**R5.运行在一台主机上的进程,使用什么信息来标识运行在另一台主机上的进程?**

见P60

使用 目的主机的IP和进程对应的端口号 来标识这台主机上的进程

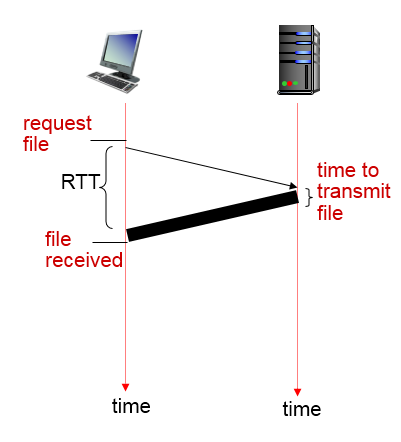
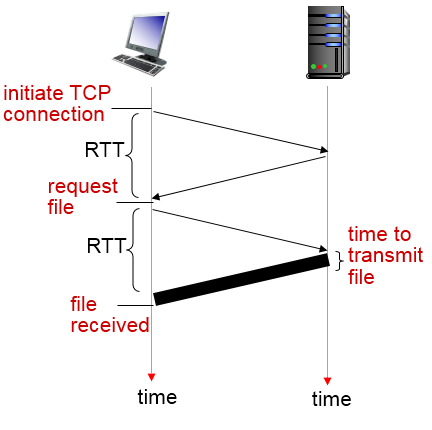
**R6. 假如你想尽快地处理从远程客户到服务器的事务,你将使用TCP还是UDP,为什么?**

见P68页

使用UDP.

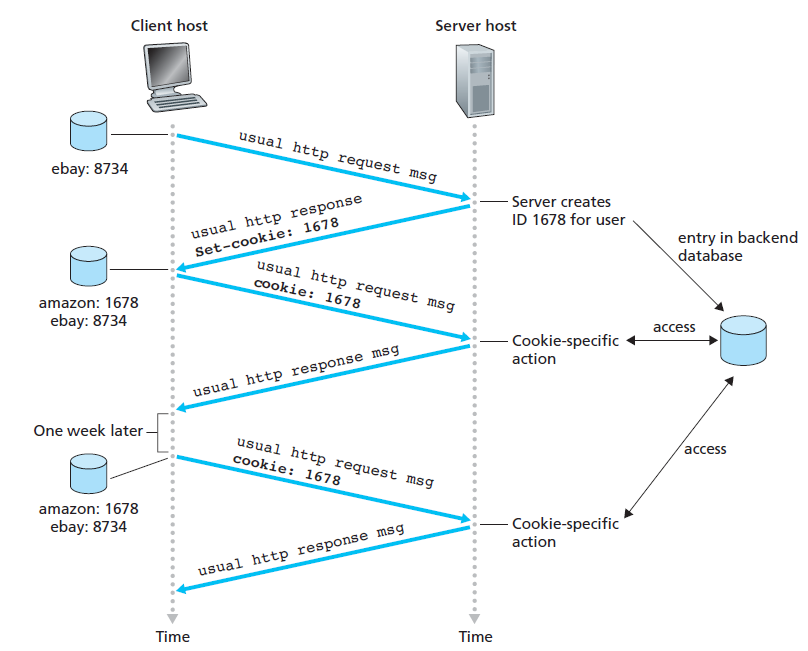
左图是使用TCP协议传输文件用时，共2RTT+传输文件用时

右图是使用UDP协议传输文件用时，共RTT+传输文件用时



**R12.考虑一个电子商务网站需要保留每一个客户的购买记录。描述如何使用Cookie来完成该功能？**

参考P73 图2-10



用户表：

|  |
| --- |
| id1 |
| id2 |
| … |
| idn |

购买记录表：

|  |  |
| --- | --- |
| id1 | 购买记录1 |
| id1 | 购买记录2 |
| … | … |

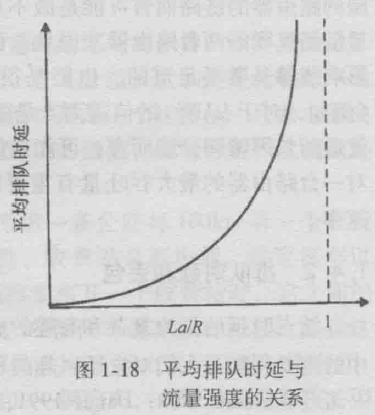
当用户第一次访问这个网站时服务器会为这个用户生成一条ID保存在服务器数据库中，同时通过响应报文在用户的浏览器中生成一条此ID对应的Cookie，并保存在此浏览器中，之后用户每次访问这个网站时，请求中都会带有cookie,网站就知道这个用户是谁，同时将其购买记录存储在数据库中

**R13.描述Web缓存器是如何减少接受被请求的对象的试验的，Web缓存器将减少一个用户请求的所有对象或只是其中的某些对象的时延吗？为什么？**

见P74划线部分

web缓存器（代理服务器）一般设置在离用户较近（用户到web缓存器的转发跳数少）的地方，它会保存用户最近访问的页面的副本，这样当用户或者其他用户访问这个页面时，直接从web缓存器中获取即可，这样就相比直接访问原始服务器减少了时延。

web缓存虽然只能缓存一部分的页面，但是因为它减少了网络上的流量强度，所以相当于它减少了所有对象的访问时延。流量强度影响的是交换机里的平均排队时延（总时延的一部分）。



**P1.是非判断题。**

**a.假设用户请求由某些文本和3幅图像组成的Web页面。对于这个页面，客户将发送一个请求报文 并接收4个响应报文。**

**b.两个不同的 Web 页面（例如，www. mit.edu/research. html 及 www. mit.edu/students.html)可以通过同一个持续连接发送。**

**c.在浏览器和初始服务器之间使用非持续连接的话，一个TCP报文段是可能携带两个不同的HTTP 服务请求报文的。**

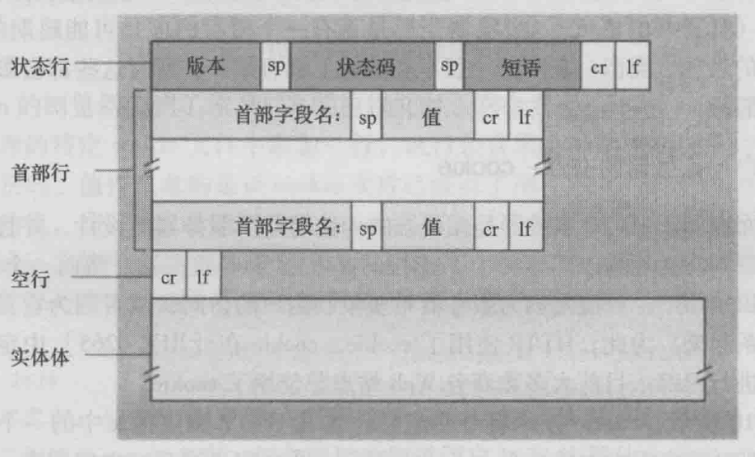
**d.在HTTP响应报文中的Date:首部指出了该响应中对象最后一次修改的时间。**

**e. HTTP响应报文决不会具有空的报文体。**

1. 错。见P67

总共发出了4个请求报文和四个响应报文。请求有：首先发出一个web页面请求，获取web页面后发现其中有三个图像链接，因此会再发送三个图像请求。

1. 对。见p69划线部分。
2. 错。见p67，p68划线部分。使用非持续连接的情况下一个TCP连接只能响应一个http服务请求。
3. 错。这个时间是服务器产生并发送该响应报文的日期和时间
4. 错。当服务器收到使用Head方法的请求时，将会使用Http报文进行相应，但是并不返回请求对象，这种方式主要是为了方便开发者进行调试。另外还有一些指示请求错误的报文，如404 not found报文，不需要在报文体里放入信息，因为没有找到请求的对象，报文头部即可指示错误类型。



**P5.下面文本中显示的是来自服务器的回答，以响应上述问题中HTTP GET报文。回答下列问题，指出你在下面报文中找到答案的地方。**

**HTTP/1.1 200 OK<cr><if>Date: Tue, 07 Mar 2008**

**12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Fedora)**

**<cr><lf>Last-Modified: Sat, 10 Dec2005 18:27:46**

**GMT<cr><lf>ETag: "526c3-f22-a88a4c80”<cr><Jf>-Accept-**

**Ranges: bytes<cr><2f>Content-Length: 3874<cr><if>**

**Keep-Alive: timeout=max=100<cr><if>Connection:**

**Keep-Alive<cr><lf>Content-Type: text/html; charset=**

**ISO-8859-l<cr><lf><cr><lf><idoctype html public "-**

**//w3c//dtd html 4 .0 transitional//en”><If><html><If>**

**<head><lf> <meta http\_equiv="Content-Type"**

**content=”text/html; charset=iso-8859-1”><lf> <meta**

**name=”GENERATOR” content="Mozilla/4.79 [ en] (Windows NT**

**5.0; U) Netscape]”><lf> <title>CMPSCI 453 / 591 /**

**NTU-ST550A Spring 2005 homepage</title><If></head><lf>**

**<much more document text following here (not shown)>**

**a.服务器能否成功地找到那个文杓？该文档提供回答是什么时间？**

**b.该文档最后修改是什么时间？**

**c.文档中被返回的字节有多少？**

**d.文档被返回的前5个字节是什么？该服务器同意一条持续连接吗?**

参考P71 HTTP响应报文格式或上题的图

1. 请求文档已成功找到。从状态码为200可以看出，表示请求成功。

回答时间是Tue,07 Mar 2008 12:39:45GMT

b．Last-Modified: Sat，10 Dec2005 18:27:46GMT

c. Content-Length: 3874

d. <!doc 因为这串字符前面都是首部行的内容（很多键值对），因此判断这里是返回对象的开始

同意持续连接，从Connection:Keep-Alive可以看出，连接还在保持

**P7.假定你在浏览器中点击一条超链接获得Web页面。相关联的URL的IP地址没有缓存在本地主机上, 因此必须使用DNS lookup以获得该IP地址。如果主机从DNS得到IP地址之前已经访问了n个DNS 服务器；相继产生的RTT依次为、…、。进一步假定与链路相关的Web页面只包含一个对象，即由少量的HTML文本组成。令表示本地主机和包含对象的服务器之间的RTT值。假定该对象传输时间为零，则从客户点击该超链接到它接收到该对象需要多长时间？**

主机获取页面IP地址所需时间=RTT1+RTT2+…+RTTn

主机获取页面时间=（建立连接，请求页面）+接受页面时间=2RTT0 （参考R6左图）

总时间 = 2RTT0 + RTT1 + RTT2+...+RTTn

**P8. 参照习题P7，假定在同一服务器上某HTML文件引用了 8个非常小的对象。忽略发送时间，在下列情况下需要多长时间：**

**a.没有并行TCP连接的非持续HTTP。**

**b.配置有5个并行连接的非持续HTTP。**

**c.持续HTTP**

a. 总时间 = 获取IP时间 + 获取页面时间 + 获取8个小对象时间

= 2RTT0 + RTT1 + RTT2 +...+ RTTn + 8\*2RTT0

1. 总时间 = 获取IP时间 + 获取页面时间 + 并行获取5个小对象时间 + 并行获取3个

小对象时间

= 2RTT0 + RTT1 + RTT2 +...+ RTTn + 2RTT0 + 2RTT0

1. 总时间 =获取IP时间 + 获取页面时间 + 直接获取8个小对象时间

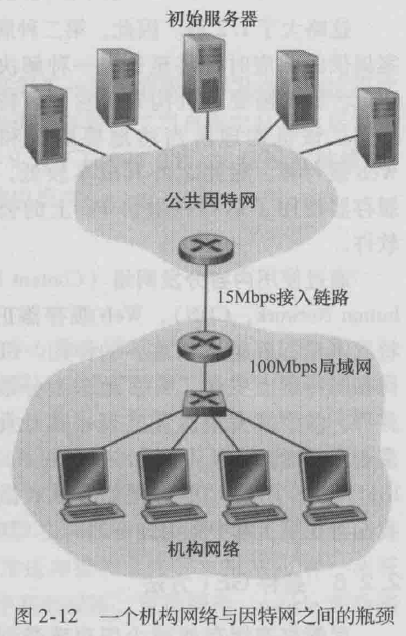
= 2RTT0 + RTT1 + RTT2 +...+ RTTn + 8RTT0

**P9.考虑图2-12,其中有一个机构的网络和因特网相连。假定对象的平均长度为850000比特，从这个机构网的浏览器到初始服务器的平均请求率是每秒16个请求。还假定从接入链路的因特网一侧的路由器转发一个HTTP请求开始，到接收到其响应的平均时间是3秒（参见2.2.5节）。将总的平均响应时间建模为平均接入时延（即从因特网路由器到机构路由器的时延）和平均因特网时延之和。对于平均接入时延，使用，式中是跨越接入链路发送一个对象的平均时间，是对象对该接入链路的平均到达率。**

**a.求出总的平均响应时间。**

**b.现在假定在这个机构LAN中安装了一个缓存器。假定命中率为0.4,求出总的响应时间。**

参考图2-12



a.

△=接入链路传输一个对象的平均时间

= 对象大小/传输速率 = 8.5\*105bit/(1.5\*107)bps = 0.0567s

平均接入时延 = △/（1-△B）=0.0567/(1-0.0567\*16)=0.61s

平均响应时间 = 平均接入时延+平均因特网时延

= 0.61 + 3 = 3.61s

b.

缓存命中为0.4即表示0.6的请求会继续通过接入链路进行传输

平均接入时延 = △/（1-△B）=0.0567/(1-0.0567\*16\*0.6)=0.124s

平均响应时间 =平均接入时延+平均因特网时延

=0.124+3 = 3.124s

**P10.考虑一条10米短链路，某发送方经过它能够以150bps速率双向传输。假定包含数据的分组是100000比特长，仅包含控制（如ACK或握手）的分组是200比特长。假定N个并行连接每个都获得1/N的链路带宽。现在考虑HTTP协议，并且假定每个下载对象是100Kb长，这些初始下载对象包含10个来自相同发送方的引用对象。在这种情况下，经非持续HTTP的并行实例的并行下载有意义吗？现在考虑持续HTTP。你期待这比非持续的情况有很大增益吗？评价并解释你的答案。**

非持续HTTP情况下的并行下载时间：

总时间 = 初始下载对象下载时间 + 10个引用对象并行下载时间

= （2\*200/150+2dprop + 200/150+dprop + 100000/150+dprop）

+(\*2+2dprop + +dprop + +dprop)

= 7377+8 dprop

\*（非持续HTTP情况下的非并行下载时间：

总时间 = （2\*200/150+2dprop + 200/150+dprop + 100000/150+dprop）

+(\*2+2dprop + +dprop + +dprop)\*10

= 7377+44 dprop）和上面基本一样

持续HTTP情况下的下载时间：

总时间=初始对象下载时间 + 依次下载10个引用对象的下载时间

= （2\*200/150+2dprop + 200/150+dprop + 100000/150+dprop）

+ （+dprop + +dprop）\*10

= 7351+24 dprop

因为链路较短，因此忽略传播时延，非持续HTTP并行下载时间为7377s,持续HTTP的下载时间为7351s

持续HTTP和非持续HTTP并行下载的时间相差不大，主要原因是持续HTTP所节省的连接建立的时间不大,持续HTTP只建立了一条连接，非持续HTTP建立了11条连接，持续HTTP节省的时间约为10条连接 **建立** 的时间，为10\*(200/150)\*2=26s, 与结果吻合。