

# 《计算机网络》讲义要点

## 第二章 应用层 Application Layer

赵增华  
天津大学智能与计算学部  
2023 年春季

### 参考教材：

“Computer networking: a top-down approach”, by Jim Kurose, Keith Ross, Pearson, 8th Edition, 2021 年。

**辅助教材：**“Computer Networks”，第 5 版英文影印版，A. Tanenbaum, 清华大学出版社，2011 年。

### 教学目标：

通过对 Internet 经典协议的学习和剖析，让学生理解网络系统设计、网络协议设计所面临的问题和常用的解决方法。课程结束后学生能够根据工作/科研的需要设计、实现相应的网络系统和协议，并进行性能评价。授课过程中强调对问题的描述（why），然后是解决方案（how）。

### 学习方法：

学而不思则罔，思而不学则殆。

多读，多思，多实践。

读：通过大量阅读理解网络、协议的基本概念，原理。阅读推荐的教材、参考书、自行在网上查找的相关内容。

思：思考为什么。网络系统/每层（个）协议设计面临什么问题？如何解决？为什么？

实践：1) 通过网络协议分析软件（如 Wireshark）深入理解各层协议的执行过程。（Wireshark Lab）

2) 通过设计、实现网络协议，掌握协议的设计方法和实现技术。（Socket programming, RDT, routing protocol）

3) 通过网络构建、配置，理解网络的实际部署和管理。（Lab Practice）

## 第二章 应用层 Application Layer

### 1. 深刻理解应用层在网络协议栈中的位置及主要功能。(2.1)

应用层协议在协议栈最上层，为网络应用提供端到端的网络服务。在端系统中实现。

网络应用是分布式的，分布在不同主机/服务器的应用程序使用“应用层协议”传输/交互数据，协作完成网络应用的功能。

“网络应用”和“应用层协议”的区别：如图 1-1 所示

网络应用包括：应用层协议和其它非数据传输的功能。

应用层协议：数据传输。

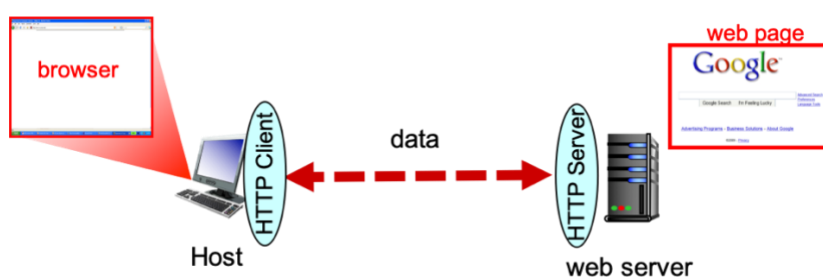


图 1-1 网络应用和应用层协议的关系，以 Web 应用和 HTTP 协议为例

### 2. 协议设计的基本方法，以应用层为例

应用层协议的设计目标：根据网络应用的需求和所需要传输数据的特点，使用传输层提供的服务，为网络应用提供满足需求的网络（数据传输）服务。

#### 1) 网络应用的需求分析

数据所在的位置：集中在服务器上/分布在参与应用的所有主机上。

需要的网络服务：可靠的数据传输/不可靠的数据传输。

达到的性能指标：需要确定性能参数。

#### 2) 协议方案设计

协议架构：C/S; P2P

传输层 Socket: TCP（面向连接）; UDP（无连接）

协议规则的设计：在数据传输过程中需要遵循的规则。

协议头部的设计：协议头部用于交互协议控制信息。需定义协议头部的格式、含义。

#### 3) 协议实现

根据设计方案来实现协议。

#### 4) 协议性能评价

协议性能评价的方法：理论分析；仿真；实验。

性能评价参数：延迟、吞吐率、丢包率等。

### 3. 理解应用层协议的两种架构：C/S、P2P。（2.2）

从数据共享的角度来看，

C/S：数据集中放在 Server 上，Client 上没有数据，只能从 Server 获取。

P2P：每个主机都可能有一部分数据，需要通过数据传输获取自己需要的数据。即主机间是对等的关系（peers），都可以做为 Server 提供数据，也可以做为 Client 获取数据。

### 4. 熟练掌握 HTTP 协议设计基本原理，两种工作机制：持续性（persistent）和非持续性（non-persistent）的传输延迟的性能分析。了解 HTTP 的扩展功能：cookies，web caching。（2.3）

以 HTTP 为例，深入理解协议设计的基本方法：需求分析；方案设计（协议规则和头部信息）；协议性能评价。

Web 服务的需求：服务器以页面形式存储了大量数据，客户端需要访问服务器获取所需要的数据。

HTTP 协议设计：

C/S 架构：使用 TCP Socket 传输数据。

从 TCP 连接的角度，HTTP 的协议规则分为两种：

Non-persistent HTTP：每个 object 传输时建立一个 TCP 连接，传输完毕后 TCP 连接关闭。

Persistent HTTP：多个 objects 共享一个 TCP 连接。

协议头部：HTTP 的 client 端和 server 端位了完成协议所需要交互的信息。

定义了格式、含义，接收方就可以解析出头部信息，根据规则做相应的处理。

性能评价方法：理论分析。根据规则建立数据传输模型。

性能评价参数：响应时间。

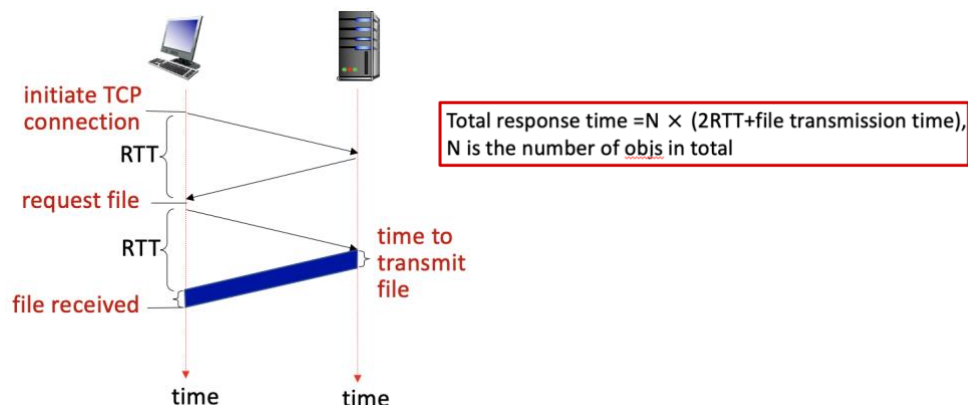


图 4-1 Non-persistent HTTP 的响应时间示意图。假设 objects 文件和 base-HTML 文件的长度都相等。

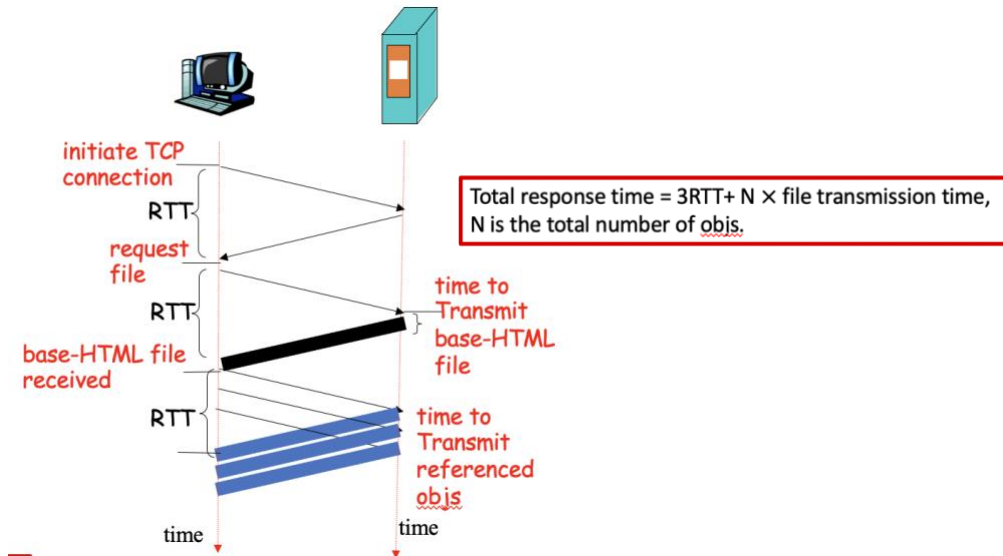


图 4-2 Persistent HTTP 的响应时间示意图。假设 objects 文件和 base-HTML 文件的长度都相等。

### Web Caching:

使用 Web proxy 把本地访问过的页面缓存起来，提高本地访问页面的速度。假设：同一个局域网（区域）里，用户访问页面的兴趣点相似，很多用户会访问相同的页面。

存在问题：缓存页面的命中率对 web caching 有很大影响。

### 5. 了解 email 的协议设计，SMTP，POP3 协议。

Email 协议的需求：像邮政系统一样，在用户间传输电子信件。

协议架构：C/S； TCP socket.

多个协议完成 email 数据传输需求：SMTP，POP3，HTTP。

SMTP: server 间传输电子信件。采用 push 的方式。用户也可以使用 SMTP 把电子信件 push 到 server。

POP3: 用户从 server 接收邮件。

HTTP: 用户也可以使用 HTTP 访问 server，实现邮件的发送和接收。

### 6. 理解 DNS 采用分布式数据库的原因。了解 DNS 两种查询方式：递归查询（recursive queries）和迭代查询（iterated queries）的工作原理。（2.5）

DNS 本质是一个分布式数据库，要解决数据的生成、存储、查询等问题。

协议需求：存储 host name 和 IP 地址的映射关系；用户可以高效查询。

协议架构：C/S； UDP socket

数据库 server 架构：层次型架构。

协议规则：数据查询方式：recursive query; iterated query.

协议头部：完成 DNS 协议需要交互的控制信息。

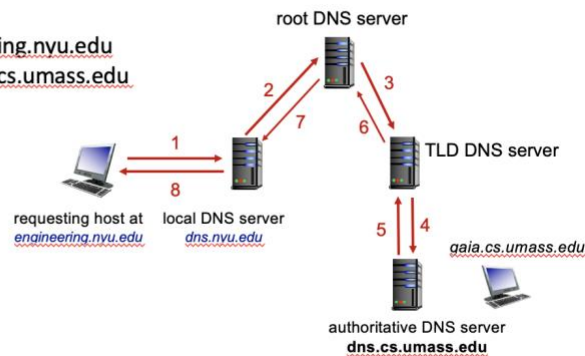
协议设计小技巧：TTL（生存期，Time To Live）。DNS 使用 TTL 来自动更新缓存的数据。

## DNS name resolution: recursive query

**Example:** host at engineering.nyu.edu wants IP address for gaia.cs.umass.edu

### Recursive query:

- puts burden of name resolution on contacted name server
- heavy load at upper levels of hierarchy?



## DNS name resolution: iterated query

**Example:** host at engineering.nyu.edu wants IP address for gaia.cs.umass.edu

### Iterated query:

- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"

