

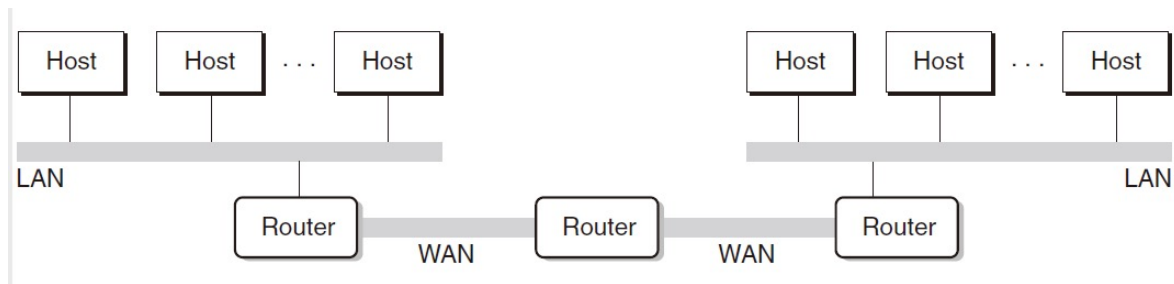
Network programming

金超 章梓立

2021-12-14

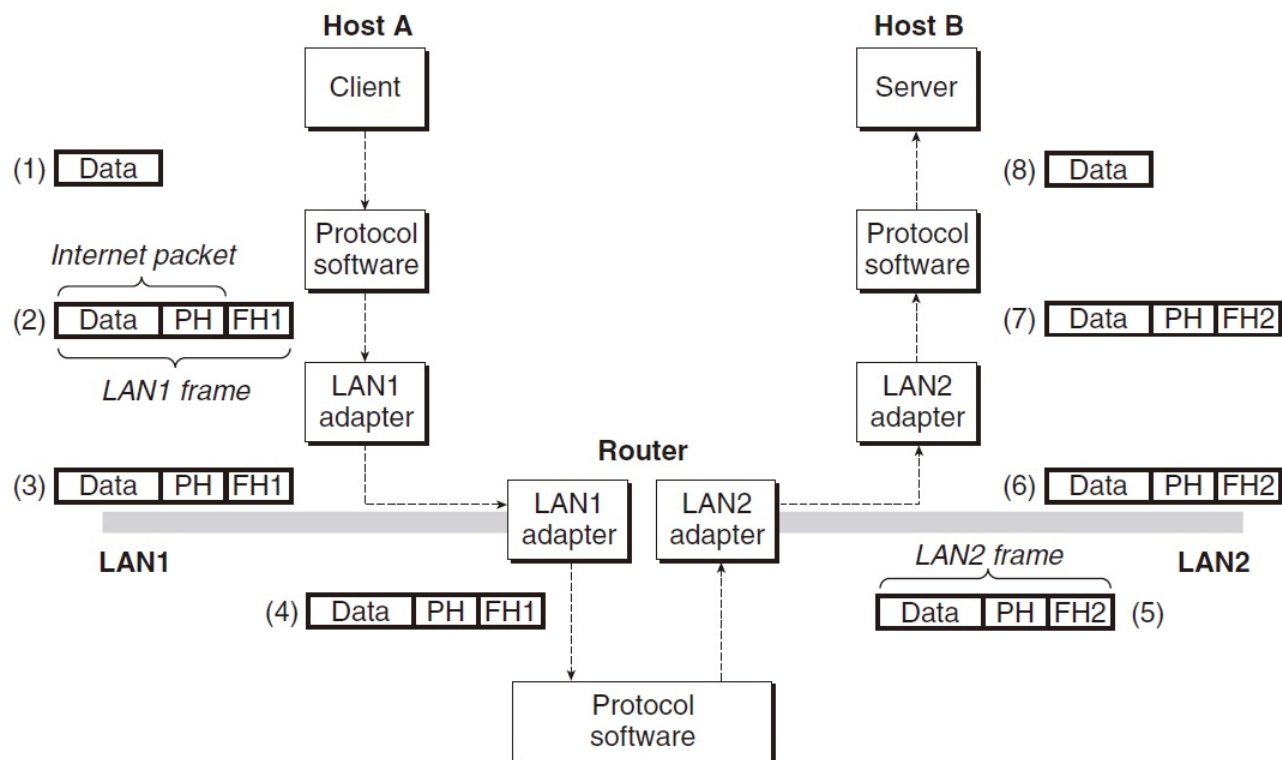
网络编程

- 客户端-服务器编程模型
 - 客户端和服务端是进程
- 网络
 - 网络是一种I/O设备
 - 按照地理远近组织的层次系统
 - 多个不兼容的网络通过路由器链接，组成互连网络
 - 通过协议软件控制主机和路由器协同工作实现数据传输
 - 协议提供命名机制和传输机制



网络编程

- LAN1帧的有效载荷是互相网络包，除此之外还包括LAN1帧头
- 互连网络包的有效载荷是用户数据，除此之外还包括互连网包头
- 书本P645页的8个基本步骤



网络编程

- TCP/IP是一个协议族
- TCP/IP函数运行在内核态
- IP协议提供命名方法和传递机制
- IP机制是不可靠的，不会试图恢复丢失数据报
- UDP是不可靠数据报协议，扩展了IP协议，包可以在进程间传送
- TCP协议是构建在IP之上的复杂协议

- 主机集合被映射为32位IP地址
- IP地址被映射为因特网域名的标识符
- 主机上的进程能够通过连接和其他主机上的进程通信。

- IPv4&IPv6

网络编程

- 网络字节顺序 大端法
- 主机字节顺序可能与网络字节顺序不一致
- UNIX通过函数转换

```
#include <arpa/inet.h>

uint32_t htonl(uint32_t hostlong);
uint16_t htons(uint16_t hostshort);
```

Returns: value in network byte order

```
uint32_t ntohl(uint32_t netlong);
uint16_t ntohs(uint16_t netshort);
```

Returns: value in host byte order

网络编程

- 点分十进制表示IP地址
- 通过下列函数实现转换

```
#include <arpa/inet.h>

int inet_pton(AF_INET, const char *src, void *dst);
    Returns: 1 if OK, 0 if src is invalid dotted decimal, -1 on error

const char *inet_ntop(AF_INET, const void *src, char *dst,
    socklen_t size);
    Returns: pointer to a dotted-decimal string if OK, NULL on error
```

网络编程

- 因特网域名
 - 层次结构
 - 第一层域名有 com、edu、gov、org、net
- DNS维护域名集合和IP地址的映射
 - DNS数据库由主机条目结构组成
 - 每个条目定义了一组域名与一组IP间的映射
 - Localhost映射为127.0.0.1
 - Hostname命令查看本机域名
 - 多个域名可以映射为同一个IP
 - 一个域名可以映射到同一组的多个IP

网络编程

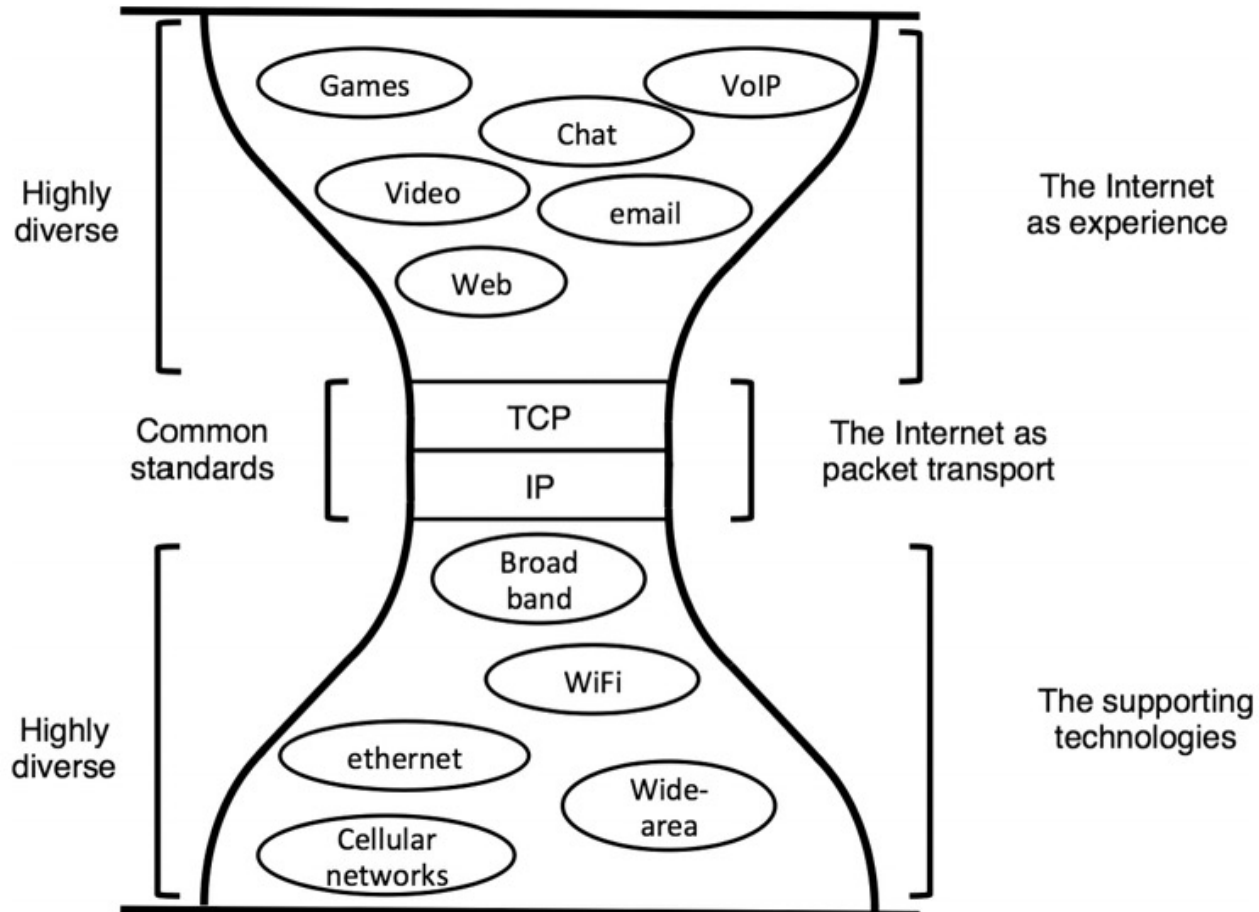
- 因特网(TCP/IP)连接
 - 点对点的、全双工的、可靠的（TCP协议是可靠的）
- 套接字
 - 一个套接字是连接的一个端点，有对应套接字地址
 - 套接字地址由地址：整数端口组成
 - 客户端的端口由内核临时分配，临时端口
 - Web服务器通常使用端口80，服务名http
 - 电子邮件服务器通常使用端口25，服务名smtp
 - /etc/services知名服务名和知名端口的映射
- 连接由两端的套接字地址唯一确定
 - (cliaddr:cliport,servaddr:servport)

Keywords

- 互联网协议(protocol): TCP/IP
 - 进程到进程 vs. 主机到主机
 - 数据包(Datagram) vs. 数据流(Data Stream)
 - 可靠吗? TCP vs. UDP
 - 实现在什么层次? (用户、内核、硬件)
- IPv4、IPv6、IP地址、域名、DNS
 - 127.0.0.1、localhost
 - 10.0.0.0/8、172.16.0.0/12、192.168.0.0/16
- 点到点、全双工、可靠
- 端口、知名端口
- 套接字
 - 套接字编程的整个过程?
- 网络服务
 - telnet、FTP、HTTP
- HTTP
 - URL、HTML
 - 静态内容、动态内容
 - HTTP请求
- Proxy (参考ProxyLab)

TCP/IP协议族

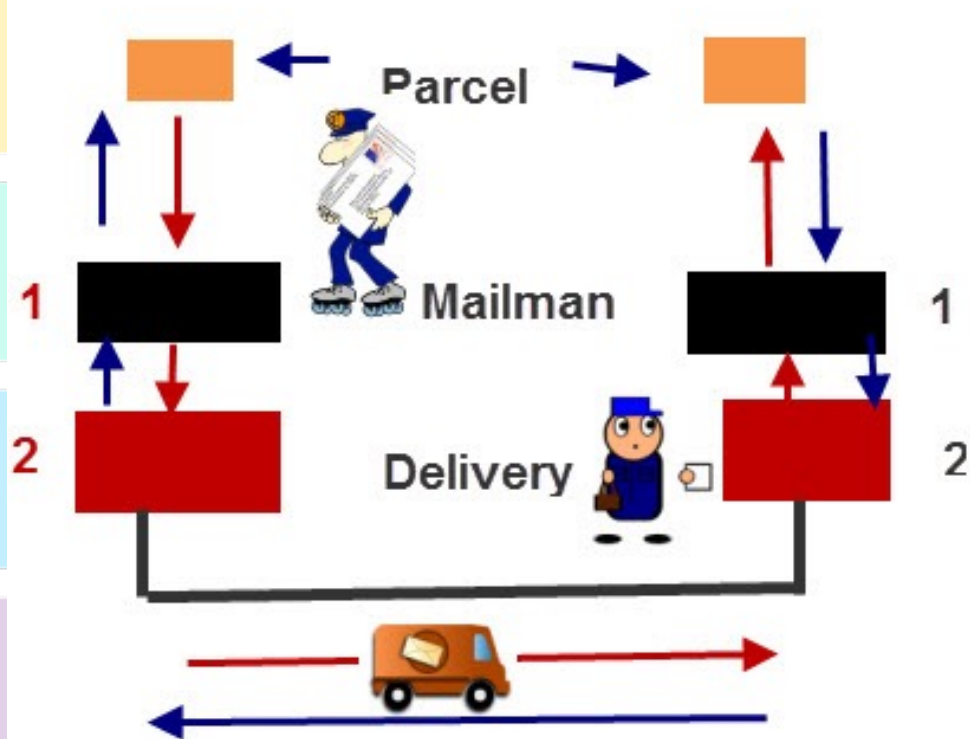
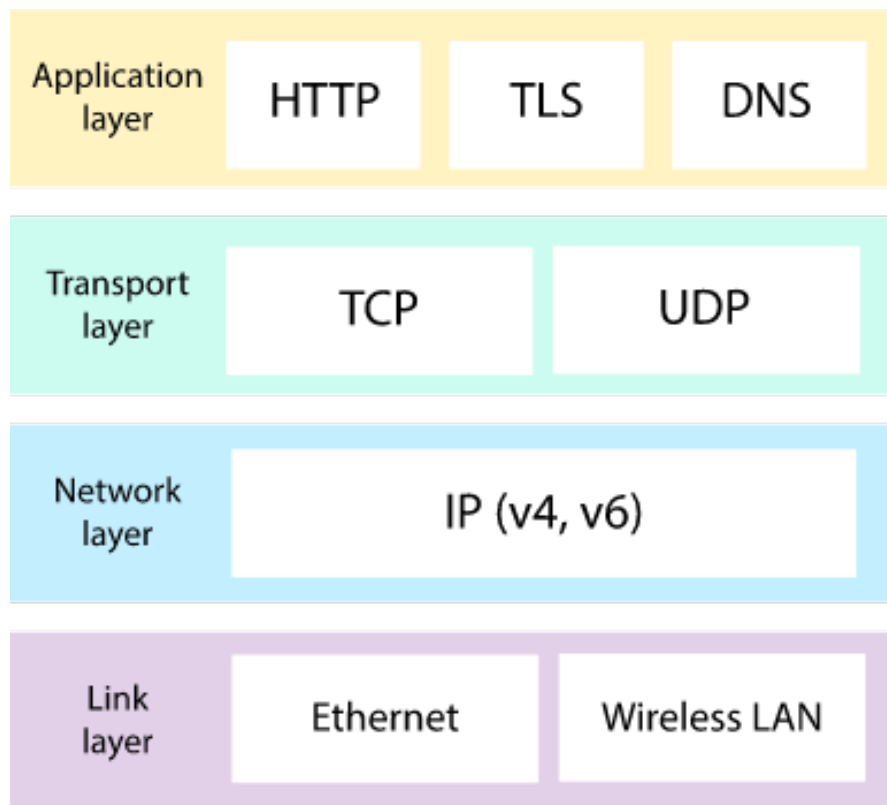
- The “narrow waist” of the Internet.



TCP/IP协议族

- IP协议提供命名方法和传递机制
 - IP机制是不可靠的，不会试图恢复丢失数据报
 - UDP是不可靠数据报协议，扩展了IP协议，包可以在进程间传送
 - TCP协议是构建在IP之上的复杂协议，提供了可靠传输机制
-
- IP 协议有 IPv4, IPv6 两个版本
 - IP 用于标记主机的 adapter (即网卡, network interface card), 而非主机

协议栈



Layered System Example

网络编程

- 网络字节顺序 大端法
- 主机字节顺序可能与网络字节顺序不一致
- UNIX通过函数转换

```
#include <arpa/inet.h>
```

```
uint32_t htonl(uint32_t hostlong);
```

```
uint16_t htons(uint16_t hostshort);
```

Returns: value in network byte order

```
uint32_t ntohl(uint32_t netlong);
```

```
uint16_t ntohs(uint16_t netshort);
```

Returns: value in host byte order

网络编程

- 点分十进制表示IP地址
- 通过下列函数实现转换

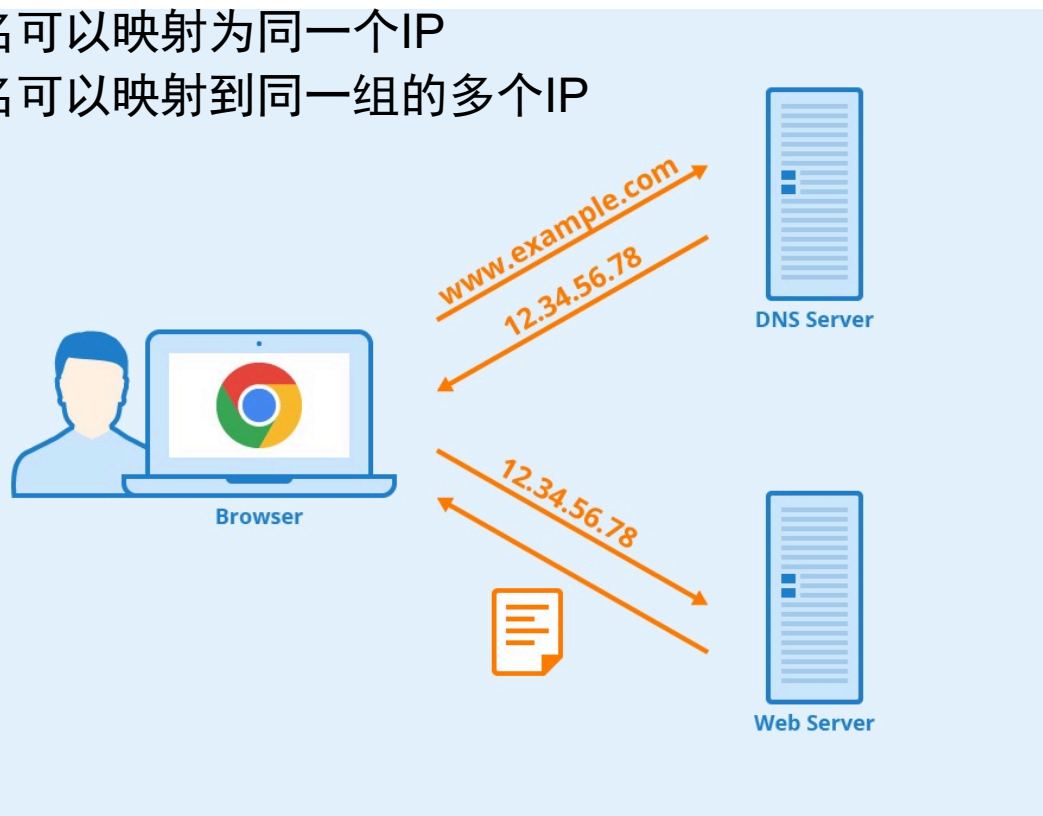
```
#include <arpa/inet.h>

int inet_pton(AF_INET, const char *src, void *dst);
    Returns: 1 if OK, 0 if src is invalid dotted decimal, -1 on error

const char *inet_ntop(AF_INET, const void *src, char *dst,
    socklen_t size);
    Returns: pointer to a dotted-decimal string if OK, NULL on error
```

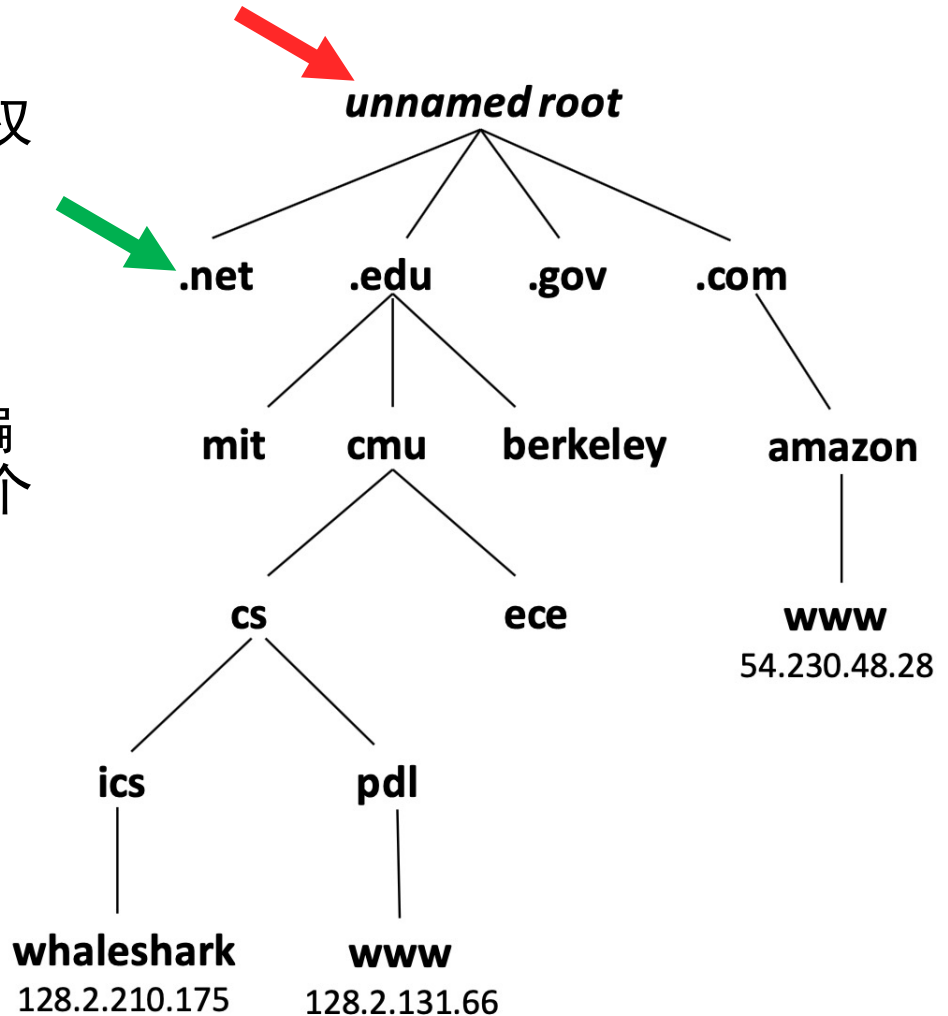
因特网域名

- 产生原因：方便人们记忆 IP 地址
- 域名到 IP 地址的映射关系利用分散在世界各地的数据库 (Domain Name System) 进行维护 特别的，localhost 映射为127.0.0.1
- 域名和IP地址并非严格的一一对应关系
 - 多个域名可以映射为同一个IP
 - 一个域名可以映射到同一组的多个IP



DNS

- **根服务器**：返回**顶级域名**的权威域名服务器的地址
- 根服务器标号：A~M (13个)
- 借由任播 (Anycast) 技术，编号相同的根服务器使用同一个 IP



DNS

- 截止 2020.12.22, 全球一共有 1367 个根服务器



<http://www.root-servers.org/>

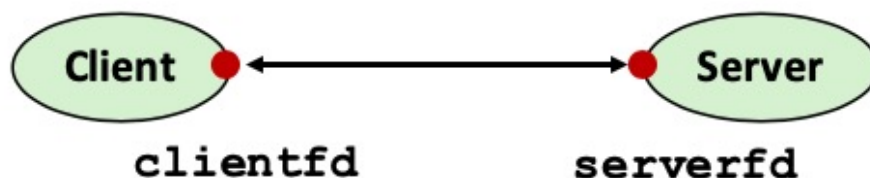
As of 12/22/2020 5:02 p.m., the root server system consists of 1367 instances operated by the 12 independent root server operators.

端口 Port

- 表示服务的类型，由 `/etc/services` 文件维护
- well-known ports:
 - 7 Echo
 - 22 SSH Remote Login Protocol
 - 25 Email Server (smtp)
 - 80 Web Server
 - 443 HTTPS

套接字(socket)编程

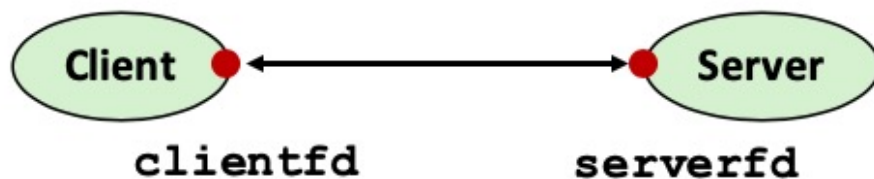
- Socket 的原意是“插座”，用以比喻计算机之间进行通信的方式
- 通常以 (hostname, port) 比如 (10.2.100.1:8080) 的形式表示“插座”的接头



- 两种主要类型：
 - 流格式套接字(Stream Sockets) 也叫“面向连接的套接字”，在代码中使用 SOCK_STREAM 表示，是一种可靠的、双向的通信数据流。
 - 数据报格式套接字(Datagram Sockets) 也叫“无连接的套接字”，在代码中使用 SOCK_DGRAM 表示。只管传输数据，不作数据校验，如果数据在传输中损坏，或者没有到达另一台计算机，是没有办法补救的。
- 课程重点：流格式套接字

连接的四元组

- 通常以 (hostname, port) 比如 (10.2.100.1:8080) 的形式表示“插座”的接头
- 用两个接头表示一个连接，比如 (10.2.100.1:8080, 10.2.100.222:22)，这种组合通常叫做四元组 (four tuple)



端口号的组合来表示。假设一个访问网页服务器的应用，客户端 IP 地址为 128.2.194.242，目标服务器端 IP 地址为 208.216.181.15，用户设置的代理服务器 IP 地址为 155.232.108.39。目标服务器同时提供网页服务（默认端口 80），和邮件服务（默认端口 25）。当客户端向目标服务器发送访问网页的请求时，下面 connection socket pairs 正确的一组是？答：（ ）

| | 客户端请求 | 代理请求 |
|---|---|--|
| A | (128.2.194.242:25, 155.232.108.39:80) | (128.2.194.242:51213, 208.216.181.15:80) |
| B | (128.2.194.242:51213, 155.232.108.39:80) | (128.2.194.242:12306, 208.216.181.15:80) |
| C | (128.2.194.242:25, 208.216.181.15:80) | (155.232.108.39:51213, 208.216.181.15:80) |
| D | (128.2.194.242:51213, 208.216.181.15:80) | (155.232.108.39:12306, 208.216.181.15:80) |

答案：D

解释：考察所有三部分内容

（客户端请求）这一段，客户端发送的请求，源地址是客户自己地址 128.2.194.242，端口号是临时端口号；目标地址仍是服务器地址 208.216.181.15:80 默认端口 80

（代理请求）这一段，代理发送的请求，源地址是代理自己地址 155.232.108.39，端口号是临时端口号；目标地址仍是服务器地址 208.216.181.15:80 默认端口 80。

所以，选项 D 正确。

（选项 A/B 中，目标地址不是代理的地址，而是目标服务器的地址。）

（选项 C 中，端口号 25 属于电子邮件应用默认端口号，不作为客户端的临时端口号。）

(1) 一个服务器拥有两个独立的固定 IP 地址，那么它在 web 应用端口 80 上最多可以监听多少个独立的 socket 连接？（2 分）

答案： $2*2^{48}$

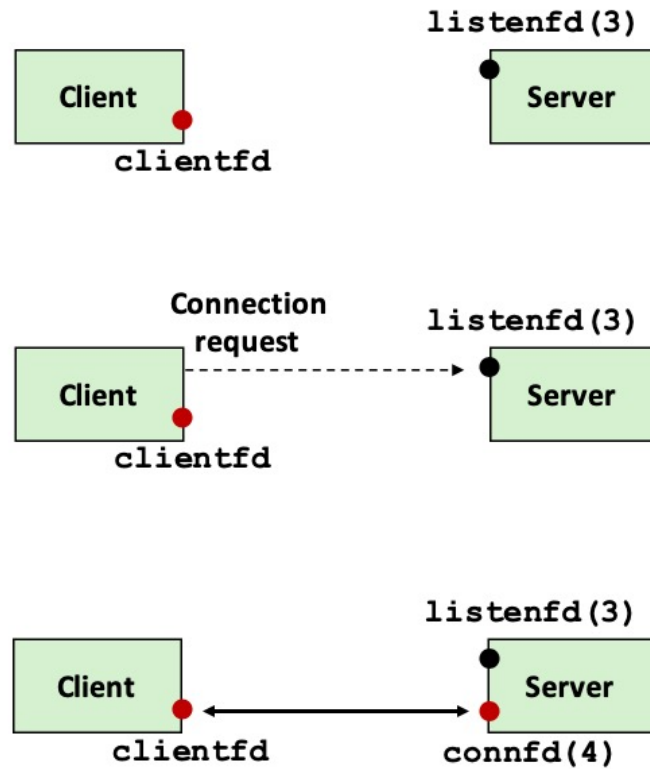
| 服务器端 | 客户端 | 结果 |
|------------|-----------------------------------|---------------|
| 2 个独立固定 IP | 任意 32 位 IP 任意 16 位 port number | $2*2^{32+16}$ |

(2) 该服务器在所有有 web 应用端口上最多可以监听多少个独立的 socket 连接？（2 分）

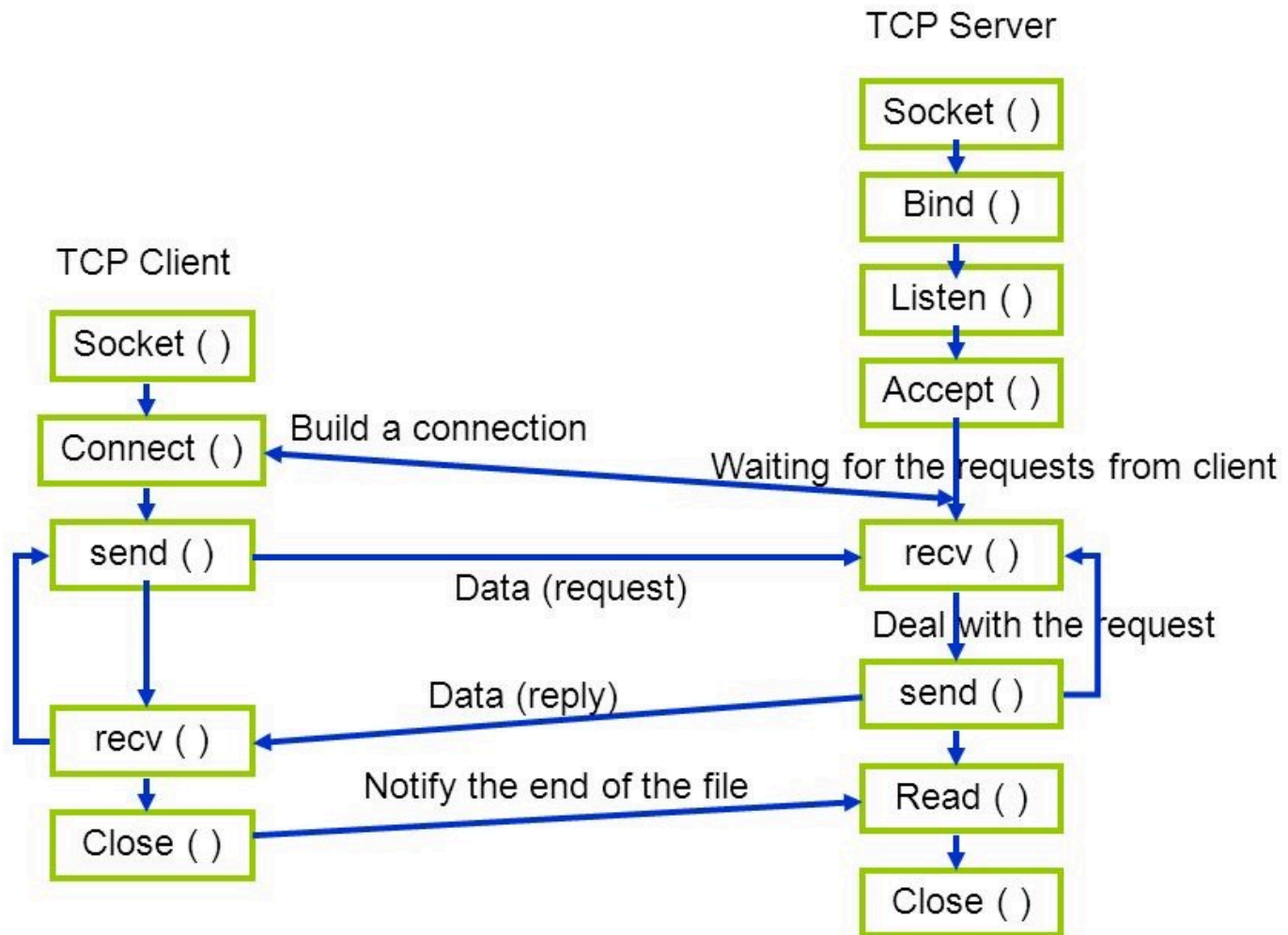
答案： $2*2^{64}$

| 服务器端 | 客户端 | 结果 |
|-----------------------------------|-----------------------------------|------------------|
| 2 个独立固定 IP 任意 16 位 port number | 任意 32 位 IP 任意 16 位 port number | $2*2^{16+32+16}$ |

connect/accept 模型



TCP Socket Interface



Web Components

- Infrastructure:
 - Clients
 - Servers
 - Proxies
- Implementation
 - HTML: formatting content
 - URL: naming content
 - HTTP: protocol for exchanging content.
Any content not just HTML!

HTML (Hypertext Markup Language)

```
1 <!DOCTYPE html>
2 <!--STATUS OK-->
3 <html>
4
5
6
7 <head>
8 <meta charset="UTF-8">
9 <meta http-equiv="X-UA-Compatible" content="IE=Edge" />
10 <meta name="referrer" content="always" />
11 <meta name="description" content="HTML称为超文本标记语言，是一种标识性的语言。它包括一系列标签。通过这些标签可
12 <title>HTML_百度百科</title>
13 <link rel="shortcut icon" href="/favicon.ico" type="image/x-icon" />
14 <link rel="icon" sizes="any" mask href="//www.baidu.com/img/baidu.svg">
15
16 <meta name="keywords" content="HTML HTML(Hyper Text Markup Language) HTML超文本标示语言 html语言 HTML
17 <meta name="image" content="https://bkssl.bdimg.com/cms/static/baike.png">
18 <meta name="csrf-token" content="">
19 <meta itemprop="dateUpdate" content="2020-12-10 19:24:04" />
20 <style>
21     . {
22         display: none !important;
23     }
24 </style>
25
```

URL Syntax

protocol : //hostname[:port]/directorypath/resource

protocol http, ftp, https, smtp, rtsp, etc.

hostname DNS name, IP address

port Defaults to protocol's standard port
e.g. http: 80 https: 443

directory path Hierarchical, reflecting file system

resource Identifies the desired resource

Can also extend to program executions:

```
http://us.f413.mail.yahoo.com/ym/ShowLetter?box=%40B%40Bulk&MsgId=2604_1744106_29699_1123_1261_0_28917_3552_1289957100&Search=&Nhead=f&YY=31454&order=down&sort=date&pos=0&view=a&head=b
```

HTTP request and response

```

whaleshark> telnet www.cmu.edu 80
Trying 128.2.42.52...
Connected to WWW-CMU-PROD-VIP.ANDREW.cmu.edu.
Escape character is '^]'.
GET /index.shtml HTTP/1.1
Host: www.cmu.edu

HTTP/1.1 200 OK
Date: Wed, 05 Nov 2014 17:37:26 GMT
Server: Apache/1.3.42 (Unix)
Transfer-Encoding: chunked
Content-Type: text/html; charset=...

1000
<html ..>
...
</html>
0
Connection closed by foreign host.

```

Client: open connection to server
Telnet prints 3 lines to terminal

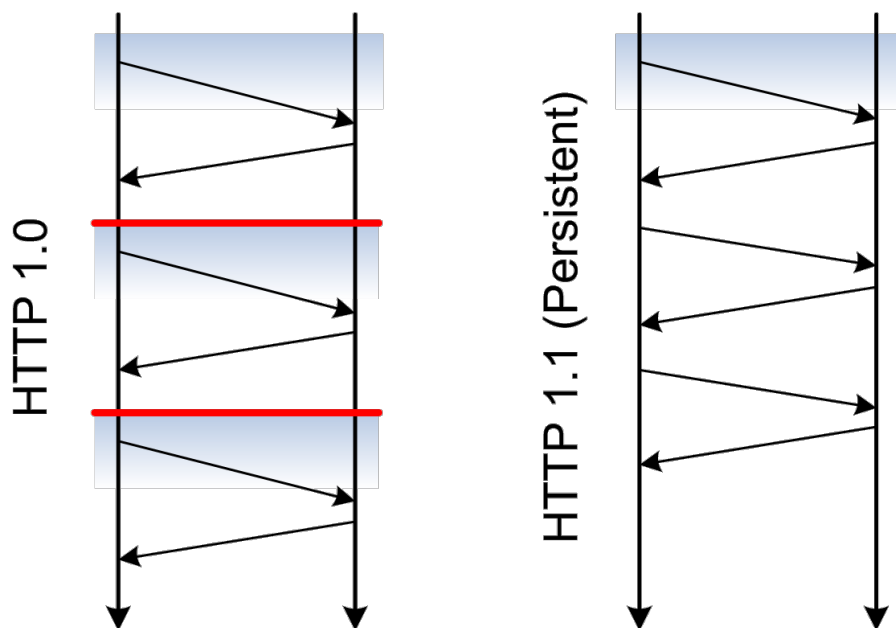
Client: request line
Client: required HTTP/1.1 header
Client: empty line terminates headers
Server: response line
Server: followed by 4 response headers

Server: empty line terminates headers
Server: begin response body
Server: first line of HTML content

Server: end response body
Server: close connection

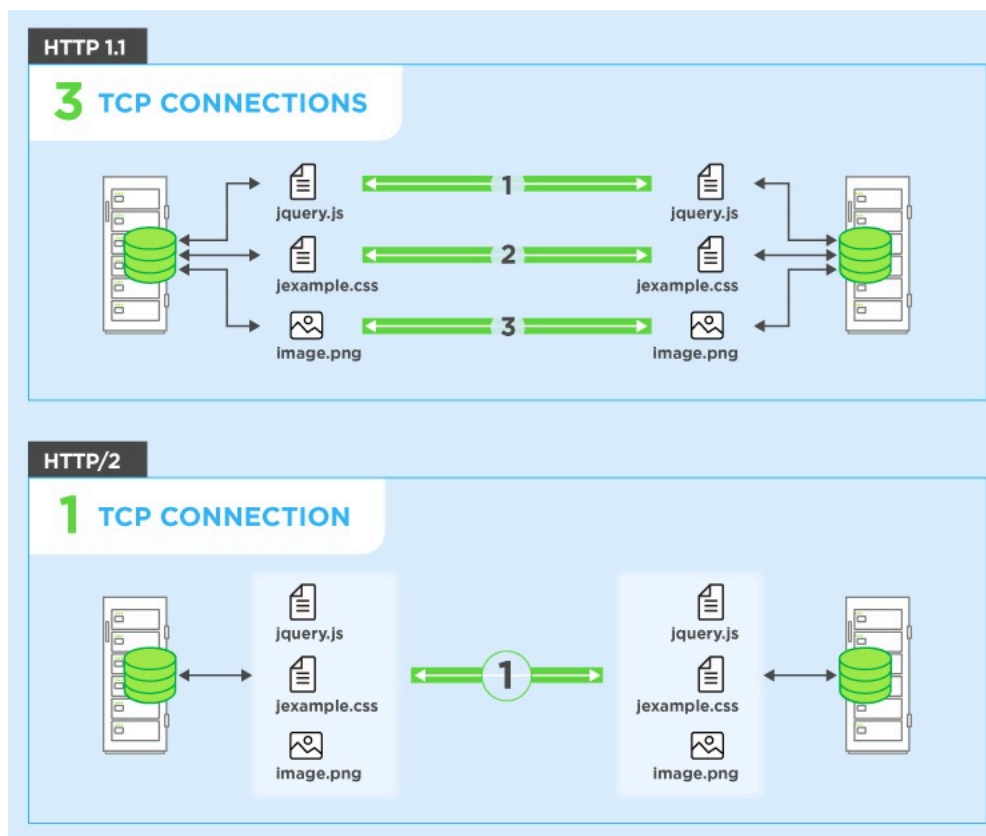
HTTP Evolution

- HTTP/1.0: Open a new connection for each request/response pair. And after each response the connection would be closed.
- HTTP/1.1: We have persistent connections which means that you can have more than one request/response on the same HTTP connection.



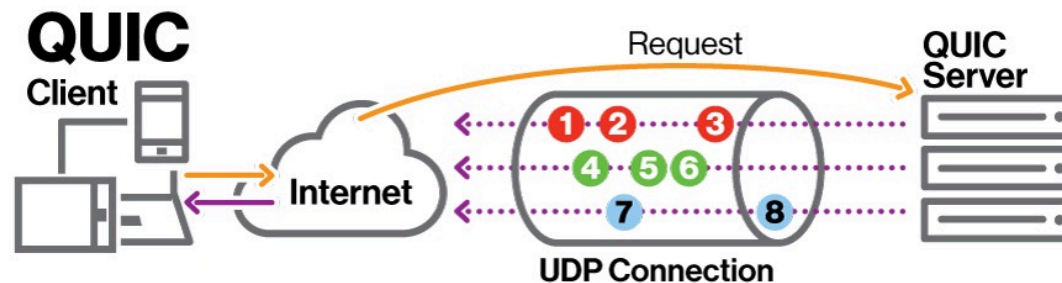
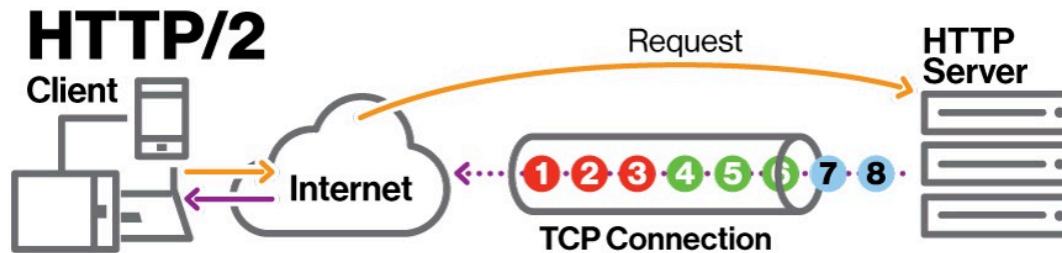
HTTP/2: multiplexing!

- SPDY (speedy) and its moral successor HTTP/2: fire off **multiple requests at once** on the same connection and receive the requests back in any order.



QUIC motivates HTTP/3

- QUIC: a multiplexed stream transport over **UDP**
- similar to TCP+TLS+HTTP/2 implemented on UDP but without head of line blocking



QUIC deployment

XQUIC是阿里巴巴淘系架构团队自研的IETF QUIC标准化协议库实现，在手机淘宝上进行了广泛的应用，并在多个不同类型的业务场景下取得明显的效果提升，为手机淘宝APP的用户带来丝般顺滑的网络体验：

- 在RPC请求场景，网络耗时降低15%
- 在直播高峰期场景，卡顿率降低30%、秒开率提升2%
- 在短视频场景，卡顿率降低20%

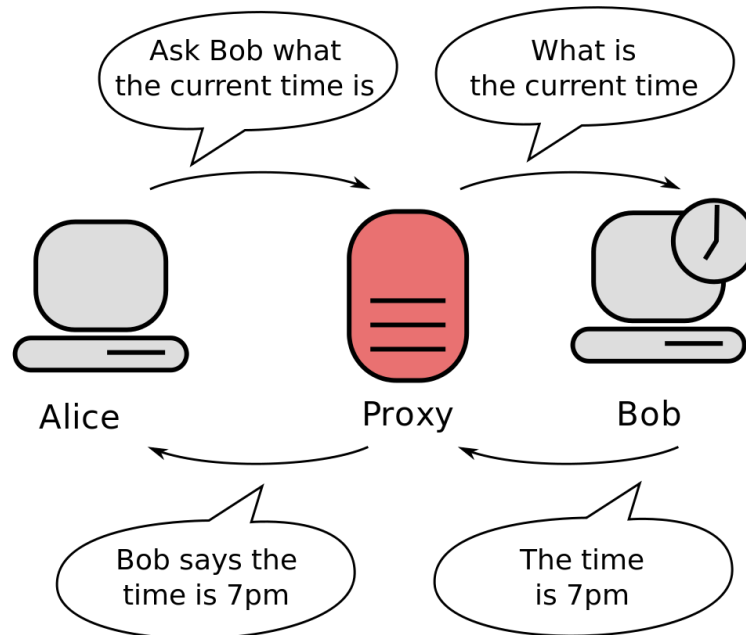
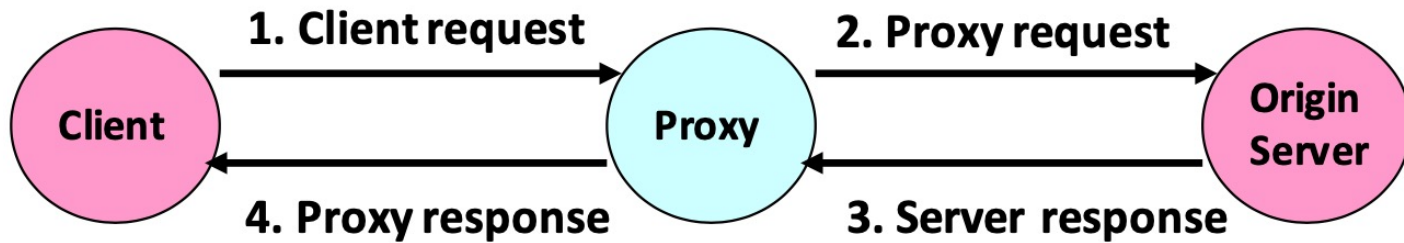
Chrome is deploying HTTP/3 and IETF QUIC

Wednesday, October 7, 2020

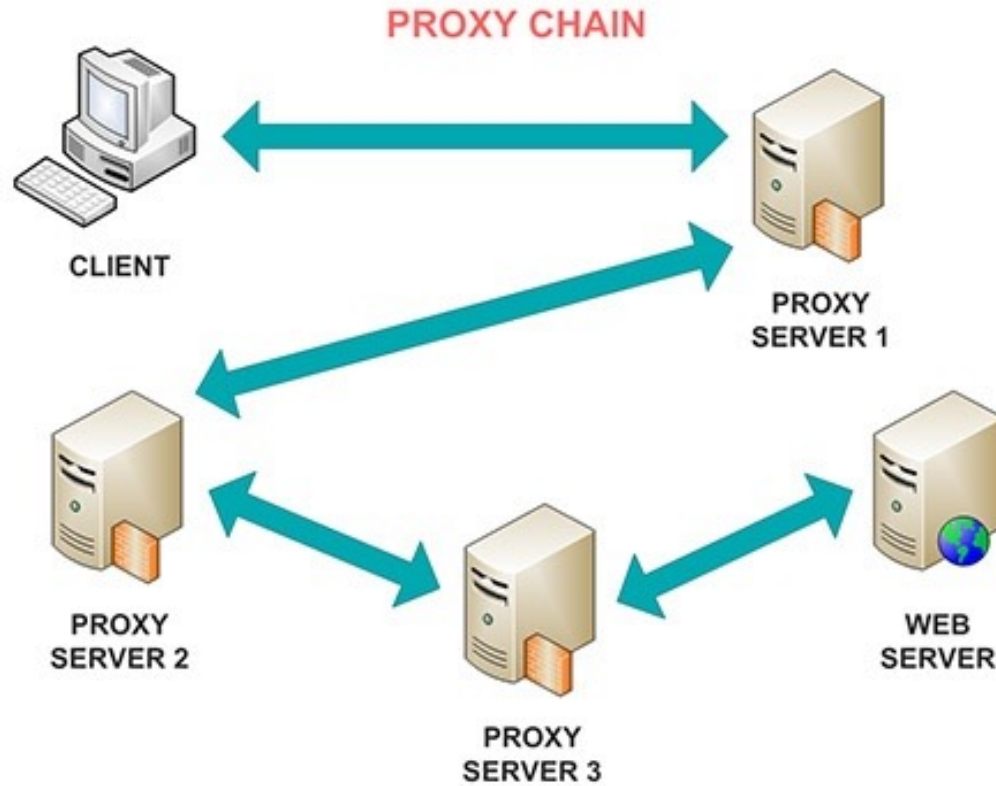
QUIC is a new networking transport protocol that combines the features of TCP, TLS, and more. HTTP/3 is the latest version of HTTP, the protocol that carries the vast majority of Web traffic. HTTP/3 only runs over QUIC.

QUIC was initially developed by Google and [first announced in 2013](#). Since then, the protocol has matured, and is now responsible for carrying over a third of Google traffic.

Proxy



Proxy chain



题目讲解

完成题目 时间 20 min

2. 请根据 web 应用在计算机网络中的定义以及其在协议栈自上而下在软件中的实现, 把以下关键字填入表格

注: 同一个关键词可能被填入多次; 不是每一个关键词都必须被填入

Streams (end to end), Datagrams, web content, IP, TCP, UDP, Kernel code, User code

| 协议 | 数据包类型 | 软件实现 |
|------|-------|------|
| HTTP | | |
| | | |
| | | |

2. 请根据 web 应用在计算机网络中的定义以及其在协议栈自上而下在软件中的实现, 把以下关键字填入表格

注: 同一个关键词可能被填入多次; 不是每一个关键词都必须被填入

Streams (end to end), Datagrams, web content, IP, TCP, UDP, Kernel code, User code

| 协议 | 数据包类型 | 软件实现 |
|------|-------|------|
| HTTP | | |
| | | |
| | | |

答案:

| 协议 | 数据包类型 | 软件实现 |
|------|-------------|-------------|
| HTTP | Web content | User code |
| TCP | Streams | Kernel code |
| IP | Datagrams | Kernel code |

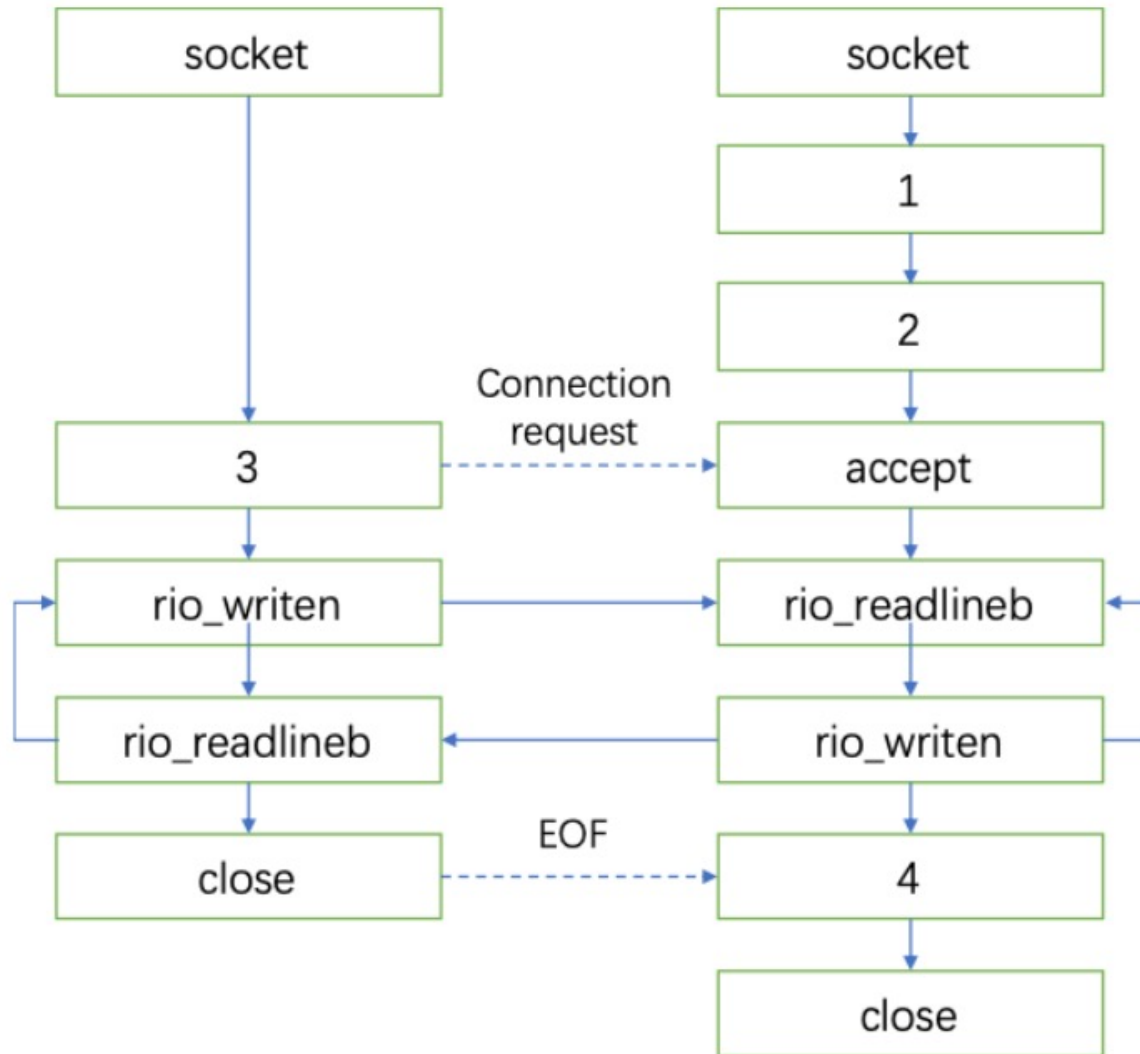
3. 以下关于互联网的说法中哪些是正确的？并简要说明原因

- A. 在 client-server 模型中, server 通常使用监听套接字 `listenfd` 和多个 `client` 同时通信
- B. 在 client-server 模型中, 套接字是一种文件标识符
- C. 准确地说, IP 地址是用于标识主机的 adapter (network interface card), 并非主机
- D. Web 是一种互联网协议
- E. 域名和 IP 地址是一一对应的
- F. Internet 是一种 internet

3. 以下关于互联网的说法中哪些是正确的？并简要说明原因

- A. 在 client-server 模型中, server 通常使用监听套接字 `listenfd` 和多个 `client` 同时通信
- B. 在 client-server 模型中, 套接字是一种文件标识符
- C. 准确地说, IP 地址是用于标识主机的 adapter (network interface card), 并非主机
- D. Web 是一种互联网协议
- E. 域名和 IP 地址是一一对应的
- F. Internet 是一种 internet

答案: B, C, F



```
bind listen connect rio_readlineb
```


2018 期末

- () 套接字接口常常被用来创建网络应用
- () Windows 10 系统没有实现套接字接口
- () `getaddrinfo()` 和 `getnameinfo()` 可以被用于编写独立于特定版本的 IP 协议的程序
- () `socket()` 函数返回的描述符，可以使用标准 Unix I/O 函数进行读写
- () 数字数据只能通过数字信号传输

对错对对错 还包括模拟信号

2015 期末

- () 在 client-server 模型中, server 通常使用监听套接字 `listenfd` 和多个 client 同时通信
- () 在 client-server 模型中, 套接字是一种文件标识符
- () 准确地说, IP 地址是用于标识主机的 adapter (network interface card), 并非主机
- () Web 是一种互联网协议
- () 域名和 IP 地址是一一对应的
- () Internet 是一种 internet

错对对错错对

说明:server 会为每一个 client 单独创建一个套接字进行通信;Web 是一种基于 HTTP 协议的互联网应用;一个域名可以对应多个 IP 地址, 一个 IP 地址也可以对应多个域名

2016 期末/2014 期末

- () HTTP 协议规定服务器端使用 80 端口提供服务
- () 使用 TCP 来实现数据传输一定是可靠的
- () Internet 上的两台的主机要通信必须先建立端到端连接
- () 在 Linux 中只能通过 Socket 接口进行网络编程
- () Web 浏览器与 web 服务器通信采用的协议是 HTML

错对错错错 80 是默认端口而非必须;IP 协议本身不基于连接,传输层中的 UDP 协议便不基于连接;可以通过网卡驱动接口直接收发数据;应该是 HTTP

【Web】使用浏览器打开网页 `www.pku.edu.cn` 的过程中，下列网络协议中，可能会被用到的网络协议有___个

- ① DNS ② TCP ③ IP ④ HTTP

4

假设有一个 HTTPS (基于 HTTP 的一种安全的应用层协议) 客户端程序想要 通过一个 URL 连接一个电子商务网络服务器获取一个文件，并且这个服务器 URL 的 IP 地址是已知的，以下哪种协议是一定不需要的？

- A. HTTP B. TCP C. DNS D. SSL/TLS

C