

# 计算机网络及应用

## Computer Networks and Applications

### 第二章 应用层

应用层协议原理; Web和HTTP; FTP; 电子邮件 Email; 域名系统 DNS; P2P文件共享; Socket programming

主讲: 清华大学 贾庆山

教材: J.F. Kurose, K.W. Ross, Computer Networking: A Top-Down Approach, Addison Wiley, 7th Edition, 2017 (机械工业出版社中文版, 2018)

清华大学 2022秋W3

Special thanks to Prof. Kurose and Prof. Ross for presentation material

1

### 单选题 1分

Web的应用层协议是

- A HTTP
- B WWW
- C Web
- D Web是什么?

清华大学 2022秋W3

2

## 应用层: 提纲

- 应用层协议原理
- **Web 和 HTTP**
- FTP
- 电子邮件 Email
- 域名系统 DNS
- P2P 文件共享
- Socket programming

清华大学 2022秋W3

3

## 2.2 Web和HTTP

- World Wide Web让因特网成为仅有数据网
- 按需操作
- Web的应用层协议是超文本传输协议(HyperText Transfer Protocol, HTTP)[RFC1945, RFC2616], 定义报文的格式、客户机和服务器如何进行报文交换
- Web页面由对象组成
- 对象: HTML文件、JPEG图片、Java applet、音频文件.....
- Web页面: 一个基本HTML文件+引用对象
- 每个对象可以通过一个URL来被寻址
- URL(Uniform Resource Locator)例子 统一资源定位符  
www.someschool.edu / someDept/pic.gif

清华大学 2022秋W3

主机名

路径名

4

## HTTP概况

- Web浏览器实现了HTTP的客户机端
- Web服务器存储对象，由URL寻址
- 支撑HTTP的运输层协议是TCP
- 提供可靠数据传输服务
- 分层体系结构优点
  - HTTP不担心数据丢失、TCP如何从网络的数据丢失和乱序故障中恢复
- HTTP是无状态协议
  - 服务器响应客户机请求时不存储客户机的任何状态信息
  - 重复请求，就重复响应
- Web使用了客户机/服务器应用程序体系结构



- 过去的历史(state)必须保留一个保留“状态”的协议是非常复杂的!
- 万一server/client意外停机了，它们对“状态”的观点会不一致，必须重新协调以实现一致

清华大学 2022秋W3

5

## 非持久连接和持久连接

- 非持久连接(non-persistent connection)
  - 每个请求/响应对经单独的TCP连接发送
- 持久连接(persistent connection)
  - 所有请求/响应对经相同的TCP连接发送
- HTTP 1.0[RFC 1945]使用非持久连接
- HTTP 1.1[RFC 2616]使用持久连接

清华大学 2022秋W3

6

## 非持久连接

假设用户输入如下URL `www.someSchool.edu/someDepartment/home.index`  
(包含文本和10个jpeg图片的引用)

- 1a. HTTP client 初始化一个到 `www.someSchool.edu` 的HTTP服务器的连接，通过端口 80
- 1b. 在主机 `www.someSchool.edu` 的HTTP 服务器在80端口等待TCP连接：“接受”连接；通知client
2. HTTP client 发送HTTP请求报文 *request message* (包含URL) 到TCP连接套接字；报文中指明client 想要 `someDepartment/home.index` 对象
3. HTTP 服务器接受请求报文，起草响应报文 *response message*，其中包含被请求的对象，然后把报文发送到它的socket

time

清华大学 2022秋W3

7

## 非持久连接(续)

4. HTTP 服务器关闭 TCP 连接
5. HTTP client 接收包含html文件的响应报文，显示html；解读html文件，发现有10个被引用的jpeg对象
6. 对每个jpeg对象再重复前面的1-5步

time

HTTP协议不管客户机如何解释一个Web页面。两个不同的浏览器(客户机)也许会以不同的方式解释同一个页面。

清华大学 2022秋W3

8

## 非持久连接：响应时间

**RTT定义：**一个短分组从客户到服务器然后再返回客户所花费的时间。

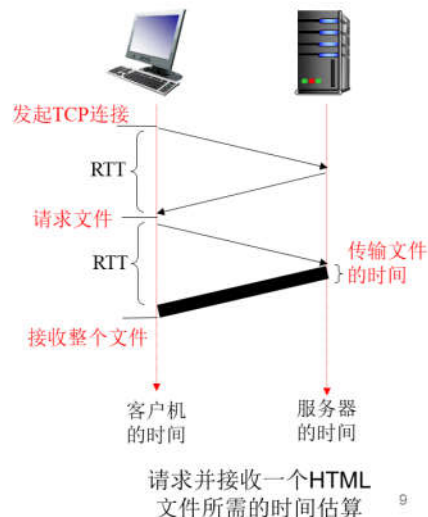
### 响应时间

- 1 RTT：初始化TCP连接
- 1 RTT：HTTP请求和返回HTTP响应的头几个字节
- 文件传输时间

总响应时间

= 2RTT + 文件传输时间

清华大学 2022秋W3



## 持久连接

### 非持久连接的缺点

- 为每个请求对象建立维护全新连接，在客户机和服务器都分配TCP的缓冲区和变量
- 每个对象的交付时延为两个RTT(建立TCP + 请求和接收对象)

### 持久连接

- 服务器在发送出响应之后保持连接在打开状态
- 随后的在同一对client/server之间的报文可以在这个打开的连接上发送

### 不带流水线的持久连接

- 仅当前一个响应已被接收以后，client才能发出新的请求
- 每个被引用对象都需要一个RTT

### 带流水线的持久连接

- HTTP/1.1的默认模式
- Client只要遇到一个被引用对象，就可以立刻发送请求
- 最好的情况下得到所有的被引用对象只需要一个RTT

清华大学 2022秋W3

10

## 题外：可穿戴式智能设备

### 智能手机 → 智能穿戴设备

- Google眼镜，智能手表(三星、小米)，智能腕带
- 医疗设备

### 三大功能

- 联网
- 传感
- 人机界面

### 启发

- 互联网无处不在
- 上述设备的成功需要建立在强大的互联网技术上
- 大数据，云计算，数据挖掘，人工智能



清华大学 2022秋W3

11

## HTTP请求报文

请求，响应

- 两种类型的HTTP 报文: *request, response*
- HTTP request message: 请求报文
  - 用普通的ASCII文本书写

请求行

方法字段	URL字段	HTTP协议版本字段
GET	/somedir/page.html	HTTP/1.1

首部行

Host: www.someschool.edu	用于Web代理高速缓存
User-agent: Mozilla/5.0	浏览器类型
Connection: close	非持久连接
Accept-language: fr	法语版本

附加回车换行符表示报文结束

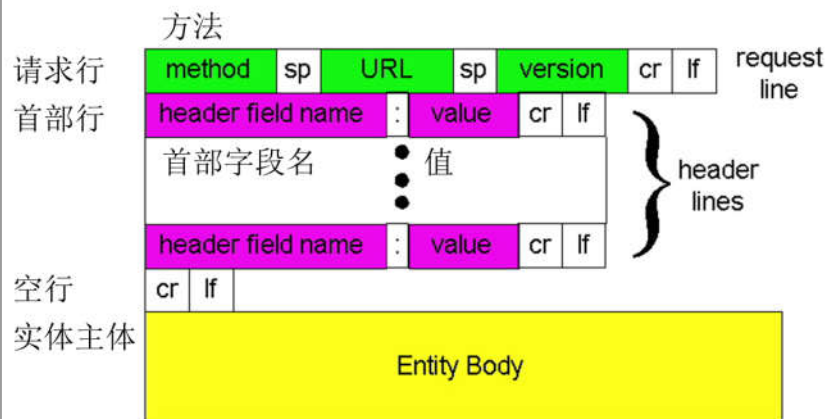
方法字段可取GET, POST, HEAD, PUT, DELETE, 一般用GET

清华大学 2022秋W3

12



## HTTP请求报文的通用格式



POST方法时使用，用户提交表单，比如搜索，  
实体主体包含的就是用户输入的搜索内容

清华大学 2022秋W3

13

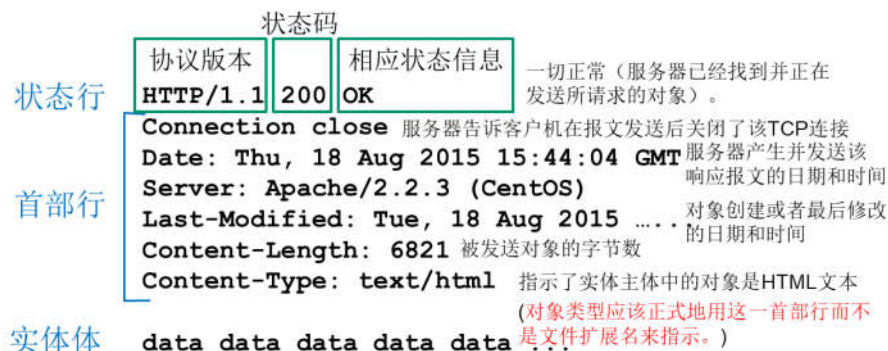
## 说明

- 用表单生成的请求报文未必使用POST方法
  - 例如用户输入monkeys和bananas
  - 可以使用扩展的URL
  - GET  
`www.somesite.com/animalsearch?monkeys&bananas`
- HEAD，服务器响应HTTP报文，但不返回请求对象。可用于调试跟踪。
- PUT，向Web服务器上传对象。
- DELETE，删除Web服务器上的对象。

清华大学 2022秋W3

14

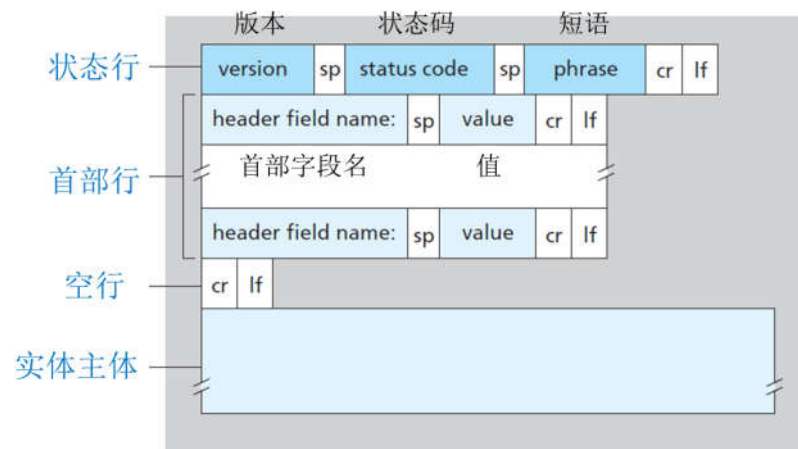
## HTTP响应报文



清华大学 2022秋W3

15

## HTTP响应报文的通用格式



清华大学 2022秋W3

16

## 常见HTTP响应状态码和短语

位于 server->client 响应报文的第一行

### 200 OK

- 请求成功，信息包含在返回的响应报文中。

### 301 Moved Permanently

- 请求的对象已经被永久转移了，新的URL定义在响应报文的Location:首部行中指定。客户机软件自动用新的URL获取该对象。

### 400 Bad Request

- 一个通用差错代码，指示该请求不能被服务器理解。

### 404 Not Found

- 被请求的文档不在服务器上。

### 505 HTTP Version Not Supported

- 服务器不支持请求报文使用的HTTP协议版本。

清华大学 2022秋W3

17

## 自己尝试进行HTTP连接 (client side)

### 1. 远程登录到你喜爱的 Web server

```
telnet www.au.tsinghua.edu.cn 80
```

Opens TCP connection to port 80  
(default HTTP server port) at www.au.tsinghua.edu.cn  
Anything typed in sent  
to port 80 at www.au.tsinghua.edu.cn

### 2. 键入一个 GET HTTP 请求

```
GET /publish/au/index.html HTTP/1.1  
Host: www.au.tsinghua.edu.cn
```

By typing this in (hit carriage return twice), you send this minimal (but complete) GET request to HTTP server

### 3. 看看HTTP服务器返回的响应报文！

注：Dos界面下若无法显示字符，键入 ctrl+] 并回车即可

清华大学 2022秋W3

18

## 用户与服务器的交互: cookie

- 许多大的Web站点都使用 cookie

### 四个组成部分:

- 1) HTTP响应报文中有一个 cookie首部行
- 2) HTTP请求报文中有一个 cookie首部行
- 3) 在用户端系统中保留有一个 cookie文件，由用户的浏览器管理
- 4) Web站点有一个后端数据库

### 例子

- Susan 总是从同一台PC访问因特网
- 她头一次访问某个电子商务站点
- 当初始 HTTP 请求到达站点时，站点生成一个唯一的标识号码(ID)并在后端的数据库中为这个ID创建一条记录

清华大学 2022秋W3

19

## 用cookie保持用户状态



清华大学 2022秋W3

20

## Cookies (续)

### Cookies 可以带来

- 授权 authorization
- 购物车
- 建议
- 用户会话状态 (Web e-mail)

### 如何保持“状态”信息

- 协议端点: 在发送方/接收方之间通过多次交互来维护状态
- cookies: http报文携带状态
- 在无状态的HTTP上建立一个用户会话层

### Cookie和隐私

- Cookies允许网站对你进行充分的学习和研究
- 你可以把姓名和e-mail提供给网站

### 存在什么问题?

用户信息安全, 伪装cookie套取用户信息 [Cookie Central 2007]

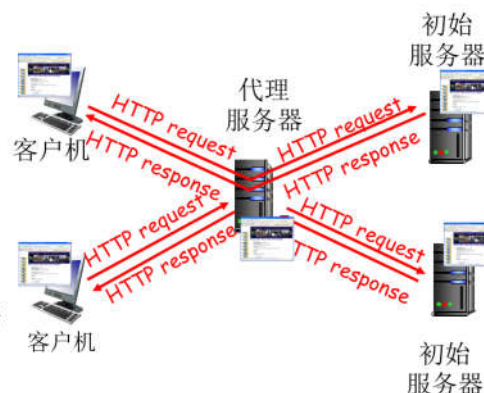
清华大学 2022秋W3

21

## Web缓存器(代理服务器)

定义: 能够代表初始Web服务器来满足HTTP请求的网络实体

- 用户设置浏览器, 所有HTTP请求首先指向Web缓存器
- 浏览器把所有的HTTP请求发送到web cache
  - 如果cache中有所需的对象: cache返回对象
  - 否则cache从原始的服务器请求该对象, 然后把该对象发送回client



清华大学 2022秋W3

22

## 关于Web缓存

- Cache 扮演的既是client又是server的角色
- 通常cache由ISP (大学、公司或住宅ISP) 安装

### 为什么要Web缓存?

- 减少客户请求的响应时间
- 减少机构接入链路的网络通信量
- 因特网上密布cache: 可以使得那些服务能力较差的内容提供商有效地发布它们的内容(也适用于P2P文件共享)

清华大学 2022秋W3

23

## 缓存的例子

$$(15 \text{ 个请求/s}) \times (1 \text{ Mb/请求}) / (100 \text{ Mbps}) = 0.15 = 15\%$$
$$(15 \text{ 个请求/s}) \times (1 \text{ Mb/请求}) / (15 \text{ Mbps}) = 1 = 100\%$$

### 假设

- 平均对象大小 = 1M bits
- 从某机构的浏览器到初始服务器的平均请求率 = 15个/sec
- 从互联网接入路由器2到任意初始服务器并返回的时延, 即Internet时延 = 2 s

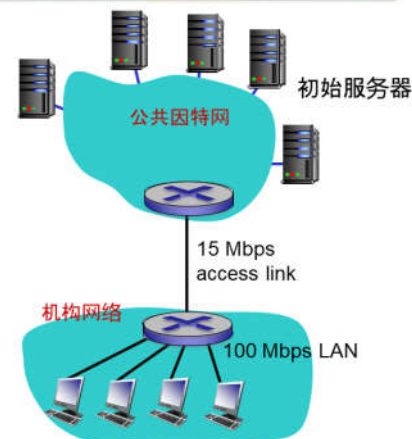
### 结果

- 局域网利用率  $\rho = 15\%$
  - 接入链路利用率  $\rho = 100\%$
  - 总时延 = Internet时延 + 接入时延 + LAN时延
- $$= 2 \text{ sec} + \text{minutes} + \text{milliseconds}$$

注: 每部分时延各包括处理/传播/传输/排队时延, 处理/传播时延忽略, 传输时延1kb/带宽, 排队时延可利用排队论知识分析

清华大学 2022秋W3

24





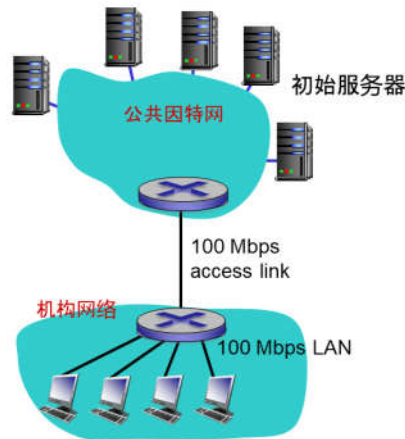
## 缓存的例子 (续1)

### 可能的解决方案

- 增加接入链路的带宽，比如说，100 Mbps

### 结果

- LAN利用率 = 15%
- 接入链路利用率 = 15%
- 总时延 = Internet时延 + 接入时延 + LAN时延  
= 2 sec + msecs + msecs
- 通常升级费用较高



清华大学 2022秋W3

25

## 缓存的例子 (续2)

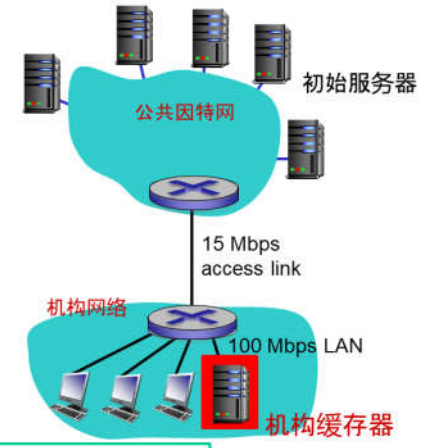
### 安装Web缓存器

- 假设命中率为0.4

### 结果

- 40% 的请求几乎立刻得到满足
- 60% 的请求由初始服务器来满足
- 接入链路的利用率下降到60%，从而产生的时延可以忽略不计(比如说100 msec)
- 总评价时延 = Internet时延 + 接入时延 + LAN时延 =  $0.6 * (2.1 \text{ secs}) + 0.4 * \text{milliseconds} \approx 1.2 \text{ secs}$

注：可利用排队论知识分析 queueing delay 的计算



响应时延更低  
费用更少

清华大学 2022秋W3

26

## 问题：缓存哪些文档？

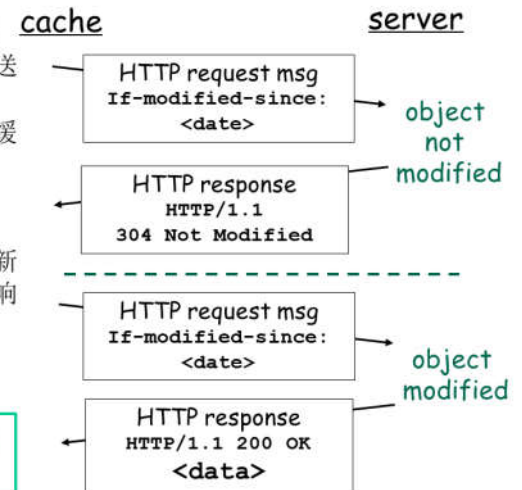
- 直观想到
  - 不变的文档
  - 热门文档
- 频繁访问的页面的集合
- 请求率/更新率  $\gg 1$  的文档可缓存

清华大学 2022秋W3

27

## 条件GET方法

- 目的：**如果cache中有对象的最新版本，就不从服务器发送该对象
  - cache:** 在HTTP请求中指明所缓存的对象拷贝的日期  
If-modified-since: <date>
  - server:** 如果缓存的拷贝是最新的，则发回不含具体对象的响应
- 动手试试IE浏览器的代理服务器设置



清华大学 2022秋W3

28

条件GET方法存在于下列哪个位置:

- ☐ A 用户代理和缓存器之间
- ☒ B 缓存器和初始服务器之间
- ☐ C 用户代理和初始服务器之间

## Web服务器建设

- 常用的Web服务器软件
  - Apache, 世界使用排名第一, 最常用的web服务器软件, 优势在开源代码开放
  - IIS, Internet Information Server, 微软公司产品, 包括Web服务器、FTP服务器、NNTP服务器和SMTP服务器
  - Nginx, 小巧高效的HTTP服务器, 支持反向代理
- 数据库网站架构
  - Apache(HTTP)+Tomcat(JSP/JAVA)+MySQL(database)
  - 其他动态网页技术: ASP, PHP, .net, CGI等
  - 如何支持大规模电子商务网站

## 提纲

- 应用层协议原理
- Web 和 HTTP
- **FTP**
- 电子邮件 Email
- 域名系统 DNS
- P2P 文件共享
- Socket programming

## 2.3 文件传输协议: FTP



- 在本地和远程站点之间传递文件
- client/server 模型
  - **client**: 发起文件传输的一方 (无论是发送到远程站点还是从远程站点下载)
  - **server**: 远程站点
- ftp: [RFC 959]
- ftp server: 控制连接端口 21、数据连接端口 20



## FTP: 控制和数据连接是相互分开的

- FTP client 通过21端口联系FTP server, 采用TCP作为传输协议
- Client 通过控制连接获得访问授权
- Client 通过控制连接发送命令来浏览远程的文件目录
- 当 server 接收到文件传输命令时, 为client打开第二个TCP连接, 数据连接
- 当一个文件传送结束后, server 关闭数据连接



- Server 打开另一个TCP数据连接来传送另一个文件
- 控制连接: “带外传输” out of band
- HTTP带内(in-band)发送控制信息
- FTP server 维护“状态”: 当前目录、早先的授权
- HTTP无状态

清华大学 2022秋W3

33

## FTP 命令和响应信息

### 命令的例子

- 通过控制连接以ASCII文本方式送
- USER username**
- PASS password**
- LIST** 返回当前目录的文件列表
- RETR filename** 访问(gets) 文件
- STOR filename** 把文件存到(puts) 远程站点

### 返回代码的例子[RFC959]

- 状态代码和短语(类似 HTTP)
- 331 Username OK, password required**
- 125 data connection already open; transfer starting**
- 425 Can't open data connection**
- 452 Error writing file**

清华大学 2022秋W3

34

## 提纲

- 应用层协议原理
- Web 和 HTTP
- FTP
- 电子邮件 Email
- 域名系统 DNS
- P2P 文件共享
- Socket programming

清华大学 2022秋W3

35

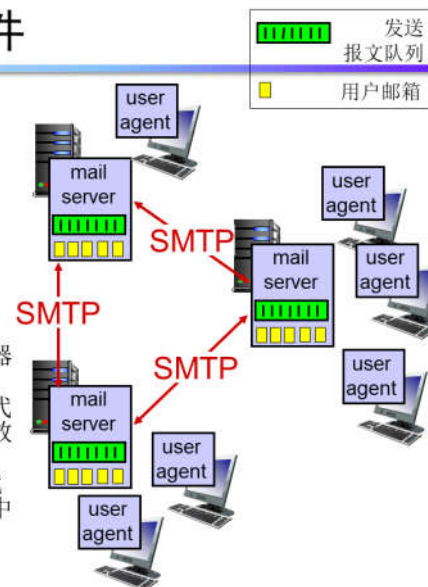
## 因特网中的电子邮件

### 三个主要组成部分

- 用户代理 user agents
- 邮件服务器 mail server
- 简单邮件传输协议 simple mail transfer protocol: SMTP

### 用户代理

- 阅读、回复、转发、保存和撰写
- 例子: Outlook, Apple Mail等
- 发送出和接收的报文存储在服务器上
- Alice完成邮件撰写时, 她的用户代理向其邮件服务器发送邮件, 被放在邮件服务器发送报文队列中。
- Bob想读取一条报文时, 其用户代理从他的位于邮件服务器的邮箱中获取该报文。



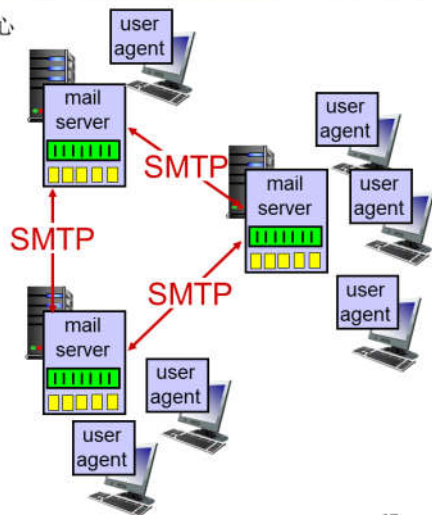
清华大学 2022秋W3

36

## 电子邮件：邮件服务器 mail servers

邮件服务器 电子邮件体系结构的核心

- 邮箱 **mailbox** 包含各个用户的已接收的邮件报文
- 待发邮件报文的**报文队列 message queue**
- **SMTP 协议** 在邮件服务器之间发送邮件报文的协议
  - client: 发送服务器
  - server: 接收服务器



清华大学 2022秋W3

37

## 电子邮件：SMTP [RFC 5321]

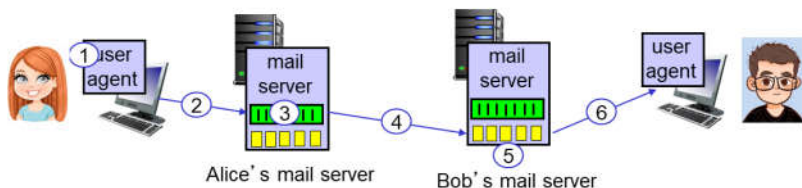
- 采用 TCP 端口25把email报文从客户端可靠传递到服务器端
- 直接传递：发送服务器到接收服务器
- 传递的三个阶段
  - 握手handshaking (greeting)
  - 报文传送
  - 关闭连接
- 命令commands /响应 responses 交互
  - **commands**: ASCII 文本格式
  - **response**: 一个回答码和一些（可选的）英文解释
- 报文的主体部分（不只首部）必须是7比特ASCII码格式！

清华大学 2022秋W3

38

## 例子：Alice给Bob发邮件

- 1) Alice 用用户代理写了一封邮件给 **bob@someschool.edu**
- 2) Alice的用户代理把邮件报文发送给她的邮件服务器；报文被排入报文队列
- 3) SMTP的Client端打开与Bob的邮件服务器的TCP连接
- 4) SMTP的client端通过TCP连接发送出Alice的邮件报文
- 5) Bob的邮件服务器把收到的报文放入Bob的邮箱
- 6) Bob调用它的用户代理来阅读邮件报文



清华大学 2022秋W3

39

## SMTP命令/响应交互的例子

```
S: 220 hamburger.edu 服务器主机名
C: HELO crepes.fr      客户机主机名
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

清华大学 2022秋W3

40

## 自己尝试 SMTP 交互命令

- ❑ telnet servername 25
  - ❑ see 220 reply from server
  - ❑ enter HELO, MAIL FROM, RCPT TO, DATA, QUIT commands
- above lets you send email without using email client (reader)

AUTH LOGIN PLAIN 用户身份认证  
使用 base64 code

清华大学 2022秋W3

41

```
C:\Users\XiaLi>telnet mail.tsinghua.edu.cn 25
220 tsinghua.edu.cn Anti-spam GT for Coremail System (tsinghua[20130305])
HELO tsinghua.org.cn
250 OK
MAIL FROM: <xiali@tsinghua.org.cn>
250 Mail OK
RCPT TO: <xial@tsinghua.edu.cn>
250 Mail OK
DATA
354 End data with <CR><LF>.<CR><LF>
From: newsletter@airtransportnews.aero
To: xial@tsinghua.edu.cn
X-Mailer: Microsoft Windows Mail 6.0.6000.16480
X-MimeOLE: Produced By Microsoft MimeOLE V6.0.6000.16480
Subject: Just a faking mail for test
Content-type: text/html
X-AIMC-AUTH: (null)
X-AIMC-MAILFROM: newsletter@airtransportnews.aero
X-AIMC-MSG-ID: lpT7zaTB
```

Hello, a fake email test, thanks.

```
.
250 Mail OK queued as DsxvpgBnCXFOWVVSfcMAA--.28803S2
QUIT
221 Bye
```

清华大学 2022秋W3

42



清华大学 2022秋W3

43

```
C:\Users\XiaLi>telnet mail.tsinghua.edu.cn 25
220 tsinghua.edu.cn Anti-spam GT for Coremail System (tsinghua[20130305])
HELO tsinghua.org.cn
250 OK
MAIL FROM: <xiali@tsinghua.org.cn>
250 Mail OK
RCPT TO: <xial@tsinghua.edu.cn>
250 Mail OK
DATA
354 End data with <CR><LF>.<CR><LF>
From: newsletter@airtransportnews.aero
To: xial@tsinghua.edu.cn
X-Mailer: Microsoft Windows Mail 6.0.6000.16480
X-MimeOLE: Produced By Microsoft MimeOLE V6.0.6000.16480
Subject: Just a faking mail for test
Content-type: text/html
X-AIMC-AUTH: (null)
X-AIMC-MAILFROM: newsletter@airtransportnews.aero
X-AIMC-MSG-ID: lpT7zaTB
```

Hello, a fake email test, thanks.

```
.
250 Mail OK queued as DsxvpgBnCXFOWVVSfcMAA--.28803S2
QUIT
221 Bye
```

清华大学 2022秋W3

44





思考：  
安全问题，可以伪装他人email地址给别人发信！

清华大学 2022秋W3

45

## SMTP 续

### 与持久HTTP相比较

- SMTP 使用持久连接
- SMTP 要求报文(header & body)必须是7位ASCII码
- SMTP 服务器使用 CRLF.CRLF 确定报文的结束 (CRLF, 回车换行符)
- HTTP: "拉" pull, 从服务器拉取信息, TCP连接由获取方发起
- SMTP: "推" push, 把文件推向服务器, TCP连接由发送方发起
- HTTP: 使用ASCII命令/响应交互和表示状态码, 但报文中可以包含非ASCII数据。
- SMTP: 使用ASCII命令/响应交互和表示状态码, 报文中非ASCII字符或二进制数据必须按照7比特ASCII码进行编码。
- HTTP: 每个对象被封装在它自己的响应报文里
- SMTP: 所有报文对象放在一个报文中

清华大学 2022秋W3

46

### 单选题 1分

Alice给Bob发邮件的过程中，下面哪个过程使用的协议不会是SMTP

- A Alice的用户代理发送报文到Alice的邮件服务器
- B Alice的邮件服务器发送报文到Bob的邮件服务器
- C Bob的用户代理收取Bob邮件服务器上的报文

清华大学 2022秋W3

47

## 邮件报文格式和MIME(Multipurpose Internet Mail Extension)

- 首部行包含环境信息，由RFC 822定义

From首部行（必须）    From: alice@crepes.fr  
To首部行（必须）      To: bob@hamburger.edu  
Subject首部行（可选） Subject: Searching for the meaning of life.

- 这些首部行与SMTP命令不同
  - SMTP命令是SMTP握手协议的一部分
  - 这里的首部行是邮件报文的一部分
- 报文首部之后紧跟一个空白行，然后是ASCII格式的报文主体。

清华大学 2022秋W3

48

## 非ASCII码数据的MIME扩展

可出现在 MIME Content-Type 说明中的七种基本类型及其意义

内容类型	子类型	说明
Text(正文)	plain	无格式的文本
	richtext	有少量格式命令的文本
Image(图像)	gif	GIF 格式的静止图像
	jpeg	JPEG 格式的静止图像
Audio(音频)	basic	可听见的声音
Video(视频)	mpeg	MPEG 格式的视频
Application (应用)	octet-stream	不间断的字节序列
	postscript	PostScript 可打印文档

清华大学 2022秋W3

49

## 非ASCII码数据的MIME扩展

Message (报文)	rfc822	MIME RFC 822 邮件
	partial	为传输将邮件分割开
Multipart (多部分)	external-body	邮件必须从网上获取
	mixed	按规定顺序的几个独立部分
	alternative	不同格式的同一样件
	parallel	必须同时读取的几个部分
	digest	每一个部分是一个完整的 RFC 822 邮件

用户代理需要根据不同的Content-type调用相应的方法来正常解释和显示邮件内容

清华大学 2022秋W3

50

## 例子：MIME首部行

From: alice@crepes.fr  
To: bob@hamburger.edu  
Subject: Picture of yummy crepe.  
MIME-Version: 1.0  
Content-Transfer-Encoding: base64  
Content-Type: image/jpeg

支持多媒体的两个  
关键MIME首部

(base64 encoded data ...  
.....  
.....base64 encoded data)

用户代理根据Content-Transfer-Encoding的值将报文主体还原成非ASCII的格式，然后根据Content-Type首部行决定应当采取何种动作来处理报文主体。

清华大学 2022秋W3

51

## 接收的报文

- SMTP接收服务器一旦接收到具有RFC 822和MIME首部行的报文，就在该报文的顶部添加一个Received:首部行

Received: from hamburger.edu by sushi.jp; 3 Jul 01 15:30.01 GMT

Received: from crepes.fr by hamburger.edu; 3 Jul 01 15:17.39 GMT

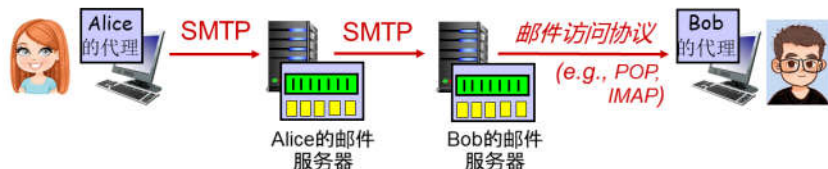
From: alice@crepes.fr  
To: bob@hamburger.edu  
Subject: Picture of yummy crepe.

...

清华大学 2022秋W3

52

## 邮件访问协议



- **SMTP**: 把邮件发送、存储到接收者的服务器
- 邮件访问协议: 从服务器提取邮件信息
  - **POP3**: 第三版邮局协议 Post Office Protocol – Version 3 [RFC 1939]
    - 授权 (agent <--> server) 和下载
  - **IMAP**: 因特网邮件访问协议 Internet Mail Access Protocol [RFC 3501]
    - 更多功能 (也更复杂)
    - 可以对服务器上存储的邮件报文进行操作
  - **HTTP**: Hotmail, Gmail, 网易邮箱 等

清华大学 2022秋W3

53

## POP3 协议

事务处理过程示例

### 特许阶段

- Client 命令
  - **user**: 声明 username
  - **pass**: password
- server 响应
  - **+OK** (有时后面跟有数据), 表示前面的命令正常
  - **-ERR**, 表示前面的命令出现差错

### 处理阶段, client:

- **list**: 列出邮件数量
- **retr**: 根据序号提取报文信息
- **dele**: 删除
- **quit**

### 更新阶段, POP3服务器

删除标记为删除的报文

```
S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on

C: list 请列出所有存储报文的长度
S: 1 498
S: 2 912
S: .
C: retr 1 取回邮件1
S: <message 1 contents>
S: .
C: dele 1 删除邮件1
C: retr 2 取回邮件2
S: <message 1 contents>
S: .
C: dele 2 删除邮件2
C: quit
S: +OK POP3 server signing off
```

清华大学 2022秋W3

54

## POP3 (续) 和 IMAP

### POP3

- 前面的例子使用了下载删除模式
- Bob如果换到另外一台计算机上将无法再阅读已经收取的e-mail
- 下载并保留: 邮件保留在服务器上, 可把邮件报文复制到多个 client上
- POP3服务器在会话期间记录哪些报文被删除, 但不在会话过程中携带状态信息, 是**无状态**的。

### IMAP [RFC 3501, IMAP 2007]

- 把所有的邮件报文放置在一个地方: 服务器
- 允许用户在文件目录中组织整理邮件报文
- **IMAP** 在会话期间保持用户的**状态**
  - 文件目录名
  - 在目录名和报文序号之间建立映射
- 允许只获取邮件报文首部
  - 低带宽, 节约流量
  - 例如: 手机email客户端

清华大学 2022秋W3

55

## 基于HTTP的邮件访问

- 更加通用, ubiquitous access
  - 只需使用浏览器, 所有数据保存在服务器
  - 客户邮件报文通过HTTP发送给邮件服务器
  - 邮件服务器之间仍使用SMTP协议
  - 如gmail, hotmail等
- 技术实现手段
  - 邮件服务器中同时运行web server, 例如apache
  - 网络编程实现web server与email server之间的数据交换
    - Java/JSP/Http网页编程, 与SMTP接口衔接

思考: 如何DIY实现自己的基于Web的邮件服务器?

清华大学 2022秋W3

56



## 提纲

- 应用层协议原理
- Web 和 HTTP
- FTP
- 电子邮件 Email
- **域名系统 DNS**
- P2P 文件共享
- 视频流和内容分发网
- Socket programming

清华大学2022秋 W4

57

## DNS, Domain Name System 域名系统: 因特网的目录服务

人: 可以有多种识别代号

- 身份证号、姓名、护照号码

□ **因特网主机、路由器:**

- IP 地址 (32 位) - 用来进行数据报寻址
- “名字”, 例如 [www.yahoo.com](http://www.yahoo.com) - 供人类使用

□ **问题:** 如何在IP地址和主机名之间建立映射?

使用CS模式在通信的端系统之间运行  
在通信的端系统之间通过下面的端到端运输层协议来传送DNS报文

**域名系统 (DNS)是**

1. 由具有分层体系结构的许多名字服务器 *name servers* 构成的分布式数据库 *distributed database*
2. **应用层协议** *application-layer protocol*: 主机和名字服务器之间相互通信来对主机名进行**解析** *resolve* (地址/名字翻译)
  - 核心的因特网功能, 实现为应用层协议
  - 网络“边缘”的复杂性

清华大学2022秋 W4

58

## DNS

**DNS 服务**

- 主机名到 IP 地址的翻译
- 主机别名 *host aliasing*
  - 规范名字 (“大名”) 与别名 (“小名”) *canonical and alias names*
- 邮件服务器别名 *Mail server aliasing*
- 负载分配
  - 互为备份的一组 Web 服务器: 为一个规范名字设置一组 IP 地址

**为什么不采用集中式的 DNS?**

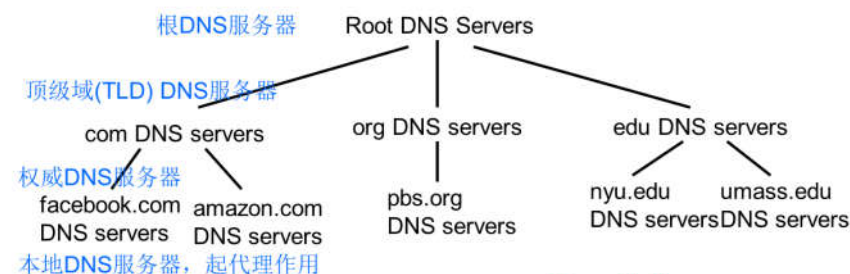
- 单点故障
- 通信容量
- 远距离的集中式数据库——严重时延
- 维护

**缺乏可扩展性!**

清华大学2022秋 W4

59

## 分布式、层次化数据库



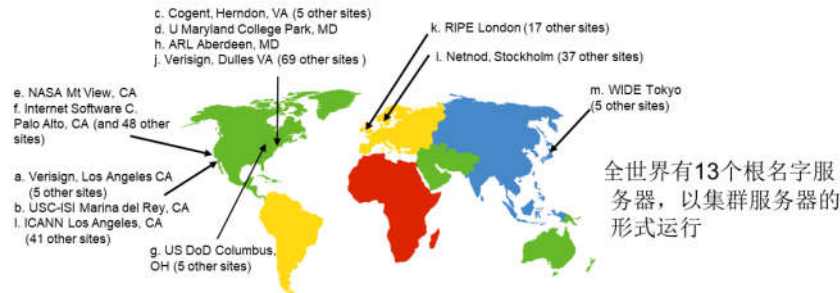
**某 Client 想知道 [www.amazon.com](http://www.amazon.com) 的 IP 地址**

- 首先询问根服务器 *root server* 查找“com”DNS 服务器
- 然后询问 com DNS 服务器来获得 *amazon.com* 的DNS 服务器
- 再询问 *amazon.com* DNS 服务器, 最终获得 [www.amazon.com](http://www.amazon.com) 的IP地址

6  
0 清华大学2022秋 W4

## 根DNS服务器

- 本地DNS服务器当发现无法解析的名字时与之联系
- 根DNS服务器
  - 如果名字映射未知则联系权威名字服务器authoritative name server
  - 获得映射信息
  - 把映射信息返回给本地名字服务器



清华大学2022秋 W4

2022年的DNS根服务器

61

## IPv6根服务器

- 亚太互联网络信息中心预测，10年内IPv4将全面退出历史舞台，互联网将全面转向IPv6
- 最后的IPv4地址储备池已于2019年11月25日完全耗尽
- 在与现有IPv4根服务器体系架构充分兼容基础上，2016年，“雪人计划”已在美国、日本、印度、俄罗斯、德国、法国等全球16个国家完成25台IPv6根服务器架设，其中1台主根和3台辅根部署在中国

“雪人计划” IPv6根服务器全球分布情况

国家	主根服务器	辅根服务器	国家	主根服务器	辅根服务器
中国	1	3	西班牙	0	1
美国	1	2	奥地利	0	1
日本	1	0	智利	0	1
印度	0	3	南非	0	1
法国	0	3	澳大利亚	0	1
德国	0	2	瑞士	0	1
俄罗斯	0	1	荷兰	0	1
意大利	0	1			

清华大学2022秋 W4

62

## 顶级域DNS服务器和权威DNS服务器

TLD and Authoritative Servers

- 顶级域DNS服务器Top-level domain (TLD) servers:
  - 负责 com, org, net, edu, gov 等和所有顶级国家域 cn, uk, fr, ca, jp.
  - Verisign Global Registry Services维护 .com TLD
  - Educause 维护 .edu TLD
- 权威DNS服务器 Authoritative DNS servers:
  - 一般是组织机构自己的DNS服务器，给该组织机构的服务器提供权威的从主机名到IP地址的映射(如Web和mail)
  - 可以由组织机构自己或服务提供商来维护

清华大学2022秋 W4

63

## 本地DNS服务器 Local DNS Server

- 严格来说不属于层次体系
- 每个ISP(住宅ISP、公司或学校)有一个
  - 也称为默认名字服务器“default name server”
- 当主机进行DNS查询时，查询先被发送到本地名字服务器
  - 实际上发挥代理(proxy)的作用，把查询转发到DNS层次体系中
  - 有name-to-address翻译的本地缓存(有可能过期!)

运行命令 `config /all` 查看自己的DHCP网络配置!

DHCP不但分配动态IP地址，同时也指定本地DNS地址，一般本地名字服务器与本机构的权威DNS服务器相同

清华大学2022秋 W4

64

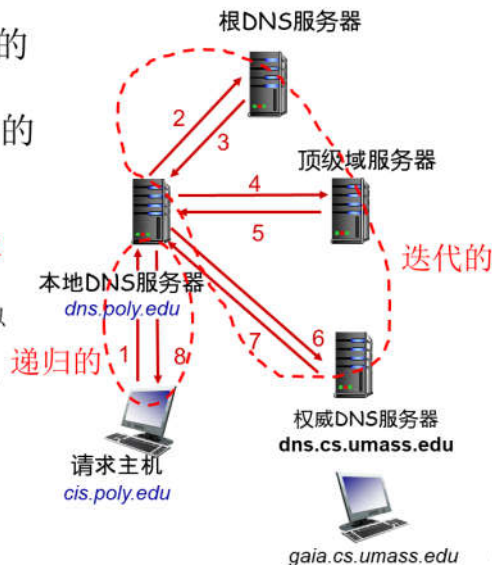


## DNS名字解析之迭代查询

- 位于 cis.poly.edu 的主机想知道 gaia.cs.umass.edu 的 IP 地址

### 迭代查询 iterative query

- 被联系的服务器返回可以联系的服务器名
- "I don't know this name, but ask this server"



清华大学2022秋 W4

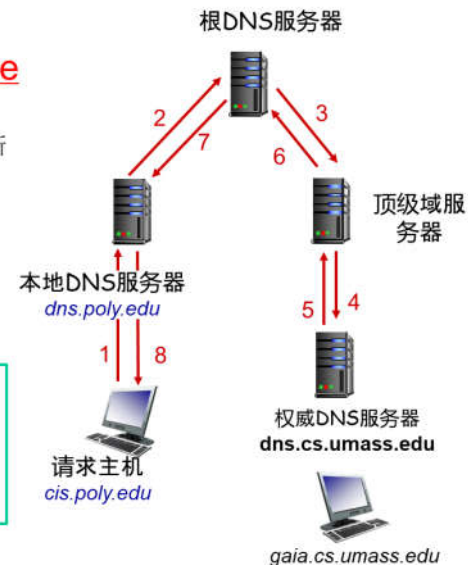
65

## DNS名字解析之递归查询

### 递归查询 recursive query

- 把名字解析的负担放到所联系的名字服务器上
- heavy load?

实际中,通常从请求主机到本地DNS服务器用递归查询,其余用迭代查询。



清华大学2022秋 W4

66

## DNS: 记录的缓存和更新

- 一旦(任何)名字服务器学习到了新的名字映射,它就把这个新的映射内容作一个缓存 **caching**
  - 经过一段时间后(TTL)缓存的内容会超时(消失)
  - 本地名字服务器通常缓存有TLD服务器的信息
    - 因而根服务器并不经常被访问
- 缓存的数据项可能会 **过期**
  - 如果被命名主机改变了其IP地址,那么全因特网可能直到所有的TTL都超时后才会知道
- IETF设计了一套更新/通知的机制
  - RFC 2136 <https://tools.ietf.org/html/rfc2136>

清华大学2022秋 W4

67

## DNS 记录

### DNS: 储存资源记录(RR)的分布式数据库

资源记录格式: (Name, Value, Type, TTL)

- Type=A
  - Name 为主机名
  - Value 为IP 地址
- Type=CNAME
  - Name 为某些“规范”名字的别名  
www.sohu.com 实际上是 fcer.a.sohu.com
  - Value 为规范名字
- Type=NS
  - Name 为域 (e.g. foo.com)
  - Value 为该域的权威DNS 服务器的主机名
- Type=MX
  - Name 为别名
  - Value 是别名为name 的邮件服务器的规范主机名
  - 可以与Web服务器具有相同别名

清华大学2022秋 W4

68