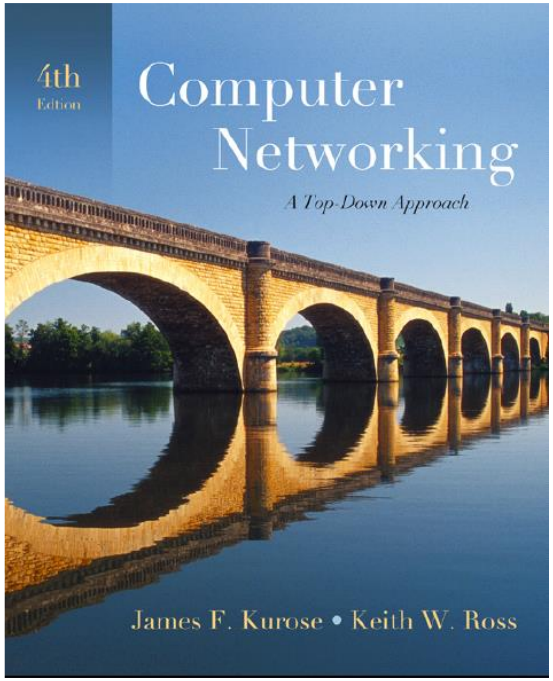
Chapter 2 应用层

- 2.1 网络应用原理
- 2.2 Web和HTTP协议
- 2.3 FTP协议
- 2.4 eMail协议
  - SMTP, POP3, IMAP
- 2.5 DNS



Computer Networking: A Top Down Approach, 4th edition.
Jim Kurose, Keith Ross
Addison-Wesley, July 2007.

2019.3.31

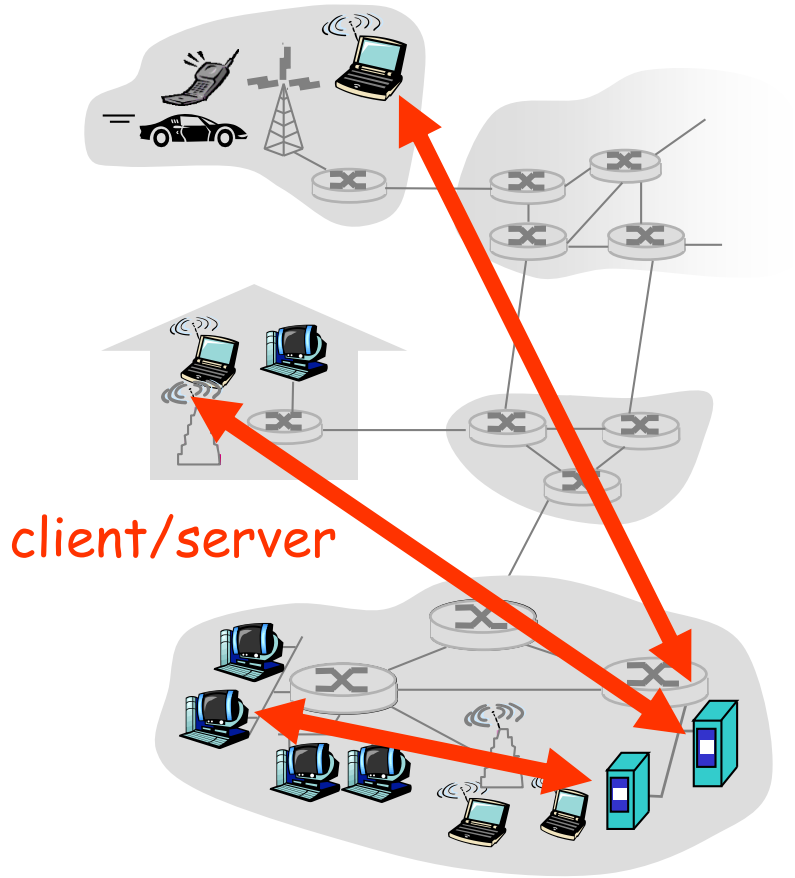
isszym@mail.sysu.edu.cn

Chapter 2: 应用层

- 2.1 网络应用原理
- 2.2 Web和HTTP协议
- 2.3 FTP协议
- 2.4 eMail协议
  - SMTP, POP3, IMAP
- 2.5 DNS

客户-服务器结构

应用层协议一般都是采用客户-服务器结构



服务器(server):

- 总是处于开机的主机
- 固定IP地址(或域名)
- 可以采用服务器群

客户端(clients):

- 与服务器通信
- 可以间歇连接
- 可以采用动态地址
- 不能直接彼此通信

定义应用层协议

- 要交换的消息(message)类型，例如，请求，响应
- 消息语法:
  - 需要什么字段以及每个字段的具体格式
- 消息语义
  - 每个字段的含义和用途
- 制定进程何时以及如何发送消息和响应的规则

公共领域的协议:

- 用RFC(Reference for Comment)定义
- 具有互操作性
- 例如, HTTP, SMTP

专属协议:

- 例如, Skype,QQ

因特网应用层协议：应用领域和传输协议

应用	应用层协议	所使用的传输协议
e-mail	SMTP [RFC 2821]	TCP
remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
file transfer	FTP [RFC 959]	TCP
streaming multimedia	proprietary (e.g. RealNetworks)	TCP or UDP
Internet telephony	proprietary (e.g., Vonage,Dialpad)	typically UDP

Chapter 2: 应用层

- 2.1 网络应用原理
- 2.2 Web和HTTP协议
- 2.3 FTP协议
- 2.4 eMail协议
  - SMTP, POP3, IMAP
- 2.5 DNS

Web和HTTP

- 网页(Web page)是由对象(objects)构成的。
- 这些对象可以是HTML文件、JPEG图像文件、MP4视频文件等。
- 网页的HTML文件中指出了所需的其他对象。
- 每个对象采用URL指明存放地址
- URL的例子:

www.someschool.edu/someDept/pic.gif

主机名　　路径名

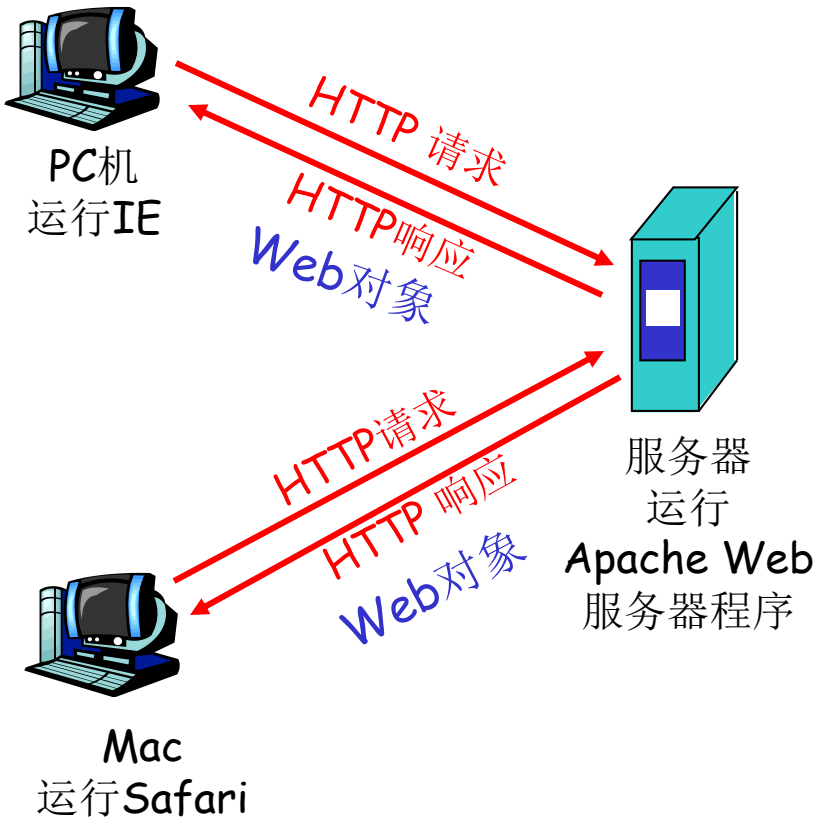
HTML文件


```
<html>
<head>
<title>中山大学 SUN YAT-SEN UNIVERSITY </title>
……
<link href="../css/layout.css" type="text/css" />
<script src="../js/display.js"></script>
<a href="zdgk/index.htm">学校概况</a>
<a href="yx/index.htm">院系设置</a>
<a href="xksz/index.htm">学科与师资</a>
……
<img src="../images/campus-珠海.jpg" alt="广州珠海校区" />
……
<img src="../images/f7.jpg" alt="附属第五医院"
border="0"/>
…
</body></html>
```

HTTP 概述

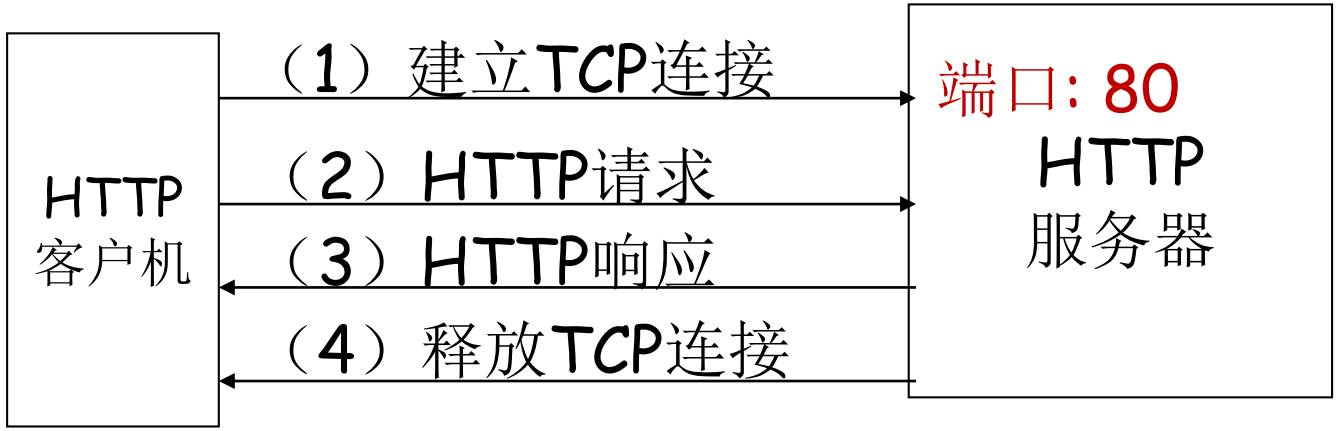
HTTP: 超文本传送协议 (hypertext transfer protocol)

- Web的一种应用层协议
- 客户/服务器模型
  - *客户端:* 浏览器，它负责请求Web对象，并接收和显示从服务器返回的Web对象。
  - *服务器:* Web服务器发送客户端请求的Web对象
- HTTP 1.0: RFC 1945
- HTTP 1.1: RFC 2616
- HTTP 2.0 RFC 7540



HTTP概述(续)

TCP-based:



HTTP 是无状态的(stateless)

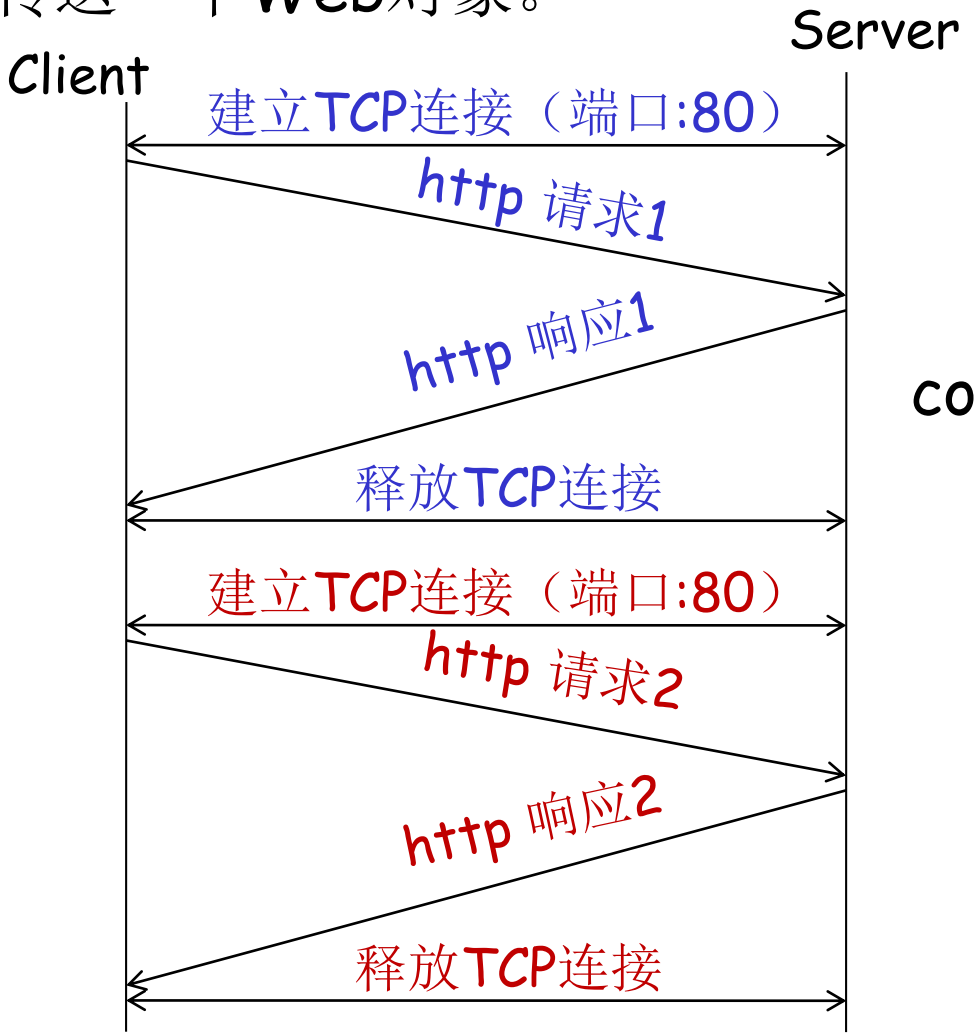
- server不保留过往客户端的任何信息

HTTP 连接

- 持久的(Persistent)，非持久的(Nonpersistent)

非持续连接的HTTP

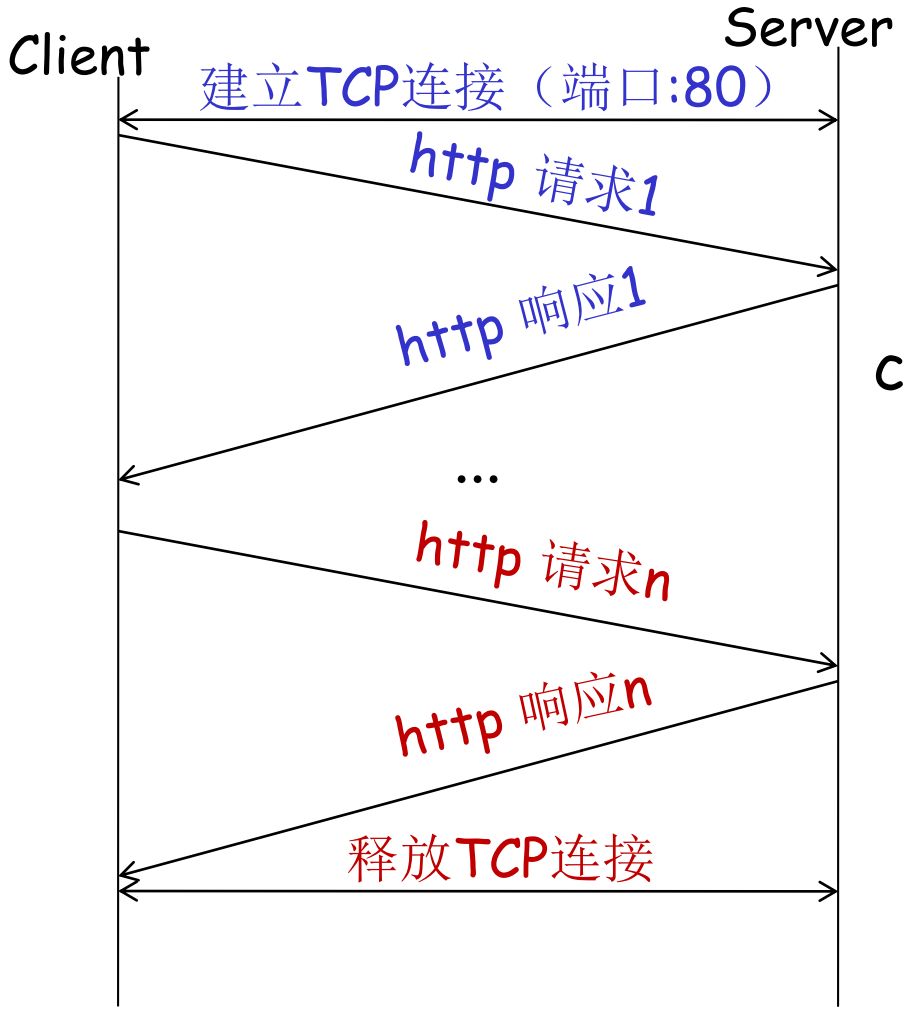
- 每个时刻最多请求一个Web对象，每建立一个连接最多只能传送一个Web对象。



connection: close

非流水式持续连接HTTP

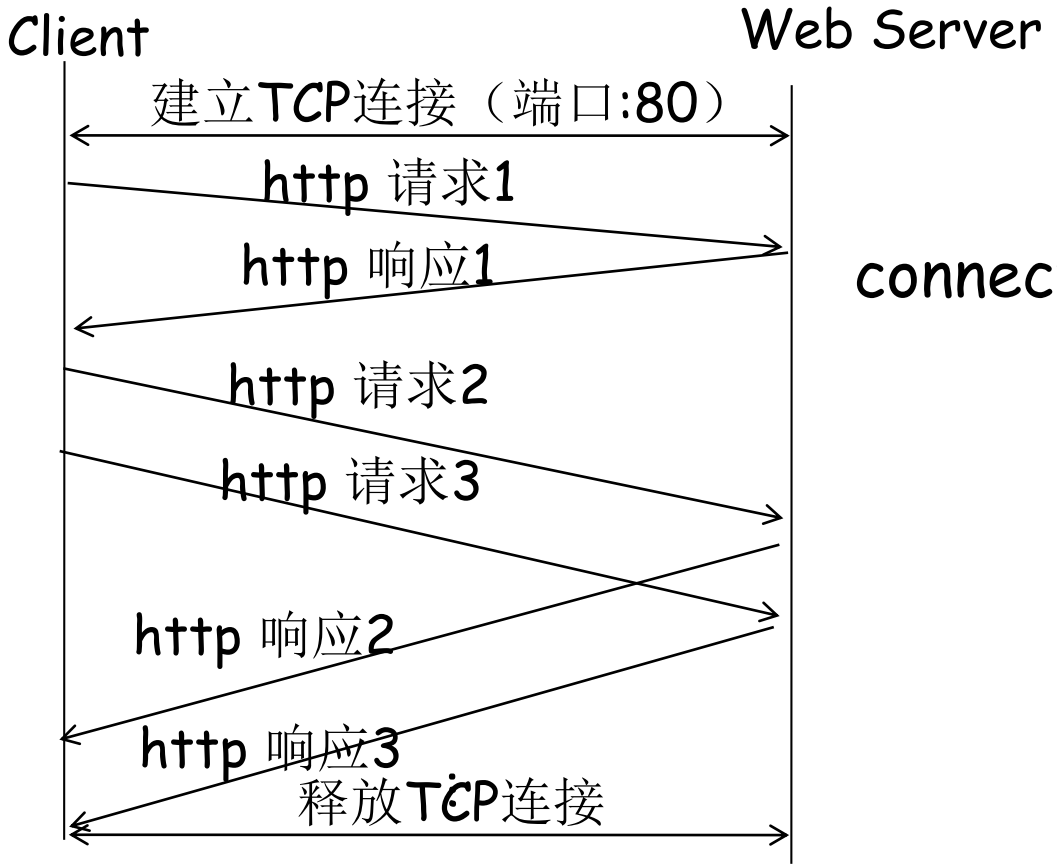
每个时刻可以请求多个Web对象，每建立一个连接可以传送多个Web对象。



connection: keep-alive

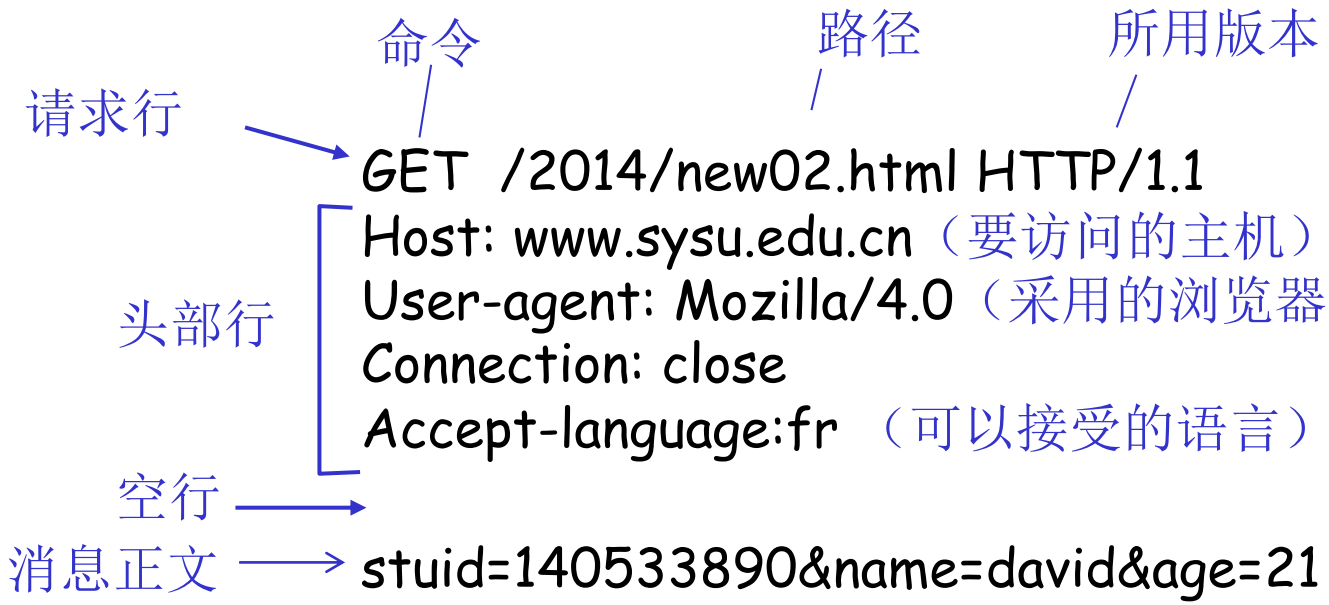
流水式持久连接HTTP

- 可以连续请求多个Web对象，每建立一个TCP连接可以传送多个 Web对象.
- HTTP/1.1 默认使用持续连接.



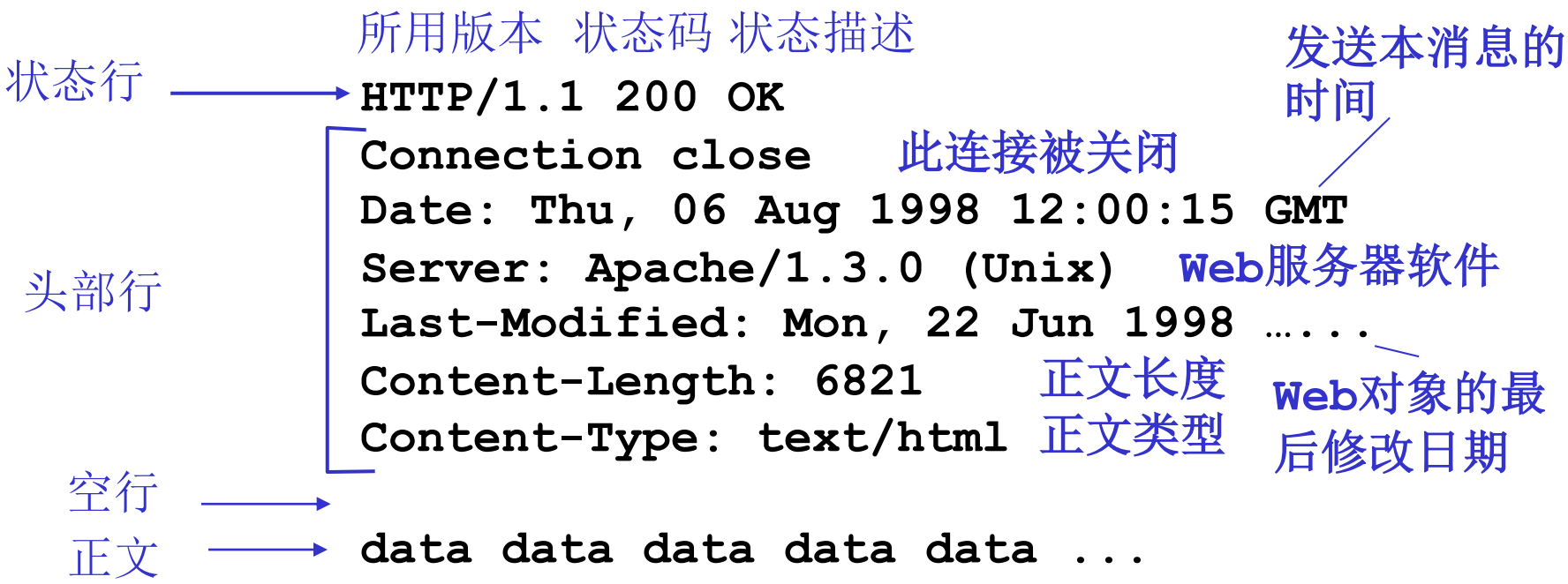
connection: keep-alive

HTTP请求消息



* 这里请求的URL为www.sysu.edu.cn/2014/new02.html
* 只有POST命令才有消息正文
* 每一行均以回车换行结束（\r\n）

HTTP 响应消息



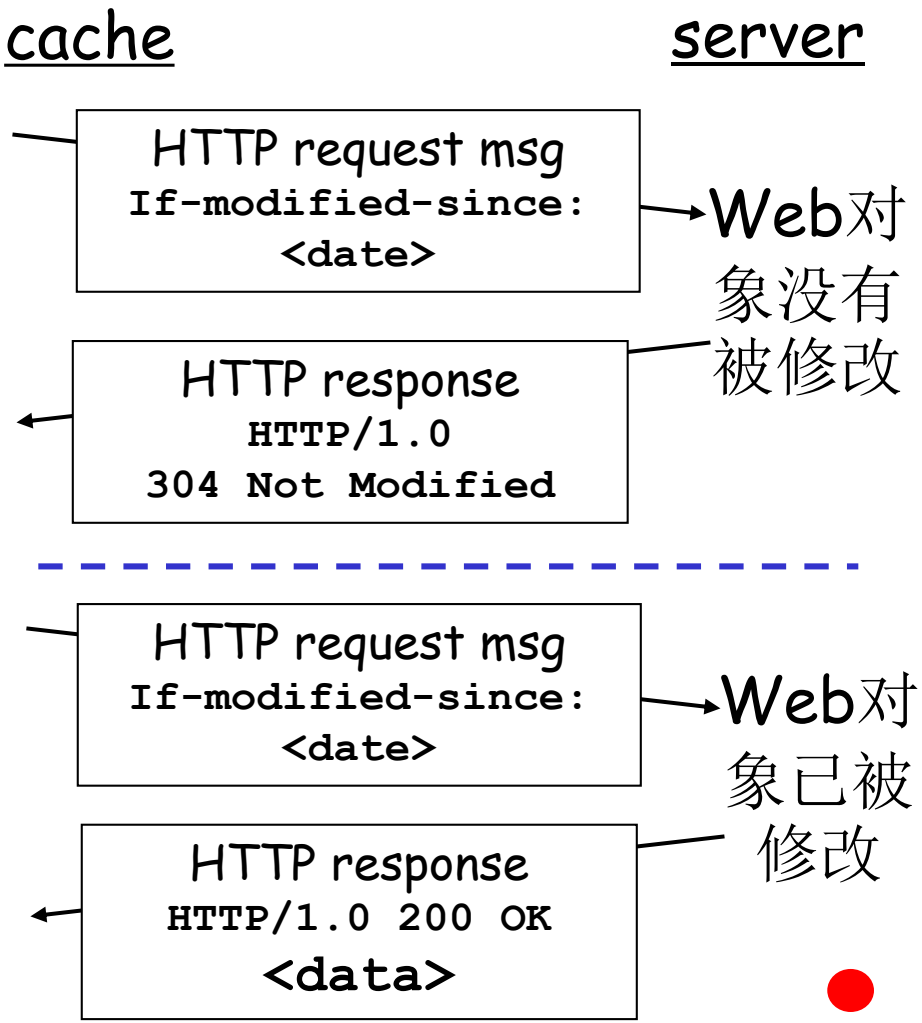
* 正文包含Web服务器发给客户端的Web对象，例如，HTML file

条件GET

- 每次取回Web对象后，浏览器都会把它缓存下来。
- 每次发出HTTP请求之前都会先查询缓存，如果所请求的Web对象存在，则会把该对象的时间date加到HTTP请求中：

  If-modified-since: <date>
- 服务器如果发现缓存的Web对象并没变化，则不需要在HTTP响应中加入该Web对象，响应码为304:

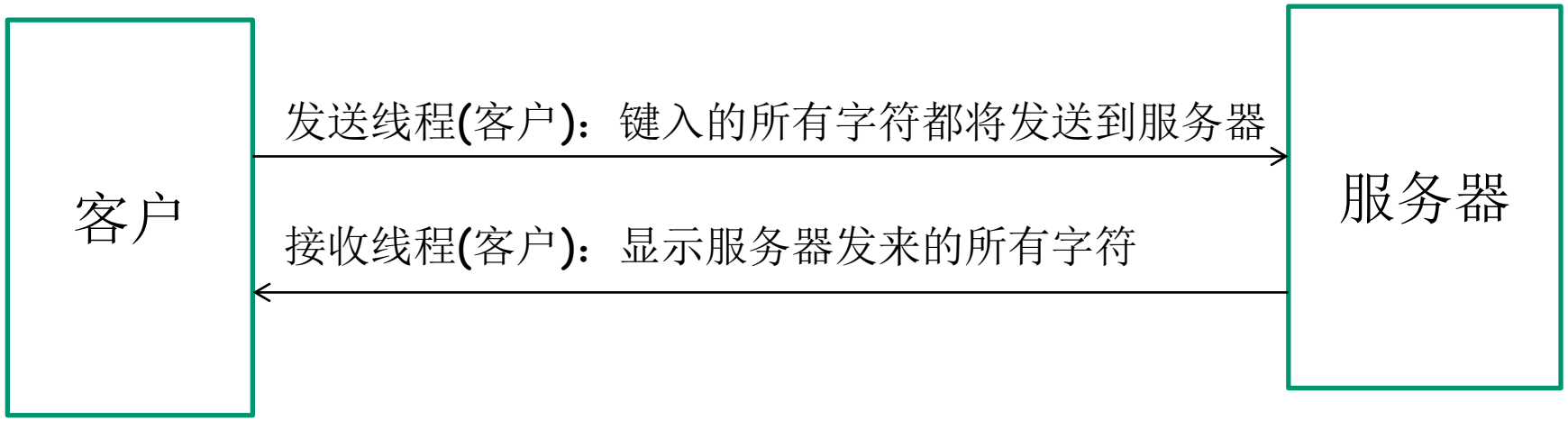
  HTTP/1.0 304 Not Modified



用telnet建立TCP连接

域名或IP地址　　端口号

建立TCP连接　c:>telnet www.sysu.edu.cn 80



HTTP协议实例

c:>telnet www.sysu.edu.cn 80

粘贴

```
GET /2012/cn/index.htm HTTP/1.1
Connection: close
Host: www.sysu.edu.cn
```

connection: keep-alive

c:>telnet www.sysu.edu.cn 80

粘贴

```
GET /2012/images/campus-east.jpg HTTP/1.1
connection: close
Host: www.sysu.edu.cn
```

请求和响应头部

- HTTP1.1对HTTP1.0的改进：

(1) 一个TCP连接可以传送多个HTTP请求和响应。
(2) 可以采用流水线方式，即多个请求和响应过程可以重叠。
(3) 增加了更多的请求头和响应头。

HTTP 2.0

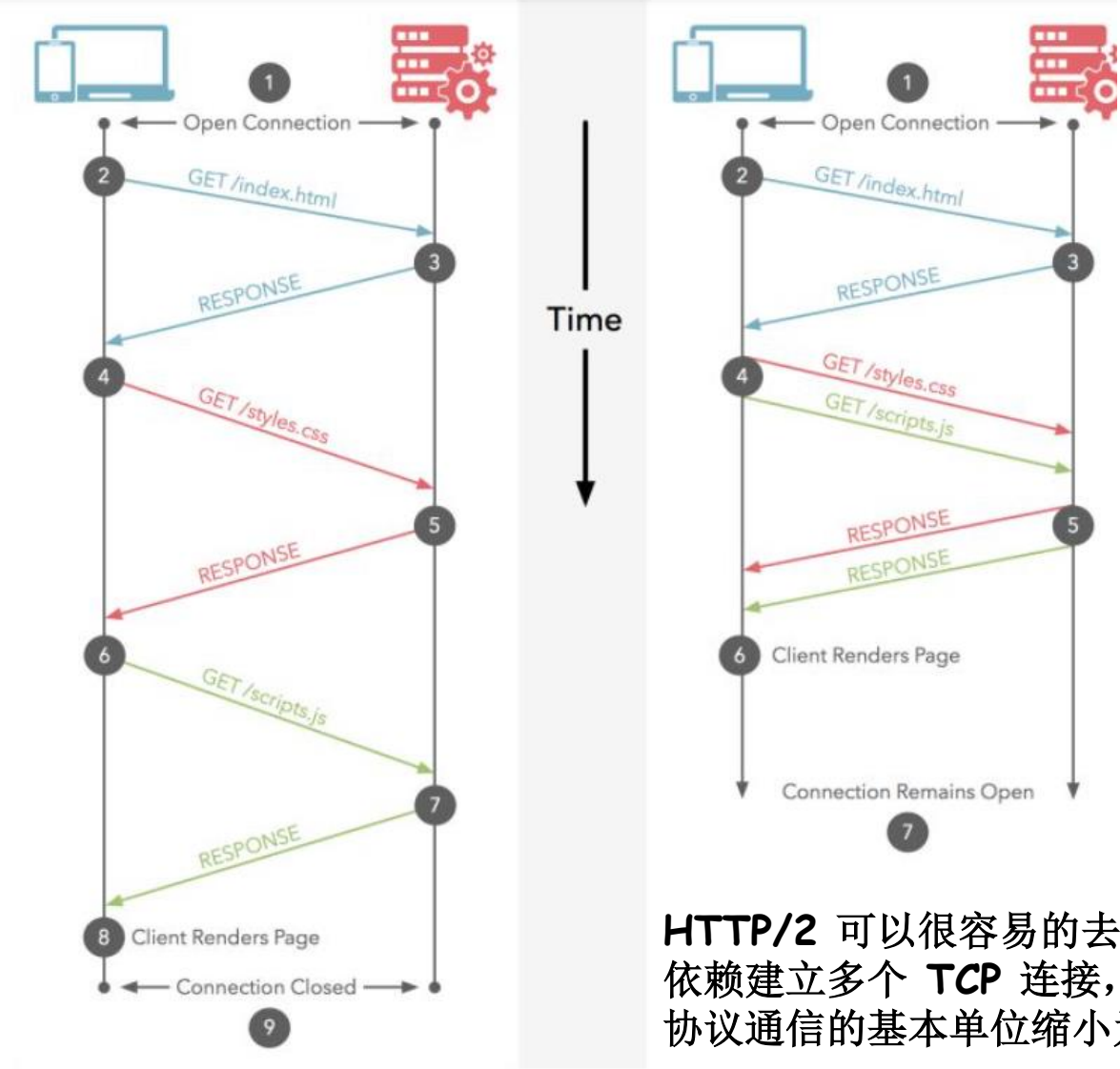
HTTP 2.0采用二进制格式而非文本格式。二进制协议解析起来更高效、“线上”更紧凑，更重要的是错误更少。

HTTP 2.0是完全多路复用的，而非有序并阻塞的。因为它能同时处理很多消息的请求和响应; 甚至可以在传输过程中将一个消息跟另外一个掺杂在一起。

HTTP 2.0使用报头压缩来降低开销。头部的轻微压缩都可以使很多http请求用一个包就可以解决，从而减少了发送数据包的数量。

HTTP 2.0让服务器可以将响应主动“推送”到客户端缓存中。当浏览器请求一个网页时，服务器将会发回HTML，在服务器开始发送JavaScript、图片和CSS前，服务器需要等待浏览器解析HTML和发送所有内嵌资源的请求。服务器推送服务通过“推送”那些它认为客户端将会需要的内容到客户端的缓存中，以此来避免往返的延迟。

HTTP/2.0 相比1.0有哪些重大改进？



HTTP/2 可以很容易的去实现多流并行而不用依赖建立多个 **TCP** 连接，**HTTP/2** 把 **HTTP** 协议通信的基本单位缩小为一个一个的帧。

请求方式

- 主要的请求方式:

GET	请求获取URI所标识的资源
POST	在URI所标识的资源后附加新的数据
HEAD	请求获取由URI所标识的资源的响应消息报头
PUT	请求服务器存储一个资源，并用URI作为其标识
DELETE	请求服务器删除URI所标识的资源
TRACE	请求服务器回送收到的请求信息，主要用于测试或诊断
CONNECT	保留将来使用
OPTIONS	请求查询服务器的性能，或者查询与资源相关的选项和应用举例

- GET请求方式的请求参数加在请求行中，一般在1K以下，在传递请求参数时，在浏览器的url地址后以"?"分隔GET的请求参数，参数之间使用"&"分隔。
- POST请求方式的请求参数在请求消息的内容中，大小无限制。
- GET方法：在浏览器的地址栏中输入网址的方式访问网页时，浏览器采用GET方法向服务器获取资源，eg:GET /form.html HTTP/1.1 (CRLF)
- POST方法要求被请求服务器接受附在请求后面的数据，常用于提交表单。

HTTP通用信息头

既能用于请求消息，也能用于响应消息，包括一些与被传输的实体内容没有关系的常用信息头字段。

(1) Cache-control:no-cache	不缓存。
(2) Connection:close	不保持连接。保持连接：keep-alive
(3) Date:Tue,11 Jul 2010 18:23:51 GMT	请求或响应时间。
(4) Pragma:no-cache	不缓存消息。
(5) Transfer-Encoding:chunked	编码方式(分段编码传输)。
(6) Via:HTTP/1.1 Proxy1,HTTP/1.1 Proxy2	指定途径的代理服务器。
(7) Keep-Alive: 300	要求服务器保持连接300秒的连接。

* Connection为keep-alive时才能用Keep-Alive要求Web服务器保持连接多少秒
* Cache-Control(请求)：见后面。

请求头

用于客户端发送的请求消息的头部。* 标准日期： Tue, 06 May 2008 02:42:43 GMT

(1) Accept:text/html,image/* 客户端可以接收的媒体类型。
(2) Accept-Charset:Unicode-1-1.ISO8859-1 客户端接受的字符编码方式。
(3) Accept-Encoding:gzip,compress 客户端支持的数据压缩格式。
(4) Accept-Language:en-gb,zh-cn 客户端支持的语言。
(5) Authorization:Basic enJEYMZQING== 验证身份(Base64编码)。
(6) Expect:100-continue 需要服务器进一步操作。
(7) Max-Forward:1 最大代理服务器数。
(8) Accept-Range:bytes 可以返回部分正文，none表示不接受
(8) Range:bytes=100-599,800- 返回100到599和800以后的正文。
(9) Referer:http://www.google.com 从哪个网页发出本http请求。
(10) User-Agent:Mozilla/4.0 指定浏览器或客户端的类型。
(11) Host:www.abc.com 客户端想访问的域名和端口号
(12) If-Match: w"1469-1461500473349" 对象的 ETag 没变时才执行请求的动作
(13) If-None-Match 对象的 ETag改变时才执行请求的动作
(14) If-Modified-Since: 标准日期 对象时间被修改时才执行
(15) If-Unmodified-Since:标准日期 对象时间被修改时才执行
(16) If-Range 没改变则按照Range发送，否则全部发送

响应头

用作实体内容的元信息。

(1) Allow:GET,POST　　允许客户端请求方式。
(2) Content-Encoding:gzip　　压缩编码方式。
(3) Content-Language:zh-cn　　服务器返回的文档语言。
(4) Content-Length:80　　实体内容大小。
(5) Content-Location:http://www.google.com　　请求资源所在位置。
(6) Content-Range:bytes 2543-4532　　返回指定部分内容。
(7) Content-Type:text/html;charset=utf-8　　实体内容的MIME类型和编码。
(8) Age:2　　代理服务器响应的实体的缓存时间(s)
(9) ETag:1469-1461500473349　　文件的标记，文件改变ETag也会改变
(10) Expired:Sat, 23 May 2009 10:02:12 GMT　　过期时间
(11) Last-Modified: Tue, 06 May 2008 02:42:43 GMT　　对象的最后修改时间
(12) Location　　告诉客户端对象的新位置
(13) Pramga: no-cache　　相当于 Cache-Control: no-cache
(14) Server：Apache/2.0.61 (Unix)　　Web服务器版本
(15) Transfer-Encoding: chunked　　响应体的编码方式
(16) Vary：Accept-Encoding　　告诉 Cache 服务器响应后续请求的条件

*ETag用于文件改变过快一直modified检测不到（秒）

扩展头

在HTTP1.1规范中没有定义的头字段，被当做实体扩展头处理。

(1) Refresh:1

告诉浏览器每1秒刷新访问1次。

Refresh:1;url=http://www.google.com

过1秒，跳转到指定页面。

(2) Content-Type:application/octet-Stream

Content-Disposition:attachment;filename=aaa.zip

让用户将响应的内容保存在一个文件中。

Cache-control

网页缓存由 HTTP消息头中的Cache-control控制，常见取值有private、no-cache、max-age、must- revalidate等，默认为private。

* Cache-Control(请求)：
  - no-cache 不要缓存的实体，要求现在从WEB服务器去取
  - max-age 只接受 Age 值小于 max-age 值，并且没有过期的对象
  - max-stale 可以接受过去的对象，但是过期时间必须小于 max-stale 值
  - min-fresh 接受其新鲜生命期大于其当前 Age 跟 min-fresh 值之和的缓存对象
  - no-store 不允许缓存

* Cache-Control(响应)：
  - public 可以用 Cached 内容回应任何用户
  - private 只能用缓存内容回应先前请求该内容的那个用户
  - no-cache 可以缓存，但是只有在跟WEB服务器验证了其有效后，才能返回给客户端
  - max-age 本响应包含的对象的过期时间
  - no-store 不允许缓存
  - must-revalidate：强制页面不缓存，作用与no-cache相同，但更严格，强制意味更明显。

HTTP状态码

(1) 100～199：表示成功接收请求，要求客户端继续提交下一次请求才能完成处理。
(2) 200～299：表示成功接收请求，并已经完成处理。
(3) 300～399：为完成请求，客户端需要进一步细化请求。
(4) 400～499：客户端请求有错误。
(5) 500～599：服务器端发送错误。

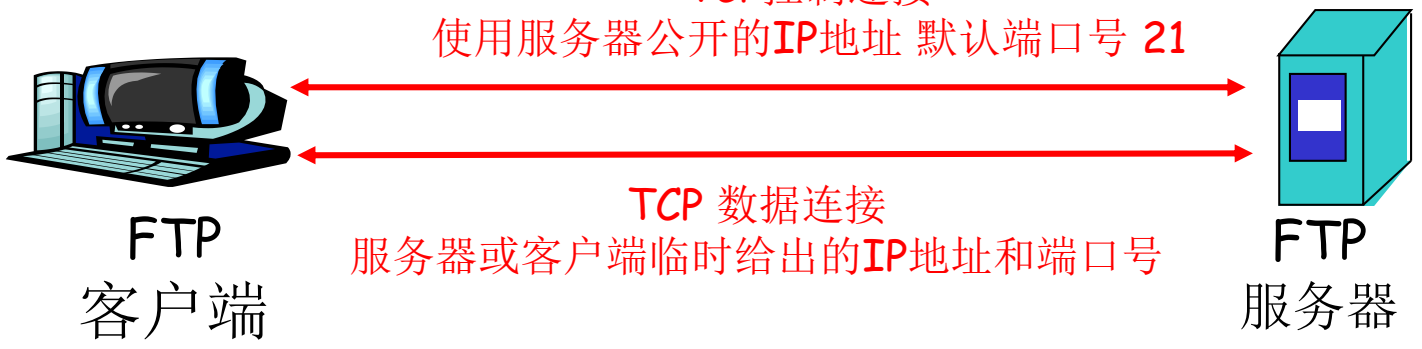
状态码	消息	描述
100	Continue	只有一部分请求被服务器接收，但只要没被服务器拒绝，客户端就会延续这个请求
101	Switching Protocols	服务器交换机协议
200	OK	请求被确认
201	Created	请求时完整的，新的资源被创建
202	Accepted	请求被接受，但未处理完
203	Non-authoritative Information	
204	No Content	
205	Reset Content	
206	Partial Content	
300	Multiple Choices	一个超链接表，用户可以选择一个超链接并访问，最大支持5个超链接
301	Moved Permanently	被请求的页面已经移动到了新的URL下
302	Found	被请求的页面暂时性地移动到了新的URL下
303	See Other	被请求的页面可以在一个不同的URL下找到
304	Not Modified	
305	Use Proxy	

状态码	消息	描述
306	*Unused*	已经不再使用此状态码，但状态码被保留
307	Temporary Redirect	被请求的页面暂时性地移动到了新的URL下
400	Bad Request	服务器无法识别请求
401	Unauthorized	被请求的页面需要用户名和密码
402	Payment Required	*目前还不能使用此状态码*
403	Forbidden	禁止访问所请求的页面
404	Not Found	服务器无法找到所请求的页面
405	Method Not Allowed	请求中所指定的方法不被允许
406	Not Acceptable	服务器只能创建一个客户端无法接受的响应
407	Proxy Authentication Required	在请求被服务前必须认证一个代理服务器
408	Request Timeout	请求时间超过了服务器所能等待的时间，连接被断开
409	Conflict	请求有矛盾的地方
410	Gone	被请求的页面不再可用
411	Length Required	"Content-Length"没有被定义，服务器拒绝接受请求
412	Precondition Failed	请求的前提条件被服务器评估为false
413	Request Entity Too Large	因为请求的实体太大，服务器拒绝接受请求
414	Request-url Too Long	服务器拒绝接受请求，因为URL太长。多出现在把"POST"请求转换为"GET"请求时所附带的大量查询信息
415	Unsupported Media Type	服务器拒绝接受请求，因为媒体类型不被支持
417	Expectation Failed	
500	Internal Server Error	请求不完整，服务器遇见了出乎意料的状况
501	Not Implemented	请求不完整，服务器不提供所需要的功能
502	Bad Gateway	请求不完整，服务器从上游服务器接受了一个无效的响应
503	Service Unavailable	请求不完整，服务器暂时重启或关闭
504	Gateway Timeout	网关超时
505	HTTP Version Not Supported	服务器不支持所指定的HTTP版本

Chapter 2: 应用层

- 2.1 网络应用原理
- 2.2 Web和HTTP协议
- 2.3 FTP协议
- 2.4 eMail协议
  - SMTP, POP3, IMAP
- 2.5 DNS

FTP: 独立的控制连接和数据连接



- 使用FTP协议首先建立控制连接，然后建立数据连接，客户端再通过控制连接发出命令，通过数据连接得到服务器返回的结果或上传数据给服务器。
- 控制连接为带外数据("out of band")
- FTP服务器会保留状态: 当前目录, 已做的认证

FTP实例--目录列表

另开一个控制台窗口

控制连接

telnet 202.116.86.101 **21**

数据连接

telnet 202.116.86.101 **3098**

220 Microsoft FTP Service

user net

331 Password required for user.

pass 123456

230 User user logged in.

pasv

227 Entering Passive Mode (202,116,86,101,12,26).

12*256+26=3098

list

125 Data connection already open; Transfer starting.

226 Transfer complete.

quit

221

port IP地址 端口号(主动模式，等待服务器来建立连接)

port 192,168,150,80,14,178

FTP实例--下载文件

控制连接 telnet 202.116.86.101 21　　数据连接 telnet 202.116.86.101 3124

220 Microsoft FTP Service
user *net*
331 Password required for user.
pass *123456*
230 User user logged in.
type *I*
200 Type set to I.
pasv
227 Entering Passive Mode
(202,116,86,101,12,52).
retr \ebook\ftp.pdf
125 Data connection already open;
Transfer starting.
226 Transfer complete.
quit
221

/ebook/exp.doc
/ebook/ftp.pdf
/text/Web.txt
/text/123456.txt

改变当前目录

cwd ebook

FTP协议例子

--下载文件(断点续传)

控制连接　telnet 202.116.86.101 21　　数据连接　telnet 202.116.86.101 3138

220 Microsoft FTP Service
user net
331 Password required for user.
pass 123456
230 User user logged in.
pasv
227 Entering Passive Mode (202,116,86,101,12,66).
rest 200
350 Restarting at 200.
retr /text/123456.txt
125 Data connection already open; Transfer starting.
226 Transfer complete.
421 Timeout (120 seconds): closing control connection.
421 Terminating connection.

/ebook/exp.doc
/ebook/ftp.pdf
/text/Web.txt
/text/123456.txt

FTP实例-上传文件

控制连接 telnet 202.116.86.101 21

220 Microsoft FTP Service

user net

331 Password required for user.

pass 123456

230 User user logged in.

pasv

227 Entering Passive Mode
(202,116,86,101,12,68).

stor sss.txt

125 Data connection already open;
Transfer starting.

226 Transfer complete.

quit

221

数据连接 telnet 202.116.86.101 3140

把一段文本粘贴进来，
然后关闭窗口

/home/ebook/exp.doc
/home/ebook/ftp.pdf
/home/text/Web.txt
/home/text/123456.txt

FTP命令

- 访问控制命令

  USER 用户名

  PASS 口令

  CWD 改变工作目录

  CDUP 回到上一层目录

  REIN 重新初始化（重置，可以再用USER命令）

  QUIT 退出登录

传输参数命令

所有数据传输参数有默认值。服务器必须记录下默认值，在FTP服务请求后，可以以任何顺序发送。下面命令传送参数：

PORT h1,h2,h3,h4,p1,p2 设置数据端口，要求服务器等待客户端进行连接： IP地址--h1.h2.h3.h4　端口号 -- p1*256+h2

PASV　　被动(Passsive)等待服务器给出数据端口的IP地址和端口

TYPE　　A-ASCII非打印字符(默认)，I-Image 二进制

STRU　　F文件结构（默认值），R 记录结构，P 页结构

MODE　　传输模式： S - 流（默认值）B - 块　C - 压缩

服务命令

RETR 获得(retrieve)文件，即把服务器的文件下载到本地

STOR 保存(store)文件，即把本地文件上传至服务器。如果文件已存在，原文件将被覆盖。如果文件不存在，则新建文件。

STOU 唯一保存(store unique)，此命令和STOR差不多，此命令要求在此目录下的文件名是唯一的，对此命令的响应必须包括产生的用户名。

APPE 附加(append)。它和STOR的功能差不多，但是如果文件在指定路径内已存在，则把数据附加到原文件尾部，如果不存在则新建文件。

ALLO 分配(allocation)此命令用于在一些主机上为新传送的文件分配足够的存储空间。参数是十进制的逻辑字节数。

REST 重新开始(restart) 参数域代表服务器要重新开始的那一点。

RNFR 重命名(rename)，没有参数指定新的文件名。

RNTO 重命名为(rename to)。 和上面的命令共同完成对文件的重命名。

ABOR 放弃(abort)。此命令通知服务中止以前的FTP命令和与之相关的数据传送。如果先前的操作已经完成，则没有动作，返回226。如果没有完成，返回426，然后再返回226。关闭控制连接，数据连接不关闭。

DELE 删除指定路径下的文件。用户进程负责对删除的提示。

RMD 删除目录

MKD 创建目录。此命令在指定路径下创建新目录。

PWD 打印工作目录(print work dir)。返回当前工作目录。

LIST 列表。如果路径指定一个目录或许多文件，服务器用ASCII返回指定路径下的文件列表。如果路径名指定一个文件，服务器返回文件的当前信息，参数为空表示用户当前的工作目录或默认目录。数据传输在或EBCDIC下进行。

NLST 名字列表(name list). 服务器用ASCII传送目录表名到用户，路径名应指定目录或其它系统指定的文件群描述子；空参数指当前目录。

SITE 站点参数。服务器用来提供服务器系统信息，信息因系统不同而不同，格式在HELP SITE命令应答中给出。

SYST 用于确定服务器上运行的操作系统。

STAT 此命令返回控制连接状态，它可以在文件传送过程中发送，服务器返回操作进行的状态，也可以在文件传送之间发送。这时命令有参数，参数是路径名。此命令的功能除了数据在控制连接上传送以外和列表命令相似。如果指定部分路径，服务器以文件名或与说明相关的属性返回；如没有参数，服务器返回服务器FTP进程的状态信息。

HELP 帮助

NOOP 等待。此命令不产生什么实际动作，它仅使服务器返回OK。

FTP响应码

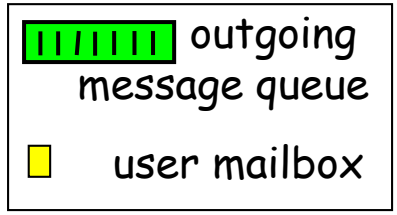
- 200 Command okay.
- 202 Command not implemented, superfluous at this site.
- 211 System status, or system help reply.
- 212 Directory status.
- 213 File status.
- 214 Help message. On how to use the server or the meaning of a particular non-standard command. This reply is useful only to the human user.
- 215 NAME system type. Where NAME is an official system name from the list in the Assigned Numbers document.
- 220 Service ready for new user.
- 221 Service closing control connection.
- Logged out if appropriate.
- 225 Data connection open; no transfer in progress.
- 226 Closing data connection.
- Requested file action successful (for example, file
- transfer or file abort).
- 227 Entering Passive Mode (h1,h2,h3,h4,p1,p2).
- 230 User logged in, proceed.
- 250 Requested file action okay, completed.
- 257 "PATHNAME" created.
- 331 User name okay, need password.
- 332 Need account for login.
- 350 Requested file action pending further information.

- 421 Service not available, closing control connection. This may be a reply to any command if the service knows it must shut down.
- 425 Can't open data connection.
- 426 Connection closed; transfer aborted.
- 450 Requested file action not taken.File unavailable (e.g., file busy).
- 451 Requested action aborted: local error in processing.
- 452 Requested action not taken. Insufficient storage space in system.
- 500 Syntax error, command unrecognized. This may include errors such as command line too long.
- 501 Syntax error in parameters or arguments.
- 502 Command not implemented.
- 503 Bad sequence of commands.
- 504 Command not implemented for that parameter.
- 530 Not logged in.
- 532 Need account for storing files.
- 550 Requested action not taken.
- File unavailable (e.g., file not found, no access).
- 551 Requested action aborted: page type unknown.
- 552 Requested file action aborted. Exceeded storage allocation (for current directory or dataset).
- 553 Requested action not taken. File name not allowed.

Chapter 2: 应用层

- 2.1 网络应用原理
- 2.2 Web和HTTP协议
- 2.3 FTP协议
- 2.4 eMail协议
  - SMTP, POP3, IMAP
- 2.5 DNS

电子邮件(Electronic Mail)



Email有三个主要组件:

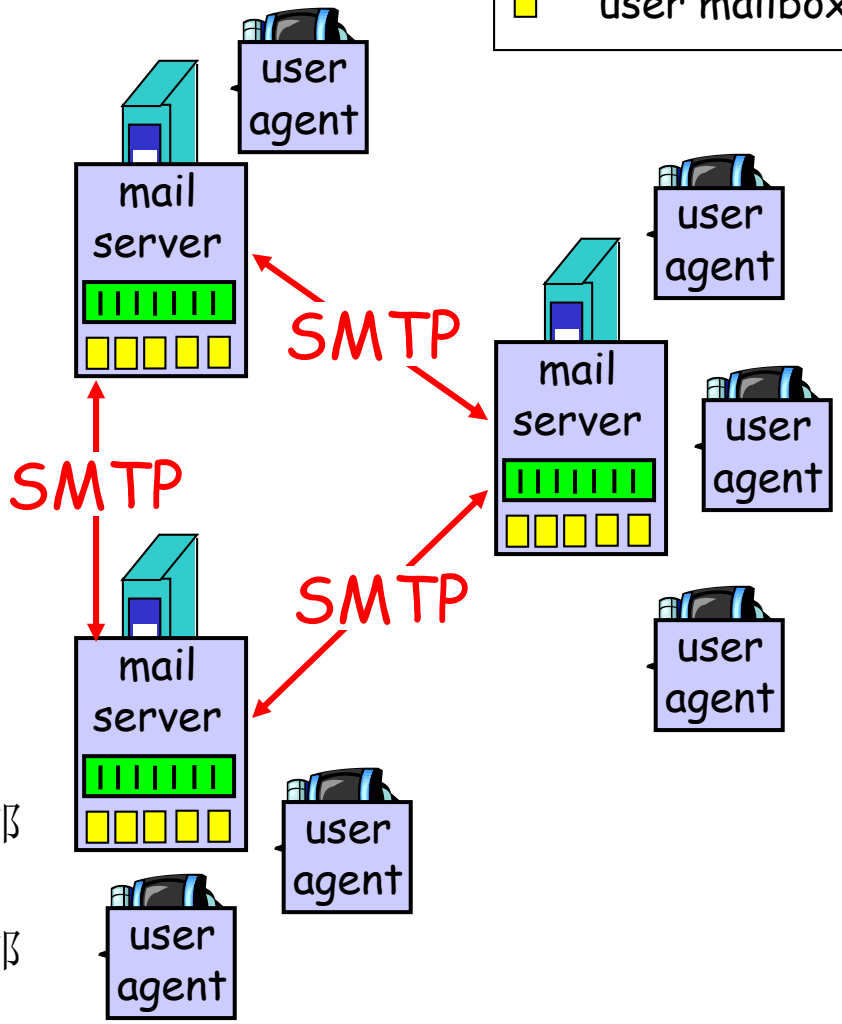
- 用户代理(user agents)
- 邮件服务器(mail servers)
- 简单邮件传送协议(simple mail transfer protocol): SMTP

用户代理(User Agent)

- 一个客户端软件:PC或Web
- 可以编写、编辑、读取邮件信息
- 例如：firefox、闪电邮、gmail

邮件服务器(Mail Server)

- 用户代理用SMTP协议把邮件发送给邮件服务器
- 邮件服务器之间采用SMTP协议传递邮件

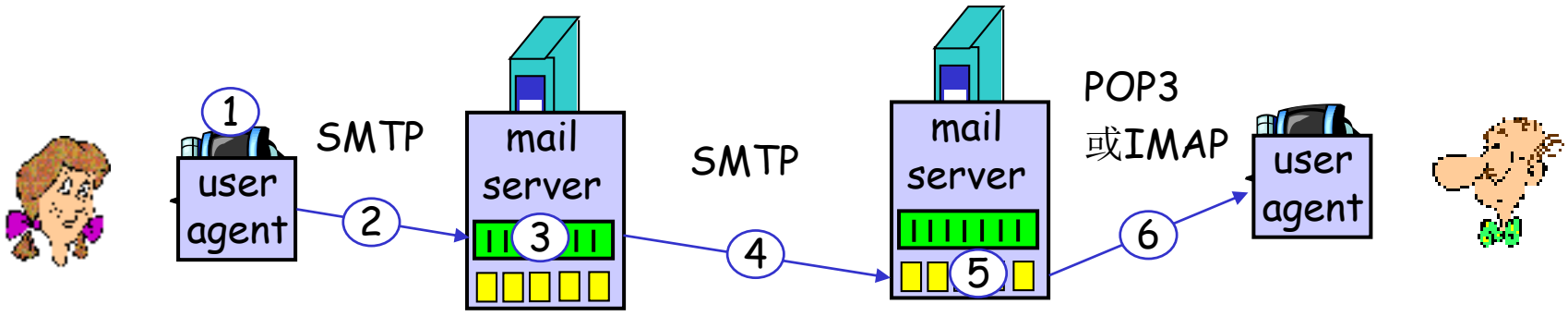


SMTP协议 [RFC 2821]

- SMTP协议采用TCP协议（端口号25）把email消息发送给邮件服务器
- 发送有三个阶段
  - 握手 (问候)和身份认证
  - 传送消息
  - 关闭
- 命令和响应
  - 命令: 采用ASCII文本
  - 响应: 状态码和短语说明
- 消息必须是7位ASCII码

Alice 发送邮件消息给Bob

1) Alice使用用户代理编写消息给 bob@someschool.edu
2) Alice用代理把她编写的消息发送给邮件服务器; 消息放在消息队列中排队
3) Alice的邮件服务器与Bob的邮件服务器建立TCP连接
4) Alice的邮件服务器把Alice的邮件消息发送给Bob的邮件服务器
5) Bob的邮件服务器把这个消息放在Bob的邮箱中
6) Bob采用POP3协议通过他的用户代理读取该邮件消息。



IMAP (Internet Mail Access Protocol [RFC 1730])协议：比POP3协议更复杂，可以对存储在邮件服务器上的消息进行操作，例如，建立文件夹并移动邮件消息，这是POP3协议所没有的功能。

简单邮件格式（RFC 822）

Date:Tue, 25 Mar 2019 14:24:01 +0800
From: "zsusender" <zsusender3@sina.com>
To: "zsureceiver" <zsureceiver3@sina.com>
Subject: SDCS 17 isszym
（空行）
This is a SDCS message in simple format.
Hello! SDCS 17 from isszym

头部

正文

SMTP实例

*可以使用网上的“base64在线编码”

与邮件服务器建立TCP连接

telnet smtp.sina.com 25

220 irxd5-201.sinamail.sina.com.cn ESMTP
HELO zsusender3 打招呼
250 irxd5-201.sinamail.sina.com.cn
AUTH LOGIN 认证注册
334 VXNlcm5hbWU6 334 Username
enN1c2VuZGVyMw== (用户名的base64编码)
334 UGFzc3dvcmQ6 334 Password
MTIzNDU2QWE= (密码的base64编码)
235 #2.0.0 OK Authenticated
MAIL FROM:<zsusender3@sina.com>
250 sender <zsusender@sina.com> ok
RCPT TO:<zsureceiver3@sina.com>
250 recipient <zsureceiver@sina.com> ok

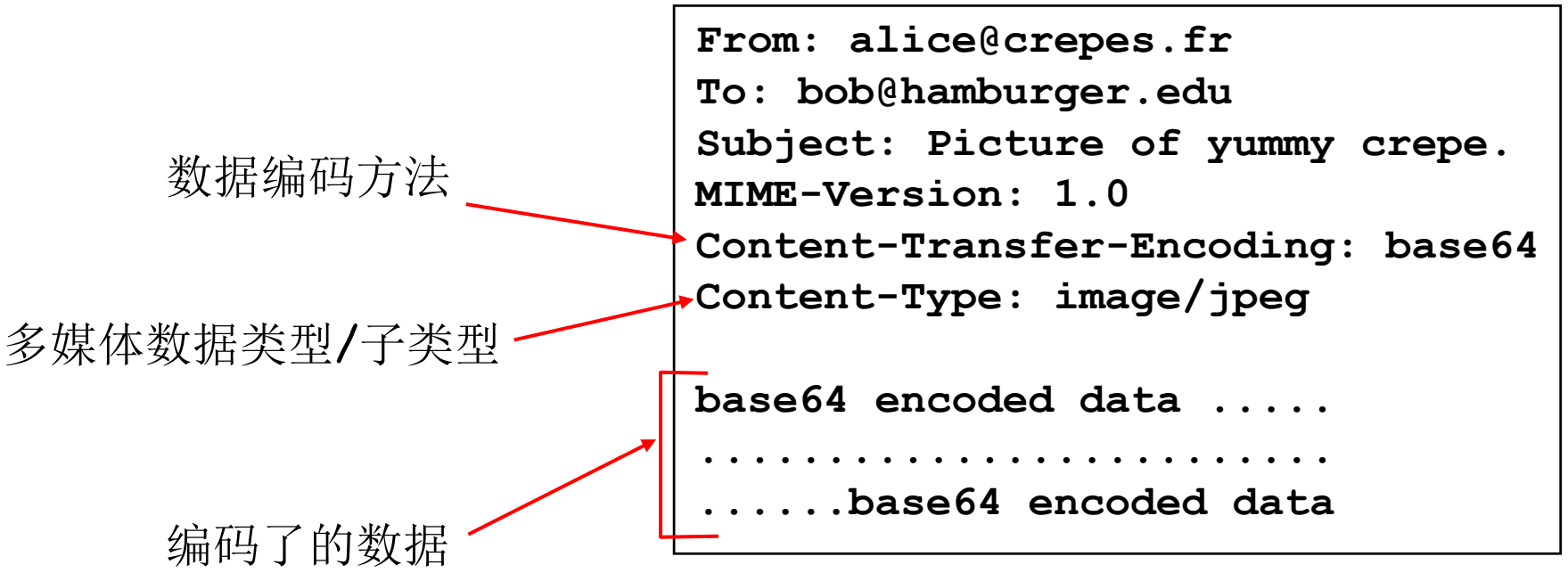
用户名：zsusender3
密码：123456Aa

data
354 go ahead
Date:Tue, 25 Mar 2019 14:24:01 +0800
From: "zsusender3" <zsusender3@sina.com>
To: "zsureceiver3"<zsureceiver3@sina.com>
Subject: SDCS 17 isszym
（空行）
This is a SDCS message in MIME format.
Hello! SDCS 17 from isszym
（以.表示邮件结束）
.

250 ok: Message 1176053204 accepted
QUIT
221 irxd5-201.sinamail.sina.com.cn

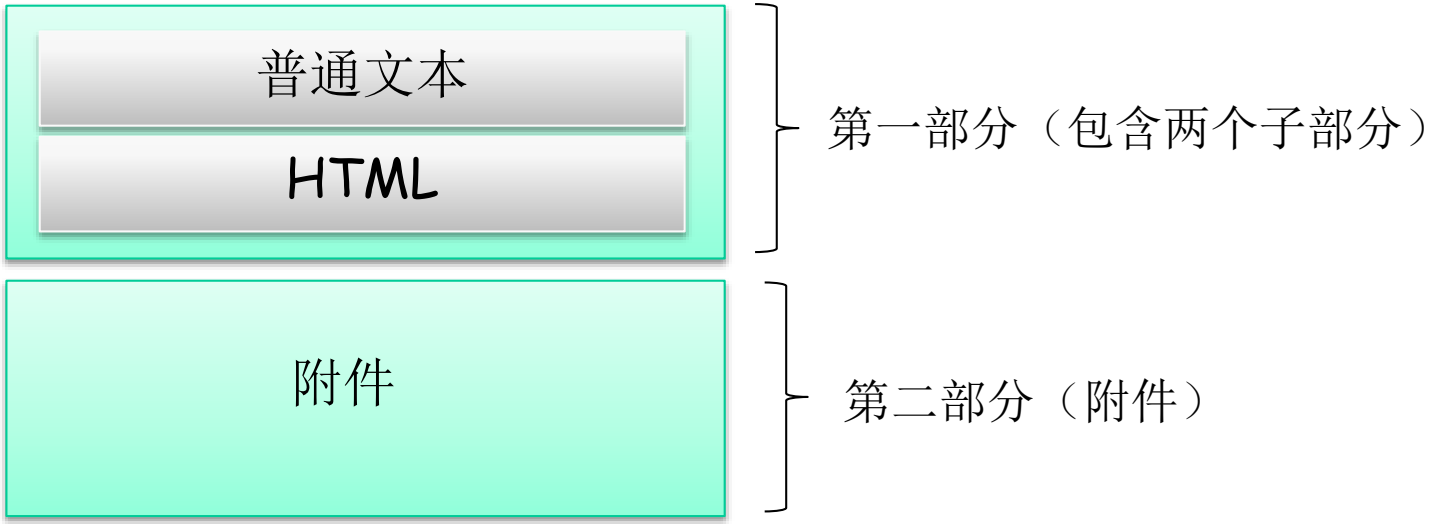
邮件消息格式: 多媒体扩展（1）

- MIME: multimedia mail extension, RFC 2045~2047
- 消息头包含MIME内容类型，内容可以包含多个部分。一般而言，包含一个文本部分（普通文本和HTML文本）和若干附件（文件）。



邮件消息格式: 多媒体扩展（2）

下面的邮件正文包含文本和附件两个部分。邮件文字分成两个子部分，分别包含普通文本和HTML两种表示，附件包含文件rfc855.pdf。这些部分都采用空行加上边界（boundary）分隔开。每一部分都有一个头部和一个正文。



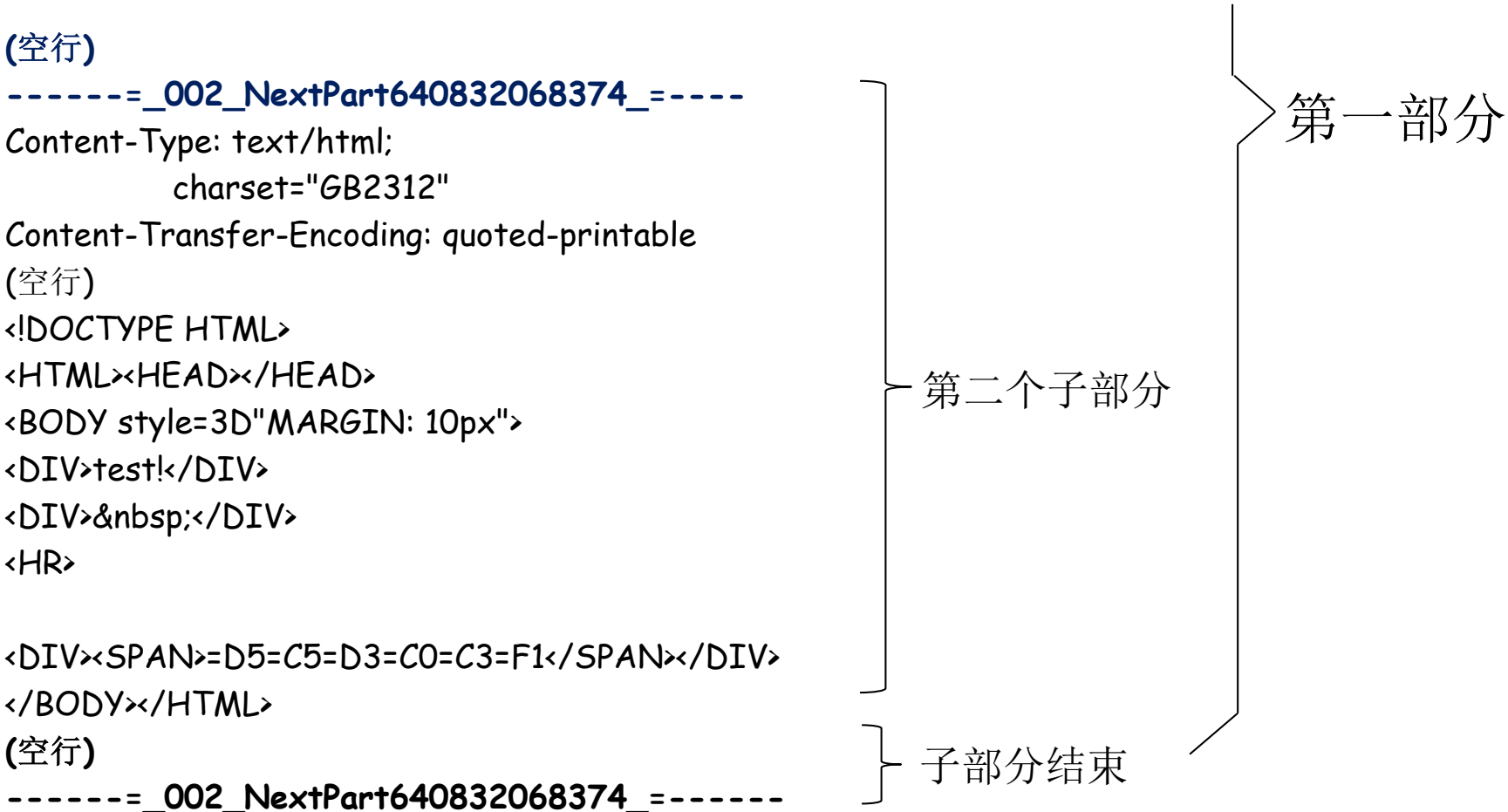
```
Date: Thu, 17 Apr 2014 20:30:07 +0800
From: =?GB2312?B?1cXTwMPx?= <zsusender3@sina.com>
To: isszym <zsureceiver3@sina.com>
Subject: rfc855
Mime-Version: 1.0
Message-ID: <20140417203005640556l@mail.sysu.edu.cn>
Content-Type: multipart/mixed;
        boundary="----=_001_NextPart061116676577_=----"
(空行)
This is a multi-part message in MIME format.
(空行)
------=_001_NextPart061116676577_=----
Content-Type: multipart/alternative;
            boundary="----=_002_NextPart640832068374_=----"
(空行)
(空行)
------=_002_NextPart640832068374_=----
Content-Type: text/plain;charset="GB2312"
Content-Transfer-Encoding: base64
(空行)
dGVzdCENCg0KDQoNCg0K1cXTwMPx
```

邮件头部

邮件正文开始

子部分开始

第一个子部分


(空行)
------=_002_NextPart640832068374_=----
Content-Type: text/html;
charset="GB2312"
Content-Transfer-Encoding: quoted-printable
(空行)
<!DOCTYPE HTML>
<HTML><HEAD></HEAD>
<BODY style=3D"MARGIN: 10px">
<DIV>test!</DIV>
<DIV> </DIV>
<HR>
<DIV><SPAN>=D5=C5=D3=C0=C3=F1</SPAN></DIV>
</BODY></HTML>
(空行)
------=_002_NextPart640832068374_=------
第二个子部分
子部分结束
第一部分

(空行)

------=_001_NextPart061116676577_=----

Content-Type: application/octet-stream;

name="rfc855.pdf"

Content-Transfer-Encoding: base64

Content-Disposition: attachment;

filename="rfc855.pdf"

(空行)

JVBERi0xLjIKJcfsj6IKNCAwIG9iago8PC9MZW5ndGggNSAwIFIvRmlsdGVyIC9GbGF0ZURlY29k

ZT4+CnN0cmVhbQp4nJVWXXMaNxR951fcpw6ewS6LE8dOpzN1iN2SiYGYTfJQ+iB2hVGrldaS1pj+

.....

aWxlcgo8PCAvU2l6ZSAxNCAvUm9vdCAxIDAgUiAvSW5mbyAxMyAwIFIKPj4Kc3RhcnR4cmVmCjQw

MzkKJSVFT0YK

(空行)

------=_001_NextPart061116676577_=------

第二部分

邮件正文结束

POP3协议

- pop3协议用于用户代理取回和处理邮件服务器的邮件访问协议。pop3协议可以通过发送命令给邮件服务器查询邮件、删除邮件、取回邮件。
- 保存在邮件服务器的每个邮件消息都有唯一标识符。可以通过这个判断是否已经取过某个邮件。
- POP3协议的每个会话在操作邮件之前要求身份认证。两个会话之间不能保持状态信息。POP3协议的会话是采用端口号110建立TCP连接实现的。
- IMAP协议是另一种邮件访问协议，它可以通过发送命令在邮件服务器上管理文件夹。IMAP在邮件服务器上保存在文件夹的名称以及消息标识符和文件夹的映射。

POP3实例

telnet pop.sina.com 110

与本地邮件服务器建立TCP连接，端口号为110

+OK pop3 proxy server ready

user zsureceiver3

+OK

pass 123456Aa

+OK

retr 1

+OK 4653 octets

X-Mda-Received: from <mx3-10.sinamail.sina.com.cn>

.....

QUIT

+OK

pop3协议的主要命令

STAT 提供信箱大小信息。

LIST 返回所有邮件的大小。 例，LIST 2 - 返回第2封邮件的大小

RETR 从服务器中取出邮件。例，RETR 2 - 返回第2封邮件的内容

TOP 取出信头和邮件的前N行。例，TOP 2 10返回第2封邮件的头10行。

DELE 删除邮件。例， DELE 2 删除邮件2。

UIDL 取出邮件的唯一标识符。
例，UIDL 1, 返回： +OK 1 MD00001:MSG:13100:3337811140

桌面邮件程序(用户代理)会把取过邮件的UIDL保存在数据库中，删除客户端邮件，这个UIDL依然存在，因此，不会再取一次，如果服务器的邮件没有删除又希望再取一次，只有建一个新的账号重新下载所有邮件。

Chapter 2: 应用层

- 2.1 网络应用原理
- 2.2 Web和HTTP协议
- 2.3 FTP协议
- 2.4 eMail协议
  - SMTP, POP3, IMAP
- 2.5 DNS

域名系统
（DNS: Domain Name System）

人: 有很多身份标识:

- 身份证号, 名字, 护照号

因特网的主机和路由器:

- IP地址 (32 bit) - 数据报中使用的地址
- 人们使用的"名字", 例如, www.yahoo.com

问题: 主机名如何转换为IP地址或相反?

域名系统:

- *分布数据库*

  由分层的很多名字服务器(*name servers*）*组成*
- *应用层协议*

  主机，路由器，名字服务器相互通信来解析名字 (地址/名字转换)
  - 注意: 这是应用层协议实现的因特网核心功能
  - 这是位于网络“边缘”的复杂功能

域名系统的服务和结构

DNS提供的服务

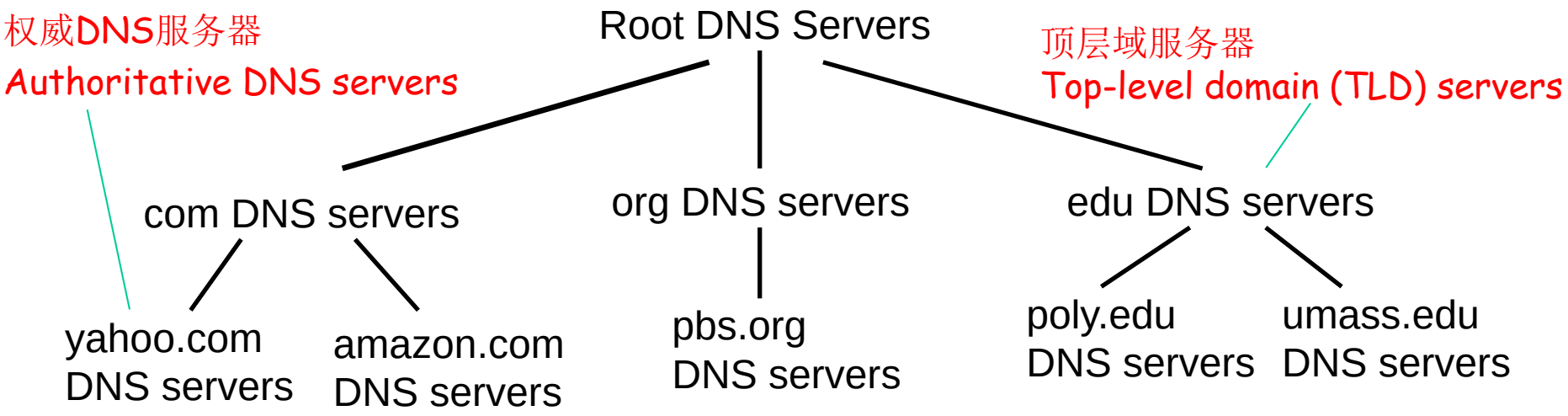
- 主机名到IP地址的转换
- 为主机取别名
  - 权威名(Canonical name), 别名(alias names)
- 为邮件服务器取别名
- 负载分配 (load distribution)
  - 重复的Web服务器: 一个权威名对应一组IP地址

为什么不采用集中式DNS?

- 单点失效
- 流量过于集中
- 可能距离数据库太远
- 维护问题

总之，不易扩大规模!

分布和分层的数据库



如果客户希望获得www.amazon.com的IP地址:

- 客户先到根服务器上查询comDNS服务器的IP地址
- 然后到com DNS服务器上查询amazon.com的DNS服务器
- 最后查询amazon.com DNS服务器获得www.amazon.com的IP地址

DNS: 根名字服务器（Root name servers）

- 公开的IP地址，不需要解析名字，由本地名字服务器直接联系
- 根名字服务器:
  - 如果不知道名字映射，把该名字所在的权威名字服务器的IP地址返回给本地名字服务器
- 根服务器主要用来管理互联网的主目录，全世界只有13台：1个为主根服务器，放置在美国。其余12个均为辅根服务器，其中9个放置在美国，欧洲2个，位于英国和瑞典，亚洲1个，位于日本。

名称	管理单位及设置地点	IP地址
A	INTERNIC.NET（美国，弗吉尼亚州）	198.41.0.4
B	美国信息科学研究所（美国，加利弗尼亚州）	128.9.0.107
C	PSINet公司（美国，弗吉尼亚州）	192.33.4.12
D	马里兰大学（美国马里兰州）	128.8.10.90
E	美国航空航天管理局（美国加利弗尼亚州）	192.203.230.10
F	因特网软件联盟（美国加利弗尼亚州）	192.5.5.241
G	美国国防部网络信息中心（美国弗吉尼亚州）	192.112.36.4

名称	管理单位及设置地点	IP地址
H	美国陆军研究所（美国马里兰州）	128.63.2.53
I	Autonomica公司（瑞典，斯德哥尔摩）	192.36.148.17
J	VeriSign公司（美国，弗吉尼亚州）	192.58.128.30
K	RIPE NCC（英国，伦敦）	193.0.14.129
L	IANA（美国，弗吉尼亚州）	198.32.64.12
M	WIDE Project（日本，东京）	202.12.27.33

顶级域服务器和权威服务器

- 顶级域服务器:
  - 顶级域名提供了权威DNS服务器的IP地址，包括com, org, net, edu, uk, fr, ca, jp等。
  - *例如，Network Solutions公司维护com顶级域服务器，Educause公司维护edu顶级域服务器* for edu TLD
- 权威DNS服务器:
  - 每个组织机构的公开可访问主机都必须提供公共可访问的DNS记录，这些记录保存在权威DNS服务器上，并把这些主机的主机名映射到IP地址上 (e.g., Web, mail).
  - 权威服务器由大型组织或服务提供商维护

本地名字服务器

- 严格说，本地名字服务器不属于DNS分层结构
- 每个ISP都可以有一个DNS服务器，例如，住宅区ISP，公司，大学等。
- 因此，本地名字服务器也成为“缺省的名字服务器”
- 一台主机会先把DNS查询送到本地DNS服务器，该服务器会作为代理到DNS分层结构中进行查询，并把查询结果返回给主机。

中大的域名服务器 -- 10.8.8.8 10.8.4.4

DNS 记录

DNS: 采用分布数据库保存资源记录(resource records, RR)

RR格式: **(name, value, type, class,ttl)**

- Type=A （Address RR）
  - name是主机名
  - value是IPv4地址
- Type=NS
  - name是域名 (e.g. foo.com)
  - value是保存这个域的权威名字服务器的主机
- Type=CNAME
  - name是别名
  - 值是规范名(canonical name)或真名(the real name)

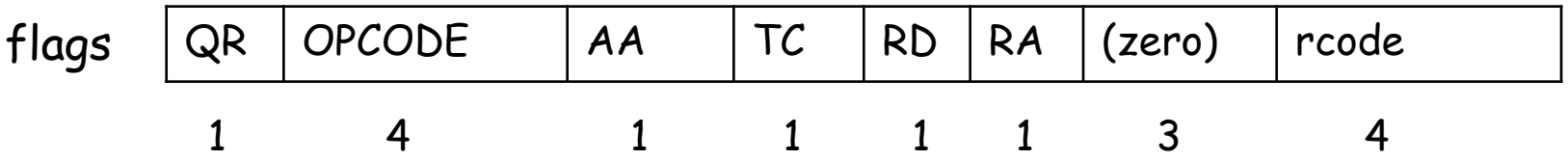
  例如，www.jazz.com是主机bp2.jazz.com的别名
- Type=MX
  - 名是邮件服务器的别名
  - 值是邮件服务器的规范名

* class均为IN(Internet). Type=*AAAA* IPv6

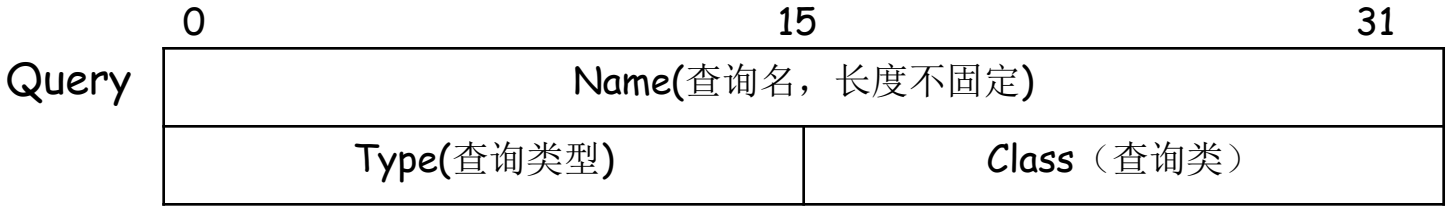
DNS协议的消息格式

DNS协议的查询和应答具有相同的消息格式。应答报文会拷贝查询报文中的标识(identification)。

DNS端口为53，可以使用TCP和UDP协议，TCP协议是用来做区域传送的，UDP协议是用来做DNS解析的



QR（1bit）	查询/响应标志，0为查询，1为响应
opcode（4bit）	0表示标准查询，1表示反向查询，2表示服务器状态请求
AA（1bit）	表示授权回答
TC（1bit）	表示可截断的
RD（1bit）	表示期望递归
RA（1bit）	表示可用递归
rcode（4bit）	表示返回码，0表示没有差错，3表示名字差错，2表示服务器错误（Server Failure）



1 - IN (Internet)

查询名为域名或IP地址（反向查询）：

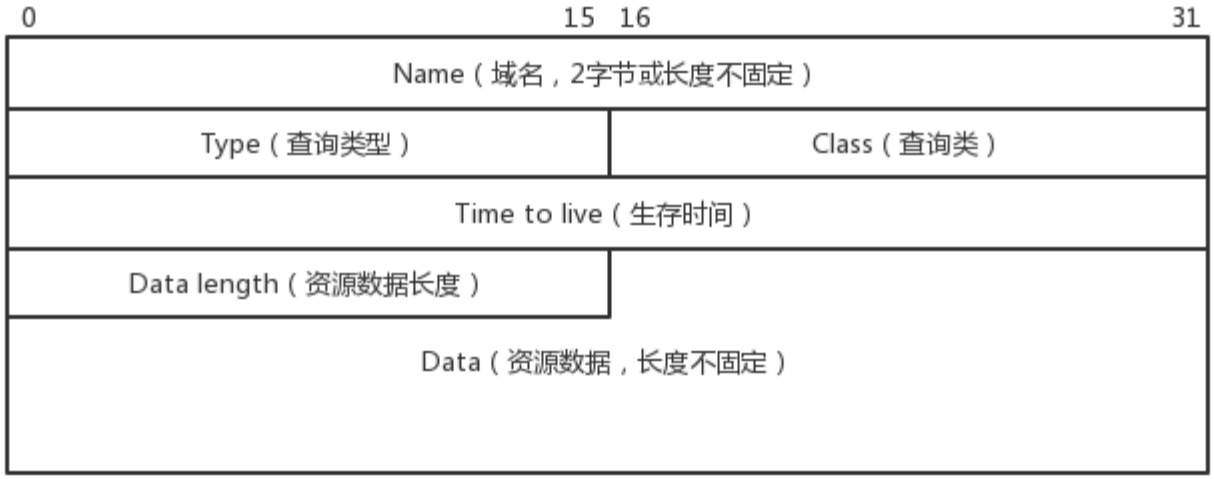
3	w	w	w	4	s	y	s	u	3	e	d	u	2	c	n	0

查询类型(Type)：

类型	助记符	说明
1	A	由域名获得IPv4地址
2	NS	查询域名服务器
5	CNAME	查询规范名称
6	SOA	开始授权
11	WKS	熟知服务
12	PTR	把IP地址转换成域名

类型	助记符	说明
13	HINFO	主机信息
15	MX	邮件交换
28	AAAA	由域名获得IPv6地址
252	AXFR	传送整个区的请求
255	ANY	对所有记录的请求

资源记录(RR)区域（应答，授权和附加）格式相同：

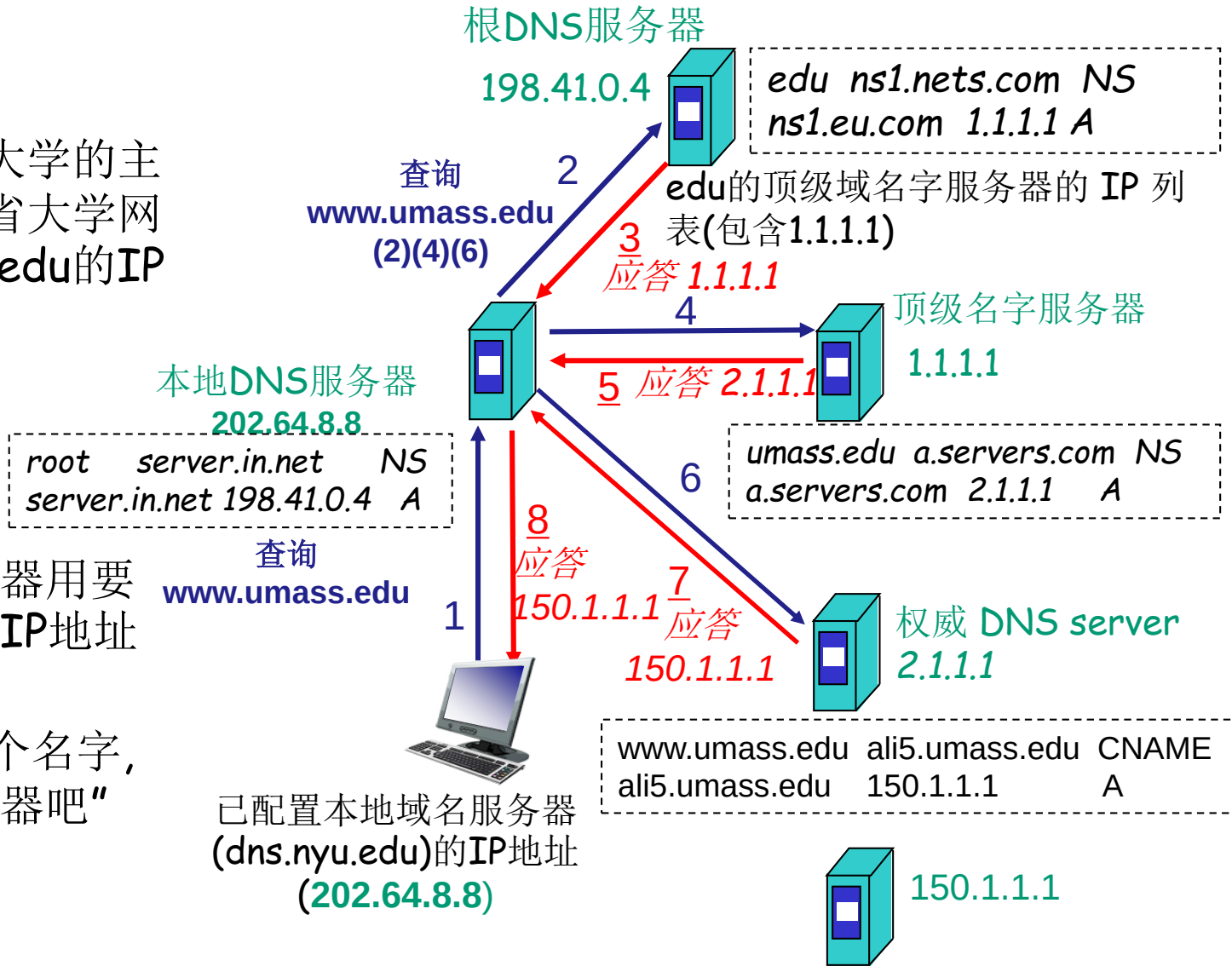


资源记录格式

域名的格式和Queries区域的查询名字字段是一样的。当报文中域名重复时候，该字段使用2个字节的偏移指针来表示。例如，1100000000001100，最前面的11表示重复域名，剩余部分的值从报文头开始计数，指向重复的域名的起始位置，这里的12正好是头部的长度，指向Queries区域的查询名字字段。生存时间（TTL）以秒为单位表示资源记录的生命周期。资源数据是一个可变长字段，表示按照查询段的要求返回的查询结果，例如，IP地址或者CNAME等。

DNS域名解析的例子

- 一台位于纽约大学的主机希望查询麻省大学网站www.umass.edu的IP地址



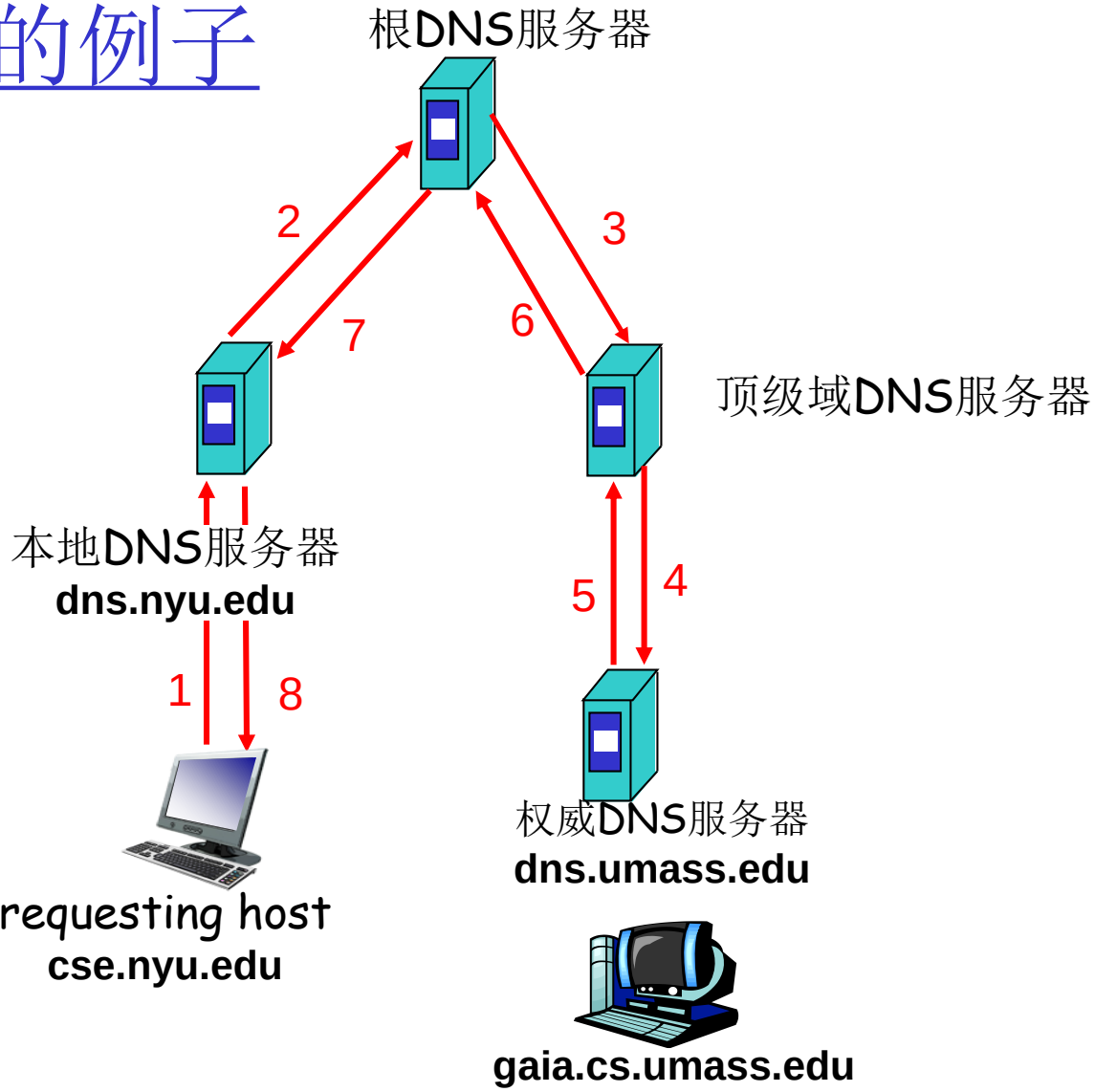
迭代查询:

- 被联系的服务器用要联系的服务器IP地址进行应答
- "我不知道这个名字，去问这个服务器吧"

DNS名字解析的例子

递归查询:

- 把域名解析的负担扔给了所联系的名字服务器
- 哪里的负载重?



DNS: 缓存和更新记录

- 一旦名字服务器获得了映射，它会被缓存起来
  - 在一段时间后缓存的映射会超时失效（消失）
  - TLD服务器通常在本地名字服务器中有缓存，因此根名字服务器并不会被经常访问
- DNS标准制定了更新/通告机制 （RFC 2136）
  - http://www.ietf.org/html.charters/dnsind-charter.html

把记录插入DNS

- 例子：注册域名group.com

  首先在*DNS注册商* (e.g., Network Solutions)
  - 提供存放该域名的权威名字服务器的名字和IP地址 (主服务器和次服务器)
  - 把两个记录插入到顶级域服务器:

  (group.com, dns1.group.com, NS)
  (dns1. group.com, 212.212.212.1, A)

- 权威名字服务器（222.196.200.1）保存四个记录:

  (www.group.com, host1.group.com, CNAME)
  (host1.group.com, 202.116.76.1, A)

  (smtp.mail.group.com, host2.group.com, MX)
  (pop3.mail.group.com, host2.group.com, MX)
  (host2.group.com, 202.116.76.2, A)