

CH2 应用层

一、应用层协议原理

1. 网络应用程序体系结构

CS(client-server)体系结构

服务器总是打开，服务各种请求。

服务器具有固定的IP地址。

客户之间不直接通信。

需要建立大型数据中心应对高强度客户端并发请求。

P2P体系结构

主机与主机相连，双方地位对等。

自扩展性强，不需要庞大的服务器基础设施。

2. 进程通信

服务器和进程使用唯一标识符通过socket软件窗口进行通信，见课程设计实验。

使用IP地址区分主机，使用端口号区分主机中的不同网络应用。

二、Web和HTTP

1. HTTP

HTTP，即超文本传输协议，由两个程序实现：客户端程序和服务器程序，“拉协议”。

无状态，不维护客户信息。

Web页面，即文档，由文件组成。

2. 非持续连接和持续连接

非持续连接：每个（请求，响应）使用一个单独的TCP发送，HTTP1.0使用

持续连接：多个（请求，响应）经相同的TCP连接发送，HTTP1.1使用

非持续连接的HTTP：

1. 客户端向服务器发起tcp连接请求，建立socket关联
2. 客户端经socket向服务器发送HTTP请求报文
3. 服务器经socket接收该请求报文，并发送响应
4. 服务器通知TCP断开该TCP连接（确认客户已收到完整响应报文）
5. 客户端接收响应报文，TCP连接关闭

RTT（往返时间）：一个短分组从客户到服务器然后再返回客户的时间

RTT包括传播时延，排队时延和处理时延

持续连接的HTTP：

非持续http的问题：为每个请求建立TCP导致服务器负担大，双倍RTT时延（tcp建立+传输）

实现：服务器在发送响应后保持TCP连接打开，包含TimeOut机制

3. HTTP响应时间模型

使用两个RTT建立TCP连接，其余为传输时间。

非持续连接：每个对象需要两个RTT

持续连接：两个RTT建立连接，其余对象每个需要一个RTT

4. HTTP报文格式

请求报文：请求行+首部行

返回报文：头部+实体

见socket实验。

5. HTTP方法类型

HTTP/1.0: GET, POST, HEAD（不包含请求对象→故障跟踪）

HTTP/1.1: GET, POST, HEAD, PUT（上载），DELETE（删除）

5. Cookie

cookie用来让服务器标识用户，跟踪用户信息。

作用：个性化推荐。

6. Web缓存

Web缓存器，又叫代理服务器，用来保存客户端最近请求过的对象的副本。

客户端↔代理服务器↔初始服务器

降低服务器响应时间，降低通信量

三、电子邮件

1. 组成部分

用户代理+邮件服务器+简单邮件传输协议（SMTP）

2. 传输过程

代理A→服务器A→SMTP→服务器B→代理B

1. 主机A撰写包含主机B的邮箱地址的报文
2. 主机A通过用户代理发送给邮件服务器，进入报文队列
3. 服务器创建TCP连接并发送报文
4. 另一台服务器收到报文，转发给主机B的代理
5. 主机B调用用户代理阅读邮件

SMTP一般不使用中间邮件服务器，“推协议”。

四、DNS

1. 概述

DNS（域名系统），是因特网的目录服务。

一个由**分层的DNS服务器**实现的分布式数据库，一个使得主机能够查询分布式数据库的应用层协议。

运行在UDP上，使用53号端口。

主机使用**主机名**标识。（网址）

DNS将主机名解析为IP地址。

- 规范主机名：（classes.tju.edu.cn）
- 主机别名：（saa.tju.edu.cn）【解析为规范主机名】
- 邮件服务器别名：（163.com/126.com）

2. 工作原理

使用**分布式层次数据库**，为了处理扩展性问题。

DNS服务器分为三种类型：根DNS服务器，顶级域DNS服务器（TLD），权威DNS服务器

顶级域:

通用：.com, .edu, .gov, .int, .mil, .net, .org, .firm, .web;

国家：.cn, .us, .nl, .jp

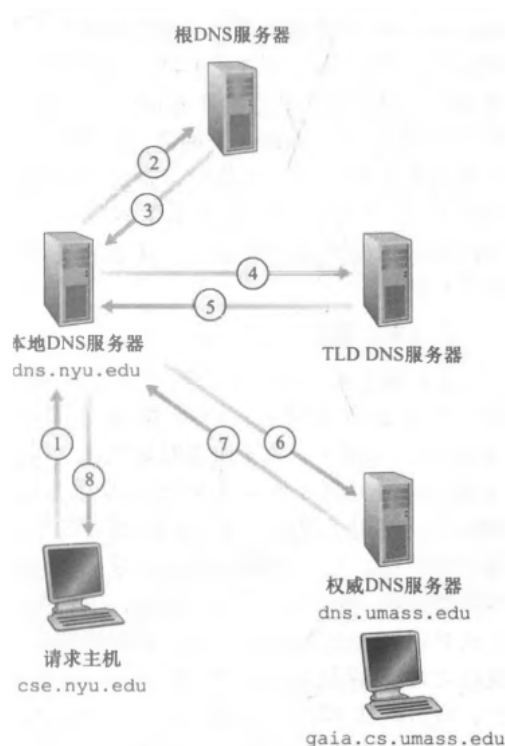
分类

根DNS服务器：提供TLD服务器的IP地址（解析.cn）

顶级域DNS服务器：返回.com对应的IP地址（.cn）

权威DNS服务器：返回amazon.com对应的IP地址（edu.cn）

本地DNS服务器：存在于每个ISP中。（tju.edu.cn）



主机请求IP地址的过程：

1. 主机向本地DNS服务器请求saa.tju.edu.cn的IP地址
2. 本地DNS服务器转发至根DNS服务器
3. 根DNS服务器注意到edu.cn的后缀，将负责edu.cn的TLD的IP地址列表返回至本地DNS服务器
4. 本地DNS服务器再次向TLD服务器发送请求报文
5. TLD服务器注意到tju.edu.cn，将对应权威DNS服务器的IP地址列表返回
6. 本地DNS服务器向权威DNS服务器发送请求报文
7. 权威DNS服务器注意到saa.tju.edu.cn，将对应IP地址返回
8. 本地DNS服务器返回saa.tju.edu.cn的IP地址，请求主机收到响应

递归查询：

```
IP = AuthorDNS(TldDNS(RootDNS(LocalDNS(addr))))
```

迭代查询：

```
1 def LocalDNS(addr):
2     IP1 = RootDNS(addr)
3     IP2 = TldDNS(IP1)
4     IP3 = AuthorDNS(IP2)
5     return IP3
6 def client():
7     addr = 'saa.tju.edu.cn'
8     IP = Local(addr)
```

上述例子中，仅1和8为递归查询，其余均为迭代查询，且一共发出/收到了四份查询报文和四份响应报文

3. DNS缓存

本地DNS服务器能够将映射（主机→IP）规则存储在缓存内，再次进行查询时能够直接从本地DNS服务器返回主机对应的IP地址。

4. DNS记录和报文

(Name, Value, Type, TTL)四元组。TTL为记录生存时间，下面不讨论。

- Type = A, 主机名→IP地址, (relay1, bar.foo.com, 145.37.93.125, A)
- Type = NS, 域→IP地址, (foo.com, dns.foo.com, NS)
- Type = CNAME, 别名主机→规范主机名, (saa.tju.edu.cn, classes.tju.edu.cn, CNAME)
- Type = MX, 别名邮件→规范主机名, (foo.com, mail.foo.com, MX)

五、P2P文件分发

1. 概述

Web, mail, DNS均采用C/S结构。

传输多媒体文件更适宜使用P2P结构。

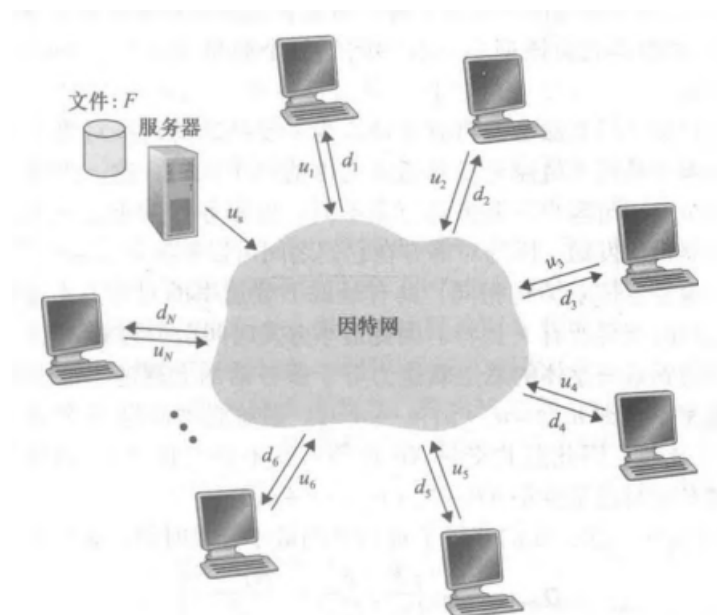


图 2-21 文件分发问题的示例图

2. 性能指标

u_s 表示服务器接入链路上载速率

u_i 表示第*i*对等方接入链路上载速率

d_i 表示第*i*对等方接入链路的下载速率

N 表示要获得的该文件副本的对等方的数量

分发时间 D_{cs} 是所有*N*个对等方得到该文件的副本所需要的时间

令 $d_{min} = \min\{d_1, d_2, \dots, d_N\}$

C/S结构: $D_{cs} = \max\{\frac{NF}{u_s}, \frac{F}{d_{min}}\}$

客户端下载时间的最大值

取服务器上载*N*个文件副本时间和下载最慢的客

P2P结构: $D_{cs} = \max\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_s + \sum u_i}\}$

据, 最大上载能力为所有*u*的求和

最后一项的含义是, 网络中一共流通*NF*字节数