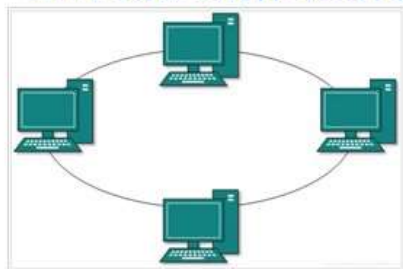


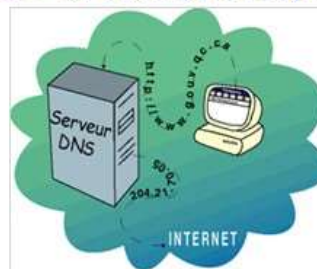
2.1 网络应用体系结构



2.2 网络应用通信原理



2.3 域名解析系统(DNS)



2.4 FTP应用



2.5 Email应用



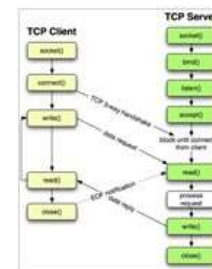
2.6 Web应用



2.7 P2P应用



2.8 Socket编程





2.5 Email应用

刘亚维

56



2.1 网络应用体系结构

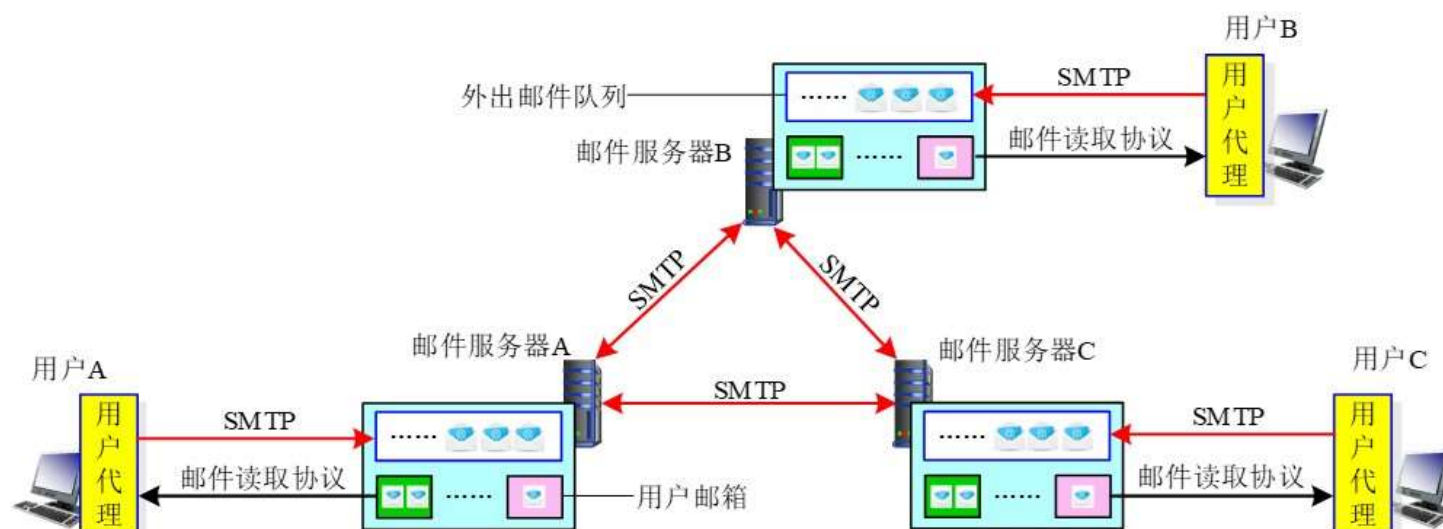
2.2 网络应用通信原理

2.3 域名解析系统(DNS)

2.4 FTP应用

2.5 Email应用

Email应用的构成(1)





- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用

2.5 Email应用



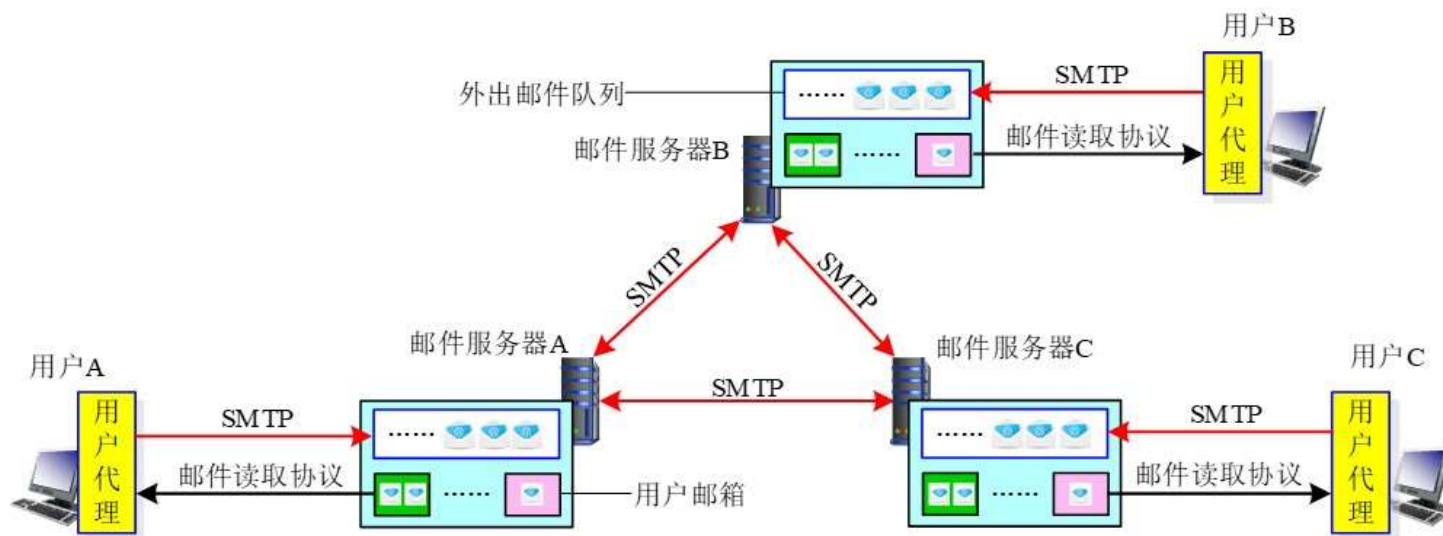
Email应用的构成(1)

❖ Email应用的构成组件

- 邮件客户端(user agent)
- 邮件服务器
- SMTP协议(Simple Mail Transfer Protocol)

❖ 用户代理 (邮件客户端)

- 读、写Email消息
- 与服务器交互, 收、发Email消息
- Outlook, Foxmail, Thunderbird
- Web客户端





- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用

2.5 Email应用



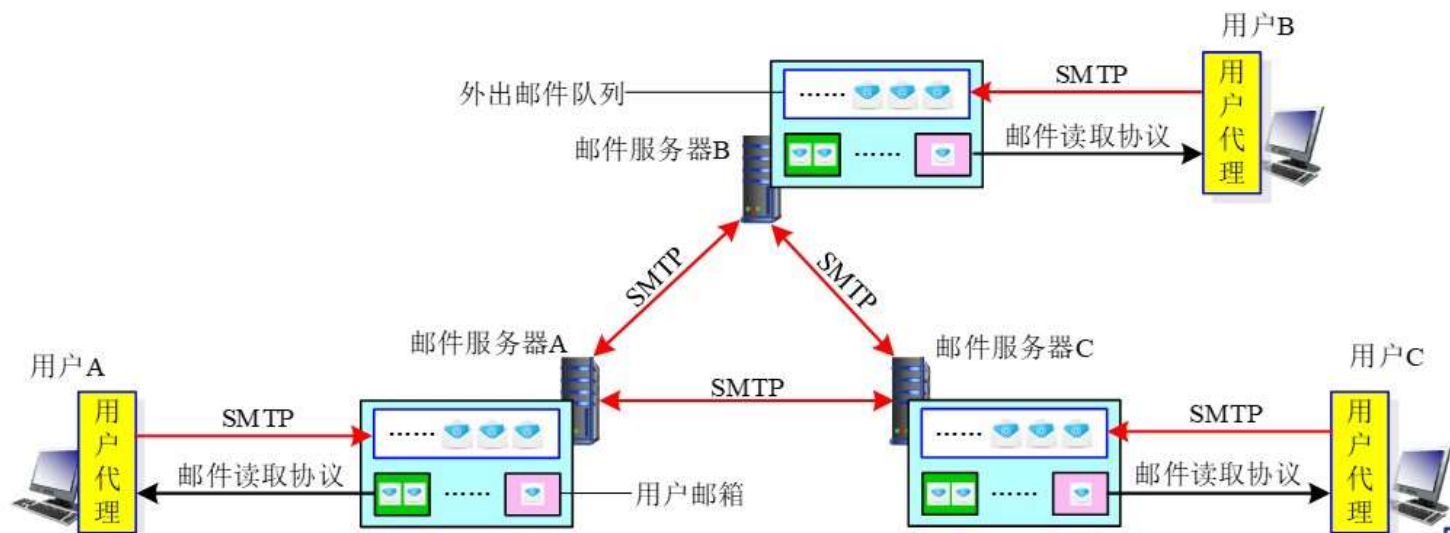
Email应用的构成(2)

❖ 邮件服务器(Mail Server)

- 邮箱：存储发给该用户的Email
- 消息队列(message queue)：存储等待发送的Email

❖ SMTP协议

- 邮件服务器之间传递消息所使用的协议
- 客户端：发送消息的服务器
- 服务器：接收消息的服务器





- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用

2.5 Email应用



SMTP协议: RFC 2821

❖ 使用TCP进行email消息的可靠传输

❖ 端口25

❖ 传输过程的三个阶段:

- 握手
- 消息的传输
- 关闭

❖ 命令/响应交互模式

- 命令(command): ASCII文本
- 响应(response): 状态代码和语句

❖ Email消息只能包含7位ASCII码



- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用

2.5 Email应用



SMTP交互示例

```
S: 220 hamburger.edu
C: HELO crepes.fr
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
```



- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用

2.5 Email应用



Email消息格式

❖ SMTP: email消息的传输/交换协议

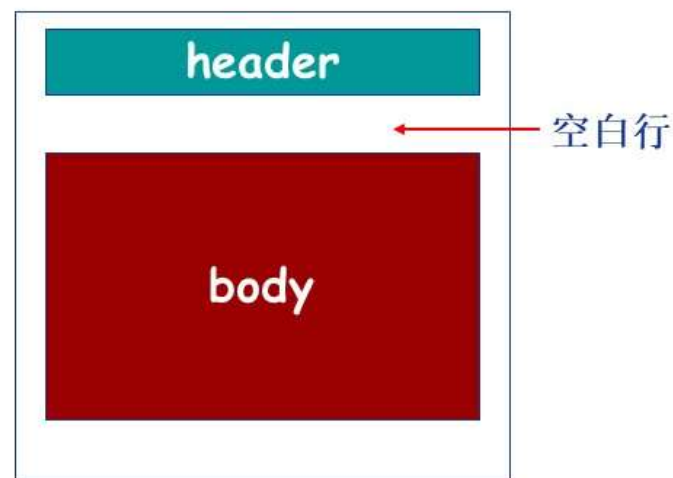
❖ RFC 822: 文本消息格式标准

■ 头部行(header)

- To
- From
- Subject

■ 消息体(body)

- 消息本身
- 只能是ASCII字符





- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用

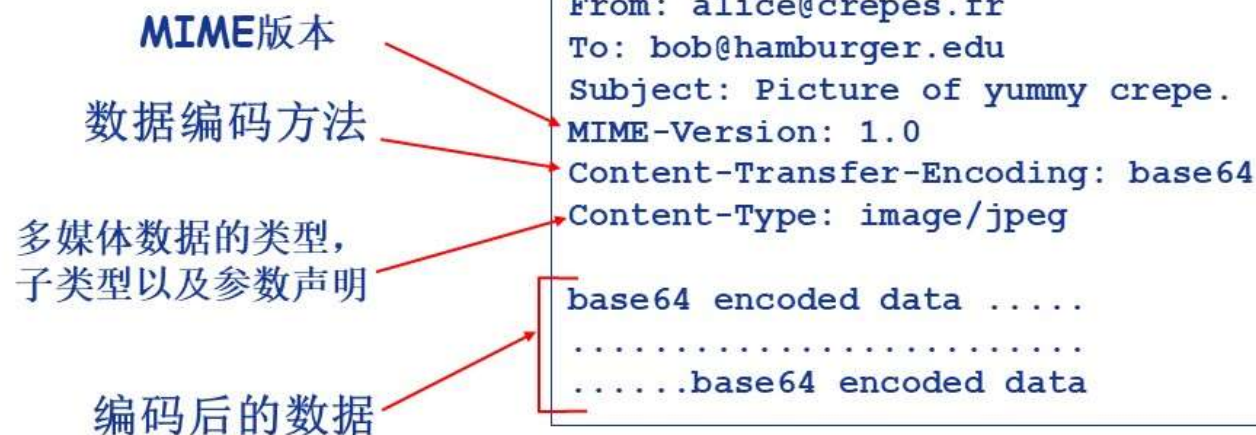
2.5 Email应用



Email消息格式：多媒体扩展

❖ MIME：多媒体邮件扩展 RFC 2045, 2056

- 通过在邮件头部增加额外的行以声明MIME的内容类型



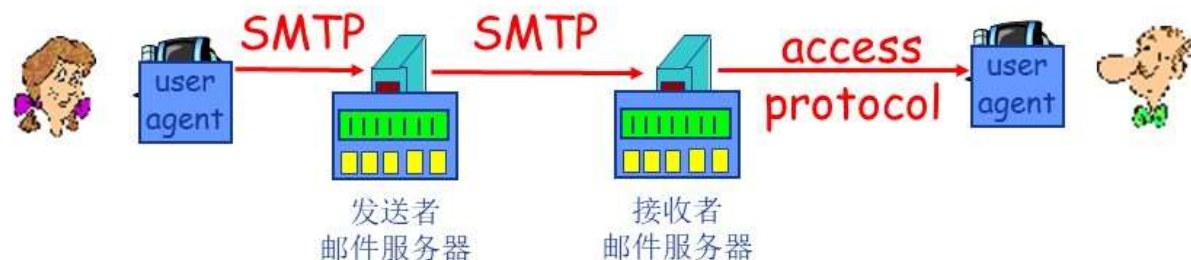


- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用

2.5 Email应用



邮件访问协议



❖ 邮件访问协议：从服务器获取邮件

- POP: Post Office Protocol [RFC 1939]
 - 认证/授权(客户端 \leftrightarrow 服务器)和下载
- IMAP: Internet Mail Access Protocol [RFC 1730]
 - 更多功能
 - 更加复杂
 - 能够操纵服务器上存储的消息
- HTTP: Web Mail



- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用

2.5 Email应用



POP协议

❖ 认证过程

■ 客户端命令

- User: 声明用户名
- Pass: 声明密码

■ 服务器响应

- +OK
- -ERR

❖ 事务阶段

- List: 列出消息数量
- Retr: 用编号获取消息
- Dele: 删除消息
- Quit

```
S: +OK POP3 server ready
C: user bob
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
```



- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用

2.5 Email应用



POP协议

❖ “下载并删除” 模式

- 用户如果换了客户端软件，无法重读该邮件

❖ “下载并保持” 模式

- 不同客户端都可以保留消息的拷贝

❖ POP3是无状态的



- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用

2.5 Email应用



IMAP协议

- ❖ 所有消息统一保存在一个地方：服务器
- ❖ 允许用户利用文件夹组织消息
- ❖ IMAP支持跨会话(Session)的用户状态：
 - 文件夹的名字
 - 文件夹与消息ID之间的映射等

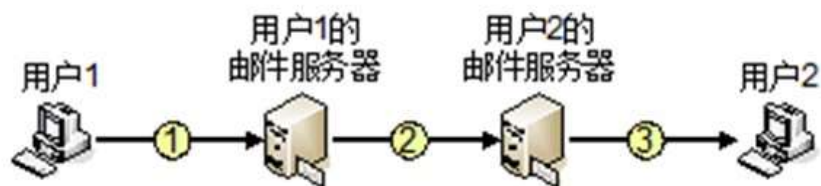
单选题 1分

- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用

2.5 Email应用



若用户1与用户2之间发送和接收电子邮件的过程如下图所示，则图中①、②、③阶段分别使用的应用层协议可以是



- A SMTP、SMTP、SMTP
- B POP3、SMTP、POP3
- C POP3、SMTP、SMTP
- D SMTP、SMTP、POP3**



2.6 Web应用

刘亚维

69



- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用



Web与HTTP

❖ World Wide Web: Tim Berners-Lee



- 网页
- 网页互相链接

❖ 网页(Web Page)包含多个对象(objects)

- 对象: HTML文件、JPEG图片、视频文件、动态脚本等
- 基本HTML文件: 包含对其他对象引用的链接

❖ 对象的寻址(addressing)

- URL (Uniform Resource Locator): 统一资源定位器 (RFC1738)
- Scheme://host:port/path

`www.someschool.edu/someDept/pic.gif`

host name

path name



- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用



HTTP协议概述(1)

❖ 万维网应用遵循什么协议?

❖ 超文本传输协议

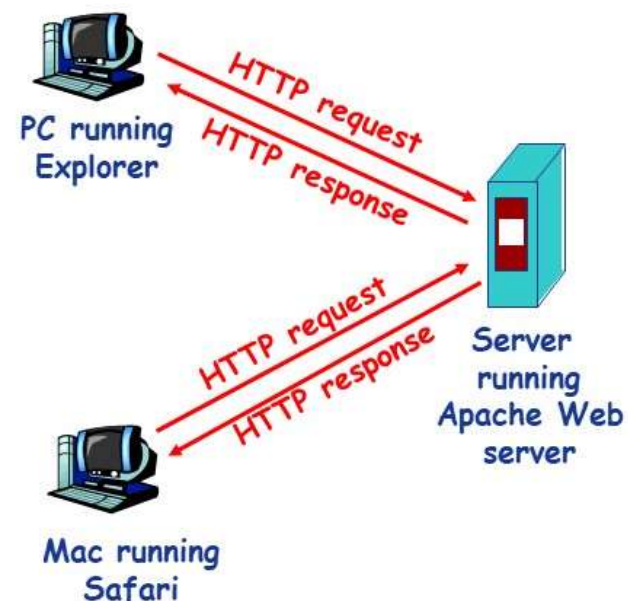
- **HyperText Transfer Protocol**

❖ C/S结构

- **客户—Browser:** 请求、接收、展示Web对象
- **服务器—Web Server:** 响应客户的请求, 发送对象

❖ HTTP版本:

- **1.0:** RFC 1945
- **1.1:** RFC 2068





- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用



HTTP概述(2)

❖ 使用TCP传输服务

- 服务器在80端口等待客户的请求
- 浏览器发起到服务器的TCP连接(创建套接字Socket)
- 服务器接受来自浏览器的TCP连接
- 浏览器(HTTP客户端)与Web服务器(HTTP服务器)交换HTTP消息
- 关闭TCP连接

❖ 无状态(stateless)

- 服务器不维护任何有关客户端过去所发请求的信息



有状态的协议更复杂:

- 需维护状态(历史信息)
- 如果客户或服务失效,会产生状态的不一致,解决这种不一致代价高

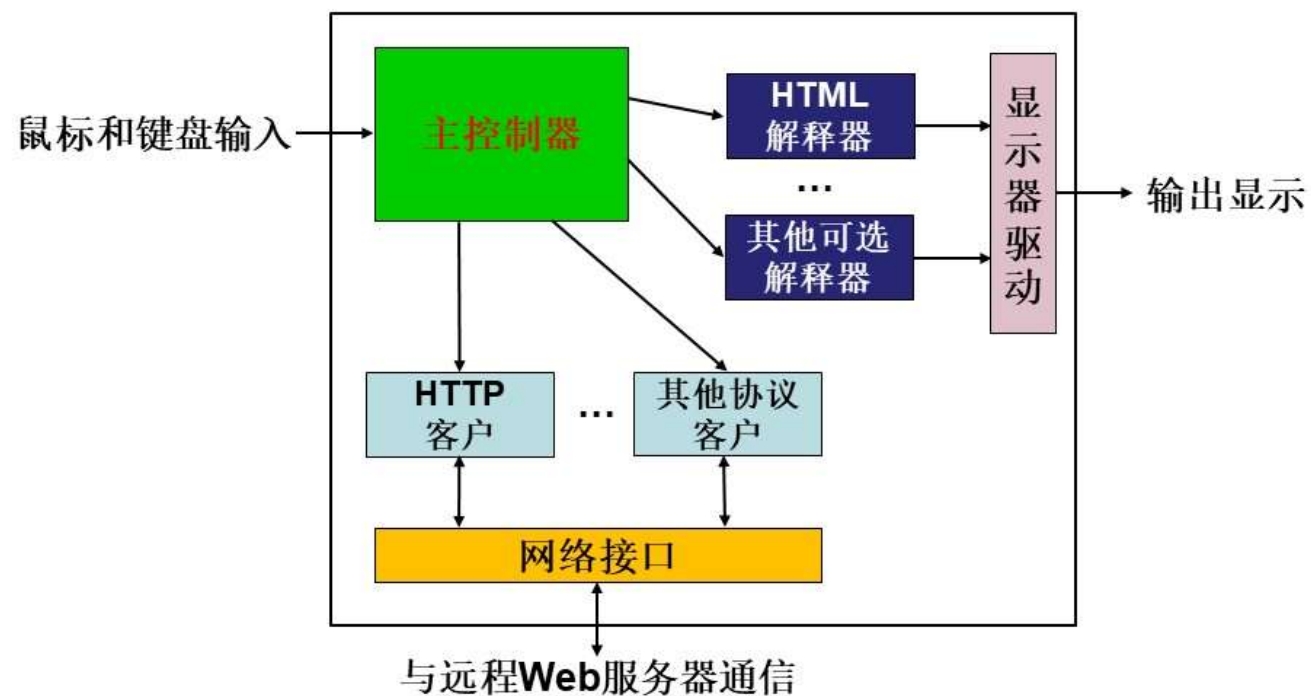


- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用



浏览器





- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用



HTTP请求消息

❖ HTTP协议有两类消息

- 请求消息(request)
- 响应消息(response)

❖ 请求消息

- **ASCII: 人直接可读**

request line
(GET, POST,
HEAD commands)

header
lines

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language: fr
```

Carriage return,
line feed indicates
end of message

(extra carriage return, line feed)



2.1 网络应用体系结构

2.2 网络应用通信原理

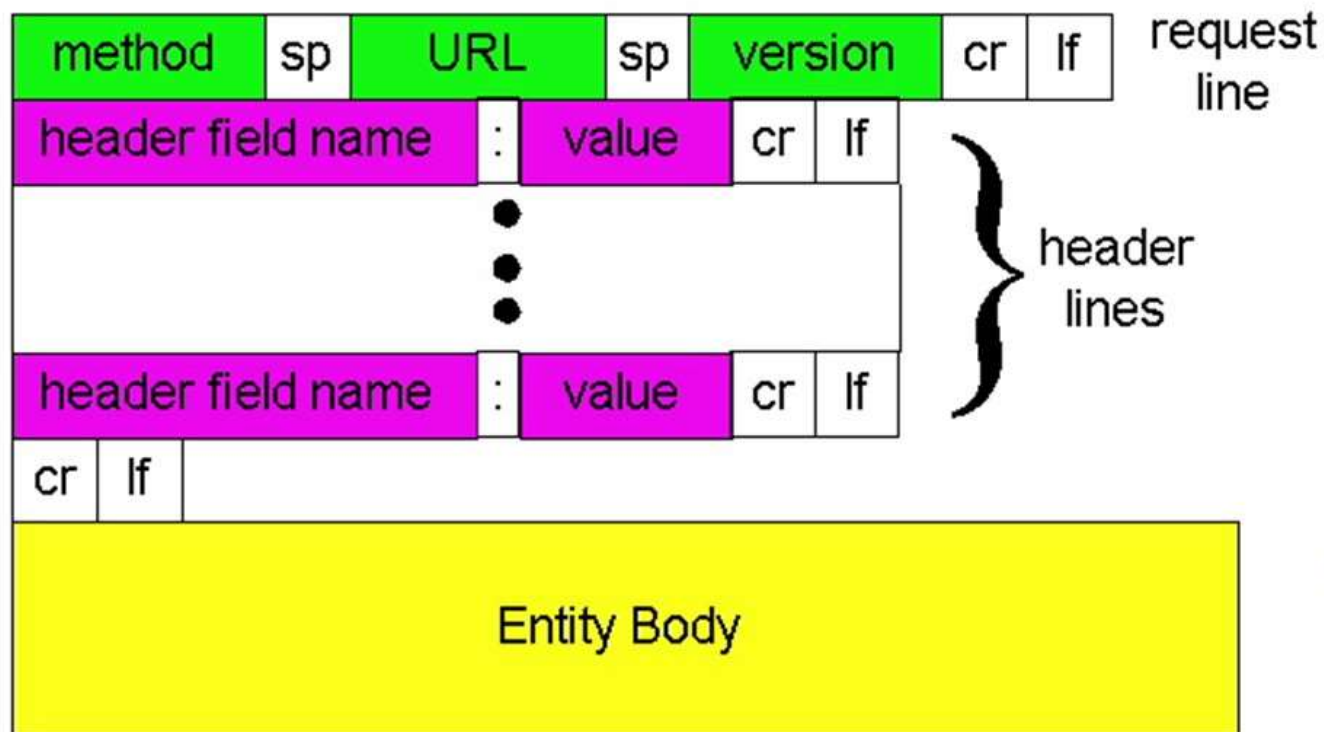
2.3 域名解析系统(DNS)

2.4 FTP应用

2.5 Email应用

2.6 Web应用

HTTP请求消息的通用格式





- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用



上传输入的方法

❖ POST方法

- 网页经常需要填写表单(form)
- 在请求消息的消息体(entity body)中上传客户端的输入

❖ URL方法

- 使用GET方法
- 输入信息通过request行的URL字段上传

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



- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用



方法的类型

❖ HTTP/1.0

- GET
- POST
- HEAD
 - 请Server不要将所请求的对象放入响应消息中

❖ HTTP/1.1

- GET, POST, HEAD
- PUT
 - 将消息体中的文件上传到URL字段所指定的路径
- DELETE
 - 删除URL字段所指定的文件



- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用



HTTP响应消息

**status line
(protocol
status code
status phrase)**

HTTP/1.1 200 OK

**header
lines**

Connection: close

Date: Thu, 06 Aug 1998 12:00:15 GMT

Server: Apache/1.3.0 (Unix)

Last-Modified: Mon, 22 Jun 1998

Content-Length: 6821

Content-Type: text/html

**data, e.g.,
requested
HTML file**

data data data data data ...



- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用



HTTP响应状态代码

❖ 响应消息的第一行

❖ 示例

- 200 OK
- 301 Moved Permanently
- 400 Bad Request
- 404 Not Found
- 505 HTTP Version Not Supported

404 Not Found

nginx



无法找到该网页

最可能的原因是:

- 在地址中可能存在键入错误。
- 当您点击某个链接时, 它可能已过期。

您可以尝试以下操作:

- 重新键入地址。
- 返回到上一页。



- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用



HTTP/1.0 请求-响应过程

❖ HTTP/1.0

- 每个TCP连接传输一个对象
- 非持久性连接(Nonpersistent HTTP)
- 该版本使用非持久性连接





- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用



HTTP/1.0 请求-响应过程(1)

假定用户在浏览器中输入URL:

`www.someSchool.edu/someDepartment/home.index`

`home.index`包含
文本和指向10个
jpeg图片的链接

1a. HTTP客户端向地址为
`www.someSchool.edu`的服务
器上的HTTP服务器进程(端口
80) 发起TCP连接请求。

1b. HTTP服务器在端口80等待
TCP连接请求, 接受连接并通
知客户端。

2. HTTP客户端将HTTP请求消
息(包含URL地址)通过TCP连
接的套接字发出, 消息中所含
的URL表明客户端需要对象
`someDepartment/home.index`

3. HTTP服务器收到请求消息,
解析, 产生包含所需要对象
的响应消息, 并通过套接字
发给客户端。

时间 5. HTTP客户端收到响应消息, 解
析html文件, 显示html文件, 发
现有10个指向jpeg对象的超连接

4. HTTP服务器关闭TCP连接。

82



- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用



HTTP/1.0 请求-响应过程(1)

2. HTTP客户端将HTTP请求消息(包含URL地址)通过TCP连接的套接字发出, 消息中所含的URL表明客户端需要对象 `someDepartment/home.index`

3. HTTP服务器收到请求消息, 解析, 产生包含所需要对象的响应消息, 并通过套接字发给客户端。

4. HTTP服务器关闭TCP连接。

时间 5. HTTP客户端收到响应消息, 解析html文件, 显示html文件, 发现有10个指向jpeg对象的超连接

6. 为每个jpeg对象重复步骤1-5。

❖ HTTP/1.0

- 每个TCP连接传输1个对象
- 非持久性连接(Nonpersistent HTTP)



- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用



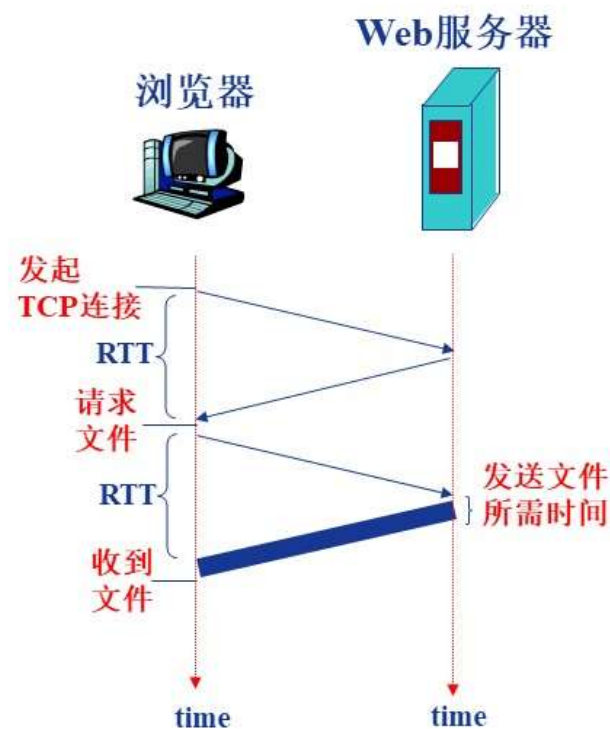
HTTP/1.0响应时间分析

❖ RTT(Round Trip Time)

- 从客户端发送一个很小的数据包到服务器并返回所经历的时间

❖ 响应时间(Response time)

- 发起、建立TCP连接: **1个RTT**
- 发送HTTP请求消息到HTTP响应消息的前几个字节到达: **1个RTT**
- 响应消息传输时间
- 响应时间:
=2RTT + 响应消息发送时间





- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用



HTTP响应时间优化-并行连接

❖ 非持久性连接的问题

- 每个对象需要2个RTT
- 操作系统需要为每个TCP连接开销资源(overhead)
- 浏览器可以怎么做?
 - 打开多个**并行TCP连接**以获取网页所需对象
 - 给服务器端造成什么影响?
- 例如：请求浏览一个引用3个JPEG**小图像**的Web页



2.1 网络应用体系结构

2.2 网络应用通信原理

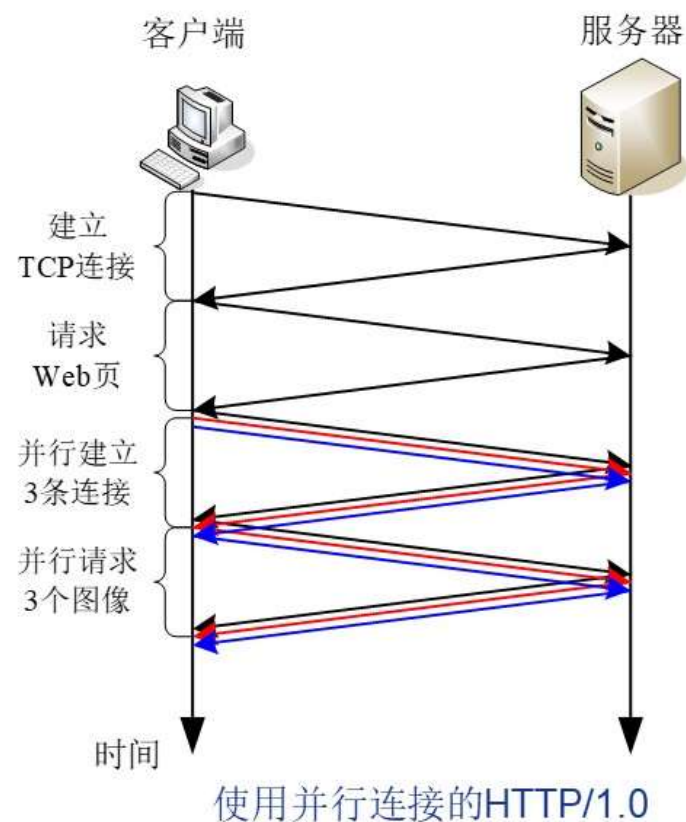
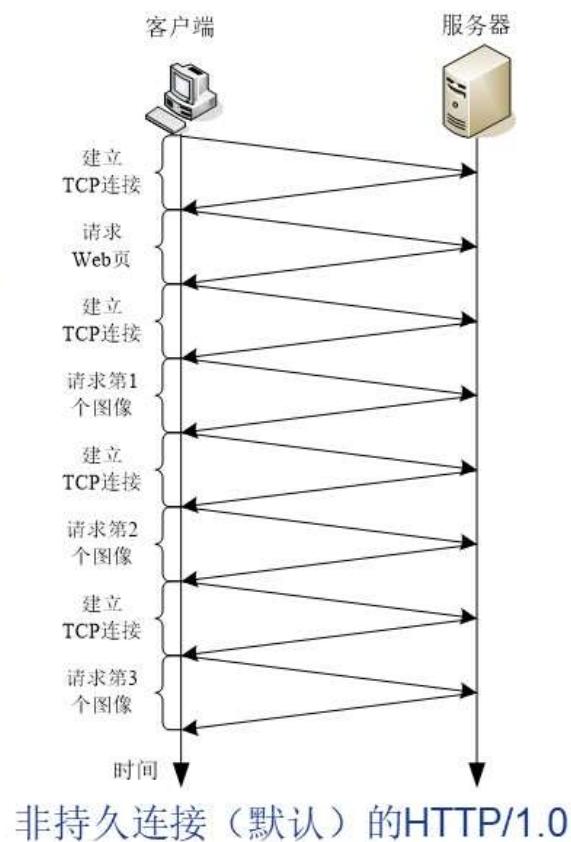
2.3 域名解析系统(DNS)

2.4 FTP应用

2.5 Email应用

2.6 Web应用

HTTP响应时间优化-并行连接





- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用
- 2.6 Web应用**



HTTP响应时间优化-持久连接

❖ 持久连接(Persistent HTTP)

- 发送响应后，服务器保持TCP连接打开
- 后续HTTP消息可以通过这个连接发送
- 每个TCP连接允许传输**多个**对象

❖ 非流水(pipelining) 持久连接

- 客户端只有收到前一个响应后才发送新的请求
- 每个被引用的对象耗时1个RTT

❖ 流水的持久连接

- 客户端只要遇到一个引用对象就尽快发出请求
- 理想情况下，收到所有的引用对象只需耗时约1个RTT
- **HTTP/1.1**版本默认使用流水的持久连接
- 例如：请求浏览一个引用3个JPEG**小图像**的Web页

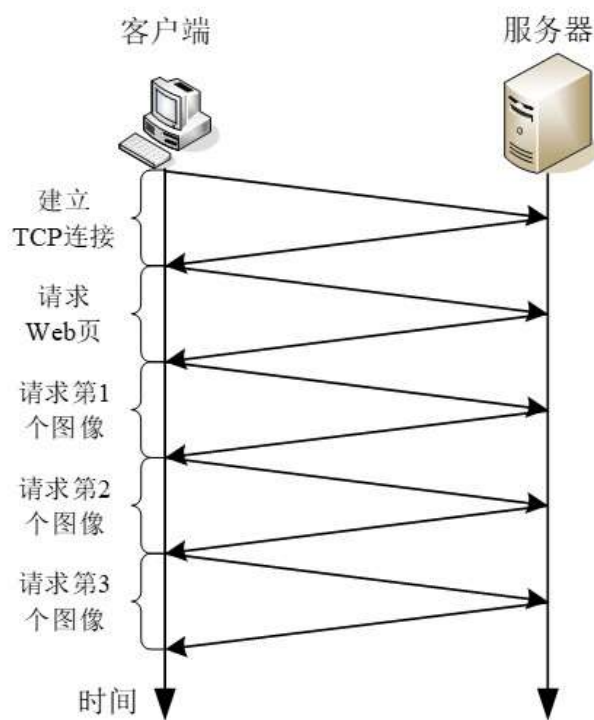


- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

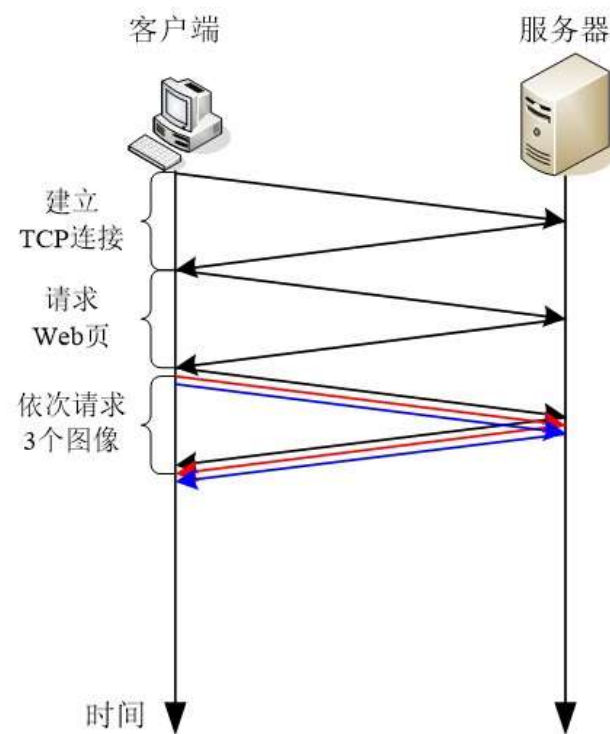
2.6 Web应用



HTTP响应时间优化-持久连接



非流水持久连接的HTTP/1.1



流水持久连接的HTTP/1.1



- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用



HTTP响应时间优化- HTTP/2

❖ HTTP/1.1有什么不足？考虑通过HTTP/1.1连续请求多个大小不同的对象，可能会发生什么问题？

- 例：通过HTTP 1.1连续请求1个大对象(如视频文件)和3个小对象





- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

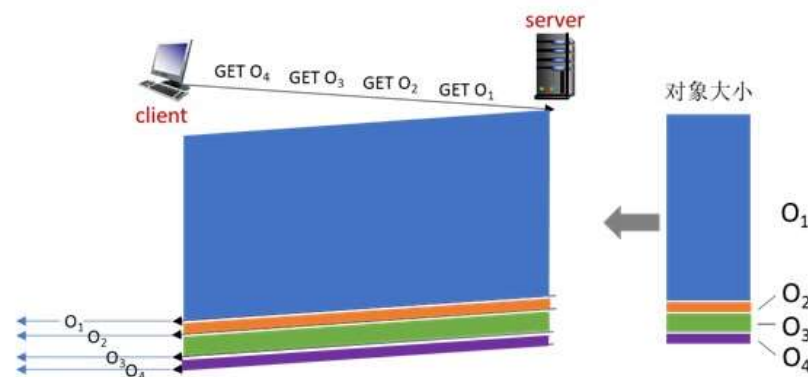
2.6 Web应用



HTTP响应时间优化- HTTP/2

❖ HTTP/1.1

- 服务器顺序响应 (FCFS调度)
- 在大对象之后请求的小对象, 需要等待大对象的传输 (**head-of-line (HOL) blocking**)
- 若发生数据丢失会暂停对象传输





- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

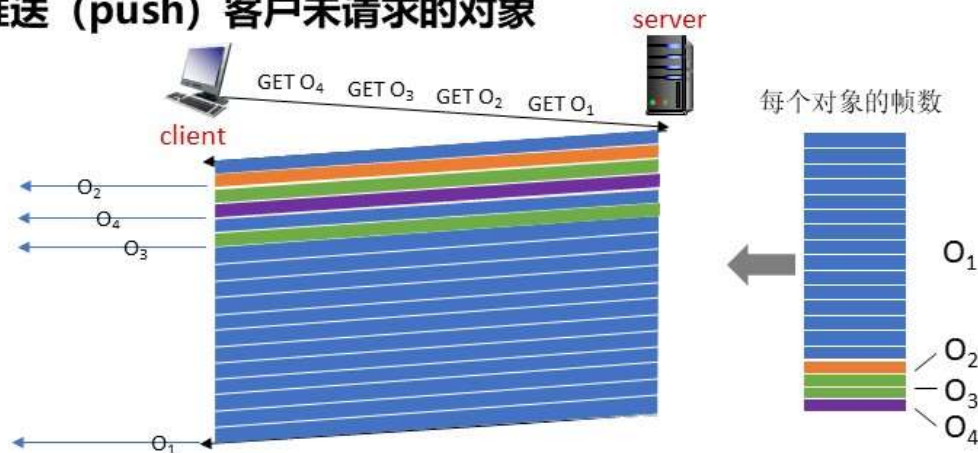
2.6 Web应用



HTTP响应时间优化- HTTP/2

❖ HTTP/2 [RFC 7540, 2015] : 减少多对象请求的时延 (响应时间)

- 将对象分割为系列“帧”，不同对象的帧交替传输
- 与HTTP/1.1相比，方法、状态码、多数首部行等没变
- 对象传输顺序基于客户定义的对象优先级（非FCFS调度）
- 推送（push） 客户未请求的对象





- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用



HTTP响应时间优化- HTTP/3

❖ HTTP/2进一步优化?

- 使用单一的TCP连接
- 丢包恢复仍然会暂停对象传输

❖ 传输层功能?

- TCP、UDP: 主要传输协议, 已超过40年
- 针对特定场景, TCP的具有不同“风格”:

场景	挑战
长、胖管道（大量数据传输）	许多数据包在“飞行”；丢包将关闭管道
无线网络	由于无线链路噪声、移动等造成的丢包；TCP 将此视为拥塞丢包
长延迟链路	超长的 RTT
数据中心网络	延迟敏感
后台流量	低优先级的“后台”TCP 流

❖ 在 UDP 之上实现应用，将传输层功能移到应用层

- HTTP/3: QUIC



- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用

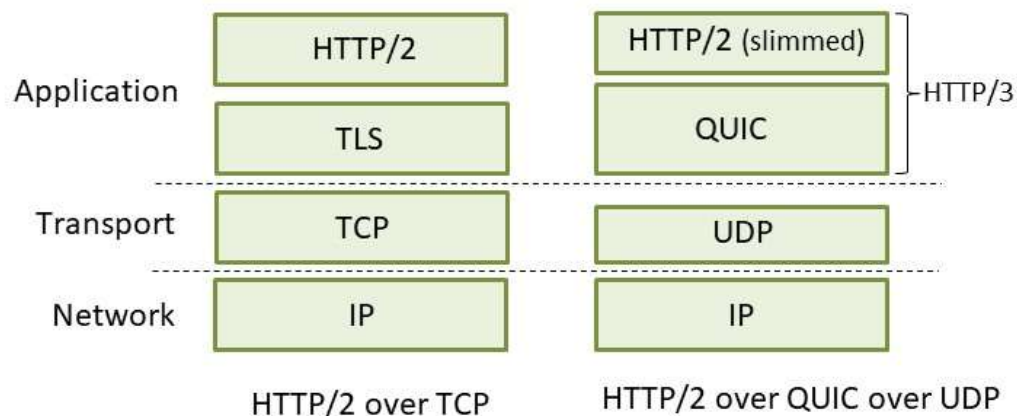


HTTP响应时间优化- HTTP/3

❖ QUIC: Quick UDP Internet Connections?

❖ 基于UDP的应用层协议

- 提高 HTTP 的性能
- 部署在许多谷歌服务器、APP应用程序 (Chrome, 移动 YouTube应用程序) 上





- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

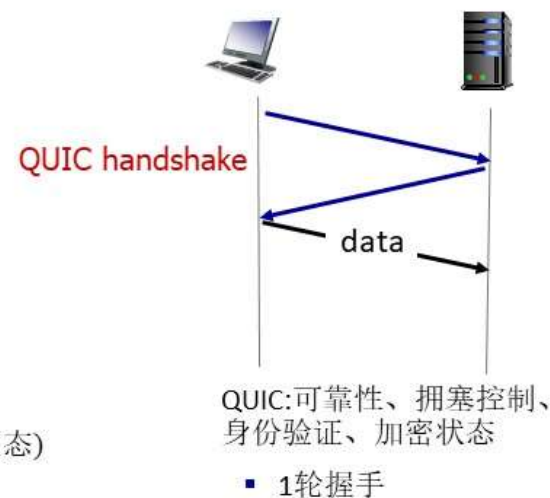
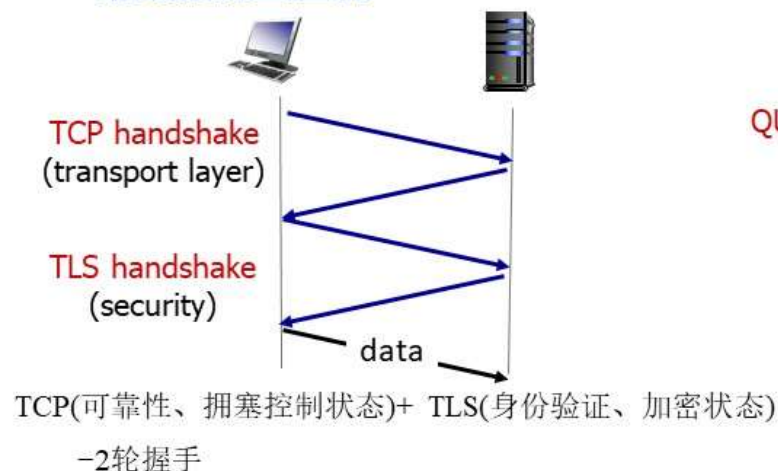
2.6 Web应用



HTTP响应时间优化- HTTP/3

❖ QUIC: Quick UDP Internet Connections ?

- 应用层实现连接建立、错误控制、拥塞控制等功能
 - **差错控制和拥塞控制**: 采用类似于 TCP 的丢失检测和拥塞控制的算法
 - **连接建立**: 可靠性、拥塞控制、身份验证、加密等在1个 RTT 建立状态
- 多个应用级“流”复用在一个 QUIC 连接
 - 可靠数据传输与安全分离
 - 常见的拥塞控制





2.1 网络应用体系结构

2.2 网络应用通信原理

2.3 域名解析系统(DNS)

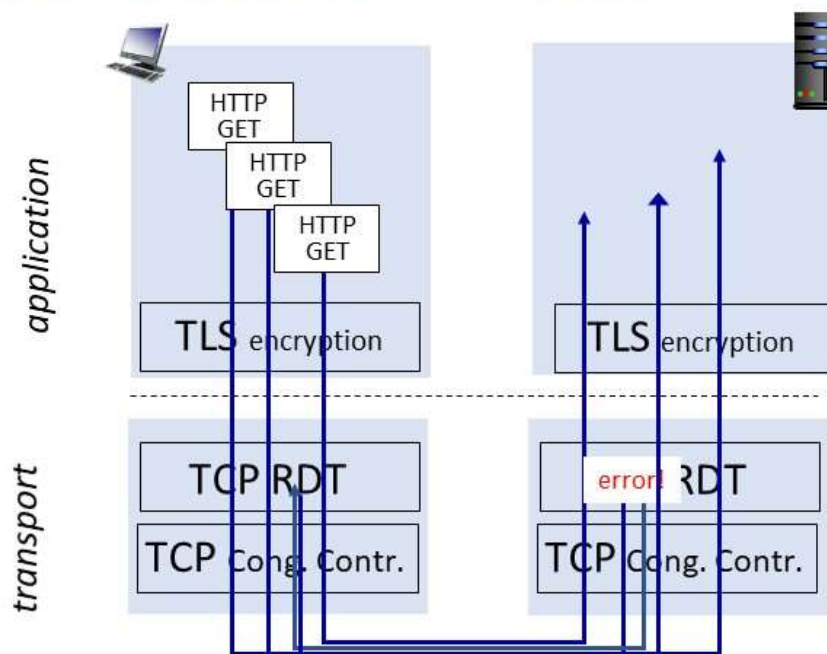
2.4 FTP应用

2.5 Email应用

2.6 Web应用

HTTP响应时间优化- HTTP/3

❖ QUIC流：并行性、无 HOL 阻塞



(a) HTTP 1.1



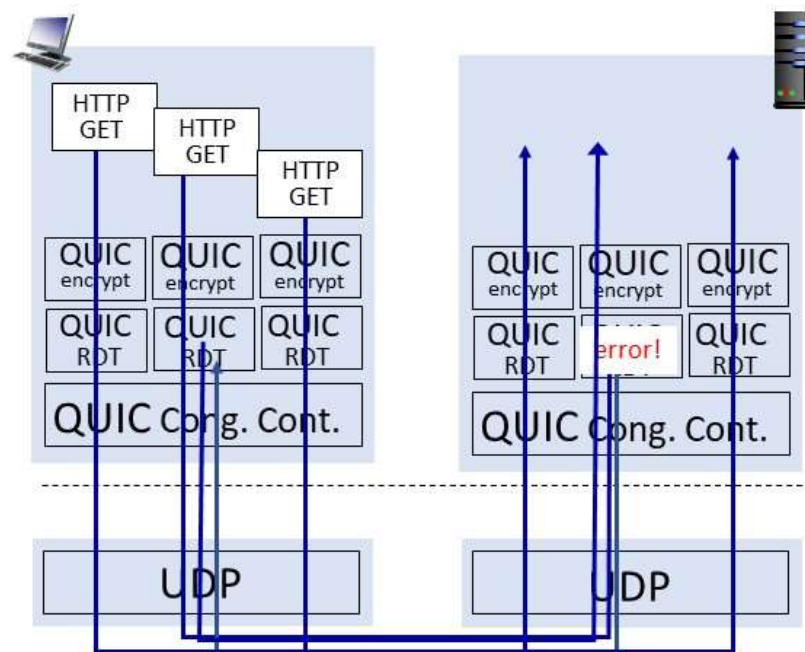
- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用



HTTP响应时间优化- HTTP/3

❖ QUIC流：并行性、无 HOL 阻塞



(b) HTTP/2 with QUIC: no HOL blocking



- 2.1 网络应用体系结构
- 2.2 网络应用通信原理
- 2.3 域名解析系统(DNS)
- 2.4 FTP应用
- 2.5 Email应用

2.6 Web应用



HTTP响应时间优化-Web缓存/代理服务器

❖ 目标

- 在不访问源服务器的前提下满足客户端的HTTP请求。

❖ 为什么要发明这种技术?

- 缩短客户请求的响应时间
- 减少机构/组织的流量
- 在大范围内(Internet)实现有效的内容分发

