

# 计算机网络

## 第二章 应用层

---

谢瑞桃

[xie@szu.edu.cn](mailto:xie@szu.edu.cn)

[rtxie.github.io](https://github.com/rtxie)

计算机与软件学院  
深圳大学



# 第二章讲解内容

---

## 1. 应用层概述

- 客户/服务器架构，P2P架构，套接字

## 2. Web 和 HTTP

- 非持续HTTP，持续HTTP，响应时间，请求/响应报文

## 3. Email

- SMTP，POP3，IMAP

## 4. DNS 域名系统

- 层级结构，迭代查询，递归查询



# 网络应用

- Web
  - Email
  - 即时信息
  - 社交网络
  - 网游
  - 流媒体
  - P2P文件共享
  - 远程实时会议
  - 搜索
  - 远程桌面
- 直播
  - .....

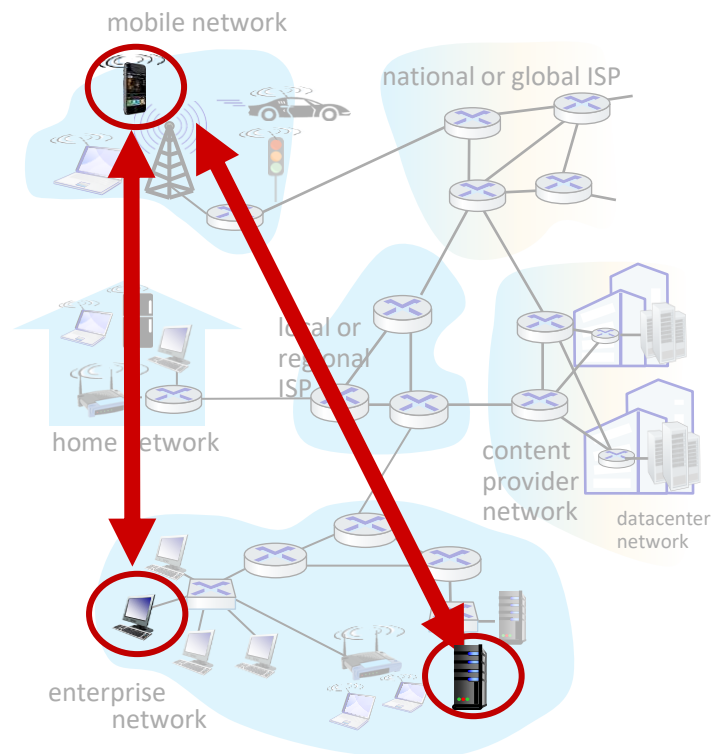
# 客户端/服务器架构

## ■ 服务器

- 一直开机并在线
- IP地址是固定的
- 一般都位于数据中心

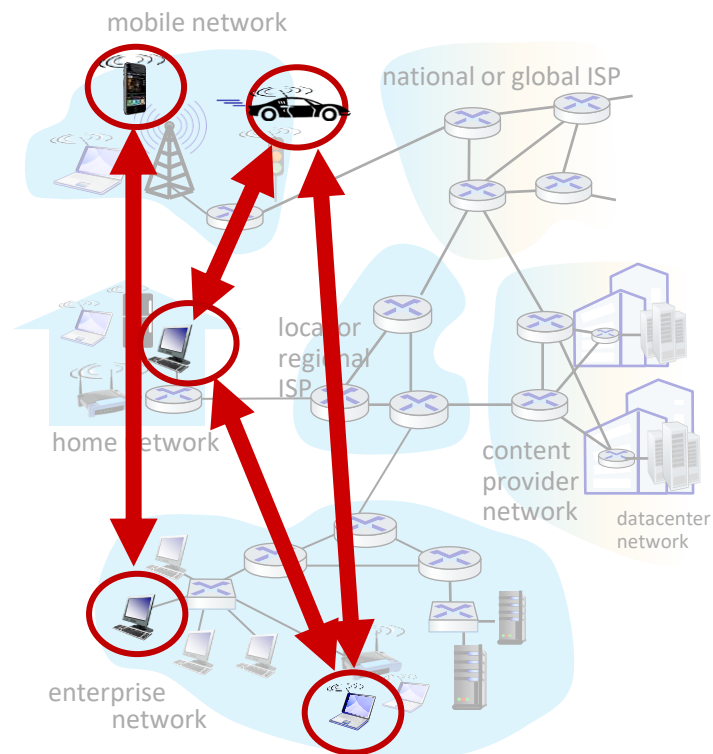
## ■ 客户端

- 发起联系
- 客户端之间不直接通信



# P2P 架构

- 通常没有服务器
- 任意的终端直接通信
- 终端互相提供服务
- 终端的网络连接是断断续续的，IP也是动态变化的，管理非常复杂



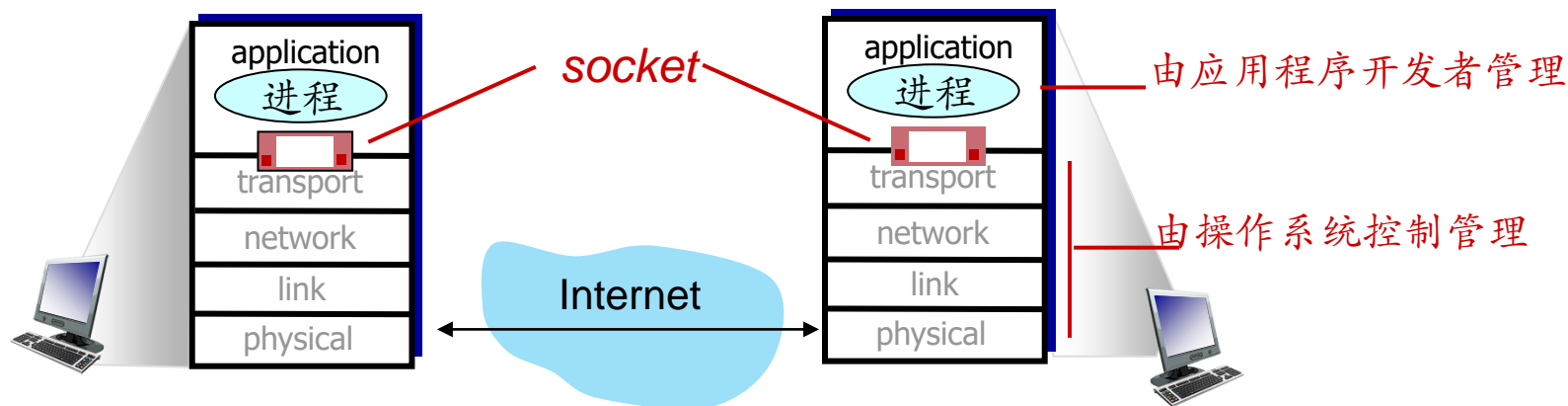


# 进程通信

- 进程：拥有独立内存的一个正在运行的程序
- 主机内的进程通信：进程间通信
- 主机间的进程通信：交换报文
- 为了接收报文，进程必须能够被识别
- 识别符：主机IP地址 + 进程端口号

# Socket 套接字

- 进程向套接字发送报文，从套接字接收报文
- 进程通过套接字使用网络传输服务





# 应用程序对网络服务的需求

应用	数据丢失	带宽	时延敏感
文件传输	不能丢失	弹性	不
e-mail	不能丢失	弹性	不
Web	不能丢失	弹性	不
实时多媒体（网络电话，视频会议）	容忍丢失	音频: 5Kbps-1Mbps 视频: 10Kbps-5Mbps	是，几十毫秒
流媒体	容忍丢失	同上	是，几秒
交互式游戏	容忍丢失	Kbps+ 云游戏几十Mbps	是，几十毫秒
讯息	不能丢失	弹性	是或不是





# 应用程序对网络服务的需求

- 两种传输层协议提供两种网络服务
- UDP
  - 不可靠传输
    - 基本裸奔
- TCP
  - 可靠传输
    - 应对丢包、乱序等问题
  - 流量控制
  - 拥塞控制
    - 两把水龙头，控制流入网络的数据量



# 应用程序对网络服务的需求

应用	应用层协议	传输层协议
文件传输	FTP	TCP
e-mail	SMTP, IMAP, POP3	TCP
Web	HTTP/1.1, HTTP/2, HTTP/3	TCP
网络电话	SIP, RTP	TCP/UDP
流媒体	HTTP, DASH	TCP/UDP
交互式游戏	WOW, FPS (proprietary)	TCP/UDP



## 第二章知识点汇总

---

- 应用层概述
  - 理解两种架构
  - 掌握使用套接字进行网络编程的方法（实验）
  - 理解应用程序对网络服务的不同需求



# 第二章讲解内容

---

## 1. 应用层概述

- 客户/服务器架构，P2P架构，套接字

## 2. Web 和 HTTP

- 非持续HTTP，持续HTTP，响应时间，请求/响应报文

## 3. Email

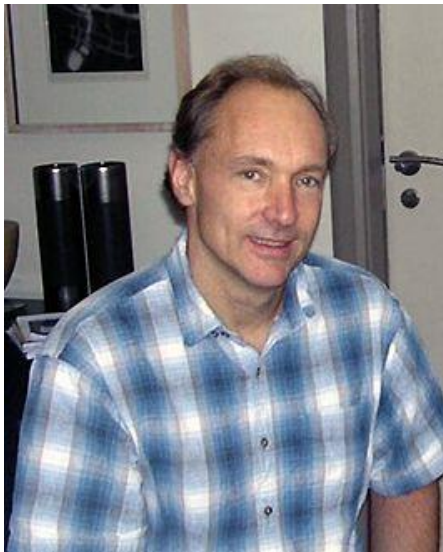
- SMTP，POP3，IMAP

## 4. DNS 域名系统

- 层级结构，迭代查询，递归查询

# World Wide Web

- **Tim Berners-Lee** 于1990年发明了万维网，于2016年获图灵奖。



- Web三件套

- Web浏览器
- Web服务器
- HTTP

- 第一个网页

<http://info.cern.ch/hypertext/WWW/TheProject.html>

# World Wide Web





# World Wide Web

---

- Web页面通常含有一个HTML文件和一些被引用的对象
- HTML文件通过URL引用对象
- 对象
  - HTML文件，图片，JavaScript，CSS文件，视频等
- URL（协议://主机/对象路径）
  - <https://www.szu.edu.cn/images/logo.png>

HTML: Hypertext Markup Language 超文本标记语言

# HTTP

- Hypertext Transfer Protocol
- 客户/服务器模型
- 传输层使用TCP





# 非持续HTTP

- 用户输入URL: `http://www.szu.edu.cn/`



1. 创建TCP套接字。HTTP客户端向服务器`www.szu.edu.cn`的80端口发起TCP连接。

2. 创建TCP套接字。主机`www.szu.edu.cn`的HTTP服务器接受连接，并通知客户端。

3. HTTP客户端向TCP套接字发送包含URL的HTTP请求报文，请求对象/`/`，即缺省主页。

4. HTTP服务器接收请求报文，将所请求的对象封装成响应报文，并发送给TCP套接字。

5. HTTP客户端收到包含HTML文件的响应报文，并显示。

在4结束后，HTTP服务器关闭连接。  
HTML所引用的其他对象用新的TCP连接传输。

time

# 非持续HTTP响应时间

- RTT (Round Trip Time) : 一个非常小的分组从客户端传输到服务器, 再返回来所需要的往返时间

这种方法好吗?

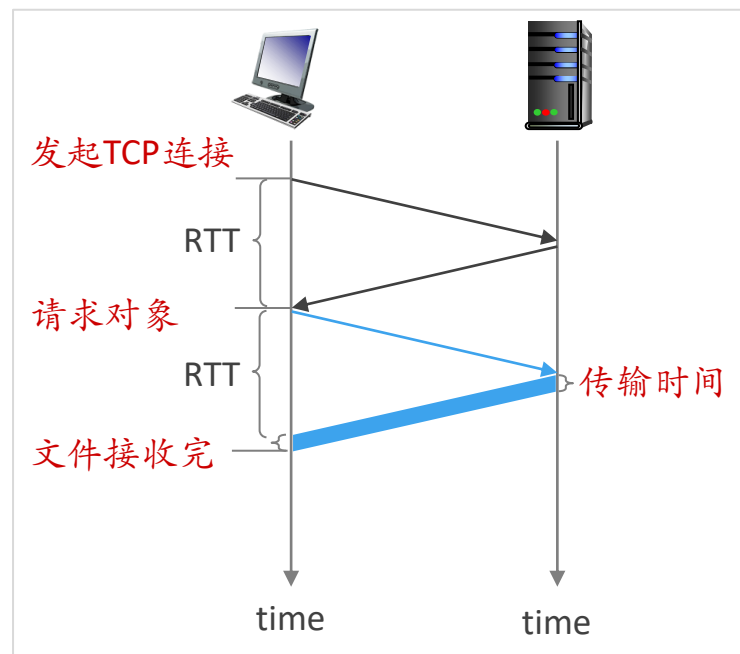
- 每个对象的HTTP响应时间

= 建立TCP连接所需的RTT

+ HTTP请求/响应所需的RTT

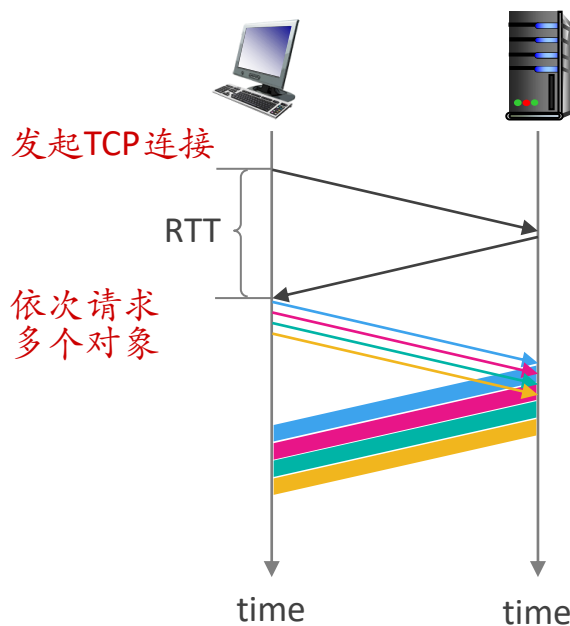
+ 对象传输时间

=  $2RTT + \text{传输时间}$



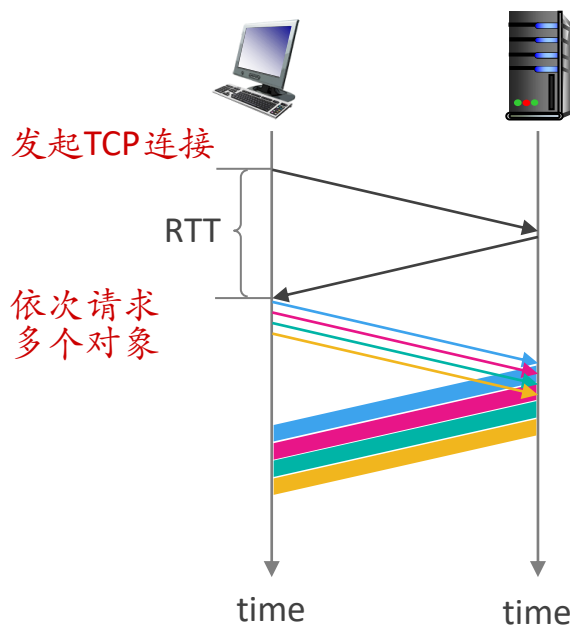
# 持续HTTP (HTTP/1.1)

- 一个TCP连接用于请求/响应多个对象
- 流水线工作方式：在一个对象请求的响应到达之前，发送其他对象的请求。



# 持续HTTP (HTTP/1.1)

- 一个TCP连接用于请求/响应多个对象
- 流水线工作方式：在一个对象请求的响应到达之前，发送其他对象的请求。
- 每个对象的HTTP响应时间 = RTT + 传输时间（除第一个对象以外）



# 持续HTTP (HTTP/1.1)

- 用户输入URL: `http://www.szu.edu.cn`
- 传输很多HTTP请求对象/响应报文
- 源地址和端口号
- 目的地址和端口号, HTTP 服务器端口80
- HTTPS服务器端口443

Time	Source	Source Port	Destination	Destination Port	Protocol	Info
18.428765	192.168.2.178	29070	210.39.12.247	80	HTTP	GET / HTTP/1.1
18.443648	210.39.12.247	80	192.168.2.178	29070	HTTP	HTTP/1.1 301 Moved Permanently (text/html)
18.509458	192.168.2.178	29071	210.39.12.247	443	HTTP	GET / HTTP/1.1
18.547112	210.39.12.247	443	192.168.2.178	29071	HTTP	HTTP/1.1 200 OK (text/html)
18.563694	192.168.2.178	29071	210.39.12.247	443	HTTP	GET /bower_components/owl.carousel/dist/assets/owl
18.579707	210.39.12.247	443	192.168.2.178	29071	HTTP	HTTP/1.1 200 OK (text/css)
18.580522	192.168.2.178	29071	210.39.12.247	443	HTTP	GET /bower_components/owl.carousel/dist/assets/owl
18.588740	192.168.2.178	29072	210.39.12.247	443	HTTP	GET /bower_components/bootstrap/dist/css/bootstrap
18.591451	192.168.2.178	29074	210.39.12.247	443	HTTP	GET /css/main.css HTTP/1.1
18.591492	192.168.2.178	29073	210.39.12.247	443	HTTP	GET /_sitegray/_sitegray_d.css HTTP/1.1
18.593054	192.168.2.178	29076	210.39.12.247	443	HTTP	GET /index.vsb.css HTTP/1.1
18.593086	192.168.2.178	29075	210.39.12.247	443	HTTP	GET /_sitegray/_sitegray.js HTTP/1.1
18.594538	210.39.12.247	443	192.168.2.178	29071	HTTP	HTTP/1.1 200 OK (text/css)
18.595106	192.168.2.178	29071	210.39.12.247	443	HTTP	GET /system/resource/js/counter.js HTTP/1.1
18.609246	210.39.12.247	443	192.168.2.178	29071	HTTP	HTTP/1.1 200 OK (application/javascript)
18.609892	192.168.2.178	29071	210.39.12.247	443	HTTP	GET /system/resource/js/dynclicks.js HTTP/1.1
18.620921	210.39.12.247	443	192.168.2.178	29075	HTTP	HTTP/1.1 200 OK (application/javascript)
18.621394	192.168.2.178	29075	210.39.12.247	443	HTTP	GET /system/resource/js/openlink.js HTTP/1.1
18.623355	210.39.12.247	443	192.168.2.178	29076	HTTP	HTTP/1.1 200 OK (text/css)
18.624044	192.168.2.178	29076	210.39.12.247	443	HTTP	GET /system/resource/js/base64.js HTTP/1.1
18.626521	210.39.12.247	443	192.168.2.178	29073	HTTP	HTTP/1.1 200 OK (text/css)
18.627054	192.168.2.178	29073	210.39.12.247	443	HTTP	GET /images/logo.png HTTP/1.1

# 持续HTTP (HTTP1.1)

- 用户输入URL: `http://www.szu.edu.cn`
- 传输很多HTTP请求对象/响应报文
- 源地址和端口号
- 目的地址和端口号, HTTP 服务器端口80
- HTTPS服务器端口443

Time	Source	Source Port	Destination	Destination Port	Protocol	Info	
18.428765	192.168.2.178	29070	210.39.12.247	80	HTTP	GET / HTTP/1.1	初始HTTP请求
18.443648	210.39.12.247	80	192.168.2.178	29070	HTTP	HTTP/1.1 301 Moved Permanently (text/html)	
18.509458	192.168.2.178	29071	210.39.12.247	443	HTTP	GET / HTTP/1.1	HTTPS请求
18.547112	210.39.12.247	443	192.168.2.178	29071	HTTP	HTTP/1.1 200 OK (text/html)	HTTPS响应
18.563694	192.168.2.178	29071	210.39.12.247	443	HTTP	GET /bower_components/owl.carousel/dist/assets/owl	
18.579707	210.39.12.247	443	192.168.2.178	29071	HTTP	HTTP/1.1 200 OK (text/css)	
18.580522	192.168.2.178	29071	210.39.12.247	443	HTTP	GET /bower_components/owl.carousel/dist/assets/owl	
18.588740	192.168.2.178	29072	210.39.12.247	443	HTTP	GET /bower_components/bootstrap/dist/css/bootstrap	多个TCP连接 并行请求对象
18.591451	192.168.2.178	29074	210.39.12.247	443	HTTP	GET /css/main.css HTTP/1.1	
18.591492	192.168.2.178	29073	210.39.12.247	443	HTTP	GET /_sitegray/_sitegray_d.css HTTP/1.1	
18.593054	192.168.2.178	29076	210.39.12.247	443	HTTP	GET /index.vsb.css HTTP/1.1	
18.593086	192.168.2.178	29075	210.39.12.247	443	HTTP	GET /_sitegray/_sitegray.js HTTP/1.1	
18.594538	210.39.12.247	443	192.168.2.178	29071	HTTP	HTTP/1.1 200 OK (text/css)	用已有的TCP连接 继续请求/响应对象
18.595106	192.168.2.178	29071	210.39.12.247	443	HTTP	GET /system/resource/js/counter.js HTTP/1.1	
18.609246	210.39.12.247	443	192.168.2.178	29071	HTTP	HTTP/1.1 200 OK (application/javascript)	
18.609892	192.168.2.178	29071	210.39.12.247	443	HTTP	GET /system/resource/js/dynclicks.js HTTP/1.1	
18.620921	210.39.12.247	443	192.168.2.178	29075	HTTP	HTTP/1.1 200 OK (application/javascript)	
18.621394	192.168.2.178	29075	210.39.12.247	443	HTTP	GET /system/resource/js/openlink.js HTTP/1.1	
18.623355	210.39.12.247	443	192.168.2.178	29076	HTTP	HTTP/1.1 200 OK (text/css)	
18.624044	192.168.2.178	29076	210.39.12.247	443	HTTP	GET /system/resource/js/base64.js HTTP/1.1	
18.626521	210.39.12.247	443	192.168.2.178	29073	HTTP	HTTP/1.1 200 OK (text/css)	
18.627054	192.168.2.178	29073	210.39.12.247	443	HTTP	GET /images/logo.png HTTP/1.1	

# HTTP 请求报文

## ■ ASCII 编码

### ▼ Hypertext Transfer Protocol

#### ▼ GET / HTTP/1.1\r\n

> [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]  
Request Method: GET  
Request URI: /  
Request Version: HTTP/1.1

请求行

Host: www.szu.edu.cn\r\n  
Connection: keep-alive\r\n  
DNT: 1\r\n  
Upgrade-Insecure-Requests: 1\r\n  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.132 Safari/537.36\r\n  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n  
Sec-Fetch-Site: none\r\n  
Sec-Fetch-Mode: navigate\r\n  
Sec-Fetch-User: ?1\r\n  
Accept-Encoding: gzip, deflate, br\r\n  
Accept-Language: en-US,en;q=0.9,zh-CN;q=0.8,zh;q=0.7\r\n

首部

\r\n

回车换行符表示首部结束



# HTTP 请求报文

---

- GET 请求
- POST 请求
  - 向服务器传输用户输入，位于实体部分
- HEAD 请求
  - 只请求首部
- PUT 请求
  - 向服务器上传新文件
  - 用实体部分的内容替换URL指定的文件



# HTTP 响应报文

状态行

## ▼ Hypertext Transfer Protocol

> HTTP/1.1 200 OK\r\n

```
Server: *****\r\n
Date: Fri, 13 Mar 2020 09:04:19 GMT\r\n
Content-Type: text/html\r\n
Transfer-Encoding: chunked\r\n
Connection: keep-alive\r\n
X-Frame-Options: SAMEORIGIN\r\n
Last-Modified: Fri, 13 Mar 2020 08:38:42 GMT\r\n
ETag: W/"f1d1-5a0b865914880"\r\n
Vary: Accept-Encoding\r\n
Cache-Control: max-age=600\r\n
Expires: Fri, 13 Mar 2020 09:00:53 GMT\r\n
Content-Language: zh-CN\r\n
Content-Encoding: gzip\r\n
\r\n
```

首部

[HTTP response 1/9]

[Time since request: 0.037654000 seconds]

[\[Request in frame: 545\]](#)

[\[Next request in frame: 563\]](#)

[\[Next response in frame: 581\]](#)

[Request URI: https://www.szu.edu.cn/]

## ▼ HTTP chunked response

### ▼ Data chunk (14253 octets)

Chunk size: 14253 octets

### ▼ Data (14253 bytes)

Data: 1f8b08000000000000003ed7d79731bc7b5efdf4e55bec304...

[Length: 14253]

Chunk boundary: 0d0a

### ▼ End of chunked encoding

Chunk size: 0 octets

\r\n

实体数据，压缩过的

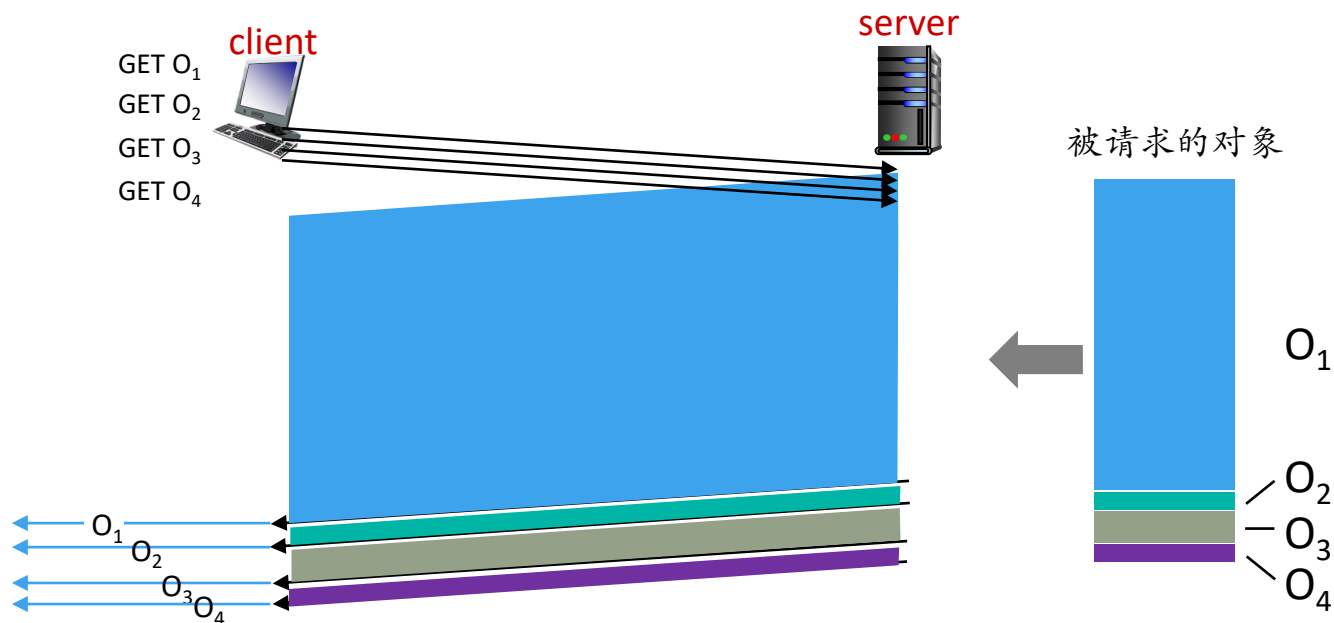


# HTTP 响应状态码

- 200 OK
  - 请求成功
- 301 Moved Permanently
  - 所请求的对象转移了，返回新地址
  - `HTTP/1.1 301 Moved Permanently\r\n`  
`Server: *****\r\n`  
`Date: Fri, 13 Mar 2020 09:04:19 GMT\r\n`  
`Content-Type: text/html\r\n`  
▼ `Content-Length: 178\r\n`  
`[Content length: 178]`  
`Connection: keep-alive\r\n`  
`Location: https://www.szu.edu.cn/\r\n`  
`\r\n`
- 400 Bad Request
  - 服务器无法解析该请求
- 404 Not Found
  - 服务器找不到所请求内容
- 505 HTTP Version Not Supported

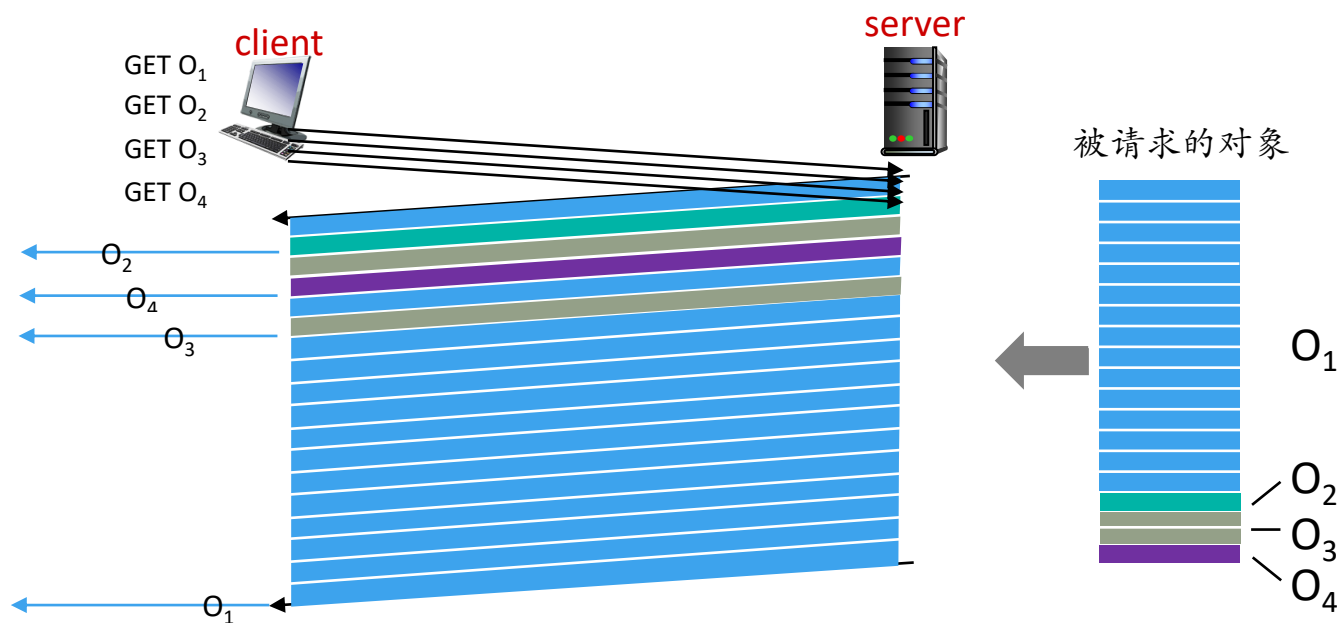
# HTTP/1.1 [RFC 2616, 1999]

- 先到先服务导致head-of-line blocking问题：小对象要等排在前面的大对象传输完，响应时间很长



# HTTP/2 [RFC 7540, 2015]

- 根据客户端指定的优先级决定对象的传输顺序
- 将对象分割成小块，交替传输



O<sub>2</sub> O<sub>3</sub> O<sub>4</sub> 很快就传完了，O<sub>1</sub> 略微延迟



# 习题

- 浏览器请求一个Web页面，其HTML文件引用了8个对象。HTML文件与对象均很小，忽略发送时间。不考虑DNS解析和页面缓存，在下列情况下，从在浏览器输入Web页面的URL并回车到接收完HTML文件和全部对象需要多少时间？设浏览器与服务之间的往返时延为RTT。

## 1. 没有并行TCP连接的非持续HTTP。

- $2RTT * 9$
- 每个对象包括HTML文件均需要2RTT，一个RTT是用来建立TCP连接，一个RTT是用来请求和接收对象。



# 习题

- 浏览器请求一个Web页面，其HTML文件引用了8个对象。HTML文件与对象均很小，忽略发送时间。不考虑DNS解析和页面缓存，在下列情况下，从在浏览器输入Web页面的URL并回车到接收完HTML文件和全部对象需要多少时间？设浏览器与服务服务器之间的往返时延为RTT。

## 2. 配置有5个并行连接的非持续HTTP。

- $2RTT + 2 * 2RTT$
- 考虑到并行，8除以5上取整为2，因此加上2个2RTT。



# 习题

- 浏览器请求一个Web页面，其HTML文件引用了8个对象。HTML文件与对象均很小，忽略发送时间。不考虑DNS解析和页面缓存，在下列情况下，从在浏览器输入Web页面的URL并回车到接收完HTML文件和全部对象需要多少时间？设浏览器与服务之间的往返时延为RTT。

## 3. 持续HTTP(流水线)。

- 2RTT+RTT
- 传输完HTML文件以后，连接保持，后续的8个对象继续使用该连接。考虑到忽略发送时间的流水线方式。



## 第二章知识点汇总

---

### ■ Web和HTTP

- 理解Web系统的构成
- 理解非持续和持续HTTP
- 掌握响应时间的计算方法
- 了解请求/响应报文的基本结构





# 第二章讲解内容

---

## 1. 应用层概述

- 客户/服务器架构，P2P架构，套接字

## 2. Web 和 HTTP

- 非持续HTTP，持续HTTP，响应时间，请求/响应报文

## 3. Email

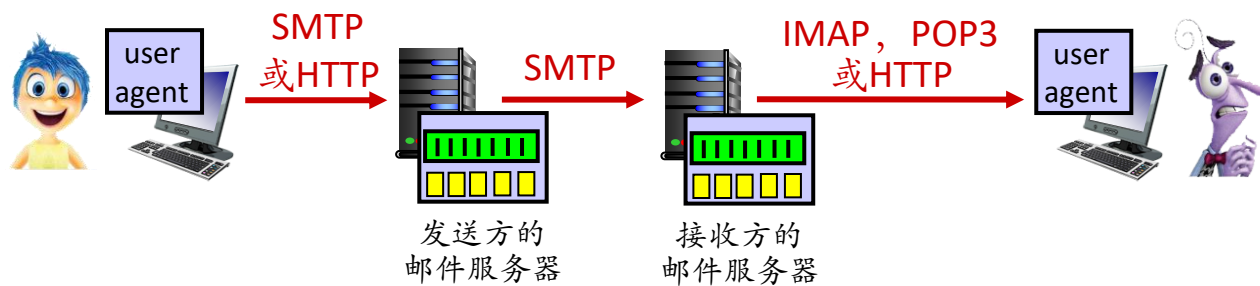
- SMTP, POP3, IMAP

## 4. DNS 域名系统

- 层级结构，迭代查询，递归查询

# Email

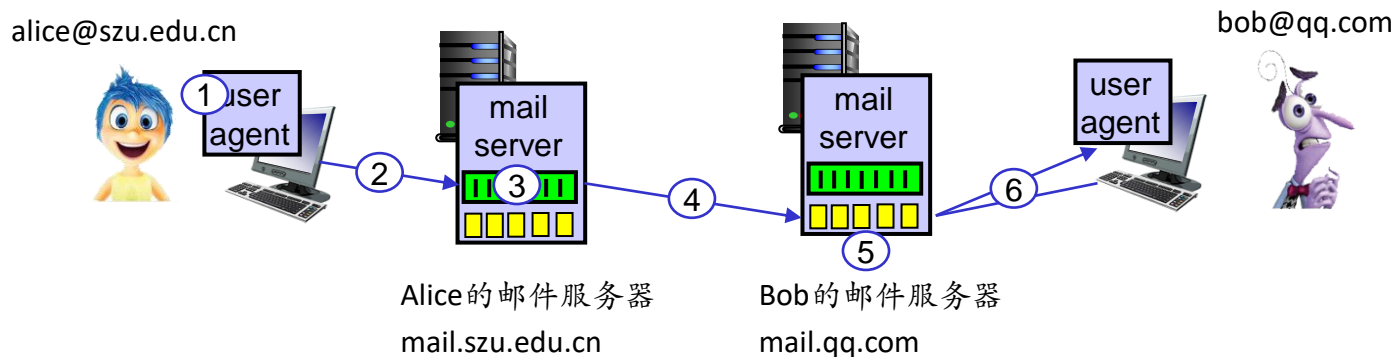
- 三件套：邮件代理、邮件服务器、邮件传输协议



- 邮件发送协议
  - SMTP: Simple Mail Transfer Protocol
- 邮件访问协议
  - IMAP: Internet Mail Access Protocol
  - POP3: Post Office Protocol-Version 3

# Email 传输

- 1) Alice用Outlook写了一封邮件，准备用邮箱alice@szu.edu.cn发给bob@qq.com。
- 2) Alice的Outlook将邮件报文发送给邮件服务器mail.szu.edu.cn，到达报文队列。
- 3) 邮件服务器mail.szu.edu.cn上的SMTP客户端建立一个与服务器mail.qq.com的TCP连接。
- 4) SMTP客户端通过该TCP连接发送邮件报文。
- 5) 邮件服务器email.qq.com将邮件报文存入Bob的邮箱。
- 6) Bob利用他的UA阅读邮件，**注意UA是客户端，先发起TCP连接。**





# SMTP 协议

---

- 使用TCP
- 报文内容为7比特ASCII编码
- 命令与响应

# SMTP 协议(RFC 5321)

## ■ TCP连接建立完以后

问  
候

- S: 220 qq.com //服务器说准备好了, 返回软件版本信息
- C: HELO szu.edu.cn //客户端自报家门, EHLO表示客户端问服务器支持哪些扩展服务
- S: 250 Hello szu.edu.cn, pleased to meet you //确认收到

传  
输  
邮  
件

- C: MAIL FROM: <alice@szu.edu.cn>
- S: 250 alice@szu.edu.cn ... Sender ok
- C: RCPT TO: <bob@qq.com>
- S: 250 bob@qq.com ... Recipient ok
- C: DATA
- S: 354 Enter mail, end with "." on a line by itself
- C: Do you like SZU?
- C: . //邮件内容结束
- S: 250 Message accepted for delivery

结  
束

- C: QUIT //客户端可以重复MAIL FROM指令继续发别的邮件
- S: 221 Bye.



# SMTP 与 HTTP 对比

---

- 都使用命令/响应方式、状态码
- 都使用TCP
- SMTP
  - Push protocol 推送
  - 多个对象放在一个报文中
  - 7比特ASCII编码
  - 特殊字符\r\n.\r\n表示数据结束
- HTTP
  - Pull protocol 去取
  - 每个对象被封装成一个报文
  - \r\n表示首部或主体结束



# 应用层协议

- 规范进程通信交换报文过程中的一切
  - 规范交互的方式与流程
  - 规范报文格式
  - 规范报文各个fields的含义
  - 规范进程什么时候发送报文
  - 规范进程如何发送/响应报文
  - .....



# 第二章知识点汇总

---

## ■ Email

- 了解邮件系统
- 了解SMTP协议的基本原理





# 第二章讲解内容

---

## 1. 应用层概述

- 客户/服务器架构，P2P架构，套接字

## 2. Web 和 HTTP

- 非持续HTTP，持续HTTP，响应时间，请求/响应报文

## 3. Email

- SMTP，POP3，IMAP

## 4. DNS 域名系统

- 层级结构，迭代查询，递归查询



# Domain Name 域名

- 为什么需要域名？
- 域名的层级结构：最右边第一级，点间隔级，向左递增，最多有128个层级。

[www.szu.edu.cn](http://www.szu.edu.cn)

- 域名的管理权限（层层授权）
  - ICANN机构管理顶级域（例如.cn）；
  - 运营“.cn”顶级域的组织（工信部）管理在“.cn”内注册的域名；
  - 注册了域名“.edu.cn”的组织（教育部）管理在“.edu.cn”下的域名；
  - 深大管理“.szu.edu.cn”下的域名。



# Domain Name 域名

---

- 注册域名

- 在域名系统中创建一个新的域，并赋予该域一个名称，即域名。

- 域名不是网址，也不是URL

# Domain Name 域名

## 顶级域发生了哪些变化？

域名

拉丁字母 a 到 z，不含  
其他语言文字或符号。

.org

.com

.uk

2009

新增的国际化域名  
不同语言文字的字符。

.عرب

.中国

.рф

(.arab)

(.china)

(.rf)

# Domain Name 域名

उदाहरण.परीक्षा

例子.测试

실례.테스트

例え.テスト

国际化域名让世界各地的人们能够使用自己的语言访问域名。

占全球互联网用户总数百分比前三位的国家/地区\*\*\*

30%



美国

9%



日本

6%



德国

2000

22%



中国

10%



美国

8%



印度

2014

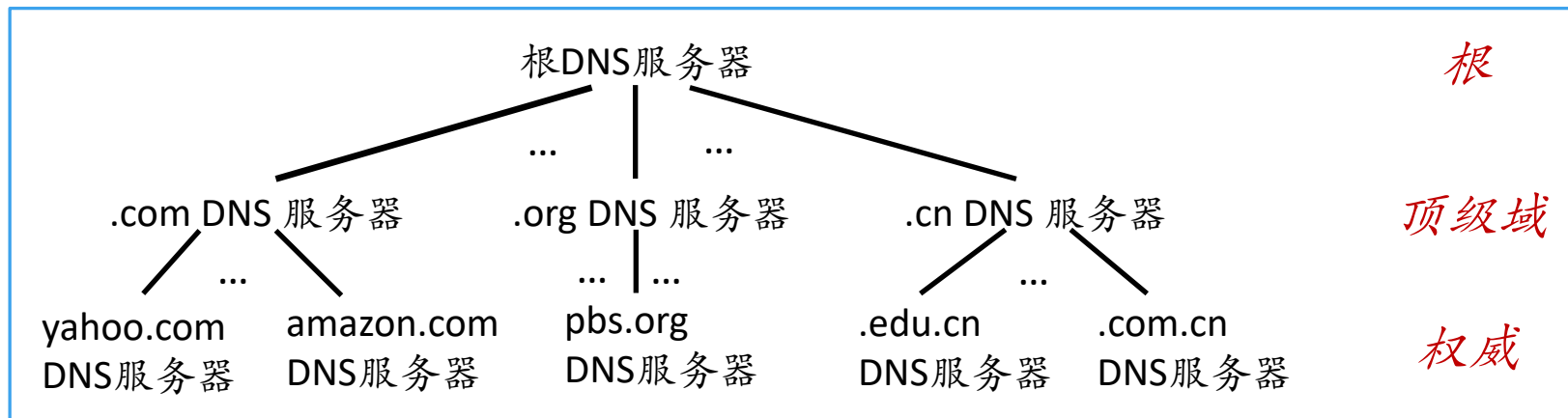


# DNS 域名系统

- 实现主机名与IP地址之间的转换
- Q: 怎么实现这个系统呢?
- A: 用一个集中式管理的DNS集群。
- 集中式方案的问题
  - 不具有扩展性
  - 单点失效问题
  - 时延问题

# DNS 域名系统

- 一个由众多的DNS服务器实现的分布式数据库



- 本地域名服务器
  - 一般ISP、企业、学校都会配置一个
  - 当主机做DNS查询时，请求会先发往本地域名服务器
    - 有域名地址映射的缓存
    - 代理查询

# DNS 域名系统

## ■ 根服务器

- 逻辑上13个
- 物理上1092个
- <https://root-servers.org>

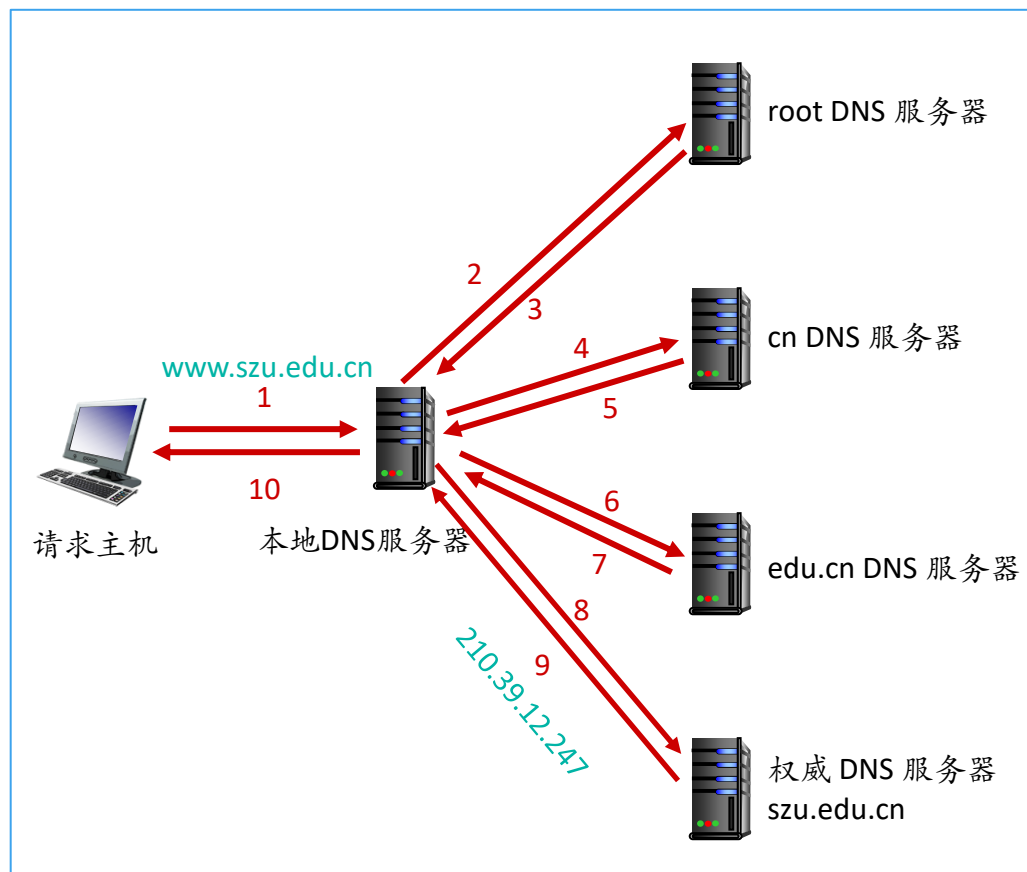
HOSTNAME	IP ADDRESSES	MANAGER
a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	VeriSign, Inc.
b.root-servers.net	199.9.14.201, 2001:500:200::b	University of Southern California (ISI)
c.root-servers.net	192.33.4.12, 2001:500:2::c	Cogent Communications
d.root-servers.net	199.7.91.13, 2001:500:2d::d	University of Maryland
e.root-servers.net	192.203.230.10, 2001:500:a8::e	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241, 2001:500:2f::f	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4, 2001:500:12::d0d	US Department of Defense (NIC)
h.root-servers.net	198.97.190.53, 2001:500:1::53	US Army (Research Lab)
i.root-servers.net	192.36.148.17, 2001:7fe::53	Netnod
j.root-servers.net	192.58.128.30, 2001:503:c27::2:30	VeriSign, Inc.
k.root-servers.net	193.0.14.129, 2001:7fd::1	RIPE NCC
l.root-servers.net	199.7.83.42, 2001:500:9f::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project





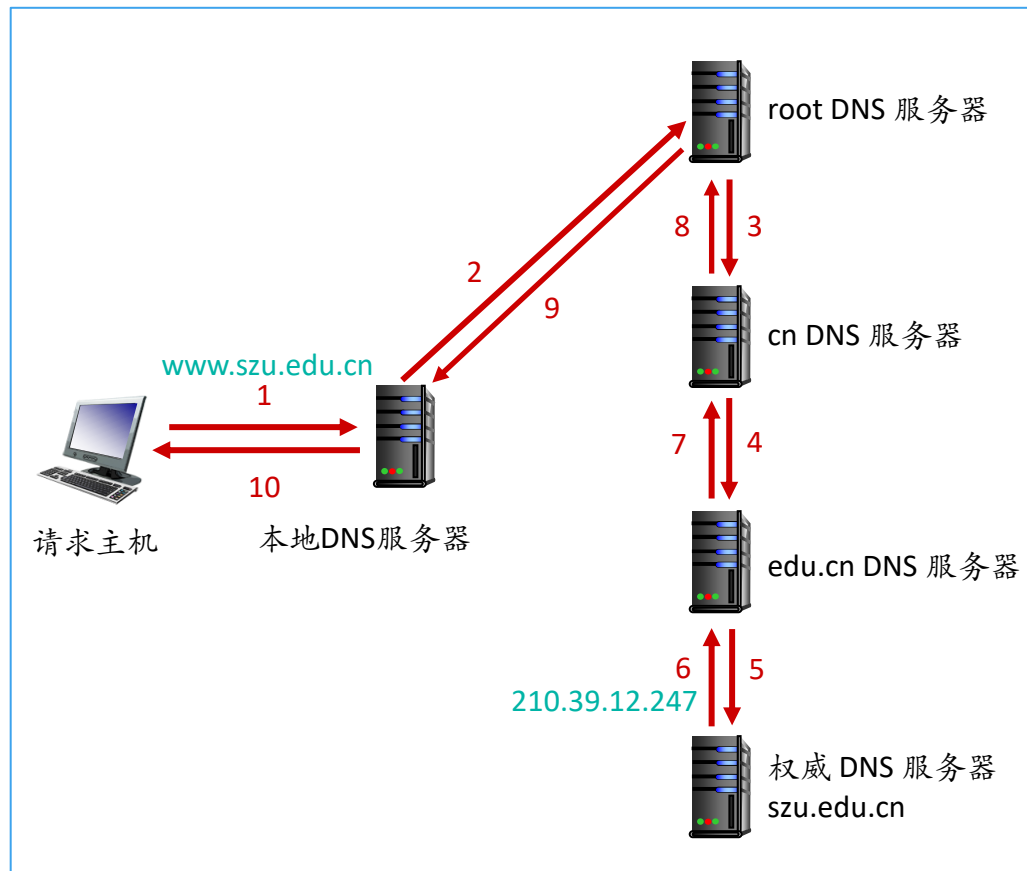
# DNS 域名解析：迭代查询

- 举例：查询 [www.szu.edu.cn](http://www.szu.edu.cn)
- 迭代查询
  - 被询问的服务器返回下一步询问的服务器地址
  - 我不知道，你去问XXX吧



# DNS 域名解析：递归查询

- 举例：查询 [www.szu.edu.cn](http://www.szu.edu.cn)
- 递归查询
  - 被询问的服务器去查询
  - 我不知道，我帮你去问吧





# DNS 域名系统

- 分布式数据库中的记录  
(name, value, type, ttl)
- **type=A**
  - Name: 主机名
  - Value: IP地址
- **type=NS**
  - Name: 域名
  - Value: 这个域的权威服务  
器的主机名
- **type=CNAME**
  - name: 主机名
  - Value: 别名
- **type=MX**
  - Name: 域名
  - Value: 邮件服务器的主机  
名



# 向DNS添加记录

- 举例：一家创业公司Network Utopia
- 向DNS注册服务机构注册域名networkuptopia.com
  - 需要提供域名，权威服务器的IP地址
  - 注册服务机构向.com 顶级域名服务器添加两条记录  
(networkuptopia.com, dns1.networkuptopia.com, NS)  
(dns1.networkuptopia.com, 212.212.212.1, A)
- 创建拥有IP地址212.212.212.1的权威服务器
  - Web服务器：为www.networkuptopia.com添加A类记录
  - Email服务器：为networkuptopia.com添加MX类型的记录



# DNS 习题

- 我们学习了迭代查询www.szu.edu.cn的IP地址的过程。利用 dig 工具验证该过程，列出迭代查询中服务器的名字和IP地址。
- dig 的安装方法 <https://nil.uniza.sk/how-install-dig-dns-tool-windows-10/>
- dig 的使用说明 <https://linux.die.net/man/1/dig>
- dig 的使用方法 <https://www.madboa.com/geek/dig>

# DNS 习题

- 举例：查询根服务器，返回cn顶级服务器。
- 请如法炮制找出其他DNS服务器。

```
命令提示符
C:\Users\ruitao>dig +norecurse @a.root-servers.net any www.szu.edu.cn

;<<>> DiG 9.14.0 <<>> +norecurse @a.root-servers.net any www.szu.edu.cn
(1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 23896
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 8, ADDITIONAL: 11

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1472
;; QUESTION SECTION:
;www.szu.edu.cn.                IN      ANY

;; AUTHORITY SECTION:
cn.                172800  IN      NS      a.dns.cn.
cn.                172800  IN      NS      b.dns.cn.
cn.                172800  IN      NS      c.dns.cn.
cn.                172800  IN      NS      d.dns.cn.
cn.                172800  IN      NS      e.dns.cn.
cn.                172800  IN      NS      f.dns.cn.
cn.                172800  IN      NS      g.dns.cn.
cn.                172800  IN      NS      ns.cernet.net.

;; ADDITIONAL SECTION:
a.dns.cn.          172800  IN      A        203.119.25.1
b.dns.cn.          172800  IN      A        203.119.26.1
c.dns.cn.          172800  IN      A        203.119.27.1
d.dns.cn.          172800  IN      A        203.119.28.1
e.dns.cn.          172800  IN      A        203.119.29.1
f.dns.cn.          172800  IN      A        195.219.8.90
g.dns.cn.          172800  IN      A        66.198.183.65
ns.cernet.net.     172800  IN      A        202.112.0.44
a.dns.cn.          172800  IN      AAAA     2001:dc7::1
d.dns.cn.          172800  IN      AAAA     2001:dc7:1000::1

;; Query time: 529 msec
;; SERVER: 198.41.0.4#53(198.41.0.4)
;; WHEN: Thu Mar 12 17:14:46 中国标准时间 2020
;; MSG SIZE rcvd: 370
```



# 第二章知识点汇总

---

- DNS域名系统
  - 理解层级结构
  - 掌握迭代查询和递归查询



# 第二章知识点汇总

## ■ 应用层概述

- 理解两种架构
- 掌握使用套接字进行网络编程的方法（实验）
- 理解应用程序对网络服务的不同需求

## ■ Email

- 了解邮件系统
- 了解SMTP协议的基本原理

## ■ Web和HTTP

- 理解Web系统的构成
- 理解非持续和持续HTTP
- 掌握响应时间的计算方法
- 了解请求/响应报文的基本结构

## ■ DNS域名系统

- 理解层级结构
- 掌握迭代查询和递归查询