



计算机通信... 与网络

COMPUTER 华中科技大学

Communication and Network



第2章 应用层

Application Layer

.....



目 录

- 应用层协议原理
- Web和HTTP
- 因特网中的电子邮件
- DNS：因特网的目录服务
- P2P文件分发



应用层协议原理

Theory of Application Layer Protocol





常见的网络应用



网络应用就为陌生的朋友聊天



Skype



QQ
Messenger



Net2Phone



QQ

微信



360

阿里旺旺



Chrome



常见的网络应用



网络应用



FTP



魔兽世界



BT



联众游戏

联众



电驴

绝地逃生



迅雷

上甘不/唯



常见的网络应用



搜索视频



Google



暴风影音



百度



腾讯视频



Bing



优酷



常见的网络应用



这么多应用通过网络充斥在我们的生活中，那么从网络的角度，它们到底具有哪些共性呢？





什么是网络应用程序

可以向网络发送数据

可以从网络接收数据

可以对数据进行处理

也许还能够

-  将数据展现在界面上，以非常友好的方式让你知道它在做什么，免得你说它怠工。
-  时不时的弹出一个窗口，提示你不要太辛勤工作，以表示对你无微不至的关怀。



决定你的网络应用所需采用的体系结构

客户机
/服务器
体系结构
(C/S)

P2P体系
结构

混合体系
结构



客户机/服务器体系结构

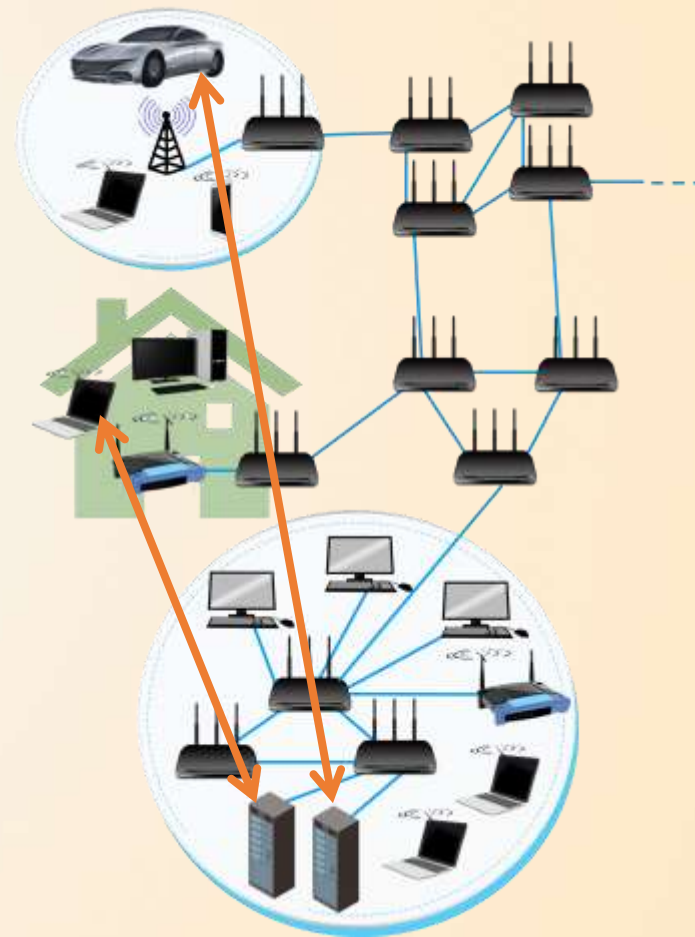
- 存在一个能够向客户机提供服务的服务器。

例如：WEB服务器。



- 存在一个或者多个主动连接服务器，试图从服务器那里获取所需服务的客户机。

例如：Edge浏览器。





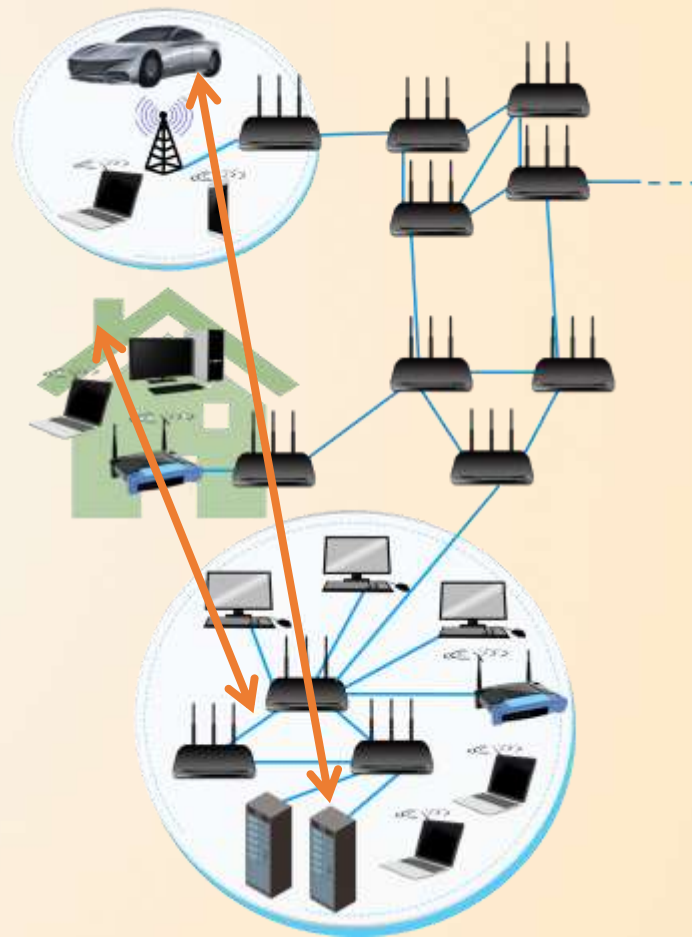
客户机/服务器体系结构

特别注意1

客户机之间不能互相通信。

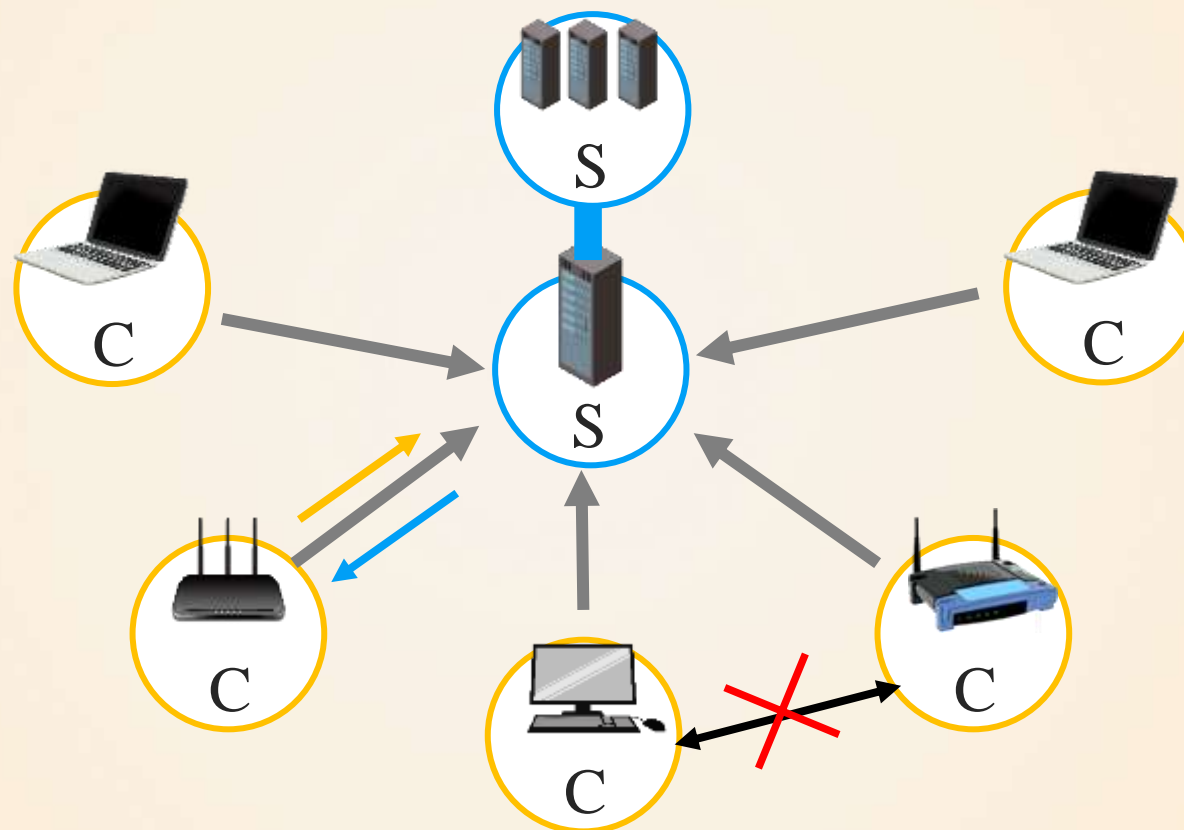
特别注意2

为提高服务器的处理能力，通常采用服务器集群（Server Farm）。





客户机/服务器体系结构





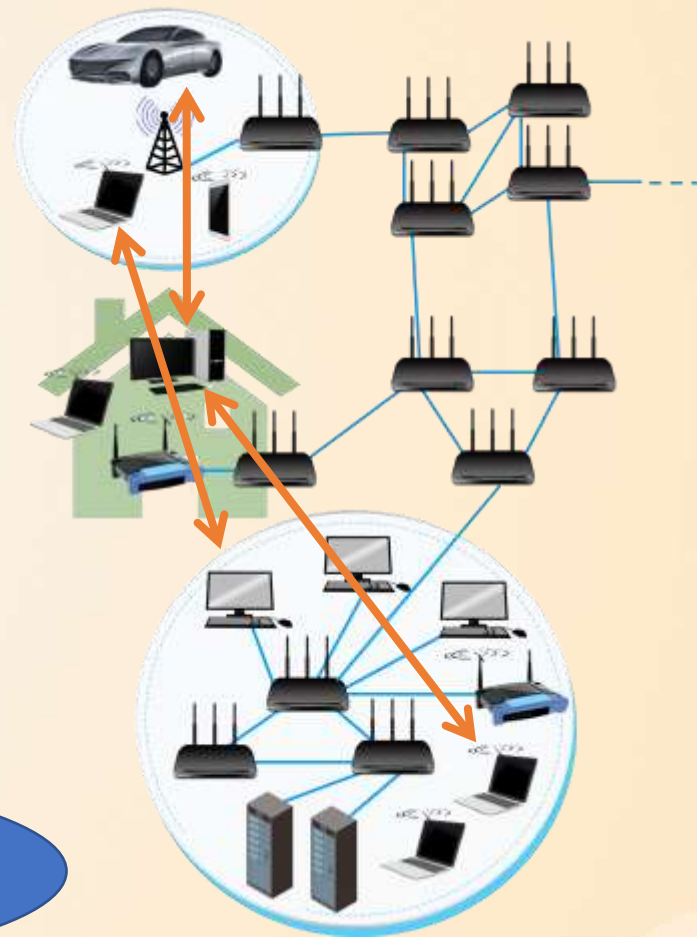
P2P体系结构

- 任何一方既提供服务又享受服务。
- 结点之间可以直接通信。
- 结点的地址以及他们之间的连接可能随时发生变化。

例如：迅雷、PPLive。

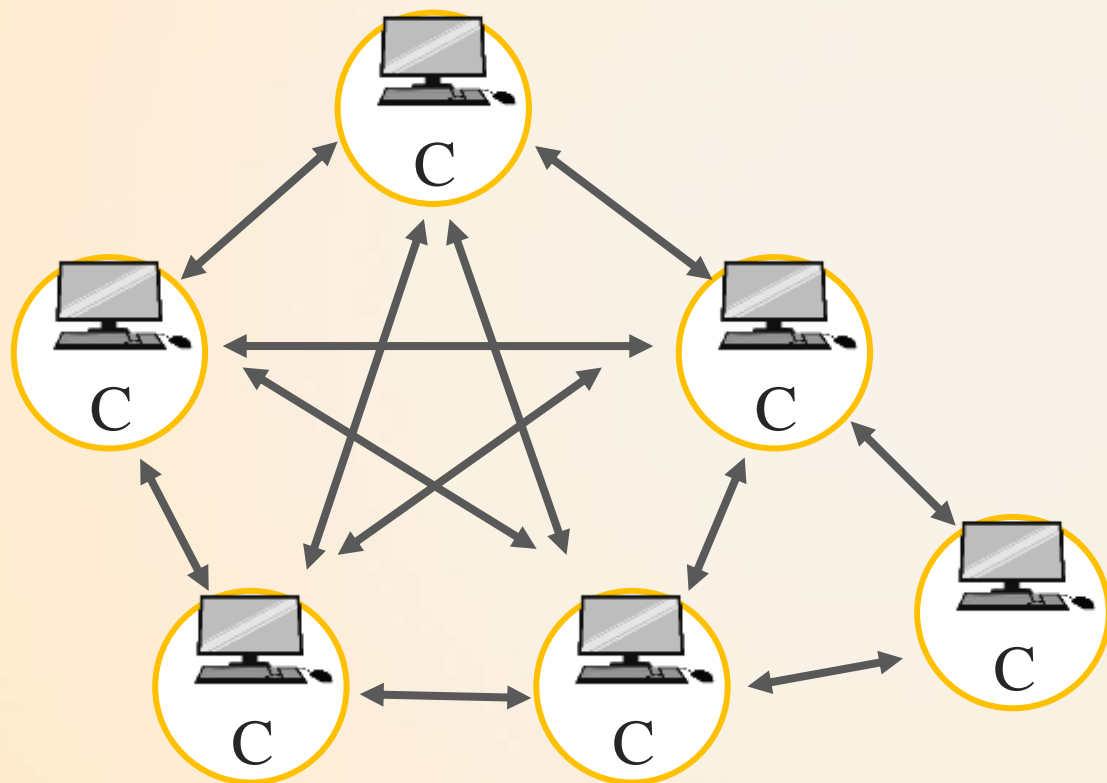


特别注意：P2P体系结构非常容易扩展，但也特别难管理





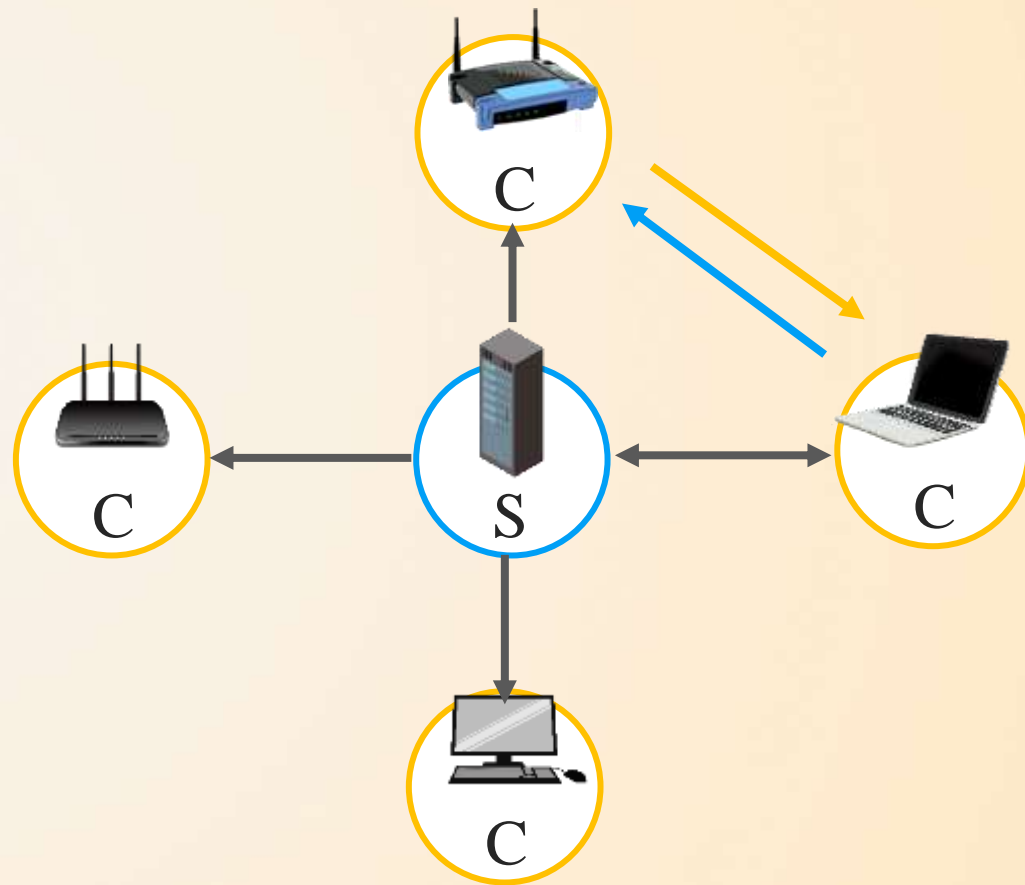
P2P体系结构





混合体系结构

那混合体系结构自然而然就是C/S体系结构和P2P体系结构的混合体喽。





网络应用会涉及到多个组成部分的交互

- 🗣️ 同一台主机上的进程之间通信的规则，由操作系统制定，和计算机网络无关，本课程就不讨论了。需要了解的，请回头看看《操作系统原理》及相关书籍。
- 🗣️ 不同主机上的进程之间通信的规则，当然就和网络相关了，这套规则在计算机网络中，称之为“**应用层协议**”，也是本章重点讨论的内容。



应用层协议定义了

- 01 交换的报文类型
- 02 各种报文类型的语法
- 03 字段的语义
- 04 进程何时、如何发送报文及对报文进行响应

应用层协议 \neq 网络应用



因特网会给网络应用提供很多不同类型的服务， 你的网络应用需要哪些服务呢？

数据的可靠传输

你的网络应用是否需要？

带宽的自动控制

你的网络应用是否带宽敏感？

传输和反馈的实时性

安全性



常见应用程序对传输服务的要求

应用程序	数据丢失	带宽	实时性
文件传输	不丢失	弹性	无
e-mail	不丢失	弹性	无
Web 网页	不丢失	弹性	无
实时音频/视频	允许丢失	音频: 5Kb-1Mb 视频: 10Kb-5Mb	100 msec
存储音频/视频	允许丢失	同上	few secs
交互式游戏	允许丢失	几 Kb/s 以上	100 msec
金融应用	不丢失	弹性	yes and no

TCP

面向

在客户端和服务端进程之间需要建立连接。

拥塞
控制

当网络超负荷时，束紧发送窗口，减缓发送速度。


可靠
传输


在发送和接收进程之间。

不提供

实时性、最小带宽承诺。
发送数据的速度决不超过接收的速度。

UDP

 在客户端和服务端进程之间实现 “不可靠的” 数据传输。

 不提供 连接建立， 可靠性保证， 流量控制， 拥塞控制， 实时性， 最小带宽承诺。



因特网常见应用采用的传输协议

应用	应用协议	所依赖的传输协议
e-mail	smtp [RFC 821]	TCP
远程终端访问	telnet [RFC 854]	TCP
Web	http [RFC 2068]	TCP
文件传输	ftp [RFC 959]	TCP
流媒体	专有协议 (e.g. RealNetworks)	TCP or UDP
远程文件服务器	NFS	TCP or UDP
IP电话	专有协议 (e.g., Vocaltec)	typically UDP



安全性

TCP/UDP天生不具备安全性

🗣️ 如果你敢把密码以明文送给TCP/UDP，TCP/UDP就敢把明文送给网络。

安全套接字层SSL

- 🗣️ 提供加密的TCP连接
- 🗣️ 数据的完整性检查
- 🗣️ 端点身份鉴别

SSL位于应用层与TCP之间



当你的网络应用程序Run起来后，就变成了网络应用进程。可能还会产生如下问题：

当你的网络应用和其它人开发的网络应用共同运行在一台主机上时，如何把不同的网络应用区分开来？

通信子网只负责把数据交付到主机，并不负责把数据交付到应用，主机如何知道数据该交付到哪个网络应用？



一个例子

你们整栋宿舍有一个信箱，栋长每天都会查看一次信箱，取走新的信件和报纸，当你有信件需要寄送时，直接投递到邮局的邮筒里。

假设

- 邮递员仅把邮件送到每栋宿舍唯一的信箱，并不负责投递到个人。
- 栋长取出新的信件和报纸后，负责将信件和报纸投递到具体的房间。



一个例子

你们整栋宿舍有一个信箱，栋长每天都会查看一次信箱，取走新的信件和报纸，当你有信件需要寄送时，直接投递到邮局的邮筒里。

问题



栋长如何区分哪封信件属于哪个房间呢？



类比到因特网，提供了类似的解决方法，那就是“套接字 (Socket)”

每个网络应用进程都有一个属于自己的套接字，该套接字在整个因特网上独一无二。

主机
地址

标识该网络应用进程运行在因特网上哪一台主机上，通常使用32位的IP地址进行标识。

端口
地址

在该主机上标识该网络应用进程，通常使用16位的端口号进行标识。

e.g., WEB Server: 80; Mail Server: 25;

所以套接字的长度为48位。

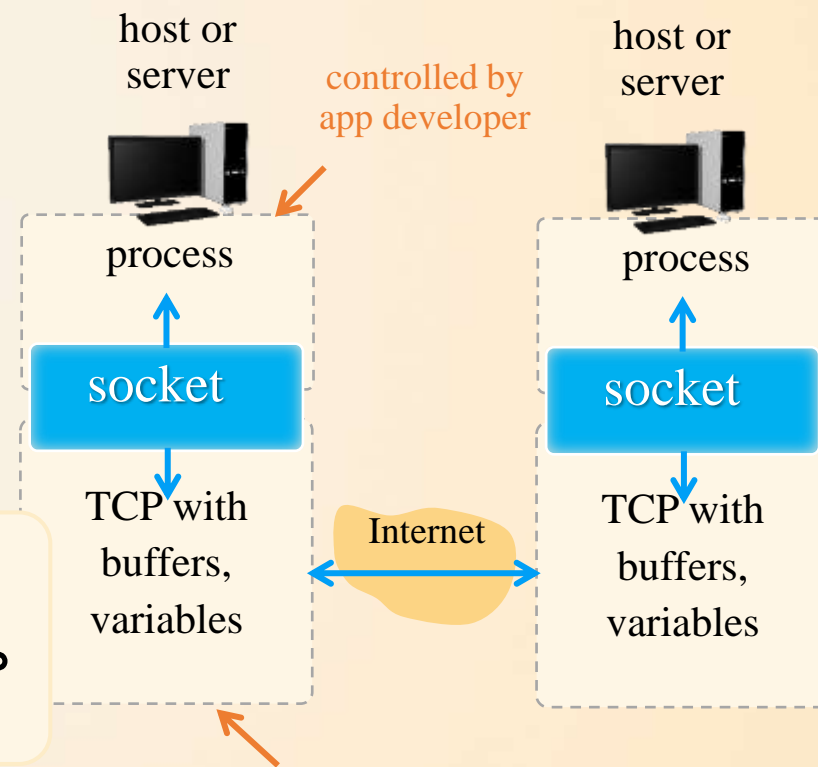


类比到因特网，提供了类似的解决方法，那就是“**套接字 (Socket)**”

- 进程通过套接字来接收和发送报文
- 套接字相当于一个通道

发送进程将报文交给套接字。

套接字将这些报文传输到接收进程的套接字。





还有.....

- 01 协议到底怎样工作
- 02 传输层的服务是如何提供的
- 03 套接字如何工作
- 04 IP地址是怎么回事
- 05 网卡和网线起了什么样的作用
- 06 如何保证网络应用的安全性和性能



本章重点讨论的网络应用

WEB

SMTP

DNS

P2P



Web和HTTP

Web and HTTP

.....



历史的回顾

- 19世纪19世纪70年代，电话的发明，扩展了人类通信的范围，增强了人类通信的实效性
- 20世纪20年代，广播收音机和电视的发明，极大的丰富了人类可获取信息
- 20世纪90年代，WEB的发明，极大的提高了人类主动获取信息的能力

广播收音机/电视机和WEB的异同点

- 都是广播和按需操作
- 你不能发布电视节目，但可以发布WEB内容



WEB的构成



WEB服务器:

IIS

Apache

TomCat.....





WEB的构成



浏览器:

IE

Maxthon

Firefox





WEB的构成



协议

信息表达的协议——HTML

信息传输的协议——HTTP

特别说明：WEB属于C/S模式





WEB内容的表达

Web 页面由一些对象组成。

- 对象可以是HTML文件、JPEG图片、音频文件、Java Applet.....
- HTML文件是Web页面的基础，它可以包括各种各样的对象，是一个容器对象。
- 任何一个对象都可以用 URL 来定位。

URL的例子

www.hust.edu.cn / cs/pic.gif
主机名 路径名



WEB内容的传输

HTTP协议



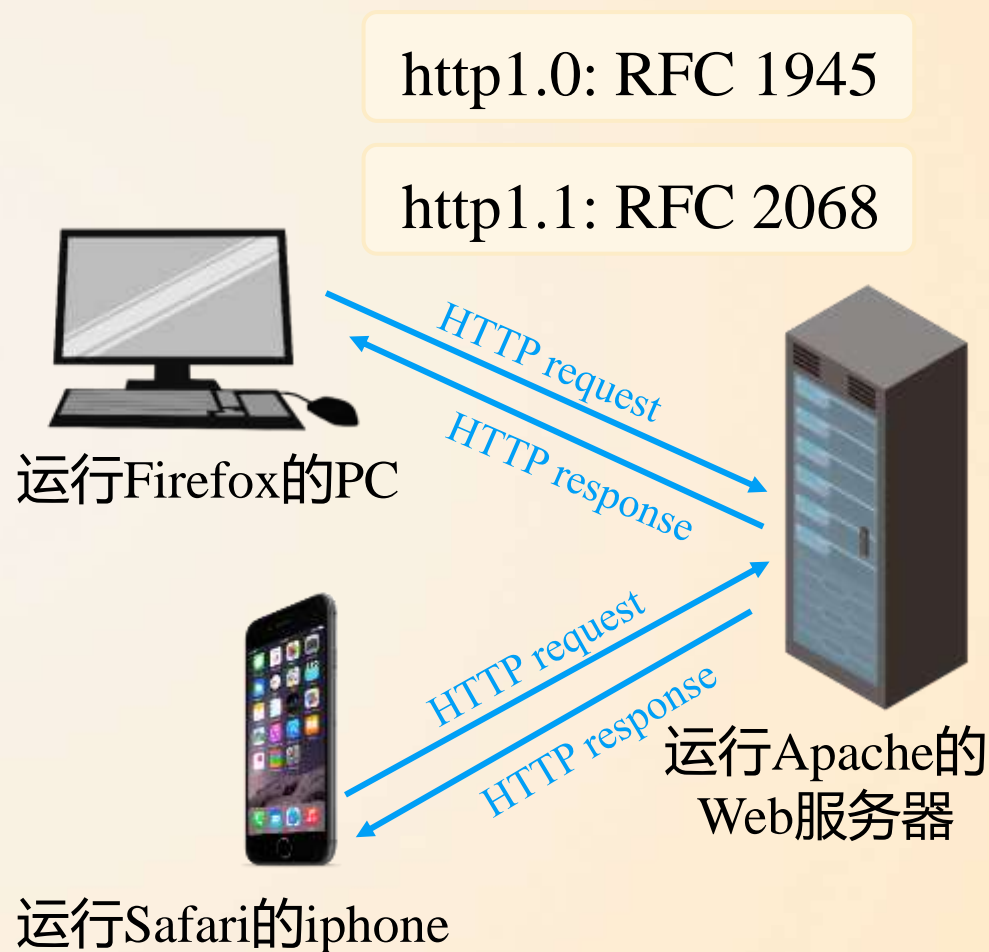
客户端/服务器模式

客户端

浏览器请求、接收、展示 Web对象（objects）

服务器





Web 服务器发送对象对请求进行响应





WEB内容的传输

http: TCP 传输服务

-  客户端启动TCP连接(创建套接字) 到服务器, 端口 80。
-  服务器接受来自客户端的 TCP 连接。
-  http 报文(应用层协议报文) 在浏览器 (http client) 和Web服务器(http server) 之间进行交换。
-  关闭TCP 连接。



HTTP1.0的传输模式

非持久性连接

假设用户键入了一个 URL
`www.hust.edu.cn/cs/home.index`

该网页包含文本并引用了10 jpeg 图片



HTTP1.0的传输模式

非持久性连接

1a. http 客户端启动 TCP 连接到 www.hust.edu.cn 上的 http 服务器（进程）。Port 80 是 http 服务器的默认端口。

2. http 客户端发送 http 请求报文（包括 URL）进入 TCP 连接插口（socket）。

1b. 在 www.hust.edu.cn 上的 http 服务器在 port 80 等待 TCP 的连接请求。“接受”连接并通知客户端。

3. http 服务器接收到请求报文，形成响应报文（包含了所请求的对象，cs/home.index），将报文送入插口（socket）。

4. http 服务器关闭 TCP 连接。

time



HTTP1.0的传输模式

非持久性连接

4. http 服务器关闭 TCP 连接。

5. http 客户端接收到了包含html文件的响应报文。
分析 html 文件, 发现 10 个引用的 jpeg 对象。

6. 对10 jpeg objects 逐个重复1-5 步。

time





非持久性连接工作机制



取对象需要2 RTTs

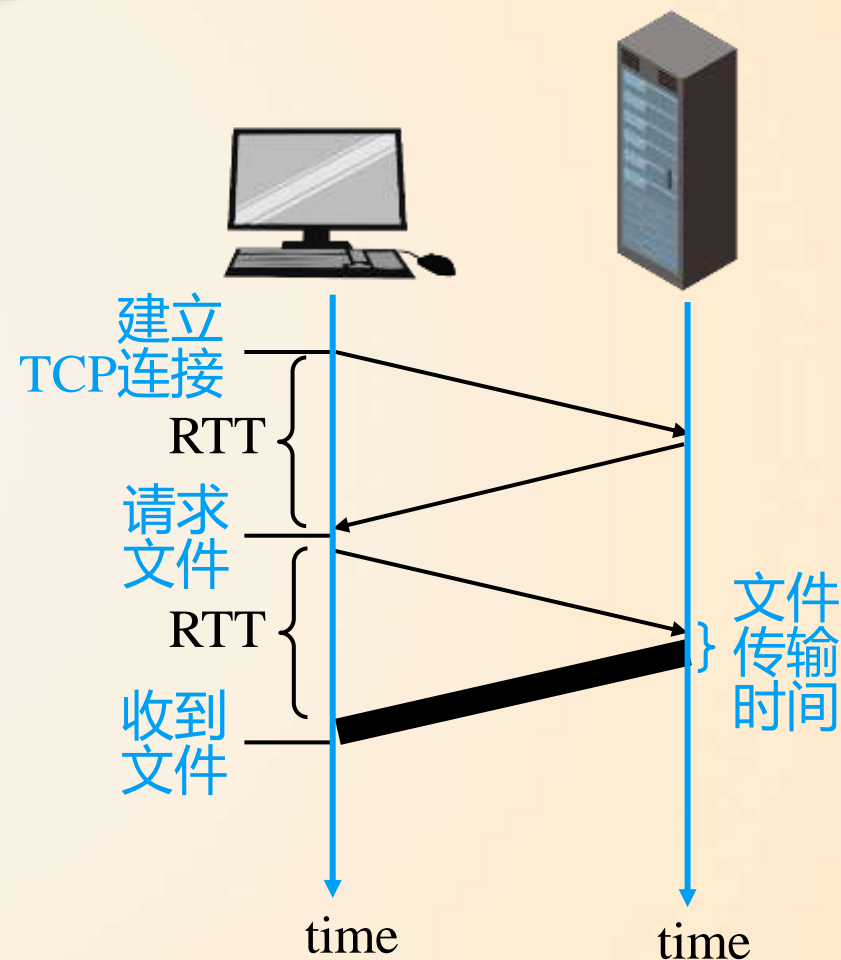
- TCP 连接
- 对象请求/传送



总时间 = $2RTT + \text{文件传输时间}$



许多浏览器同时打开多个**并行的连接**来改善性能





HTTP1.1引入的新传输模式

持久连接

服务器在发送响应后，不再断开TCP连接，而是保持该连接，用于后续对象的传送，直至该连接“休息”了一个较长的时间后，方断开该连接。

减少了对服务器端连接数的需要，从而减少了对服务器端套接字资源的占用，提高了服务器的负载能力。

持久连接又可以分为

非流水线方式 一个对象传输完成方能传输下一个

流水线方式 可以一次性发送所有请求，慢慢接收



HTTP报文类型

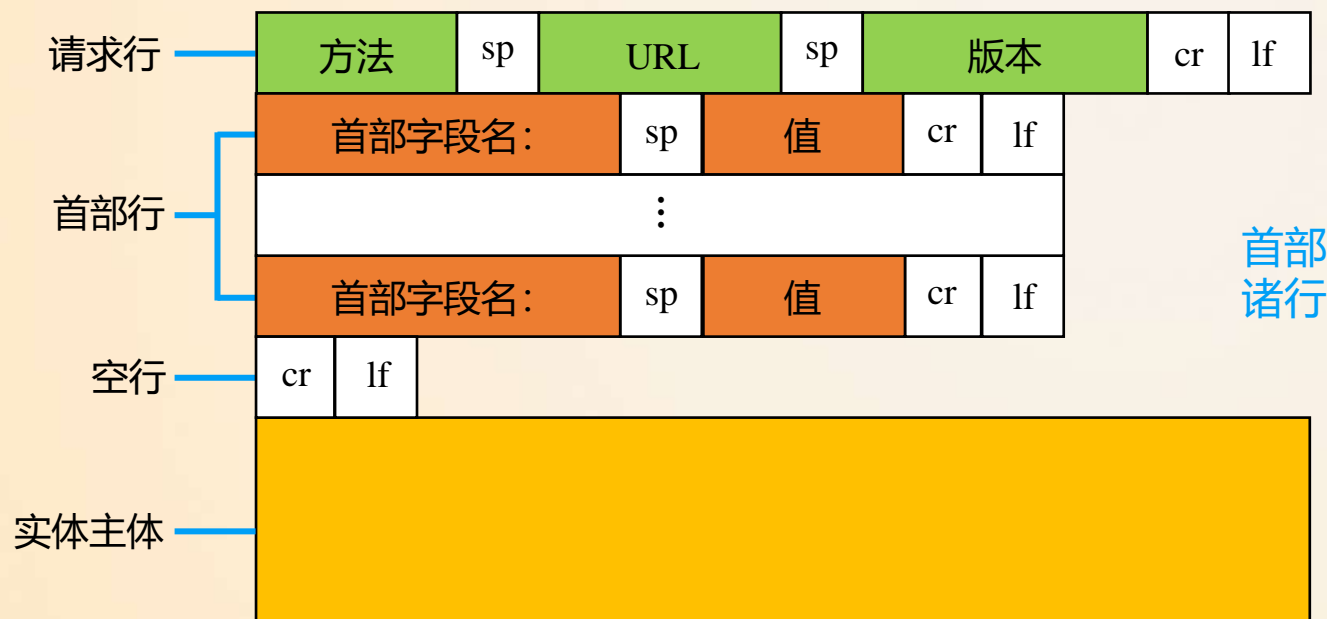
HTTP
请求报文

HTTP
响应报文



HTTP请求报文

HTTP请求报文的一般格式



一段典型的HTTP请求报文 (ASCII)

请求行
(GET, POST, HEAD 命令)

回车符
换行符

```
GET /index.html HTTP/1.1\r\nHost: www-net.cs.umass.edu\r\nUser-Agent: Firefox/3.6.10\r\nAccept: text/html,application/xhtml+xml\r\nAccept-Language: en-us,en;q=0.5\r\nAccept-Encoding: gzip,deflate\r\nAccept-Charset: ISO-8859-1,utf-8;q=0.7\r\nKeep-Alive: 115\r\nConnection: keep-alive\r\n\r\n
```

首部
诸行

单独一行回车、换行表示报文首部结束



请求行支持的方法

HTTP 定义的方法

GET

向服务器请求指定URL的对象

POST

用于向服务器提交表单数据也可以同时请求一个WEB页面

HEAD

返回响应报文，不包含请求的对象

PUT

上传的文件放在实体主体字段中，目标路径由URL字段标明

DELETE

删除URL字段中指定的文件

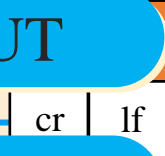
请求行



首部行



空行



实体主体



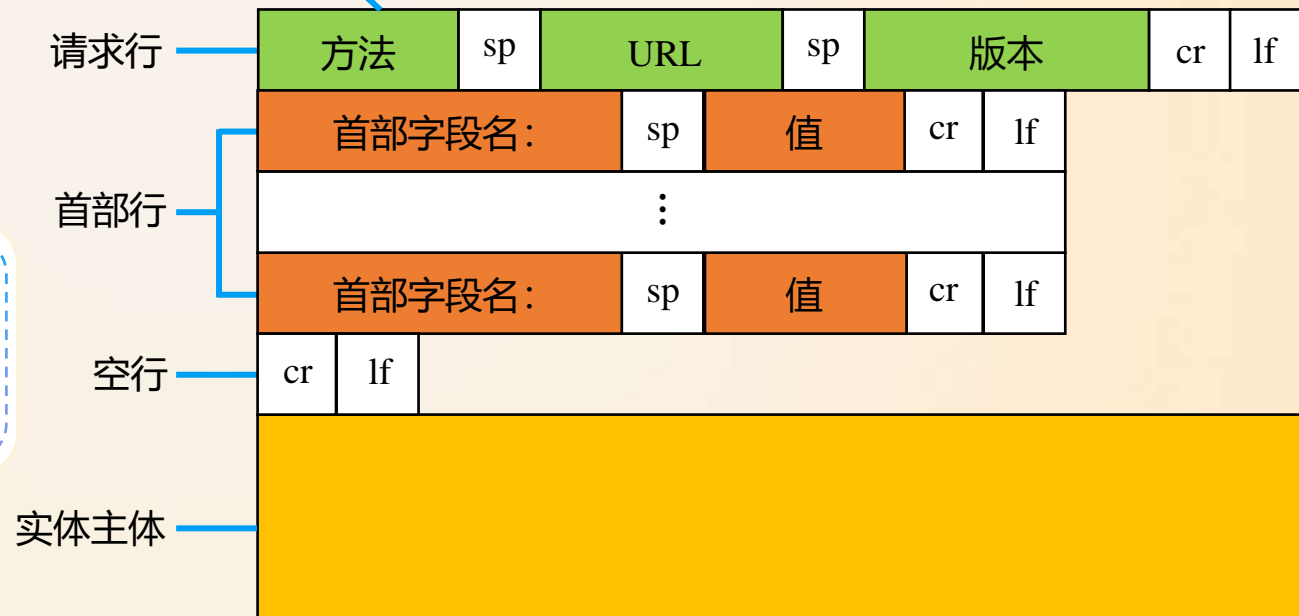


请求行支持的方法

另一种上传数据的方式

- 使用GET方法
- 将需要上传的数据放到URL中

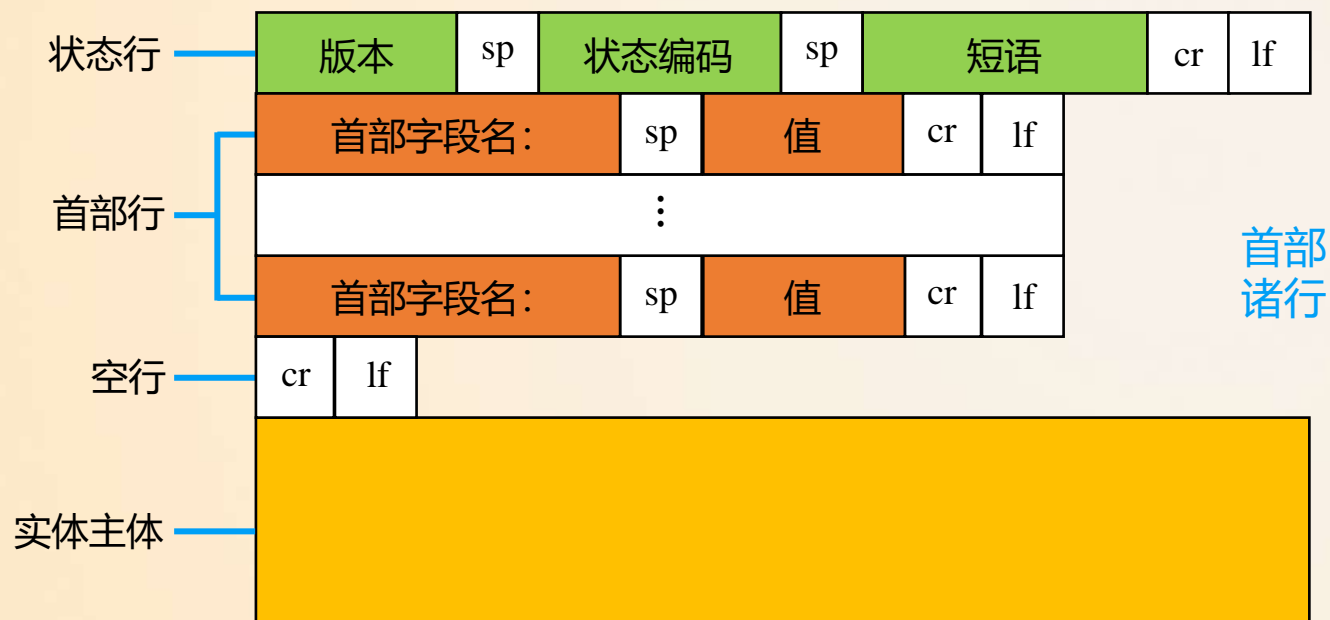
`www.somesite.com/animalsearch?`
`monkeys&banana`





HTTP响应报文

HTTP响应报文的一般格式



一段典型的HTTP响应报文 (ASCII)

状态行

(协议状态码状态短语)

HTTP/1.1 200 OK\r\n

Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n

Server: Apache/2.0.52 (CentOS)\r\n

Last-Modified: Tue, 30 Oct 2007 17:00:02 GMT\r\n

ETag: "17dc6-a5c-bf716880"\r\n

Accept-Ranges: bytes\r\n

Content-Length: 2652\r\n

Keep-Alive: timeout=10, max=100\r\n

Connection: Keep-Alive\r\n

Content-Type: text/html; charset=ISO-8859-1\r\n

\r\n

data data data data data ...

数据, e.g., 被请求的html文件



常见的HTTP响应状态码和短语

200 OK

请求成功, 被请求的对象在报文中

301 Moved Permanently

被请求的对象被移动过, 新的位置在报文中说明



400 Bad Request

服务器不懂请求报文



404 Not Found

服务器上找不到请求的对象

505 HTTP Version
Not Supported

服务器不支持请求报文使用的HTTP协议版本



用户-服务器交互：Cookie



WEB站点使用Cookie的目的

- 限制用户的访问
- 把内容和用户身份关联起来

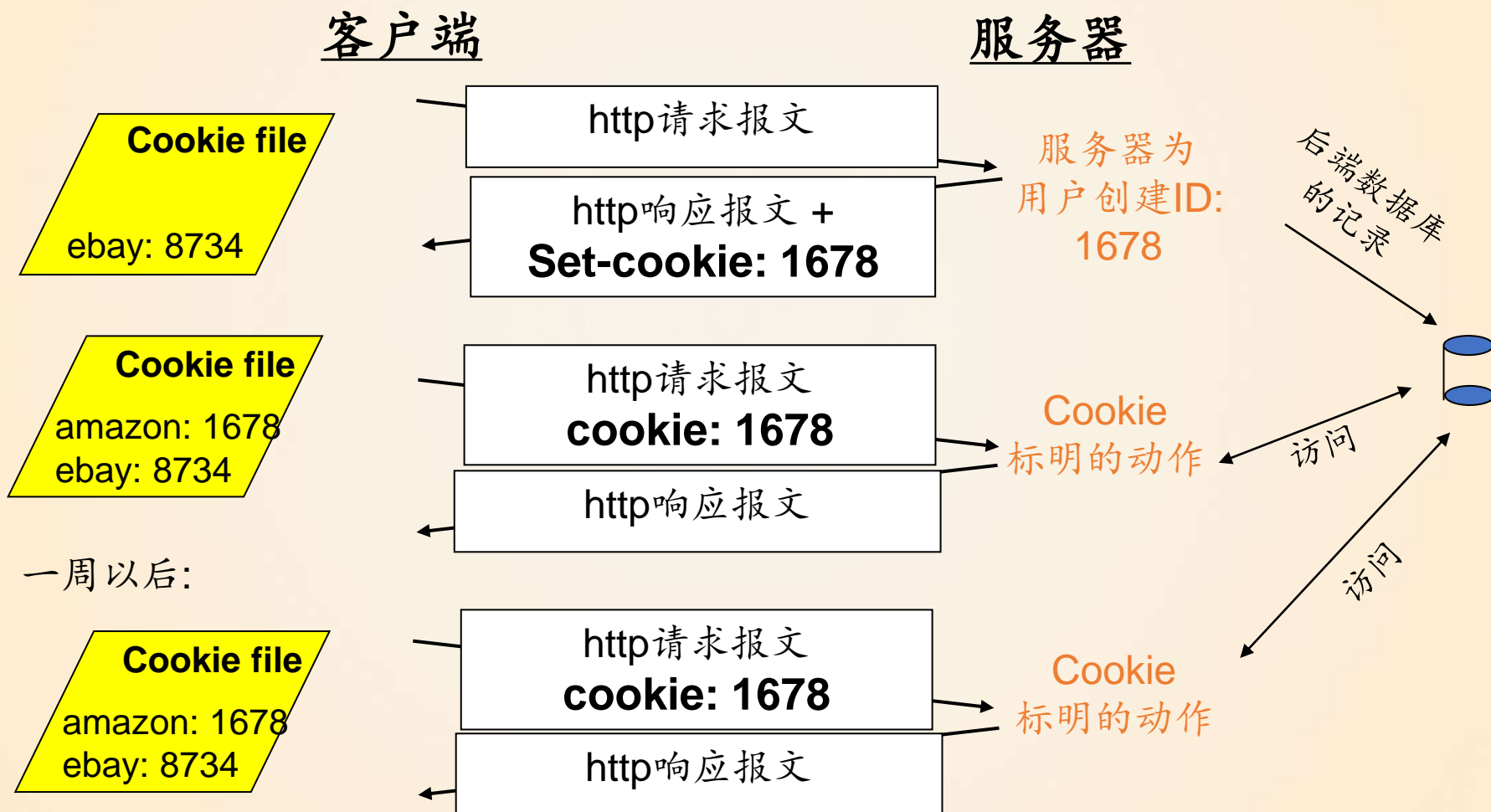


Cookie技术的组成部分

- 在HTTP响应报文中有一个Cookie首部行
- 在HTTP请求报文中也有一个Cookie首部行
- 在用户的端系统中保留了一个Cookie文件，由用户浏览器负责管理
- 在Web站点有一个后端数据库



Cookie工作流程





用户-服务器交互：Cookie



Cookies能给我们带来什么好处呢？

认证 “购物车” “推荐” 用户会话状态
(Web e-mail)

Cookies和私密性

- Cookies允许网站获得相当多的用户的信息
- 你可能会向网站提供你的姓名和E-Mail地址
- 搜索引擎也可以使用cookie和重定向技术获得很多的信息
- 广告公司也可以通过用户访问过的网站来获得用户的相关信息



因特网中的电子邮件

E-mail in the Internet

.....

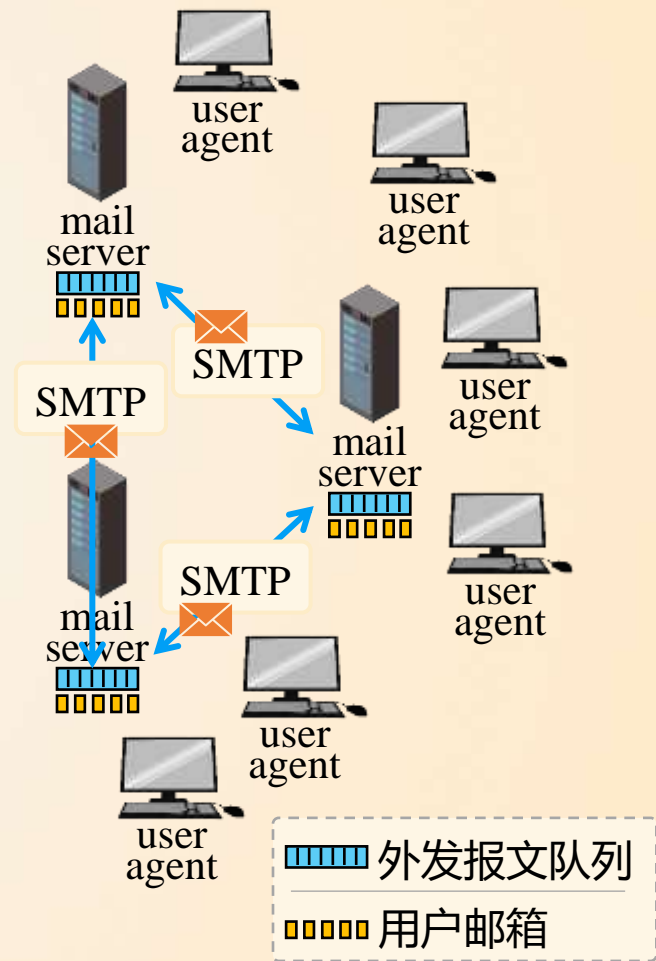


电子邮件系统的构成

用户代理

邮件服务器

简单邮件传输协议: SMTP





电子邮件系统的构成



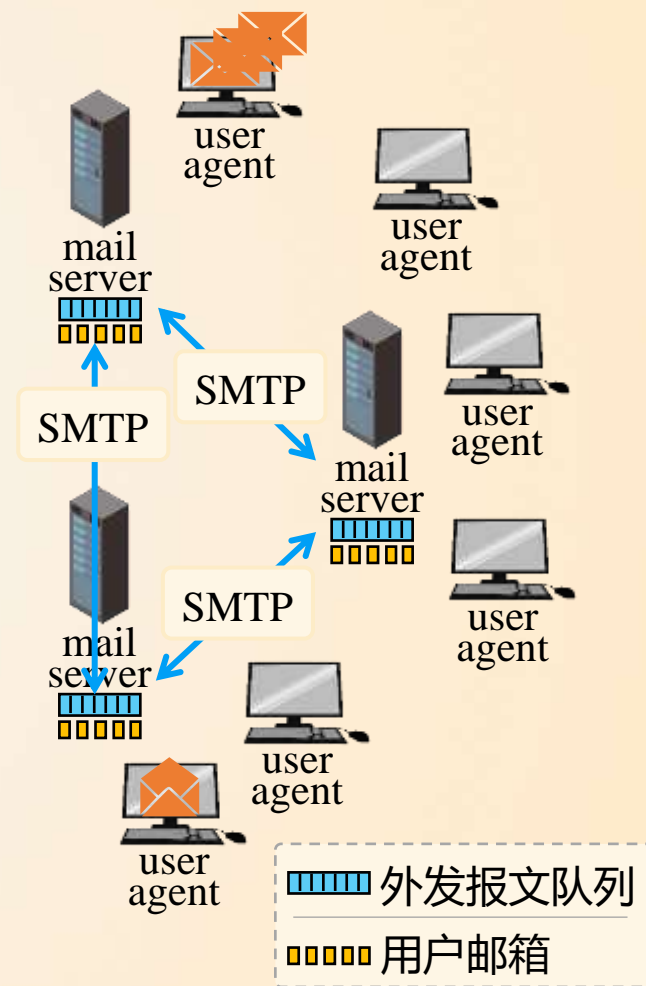
用户代理

- 写作, 编辑, 阅读邮件报文
- e.g. OE、FoxMail



邮件服务器

- 邮箱 包含了收到的用户邮件 (尚未被阅读)
- 报文 队列包含了外发的 邮件报文



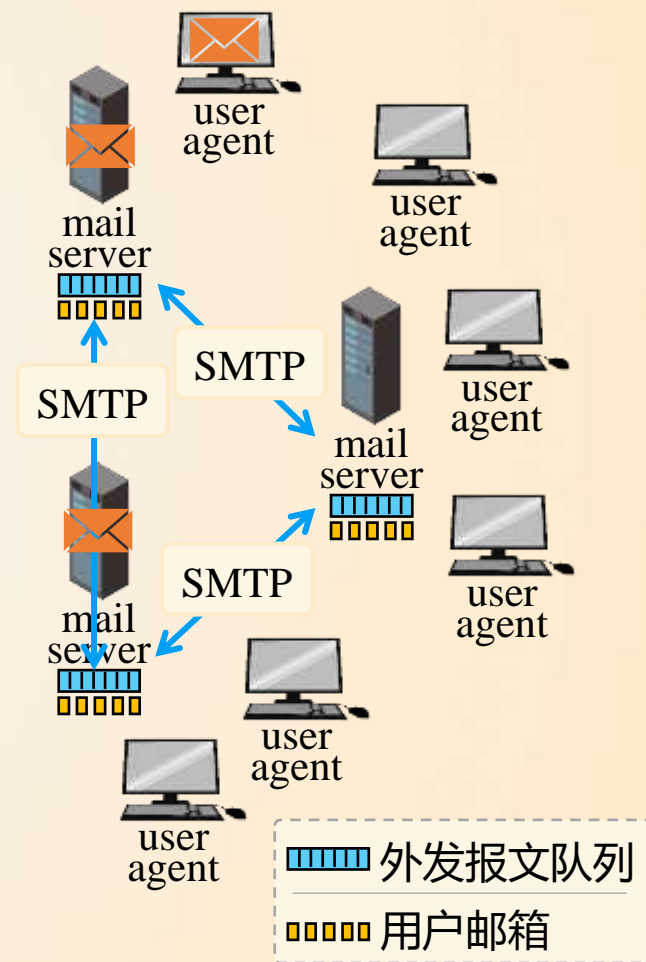


电子邮件系统的构成



SMTP 协议用在邮件服务器之间发送邮件

- 客户端: 将邮件发送到邮件服务器。
- 服务器: 接收和转发邮件。





SMTP协议

使用 TCP可靠的传送邮件报文，端口25

直接传输: 发送服务器到接收服务器

传输的三个阶段

- 握手（打招呼）
- 报文传输
- 结束

命令/响应交互

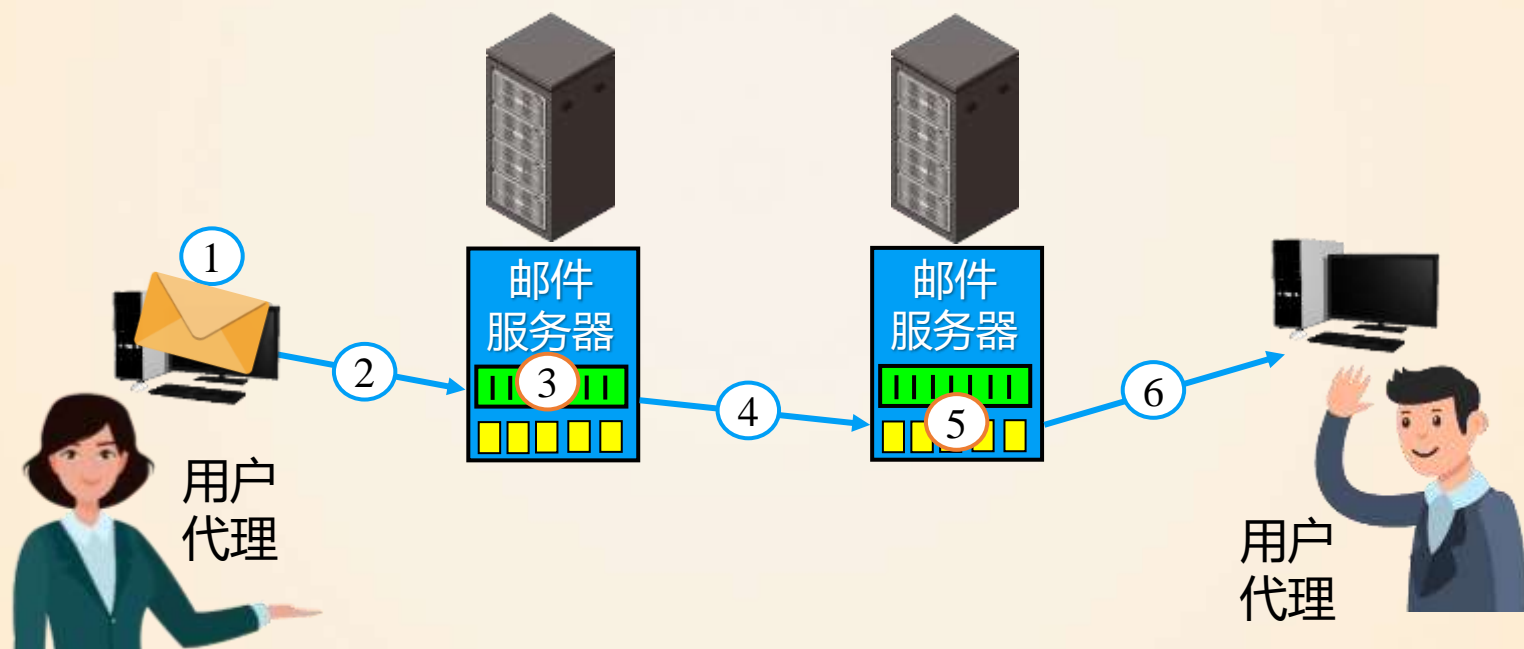
- 命令: ASCII文本
- 响应: 状态码和短语

邮件报文必须使用7-bit ASCII表示



SMTP协议

一次邮件传送过程





SMTP评述

一次邮件传送过程

- SMTP使用持续连接
- SMTP要求报文 (首部 & 信体) 全部使用 7-bit ASCII码
- 某些代码组合不允许出现在报文中 (e.g., CRLF.CRLF). 此类数据必须进行编码
(通常使用 base-64 或 quoted printable)
- SMTP服务器用 CRLF.CRLF 表示邮件报文的结束



SMTP vs HTTP

- ① 都使用 ASCII 命令/响应交互，状态码
- ② HTTP: pull (拉) and SMTP: push (推)
- ③ HTTP协议不限制报文编码格式，SMTP协议要求报文必须使用7bit ASCII 码格式
- ④ HTTP: 每个对象分装在各自己的响应报文中
SMTP: 多个对象在一个多分部的报文中传送



邮件报文格式 (RFC 822)

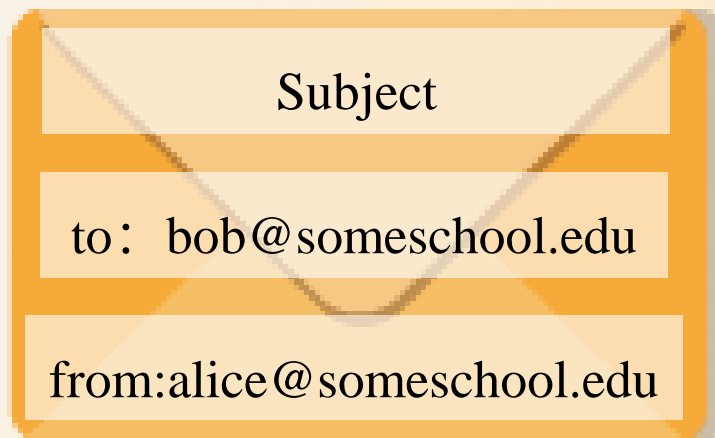
首部诸行, e.g.

- To:
- From:
- Subject:

不同于smtp 命令!

信体

即 “报文” , 7bit ASCII characters
only





邮件报文格式 (RFC 822)

首部诸行, e.g.

- To:
- From:
- Subject:

不同于smtp 命令!

信体

即 “报文” , ASCII characters only





邮件报文格式 (RFC 822)

首部诸行, e.g.

- To:
- From:
- Subject:

不同于smtp 命令!

信体

即 “报文” , ASCII characters only





非ASCII码数据的MIME扩展

MIME 版本

数据编码方法

**多媒体类型, 子类型,
参数声明**

编码后的数据

```
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

base64 encoded data .....
.....base64 encoded data
```



客户机获取邮件的方法

POP3协议

IMAP协议

HTTP



POP3协议的认证阶段



客户端命令

user: 用户名
pass: 口令

+OK
-ERR



服务器响应

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



POP3协议的交互命令

list

列出报文号码

retr

用报文号码取信

dele

用报文号码删信

quit

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

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```



POP3评述

“下载-删除”

用户如果更换客户机无法再次阅读原来的邮件。

“下载-保存”

在不同的客户机上保存邮件的副本。

POP3会话是没有状态的

用户使用POP3协议无法在邮件服务器上对自己的邮件进行重组织，只能将邮件下载到本地计算机进行重组织。



IMAP协议

- ① 将所有邮件都保存在服务器上
- ② 允许用户在服务器上组织自己的邮件目录
- ③ IMAP维护了IMAP会话的用户信息:

目录名以及报文ID与目录名之间的映射关系。



DNS：因特网的目录服务

DNS:Directory Service in the Internet

.....



DNS的由来





DNS简况

- 🗣️ DNS是一个分布式数据库，由很多DNS服务器按照层次结构组织起来
- 🗣️ DNS运行在端到端系统上，且使用UDP协议（53号端口）进行报文传输，因此DNS是应用层协议
- 🗣️ DNS以C/S的模式工作
- 🗣️ DNS不直接和用户打交道，而是因特网的核心功能



一次最简单的DNS解析过程

- 在浏览器中输入 www.hust.edu.cn/index.html 链接，从该链接中取出 www.hust.edu.cn 部分，发送给DNS客户机
- DNS客户机向DNS服务器发送包含域名 www.hust.edu.cn 的查询请求报文
- DNS服务器向DNS客户机返回一个包含对应IP地址（202.114.0.245）的响应报文
- DNS客户机将获得的IP地址传送给浏览器
- 浏览器向IP地址所在WEB服务器发起TCP链接



DNS的实现



最简单的方法——单台DNS服务器

- 单点故障的问题：一旦崩溃，因特网何以堪
- 数据的流通量：使得DNS服务器不堪重负
- 远程的集中式数据库：带来严重的延时
- 维护量巨大：DNS服务器不得不拼命的更新以适应因特网上主机的增加与减少

显然，这种方法是世界上最笨的方法！！！！

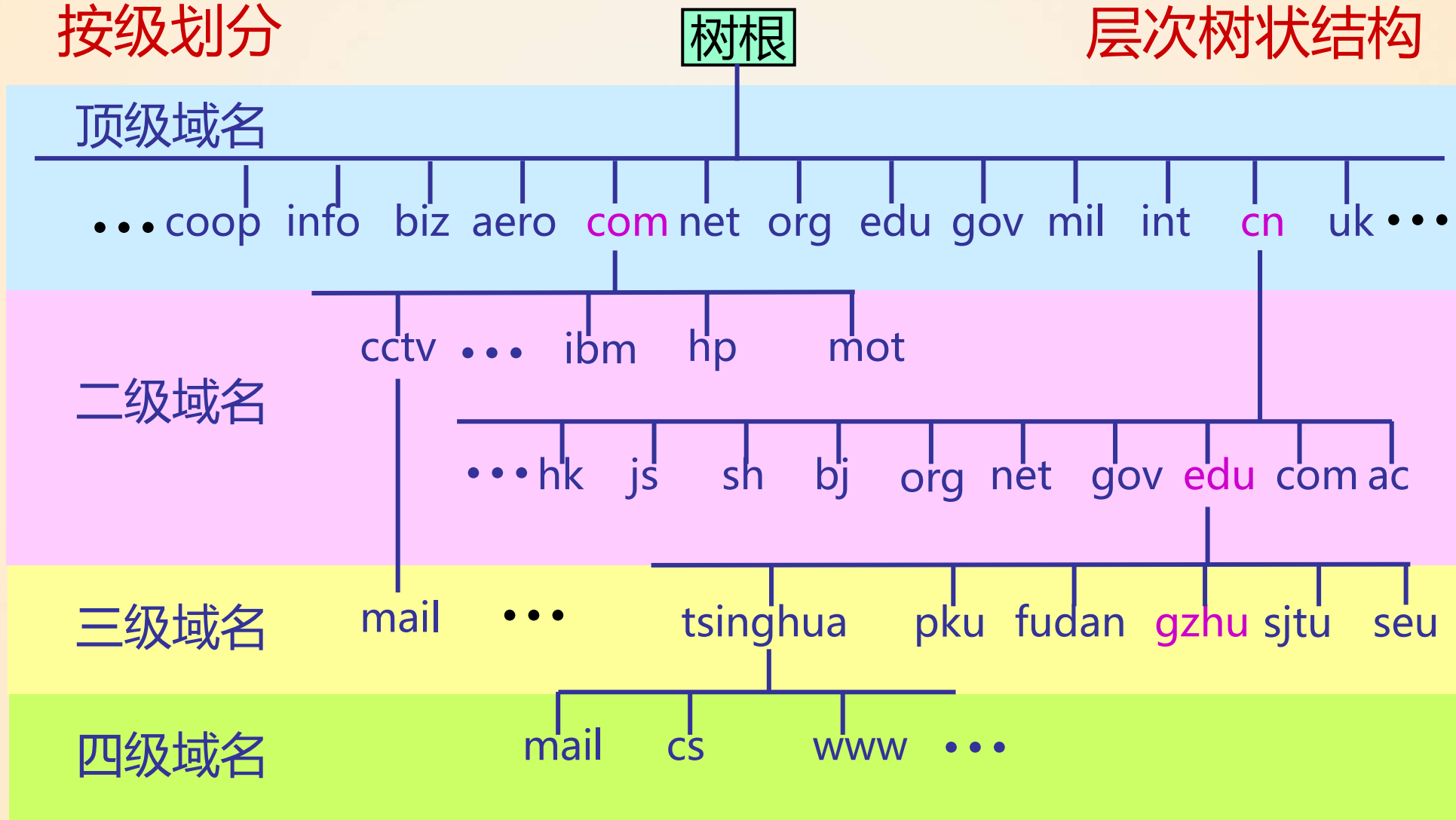


域名系统名字空间和层次结构

按级划分

层次树状结构

域
树





域名系统名字空间和层次结构



Internet的域名结构

- Internet的域名结构采用了层次树状结构的命名方法
- 域名的结构由若干个分量组成，各分量之间用小数点(.)隔开，总长不超过255个字符
- 各分量分别代表不同级别的域名。(≤63字符)
- 合法域名中，点“.”的个数至少为一个
- 通常，点“.”对应的英文单词为dot，也可以读为point

... 三级域名.二级域名.顶级域名



域名系统名字空间和层次结构



顶级域名TLD (Top Level Domain) 一般有三类

- 国家或地区顶级域nTLD，也记为ccTLD (cc: country code)

- 例如.cn 表示中国，.us 表示美国，.uk 表示英国。目前有300多个

- 基础设施域.arpa (Address and Routing Parameter Area)

- 专用于Internet基础设施目的

- 目前有二级域ip6.arpa; iris.arpa; in-addr.arpa; uri.arpa; urn.arpa; home.arpa; as112.arpa; in-addr-servers.arpa; ipv4only.arpa等

- 通用顶级域gTLD

- 早期规定了20个通用顶级域名，2011年批准新通用顶级域名(New Generic Top-level Domain, New gTLD)

- 截至2020年，已注册有1200多个通用顶级域名



域名系统名字空间和层次结构



早期的通用顶级域名

- .com 表示公司企业 (commerce)
- .net 表示网络服务机构, 例如NIC和NOC (network)
- .org 表示非赢利性组织 (organization)
- .edu 表示教育机构(美国专用) (education)
- .gov 表示政府部门(美国专用) (government)
- .mil 表示军事部门(美国专用) (military)
- .int表示政府间国际合约建立的国际性组织 (international)
- .mobi 用于提供移动产品和服务的用户和供应商 (mobile)



域名系统名字空间和层次结构



早期的通用顶级域名

- .aero 用于航空运输企业
- .biz 用于商业 (business)
- .cat 用于(西班牙)加泰罗尼亚语言和文化团体
- .coop 用于合作团体
- .info 适用于各种情况
- .jobs 用于人力资源管理者



域名系统名字空间和层次结构



早期的通用顶级域名

- .museum 用于博物馆
- .name 用于个人
- .pro 用于有资质的专业人员及其实体，例如会计、律师和医师等自由职业者
- .tel 用于商业和个人公布其联系方式
- .travel 用于旅游业实体
- 2012年开始，公司名可以作为新的顶级域名(18万美元)
- 也可以注册 **.中国、.公司、.网络** 等顶级域名(多语种域名国际标准 RFC3454、RFC3490、RFC3491、RFC3492)



域名系统名字空间和层次结构



国家顶级域名 .cn 下的二级域名分为三类

■ 类别域名 7 个

□ .edu.cn 教育

□ .com.cn 工商金融等企业

□ .gov.cn 政府

□ .ac.cn 科研

□ .org.cn 非营利组织

□ .mil.cn 国防机构

□ .net.cn 网络服务

■ 行政区域名 34 个：省、直辖市、自治区、特区等行政区域名，每个行政区域名为两个字母，例如北京bj、河北he等

■ 无类别域名：例如 www.google.cn、www.tianya.cn 等



域名系统名字空间和层次结构



域名的管理

■ 域名管理机构分级负责域名注册

□ Internet的域名管理机构： ICANN (Internet Corporation for Assigned Names and Numbers)
www.icann.org

□ ccTLD下的二级域名该国自行确定

□ 三级域名注册由其所属二级域名机构负责，以此类推

■ .edu.cn下三级域名注册由CERNET负责

■ 我国的其它二级域名注册由中国互联网络信息中心(CNNIC)负责

申请注册域名



注册服务商



注册管理机构



ICANN

注册管理机构与注册服务商之间的关系

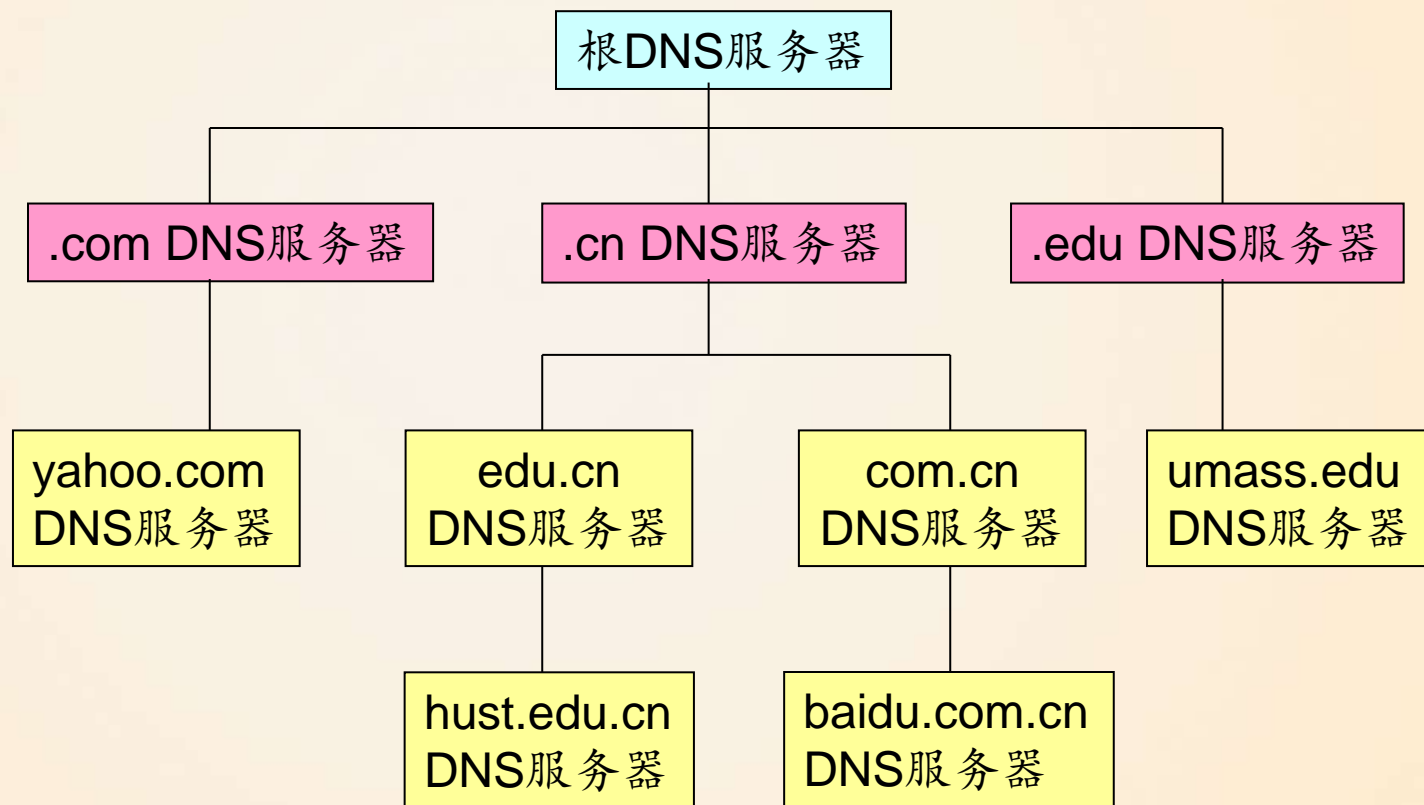


真正DNS的实现

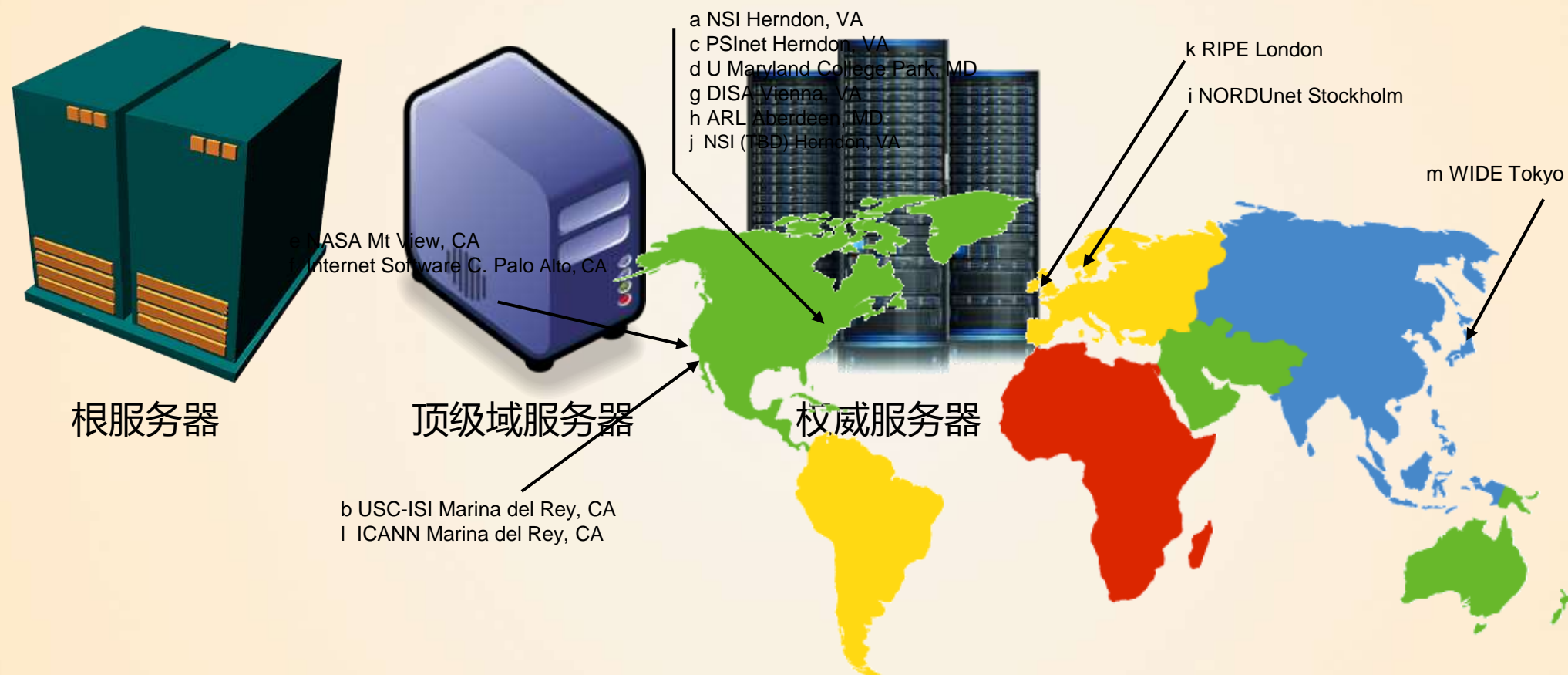
根DNS服务器

顶级域 (TLD)
服务器

权威DNS
服务器



13个



顶级域DNS服务器

负责顶级域名和所有国家的顶级域名解析工作，
例如：com, org, net, gov, uk, cn, jp等



VeriSign负责维护com顶级域DNS服务器



EDUCAUSE负责维护edu顶级域DNS服务器

权威DNS服务器

属于某个组织的DNS服务器, 为组织的服务器提供一个权威的域名到IP地址的映射服务 (例如: Web 和 mail)



这些DNS服务器一般由所属组织或者服务提供商负责维护



本地DNS服务器



每一个网络服务提供商都会提供一个本地DNS服务器

有时候，我们将其称为 “默认DNS服务器”



当一台主机需要做一个域名查询的时候，查询请求首先被发送到本地域名服务器

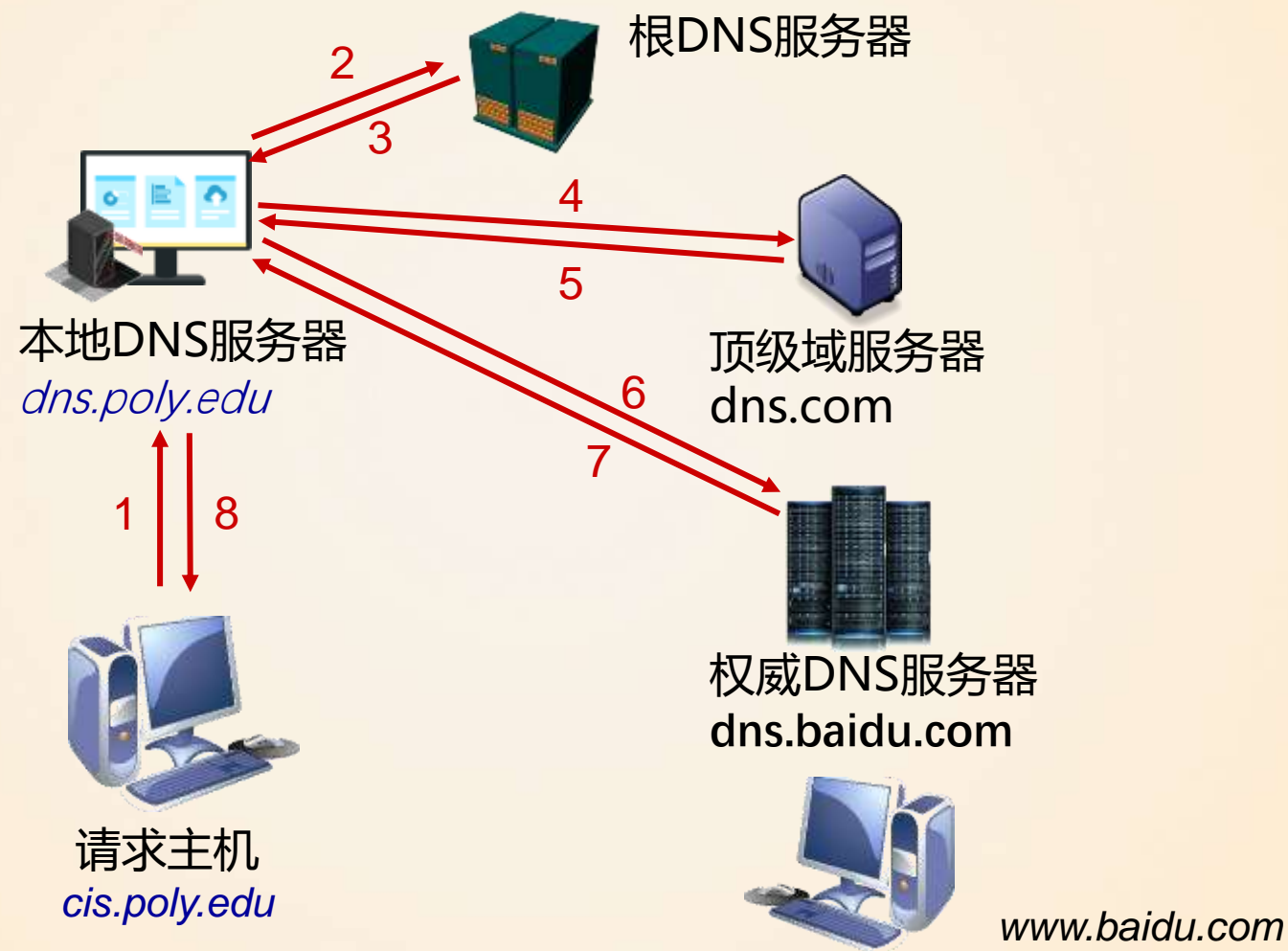
本地域名服务器的行为就像一个代理，它会向域名的层次体系中进行进一步的域名查询。



严格的讲，本地DNS服务器其并不属于DNS层次结构中的一层



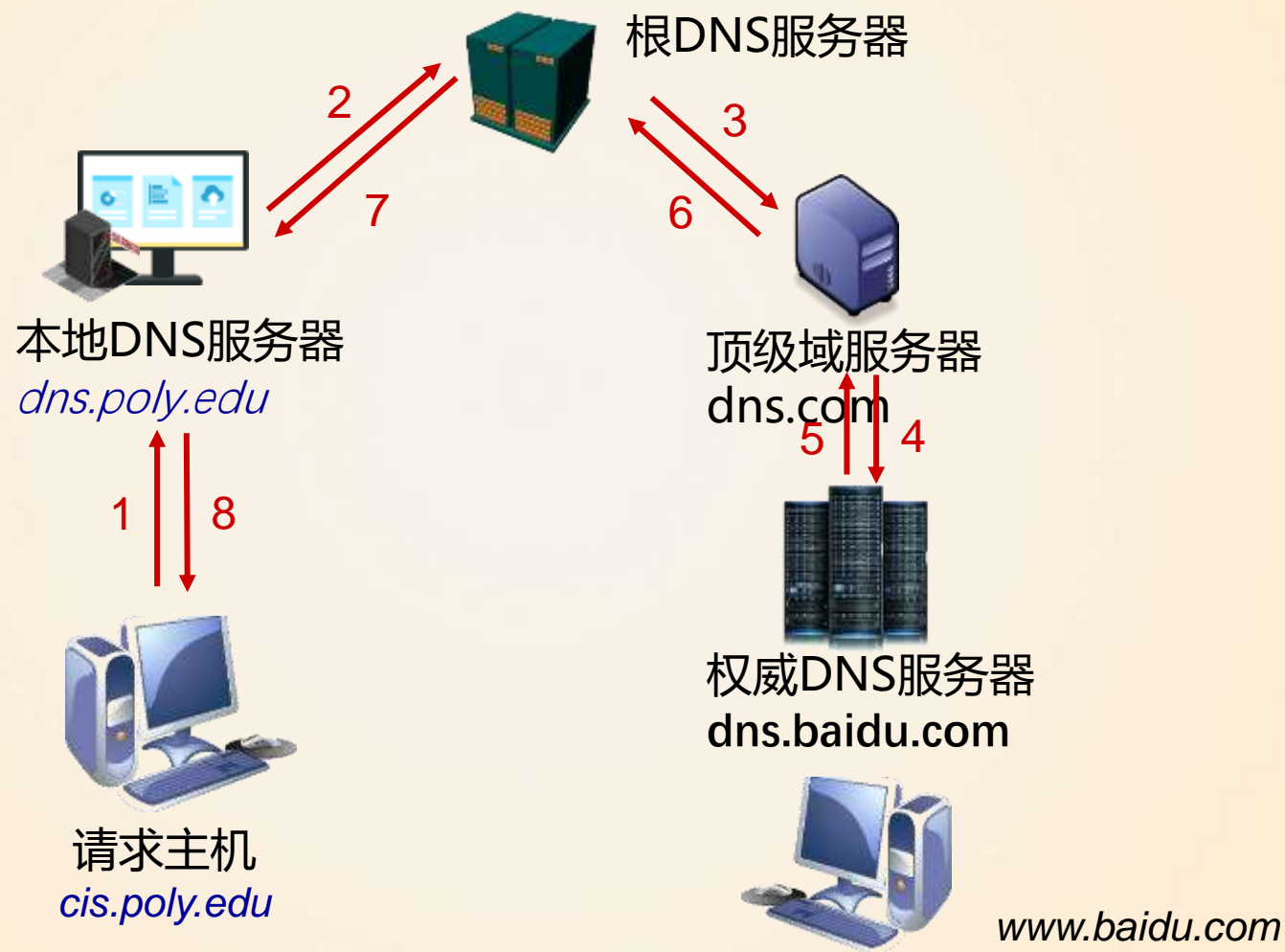
DNS解析过程——递归+迭代





DNS解析过程——纯递

归





DNS缓存



一旦 (任何) 域名服务器得知了某个映射, 就将其 **缓存**

在一定的时间间隔后缓存的条目将会过期(自动消除)

TLD DNS服务器通常被缓存在本地DNS服务器
中

这样可以减少根DNS的负载



DNS可提供的服务

域名到IP地址的转换

主机/邮件服务器别名



为不好记的规范主机/邮件服务器名提供一个易记的别名

e.g. `www.hotmail.com` -> `www.hotmail.aate.nsatc.net`



DNS可提供的服务

负载均衡



一个域名对应多个IP



DNS服务器在多个IP中进行轮转



DNS记录的格式

RR 格式: (name, value, type, ttl)

■ Type=A

- name = 主机名
- value = IP 地址

■ Type=NS

- name = 域 (如foo.com)
- value = 该域权威域名服务器的 **主机名**

■ Type=CNAME

- Name= 别名
www.ibm.com
- value = 真名
servereast.backup2.ibm.com


■ Type=MX

- value = 与 name 相关的邮件服务器域名




攻击DNS服务器

DDoS攻击

 通过ICMP Ping洪泛攻击根DNS服务器——难以成功

根服务器通常配备分组过滤器

大多数本地DNS服务器缓存了TLD DNS服务器的地址



 通过DNS请求报文洪泛攻击TLD DNS服务器

难以过滤



攻击DNS服务器

重定向攻击

-  中间人攻击——攻击者截获来自主机的请求并返回伪造的回答。
-  DNS毒害攻击——攻击者向一台DNS服务器发送伪造的回答，诱使服务器在它的缓存中接收伪造的记录。

利用DNS服务器对目标主机采用DDoS攻击——反射攻击



P2P文件分发





File Distribution in the P2P





P2P传输





一次传输的场景

-  Alice在她的笔记本电脑上运行了一个P2P客户端应用
-  她不定期的连接到因特网上，每次都获得一个不同的IP地址
-  她希望获得这样的资源 “Hey Jude”
-  这个应用显示出拥有这个资源的所有其它计算机



P2P传输

一次传输的场景

-  Alice从中选择了Bob
-  这个资源从Bob的PC拷贝到了Alice的笔记本电脑上
-  当Alice在下载资源的时候，其它用户也在向Alice的机器上传其它资源
-  Alice的计算机既是一个客户机，也是一个服务器。

所有的计算机都是服务器 = 高可扩展性



文件分发



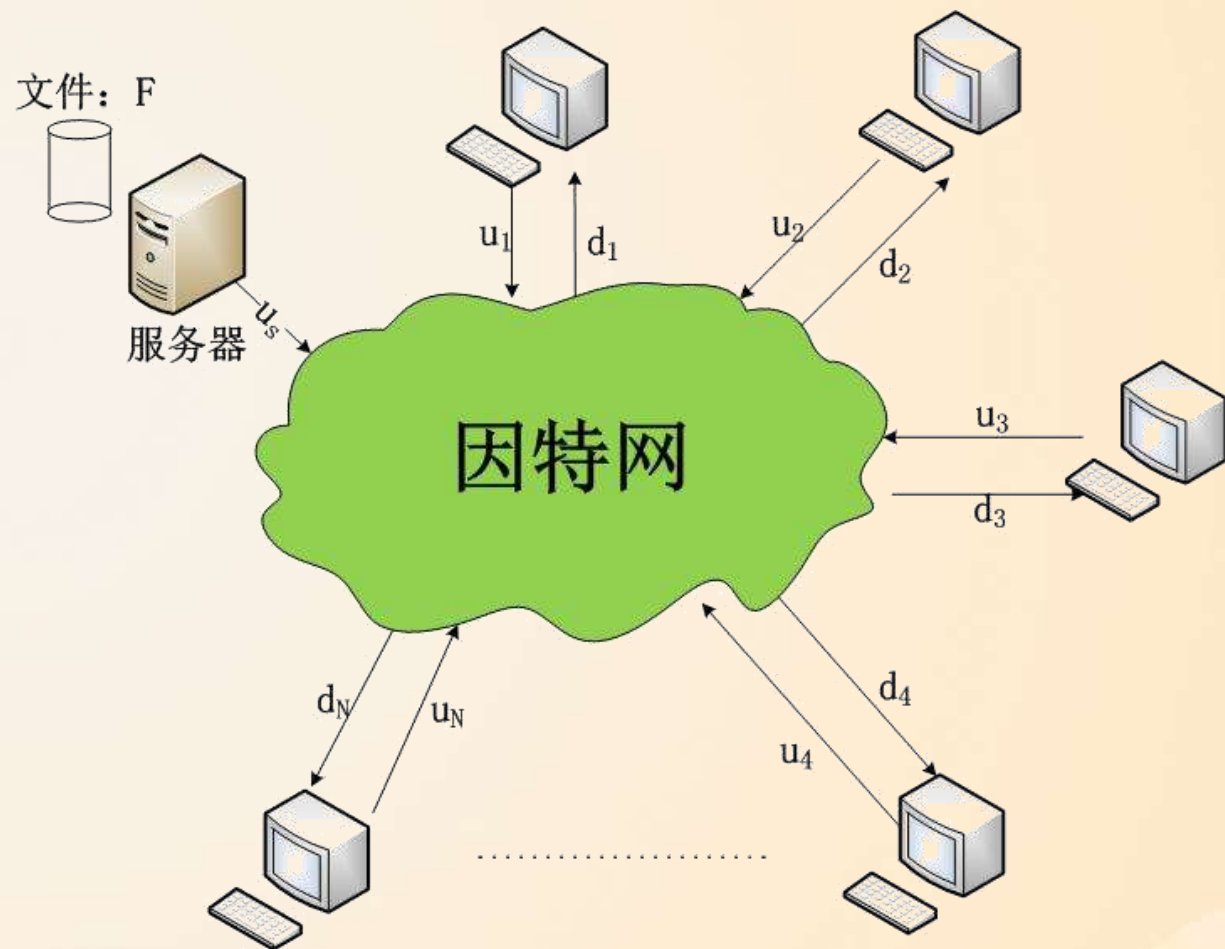
C/S模式

$$T_{cs} \geq \max\left\{\frac{NF}{u_s}, \frac{F}{\min\{d_1, d_2, \dots, d_N\}}\right\}$$



P2P模式

$$T_{P2P} \geq \max\left\{\frac{F}{u_s}, \frac{F}{d_{\min}}, \frac{NF}{u_s + \sum_{i=1}^N u_i}\right\}$$

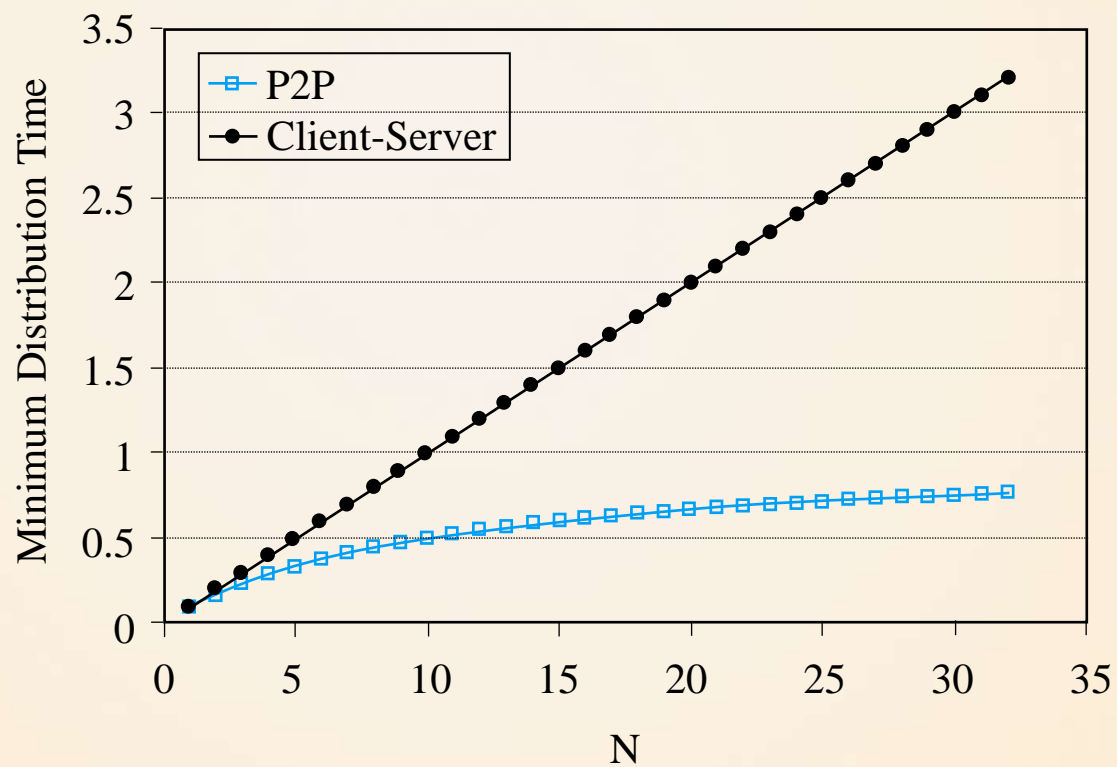




文件分发



C/S和P2P体系结构的文件分发时间





P2P的基本概念

BitTorrent的基本概念

洪流(torrent)

参与一个特定文件分发的所有对等方的集合

追踪器(tracker)

跟踪正参与在洪流中的对等方

文件块(chunk)

256KB



P2P的基本概念

BitTorrent的基本工作机制



向邻居请求哪些块——最稀罕优先



优先响应哪些请求——对换算法 (4+1)

- 每10秒重新确定4个最高速率对等方
- 每30秒随机选择1个新的邻居



本章小结

□ 应用程序架构

- 客户-服务器
- P2P

□ 应用服务要求:

- 可靠性、带宽、延迟

□ 互联网传输服务模式

- 面向连接、可靠: TCP
- 不可靠、数据报: UDP

□ 具体协议:

- HTTP
- SMTP、POP、IMAP
- DNS
- P2P: BitTorrent



本章小结

□ 典型的请求/回复消息交换:

- 客户端请求信息或服务
- 服务器用状态代码, 数据响应,

□ 消息格式:

- 首部: 提供数据信息的字段
- 数据: 正在通信的信息 (有效载荷)

重要主题

□ 控制 vs. 消息

- 带内, 带外

□ 集中 vs. 分散

□ 无状态 vs. 有状态

□ 可靠 vs. 不可靠的邮件传输

□ “网络边缘的复杂性”



课后思考题

□ 复习题 2、5、10、11、16、20、21、23

□ 习题 1、4~11、17、22~24、26



作业

习题

□ 4、7、9