

Homework-1

21307289 刘森元

Prob. 1

用户 A 与 B 的邮箱分别为 `a@XXX.com` 与 `b@XXX.com`，请从应用角度简述 A 发送邮件给 B 的过程以及在这个过程中涉及的应用层协议如何协作完成该任务。

Answer:

过程中涉及的应用层协议有 SMTP、IMAP/POP3，过程简述如下：

1. 用户 A 使用邮件客户端应用程序（如Outlook、Gmail等），或使用 HTTP 连接到网页编写邮件，并指定用户 B 的邮箱地址（`b@XXX.com`）作为收件人。
2. 用户 A 的邮件客户端应用程序将邮件内容和收件人信息发送给用户 A 的邮件服务器。
3. 用户 A 的邮件服务器使用发送邮件的应用层协议，如 SMTP，将邮件传输到用户 B 的邮件服务器。
4. 用户 B 的邮件服务器接收到来自用户 A 的邮件，并使用接收邮件的应用层协议，如 POP3/IMAP，将邮件存储在用户 B 的邮箱中。
5. 用户 B 的邮件客户端应用程序通过使用接收邮件的应用层协议，如 POP3/IMAP，从用户 B 的邮件服务器中检索新的邮件。
6. 用户 B 在其邮件客户端应用程序中查看和读取用户 A 发送的邮件。

Prob. 2

SMS、iMessage和WhatsApp都是智能手机即时通信系统。在因特网上进行一些研究后，为这些系统分别写一段他们所使用协议的文字。然后撰文解释他们的差异所在。

Answer:

- SMS（短信）是一种智能手机即时通信系统，它使用GSM（全球系统移动通信）协议。SMS使用移动网络运营商的信令通道来传输文本消息。它采用了一种简单的文本编码方式，允许用户发送和接收包含有限字符数的短信。SMS是一种基于电信运营商网络的通信方式，可以跨不同手机平台和操作系统进行交流。
- iMessage是苹果公司的智能手机即时通信系统，专为苹果设备开发。它使用了多种协议，包括XMPP（可扩展通信和表示协议）和APNS（苹果推送通知服务）。iMessage通过互联网连接进行通信，并允许苹果设备用户发送文本消息、照片、视频和其他多媒体内容。iMessage还支持端到端加密，确保消息在传输过程中的安全性。
- WhatsApp是一种跨平台智能手机即时通信系统，可以在不同操作系统上使用。它使用自有的协议，称为WhatsApp协议。WhatsApp通过互联网连接进行通信，允许用户发送文本消息、语音消息、图片、视频和其他多媒体内容。WhatsApp还支持端到端加密，保护用户的隐私和消息安全。

这些系统之间的主要差异在于协议和功能。

- SMS是一种基本的文本消息传输协议，适用于所有支持短信功能的手机。
- iMessage和WhatsApp则提供了更丰富的功能，包括多媒体内容的发送、端到端加密和互联网连接。
iMessage专为苹果设备开发，而WhatsApp是跨平台的通信系统，可在不同操作系统上使用。此外，iMessage和WhatsApp还提供了更多的交流方式和功能，如语音通话、视频通话和群组聊天等。

总的来说，这些智能手机即时通信系统在协议和功能上有所不同，适用于不同的用户需求和平台。用户可以根据自己的喜好和需求选择适合自己的通信系统。

Prob. 3

考虑当浏览器发送一个HTTP GET报文时，通过Wireshark俘获到下列ASCII字符串（即这是一个HTTP GET报文的实际内容）。字符

是回车和换行符（即下面文本中斜体字符串表示了单个回车符，该回车符包含在HTTP首部中的相应位置）。回答下列问题，指出你在下面HTTP GET报文中找到答案的地方。

```
GET /cs453/index.html HTTP/1.1<cr><lf>Host: gai
a.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0 (
Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gec
ko/20040804 Netscape/7.2 (ax) <cr><lf>Accept:ex
t/xml, application/xml, application/xhtml+xml, text
/html;q=0.9, text/plain;q=0.8,image/png,*/*;q=0.5
<cr><lf>Accept-Language: en-us,en;q=0.5<cr><lf>Accept-
Encoding: zip,deflate<cr><lf>Accept-Charset: ISO
-8859-1,utf-8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300<cr>
<lf>Connection:keep-alive<cr><lf><cr><lf>
```

Answer:

a. 由该浏览器请求的文档的URL是什么？

文档的URL为 `/cs453/index.html`

根据HTTP报文格式，第一行为请求行， `请求方法` `URL` `协议版本`，即图中 `GET /cs453/index.html HTTP/1.1`

b. 该浏览器运行的是HTTP的何种版本？

HTTP/1.1

由a题中可得

c. 该浏览器请求的是一条非持续连接还是一条持续连接？

持续连接

HTTP/1.1默认使用持续连接，这里报文显式通过请求确定了持续连接，其中 `Connection:keep-alive` 和 `Keep-alive: 300` 表示服务器在 300 秒内维持连接

d. 该浏览器所运行的主机的IP地址是什么？

未知

`Host` 请求头紧跟着的是目标服务器的地址，并不包括本机地址，`User-Agent` 也不包括本机地址

e. 发起该报文的浏览器的类型是什么？在一个HTTP请求报文中，为什么需要浏览器类型？

`User-Agent` 头指明了一下信息：

- 浏览器类型：Mozilla/5.0
- 操作系统：Windows; U; Windows NT 5.1; en-US
- 浏览器版本：rv: 1.7.1
- 浏览器引擎：Gecko/20040804
- 浏览器名称：Netscape/7.2 (ax)

Prob. 4

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><lf>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><lf>Accept-
Ranges: bytes<cr><lf>Content-Length: 3874<cr><lf>
Keep-Alive: timeout=max=100<cr><lf>Connection:
Keep-Alive<cr><lf>Content-Type: text/html; charset=
ISO-8859-1<cr><lf><cr><lf><!doctype html public "-
//w3c//dtd html 4.0transitional//en"><lf><html><lf>
<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-ST550ASpring 2005 homepage</title><lf></head><lf>
<much more document text following here (not shown)>
```

Answer:

a. 服务器能否成功地找到那个文档？该文档提供回答是什么时间？

服务器成功找到了文档，时间为 2008/03/07 12:39:45 GMT

根据报文格式，服务器返回状态 200 OK 即请求被成功接受，后续为时间

b. 该文档最后修改是什么时间？

文档最后修改时间为 2005/12/10 18:27:46

见 Last-Modified

c. 文档中被返回的字节有多少？

返回 3874 Bytes

见 Content-Length

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？

前 5 个字节为 <!doc ，服务器同意一条持续连接，最大时间为 100

将报文整理后有

```
HTTP/1.1 200 OK // HTTP协议版本和状态码
Date: Tue, 07 Mar 2008 12:39:45 GMT // 响应报文的生成日期和时间
Server: Apache/2.0.52 (Fedora) // 服务器软件信息
Last-Modified: Sat, 10 Dec 2005 18:27:46 GMT // 资源的最后修改日期和时间
ETag: "526c3-f22-a88a4c80" // 资源的实体标签
Accept-Ranges: bytes // 服务器支持的字节范围请求
Content-Length: 3874 // 响应报文正文的长度
Keep-Alive: timeout=max=100 // 保持连接的超时时间
Connection: Keep-Alive // 连接类型为保持连接
Content-Type: text/html; charset=ISO-8859-1 // 响应报文正文的媒体类型和字符编码
<!doctype html public "-//w3c//dtd html 4.0transitional//en"> // HTML文档类型声明
<html> // HTML文档的开始标签
<head> // HTML文档头部的开始标签
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"> // HTML文档
中的元数据，指定字符编码
<meta name="GENERATOR" content="Mozilla/4.79 [en] (windows NT5.0; u) Netscape]">
// HTML文档中的元数据，指定生成器
<title>CMPSCI 453 / 591 NTU-ST550A Spring 2005 homepage</title> // HTML文档的标题
</head> // HTML文档头部的结束标签
<much more document text following here (not shown)> // HTML文档的更多内容（未显示）
```

综上可知文档返回的前五个字符为 <!doc

Connection: KeepAlive 和 Keep-Alive: timeout=max=100 可知支持持续连接

Prob. 5

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 RTT_1 、 \dots 、 RTT_n 。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 RTT_0 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,则从该客户点击该超链接到它接收到该对象需要多长时间?

从本地主机到包含对象的服务器之间有两个往返的过程,即握手和请求连接的过程,所以有两个 RTT_0 ,因此总共需要的时间为:

$$2RTT_0 + \sum_{i=1}^n RTT_i$$

Prob. 6

假定你能够访问所在系的本地DNS服务器中的缓存。你能够提出一种方法来粗略地确定在你所在系的用户中最为流行的Web服务器(你所在系以外)吗?解释原因。

1. 分析DNS缓存记录:查看本地DNS服务器的缓存记录,特别是针对Web服务器的记录。这些记录通常包含了已解析的域名和对应的IP地址。通过分析这些记录,我们可以得出在所在系的用户中最为流行的Web服务器的IP地址。
2. 统计IP地址频率:根据DNS缓存记录中的IP地址,统计每个IP地址出现的频率。频率较高的IP地址表示被访问较多的Web服务器,因此可以认为该Web服务器在所在系的用户中最为流行。

Prob. 7

考虑使用一种客户-服务器体系结构向 N 个对等方分发一个 F 比特的文件。假定一种某服务器能够同时向多个对等方传输的流体模型,只要组合速率不超过 u_s ,则以不同的速率向每个对等方传输。

- a. 假定 $u_s/N \leq d_{\min}$ 。定义一个具有 NF/u_s 分发时间的分发方案。
- b. 假定 $u_s/N \geq d_{\min}$ 。定义一个具有 F/d_{\min} 分发时间的分发方案。
- c. 得出最小分发时间通常是由 $\max\{NF/u_s, F/d_{\min}\}$ 所决定的结论。

- a. 当 $u_s/N \leq d_{\min}$ 时,每个对等方的传输速率都小于等于 d_{\min} 。因此,我们可以同时向所有对等方传输文件,每个对等方的传输速率为 u_s/N 。在这种情况下,分发时间为 NF/u_s 。
- b. 当 $u_s/N \geq d_{\min}$ 时,每个对等方的传输速率都大于等于 d_{\min} 。由于服务器的传输速率限制为 u_s ,我们需要将文件分割成多个部分,以便在每个时间段内只向一个对等方传输。每个对等方的传输时间为 F/d_{\min} 。在这种情况下,分发时间为 F/d_{\min} 。
- c. 最小分发时间通常由 $\max\{NF/u_s, F/d_{\min}\}$ 决定。这是因为我们需要同时考虑两个因素:每个对等方的传输速率和服务器的传输速率。如果 NF/u_s 大于 F/d_{\min} ,则分发时间取决于 NF/u_s ;如果 F/d_{\min} 大于 NF/u_s ,则分发时间取决于 F/d_{\min} 。因此,最小分发时间为 $\max\{NF/u_s, F/d_{\min}\}$ 。

Prob. 8

假定Bob加入BitTorrent，但他不希望向任何其他对等方上传任何数据（因此称为搭便车）。

a. Bob声称他能够收到由该社区共享的某文件的完整副本。Bob所言是可能的吗？为什么？

b. Bob进一步声称他还能够更为有效的进行他的“搭便车”，方法是利用所在系的计算机实验室中的多台计算机（具有不同的IP地址）。他怎样才能做到这些呢？

a. Bob声称他能够收到由该社区共享的某文件的完整副本是不可能的。BitTorrent是一种点对点文件共享协议，其中每个对等方都负责上传和下载文件的不同部分。如果Bob不上传任何数据，其他对等方没有理由将完整的文件发送给他。根据BitTorrent的工作原理，对等方之间的上传和下载是相互依赖的，没有上传就没有下载。

b. Bob声称他可以更有效地进行“搭便车”，利用所在系的计算机实验室中的多台计算机。要实现这一点，Bob可以在所在系的计算机实验室中安装BitTorrent客户端，并使用不同的IP地址为每台计算机创建独立的BitTorrent身份。这样，他可以同时在多个计算机上连接到BitTorrent网络，并利用每个计算机的上传和下载带宽来更有效地进行“搭便车”。