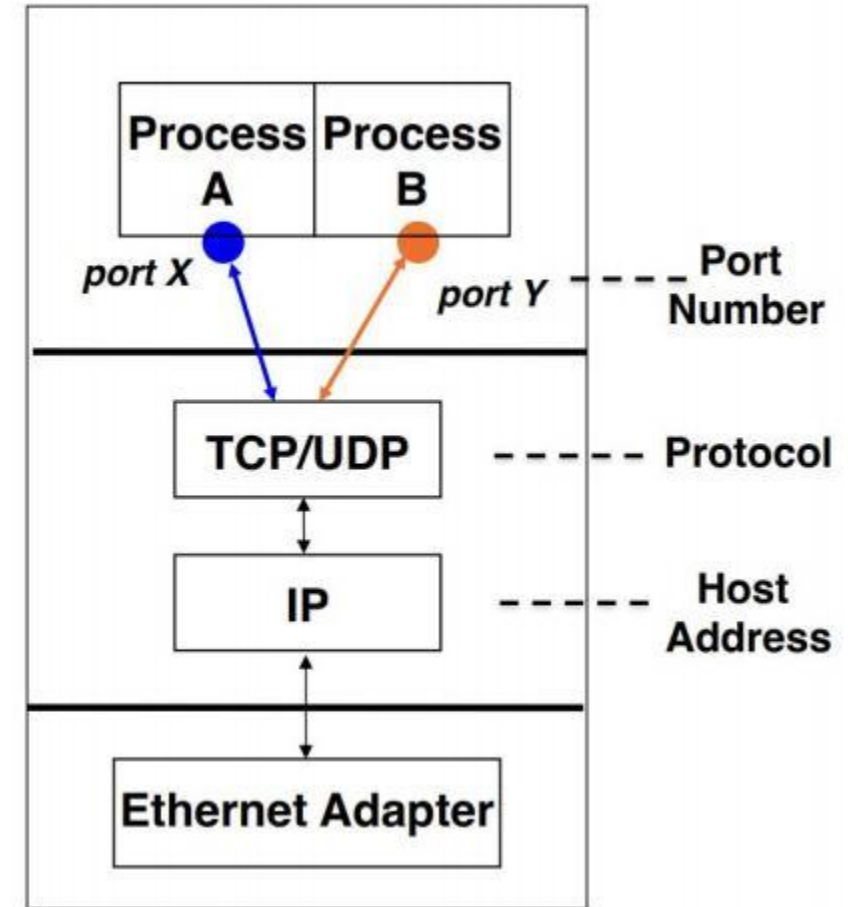
Socket(套接字)



中间软件抽象层

Socket(套接字)

- Receiving host
 - Destination **address** that uniquely identifies host
 - **IP address**: 32-bit quantity (“1.2.3.4”)
- Receiving socket
 - Host may be running many different processes
 - Destination **port** that uniquely identifies socket
 - **Port number**: 16-bits (“80”)

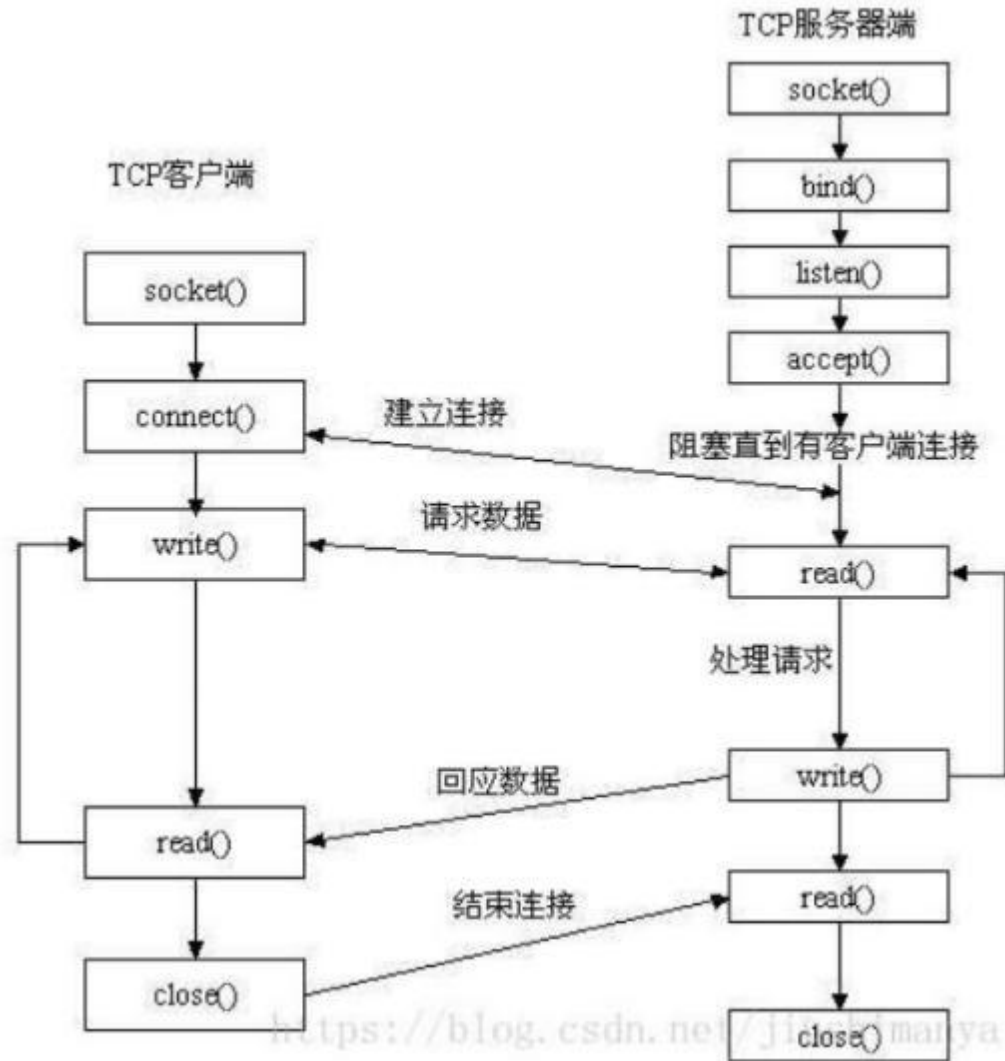


Socket programming

- 应用程序比如浏览器、电子邮件、文件传输服务器等产生的数据，下一步会通过传输层协议进行传输。
- 但应用程序是不会和传输层直接建立联系的，而是有一个能够**连接应用层和传输层之间的套件**，这个套件就是 **Socket**
- 应用程序中有一个 socket 组件，在应用程序启动时，会调用 socket 申请创建Socket，会返回一个**Socket描述符**给应用程序，相当于是**区分不同Socket的号码牌**。根据这个描述符，应用程序在委托协议栈收发数据时就需要提供这个描述符。

Socket 流程

1. 初始化socket
2. 连接服务器
3. 发送数据请求
4. 接收数据
- ... 关闭socket连接



1. 初始化socket
2. socket绑定端口
3. socket监听端口
4. 等待客户端连接
5. 接收数据请求
6. 返回数据
- ... 关闭socket连接

Socket Programming

- Assignment 1 将进行基本的套接字编程。
- 需要编写两对 TCP 客户端和服务端程序，用于在互联网上发送和接收文本信息。其中一对客户端/服务端程序必须用 C 语言编写，另一对可以用 Python 或 Go 语言编写（选择其中一种语言）。

Server 说明

- 每个Server程序都应监听一个套接字，等待Client连接，接收Client发送的消息，将消息打印到标准输出 stdout，然后无限期地等待下一个Client。
- 每个Server都应接受一个命令行参数：监听Client连接的端口号。
- 每个Server都应在无限循环中接受并处理Client通信，允许多个Client向同一Server发送信息。Server只应在响应外部信号（如按下 ctrl-c 产生的 SIGINT）时退出。
- 每个Server应维护一个短的 (5-10个) Client队列，并依次处理多个Client连接尝试。在实际应用中，TCP服务器将派生一个新进程来并发地处理每个客户端连接，但这对于本次任务来说不是必需的。

Client 说明

- 每个client程序都应**与server进行连接**，从 `stdin` 中读取信息，发送信息，然后退出。
- 每个client程序都应按照信息在 `stdin` 中的显示方式读取和发送信息，直至到达 `EOF`（文件结束）
- 每个client程序都应接受两个命令行参数：**服务器的 IP 地址和服务器的端口号**。
- 每个client都应能处理任意大的信息，方法是反复读取和发送信息块，而不是先将整个信息读入内存。
- 每个client都应能处理**部分发送**（当套接字只发送上次发送调用中给出的部分数据时）。方法是**尝试重新发送其余数据**，直到全部发送完毕。

测试 说明

- 将尝试在所有4种客户端和服务端（ C/Python client to C/Python server, etc.） 之间发送几条不同的消息
- 1. 短消息 “Go Tigers! \n”;
- 2. 随机生成的长的字母数字消息;
- 3. 随机生成的长的二进制消息;
- 4. 从不同的客户端顺序地发送到一个服务器的几条短消息;
- 5. 从不同的客户端并发地发送到一个服务器的几个长的、 随机的字母数字消息。

作业验收及提交

- 实践课现场验收（10.23），报告提交（10.26）
- 发送TA邮箱：
 - 点名册1-10：杨嘉莉 (10234804407@stu.ecnu.edu.cn)
 - 点名册11-23：刘蔚美 (51285903074@stu.ecnu.edu.cn)
- 邮件主题：CN-ASS1-学号-姓名
- 邮件附件：CN-ASS1-学号-姓名.zip
- 提交内容：
 - ✓ 全部代码
 - ✓ 实验报告：需要包括关键代码解析、实验结果截图与分析

严禁抄袭!!!