

第二章

R4

不同意这种说法。第七版书中 57 页给出了客户和服务器的定义。在 P2P 文件共享应用程序中，接收文件的对等点通常是客户机，发送文件的对等点通常是服务器。

R5

使用目标主机的 IP 地址和目标进程中套接字的端口号来标识运行在另一台主机上的进程。（教材 P59）

R6

使用 UDP。使用 UDP，可以在一次往返时间(RTT)内完成——客户端将事务请求发送到 UDP 套接字，服务器将响应发送回客户端 UDP 套接字。对于 TCP，至少需要两个往返时间(RTT)——一个用于设置 TCP 连接，另一个用于客户机发送请求，服务器发送响应。

R9

SSL 运行在应用层。

用 SSL 强化 UDP 没有意义。若是用 SSL 来强化 TCP，则需要在应用程序中包含 SSL 代码。（教材 P61）

P1

- a. 错误。客户将会发送四个请求报文。
- b. 正确。
- c. 错误。第七版书中 66 页“值得注意的是每一个 TCP 连接只传输一个请求报文和一个响应报文。
- d. 错误。第七版书中 69 页“Date: 首部行指示服务器产生并发送该响应报文的日期和时间。
- e. 错误。在条件式请求时，对象在服务器端没有改变，HTTP 响应报文的头部代码是: 304 时，为空的报文体。

R13

Web 缓存器可以使得用户期望的内容离用户更“近”，用户可以在更小的时延内访问缓存器中的内容，而不用访问远端的服务器；除更“近”外，客户与 Web 缓存器之间的瓶颈带宽通常远大于客户与服务器间。

Web 缓存器将减少用户请求**所有对象**的时延。对于命中的内容，通过缓存器

发送可以减小时延。与此同时，缓存器减少了用户和服务器链路上的流量强度，所以它也可以减少访问没有被命中的对象的时延。

(CH1 P14 画过排队时延与流量强度的关系图)

R16

因为 Ailce 的电子邮件账户基于 Web，所以她通过 HTTP 协议访问自己的邮件服务器。Alice 的邮件服务器通过 SMTP 协议将邮件推至 Bob 的邮件服务器。

由题目知，Bob 通过 POP3 协议将邮件从服务器拉至主机。

(本题 HTTP 协议见书 P83)

P5

1. 可以成功的找到文档 (200 OK) , Tue, 07 Mar 2008 12:39:45 GMT (Date 处)
2. Sat, 10 Dec 2005 18:27:46 GMT (Last-Modified 处得出)
3. Content-Length: 3847
4. <!doc 同意一条持续连接 (Keep-Alive) (Content-Type 后的内容)

P9

(a) 跨越接入链路发送一个对象的平均时间 $\Delta = L/R = \frac{850000}{15000000} s = 0.0567s$

对象对该接入链路的平均到达率 $\beta = 16 \text{ resquest/s}$

所以平均接入时延 $t_a = \Delta / (1 - \Delta\beta) = 0.6s$

总的平均响应时间 $t = t_a + t_r = 0.6s + 3s = 3.6s$ 其中 t_r 为接收到响应的平均时间

- (b) 因为命中率为 0.4，所以接入链路上的流量强度减少了 40%

平均接入时延变为 $t_a = \Delta / (1 - (1 - 0.4) * \Delta\beta) = 0.124s$

未命中对象的平均响应时间变为 $t_1 = t_a + t_r = 0.124s + 3s = 3.124s$

因为缓存器在 LAN 处，所以命中对象的平均响应时间可以看作为 $t_2 = 0s$

所以总的响应时间为 $t = 0.6 * 3.124s + 0.4 * 0s = 1.874s$

P10

发起一次 TCP 连接要经过三次握手，即要经过三次控制数据传输的时间（1.5RTT），之后再行数据的传输，设 T_p 为传播时延。

(1) 首先考虑非持续 HTTP，在进行初始对象下载后建立 10 个并行 HTTP 下载引用对象在载初始对象的时间为

$$t_1 = 3 * \left(\frac{200}{150} + T_p \right) + \left(\frac{100000}{150} + T_p \right)$$

上式中前一部分为建立连接所用的时间，后一部分为数据传输所用的时间
下载引用对象的时间为

$$t_2 = 3 * \left(\frac{200}{150/10} + T_p \right) + \left(\frac{100000}{150/10} + T_p \right)$$

上式中前一部分为建立连接所用的时间，后一部分为数据传输所用的时间
所以使用非持续 HTTP 的总时间为

$$T_1 = t_1 + t_2 = 7377 + 8T_p(s)$$

(2) 再考虑持续 HTTP，下载初始对象所用时间和非持续 HTTP 相同

$$t_3 = 3 * \left(\frac{200}{150} + T_p \right) + \left(\frac{100000}{150} + T_p \right)$$

因为是持续连接，所以不用再建立 TCP 连接了，下载引用对象的时间为

$$t_4 = 10 * \left(\frac{200}{150} + T_p + \frac{100000}{150} + T_p \right)$$

所以使用持续 HTTP 的总时间为

$$T_2 = t_3 + t_4 = 7351 + 24T_p(s)$$

由于是 10 米短链路，所以传播时延 T_p 可以忽略不计。

可以发现持续和非持续 HTTP 下载所用时间几乎相同，相差不超过 1%，所以持续连接没有比非持续连接得到更多增益。因为从上式中可以看出，持续连接仅仅比非持续连接减

少了两次控制数据传输时间 $2 * \frac{200}{150/10}$ 。在主要部分数据传输上并没有增益。

P11

- a. 能， Bob 的并行连接能使他获得更大的链路带宽。
- b. 仍有好处，若 Bob 采用非并行连接，则相比其他 4 个采用并行连接的用户， Bob 获得的带宽比较小。

R19

一个机构的 Web 服务器和邮件服务器可以有完全相同的主机名别名（第 7 版课本 P89）。

MX 记录维护邮件服务器的别名到 IP 地址的映射关系。

R22

每过 30 秒，BitTorrent 中的对等方将**随机**地选择一个邻居向其发送块，一旦 Alice 被选中，即可得到她的第一个文件块。

R23.

覆盖网络指应用层上的逻辑网络，包括所有活动的对等方和边。

不包括路由器。

边指应用进程间的 TCP 连接，即逻辑的指向关系。

P25

在对应的覆盖网络中，共有 N 个节点， $N*(N-1)/2$ 条边。

P22

1) C-S 模式

U\N	10	100	1000
300bps	7680	51200	512000
700bps	7680	51200	512000
2Mbps	7680	51200	512000

$$D_{cs} = \max \left\{ \frac{NF}{u_s}, \frac{F}{d_{\min}} \right\}$$

2)

U\N	10	100	1000
300bps	7680	25600	46545.5
700bps	7680	15360	21041.1
2Mbps	7680	7680	7680

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

P23

a. 分发方案：服务器以 u_s/N 的速率并行地将文件发送到每个客户端。

由于 $u_s/N \leq d_{\min}$ 此速率不大于客户端每个的下载速率。每个客户端可以以速率 u_s/N 接收，因此每个客户端接收整个文件的时间是 $F/(u_s/N) = NF/u_s$ 。所有客户端都在以 NF/u_s 接收文件，因此整体分发时间也是 NF/u_s 。

b. 分发方案：服务器以 d_{\min} 速率并行地将文件发送到每个客户端。由于 $(u_s/N) \geq d_{\min}$ ，则 $N \cdot d_{\min}$ 不大于服务器的上载速率 u_s ，由于每个客户端以速率 d_{\min} 接收，每个客户端接收整个文件的时间是 F/d_{\min} ，此时所有客户端都收到文件，因此总的分发时间也是 F/d_{\min} 。

c. 由书中 2.5 节可知： $D_{CS} \geq \max \left\{ \frac{NF}{u_s}, \frac{F}{d_{\min}} \right\}$;

在 a 中， $u_s/N \geq d_{\min}$ ，可得 $NF/u_s \geq F/d_{\min}$;

在 b 中， $u_s/N \leq d_{\min}$ ，可得 $F/d_{\min} \geq NF/u_s$;

综上最小分发时间由 $\max \{NF/u_s, F/d_{\min}\}$ 所决定。

P24

a. 分发方案：设 $u = u_1 + u_2 + \dots + u_N$ 则根据条件有：

$u_s \leq (u_s + u)/N$ 将文件分为 N 部分，第 i 部分大小为 $(u_i/u) F$ ，服务器以速率 $r_i = (u_i/u) u_s$ 将第 i 个部分发给对等方。由于 $r_1 + r_2 + \dots + r_N = u_s$ ，聚合转发速率不会超过服务器的上载速率。每个对等点以速率 r_i 将它接受的比特发给其他 $N-1$ 个对等方，则结点 i 的聚合转发速率为 $(N-1) r_i$ ，根据题中条件 $u_s \leq (u_s + u)/N$ 可得：

$$(N-1)r_i = (N-1)(u_s u_i)/u \leq u_i$$

因此对等方聚合转发速率不大于其上载速率 u_i ，在该分发方案中，对

等方接收比特的总速率为：
$$r_i + \sum_{j \neq i} r_j = u_s$$

因此每个对等方接收文件的时间为 F/u_s 该分发方案的分发时间为 F/u_s

b. 分发方案：设 $u = u_1 + u_2 + \dots + u_N$ 则根据条件有：

$u_s \geq (u_s + u)/N$, 令 $r_i = u_i/(N-1)$, $r_{N+1} = (u_s - u/(N-1))/N$. 在此分发方案中, 将文件分为 $N+1$ 份。服务器以速率 r_i 将第 i 个部分发送到第 i 个对等体 ($i = 1, 2, \dots, N$)。每个对等体 i 将接收到的比特以速率 r_i 发给其余 $N-1$ 个对等体。此外, 服务器将第 $N+1$ 个部分的比特以速率 r_{N+1} 发送给 N 个对等方。对等方不转发来自第 $N+1$ 部分的比特。服务器的聚合发送速率是:

$$r_1 + r_2 + \dots + r_N + N \cdot r_{N+1} = u/(N-1) + u_s - u/(N-1) = u_s$$

因此服务器聚合转发速率不会超过服务器的上载速率。对等方的聚合转发速率为:

$$(N-1)r_i = u_i$$

因此, 每个对等方的聚合转发速率不会超过其上载速率。在这种分发方案中, 对等方接收比特的总速率为:

$$r_i + r_{N+1} + \sum_{j \neq i} r_j = u/(N-1) + (u_s - u/(N-1))/N = (u_s + u)/N$$

因此每个对等方接收文件的时间为 $NF/(u_s + u)$ 该分发方案的分发时间为 $NF/(u_s + u_1 + u_2 + \dots + u_N)$.

c. 同理 P23. c: 由书中

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

由于 d_{\min} 很大, 所以 F/d_{\min} 不考虑。结合 a, b 不难得出最小分发时间由 $\max \{F/u_s, NF/(u_s + u_1 + u_2 + \dots + u_N)\}$ 决定。