R1. List five nonproprietary Internet applications and the application-layer

protocols that they use.

答：

电子邮件（Email） - 电子邮件使用几种协议，包括用于发送信息的SMTP（简单邮件传输协议），用于接收信息的POP3（邮局协议版本3）或IMAP（互联网信息访问协议），以及用于格式化信息的MIME（多用途互联网邮件扩展）。

万维网（WWW）--万维网使用超文本传输协议（HTTP）进行网络服务器和客户端之间的通信。HTTPS（HTTP安全）是HTTP的一个安全版本，使用加密来保护敏感数据。

文件传输协议（FTP） - FTP用于在网络上的计算机之间传输文件。它使用文件传输协议（FTP）应用层协议。

域名系统（DNS） - DNS用于将人类可读的域名（如www.example.com）翻译成计算机可以用来相互通信的IP地址。DNS使用域名系统（DNS）应用层协议。

简单网络管理协议（SNMP） - SNMP用于管理和监控网络设备，如路由器、交换机和服务器。它使用简单网络管理协议（SNMP）应用层协议。

R11. Why do HTTP, FTP, SMTP, and POP3 run on top of TCP rather than on UDP?

答：

HTTP、FTP、SMTP和POP3都是应用层协议，用于在网络上不同设备之间交换数据。这些协议旨在提供可靠的数据传输，这意味着数据必须在没有错误的情况下按正确的顺序传递。

因此，HTTP、FTP、SMTP和POP3运行在TCP而不是UDP之上，因为它们需要可靠的数据传输。如果它们运行在UDP之上，就不能保证数据的传输没有错误或顺序正确，这可能导致数据损坏、数据丢失或其他错误。

P1. True or false?

a. A user requests a Web page that consists of some text and three images.

For this page, the client will send one request message and receive four

response messages.

True.

当用户请求一个由一些文本和三个图像组成的网页时，客户端将向服务器发送一个HTTP请求消息，要求提供该网页。然后，服务器将以四个HTTP响应消息来回应，一个是HTML文件，另一个是三个图像中的每一个。

b. Two distinct Web pages (for example, www.mit.edu/research.html

and www.mit.edu/students.html) can be sent over the same persistent

connection.

True.

当客户端请求一个网页时，它与服务器建立了一个TCP连接。如果连接是持久的，客户端可以在同一个连接上为不同的网页发送多个请求，而服务器可以响应相应的网页。

c. With nonpersistent connections between browser and origin server, it is possible

for a single TCP segment to carry two distinct HTTP request messages.

False.

在浏览器和源服务器之间的非持久性连接中，一个TCP段不可能承载两个不同的HTTP请求信息。由于每个HTTP请求信息都在自己的TCP段中发送，所以一个TCP段不可能承载两个不同的HTTP请求信息。

d. The Date: header in the HTTP response message indicates when the

object in the response was last modified.

False.

HTTP响应消息中的Date: header表示服务器生成响应的日期和时间。它并不表示响应中的对象最后被修改的时间。

e. HTTP response messages never have an empty message body.

False.

HTTP响应消息可以有一个空的消息体。许多HTTP响应消息根本就没有消息体。

P4. Consider the following string of ASCII characters that were captured by

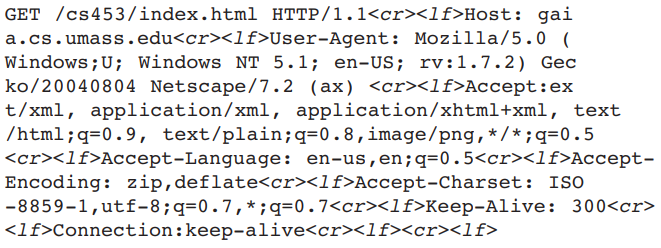
Wireshark when the browser sent an HTTP GET message (i.e., this is the actual

content of an HTTP GET message). The characters <cr><lf> are carriage

return and line-feed characters (that is, the italized character string <cr> in

the text below represents the single carriage-return character that was con-tained at that point in the HTTP header). Answer the following questions,

indicating where in the HTTP GET message below you find the answer.



1. What is the URL of the document requested by the browser?

/cs453/index.html

1. What version of HTTP is the browser running?

HTTP/1.1

1. Does the browser request a non-persistent or a persistent connection?

The browser request a persistent connection.

From:’Connection:Keep-alive’

1. What is the IP address of the host on which the browser is running?

The IP address of the host is: gaia.cs.umass.edu

From: ‘Host: gaia.cs.umass.edu’

e. What type of browser initiates this message? Why is the browser type

needed in an HTTP request message?

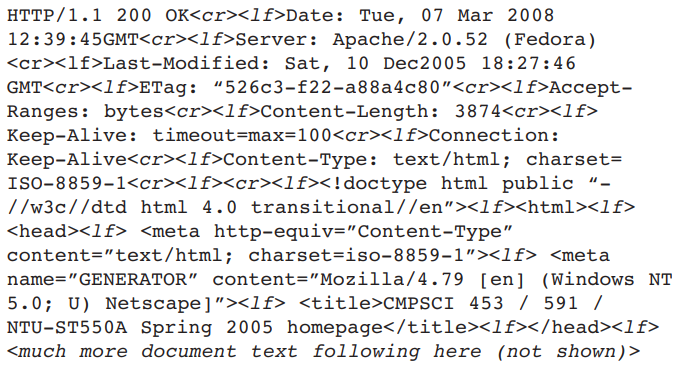
The type of browser initiates this message is: Mozilla/5.0

From: ‘User-Agent: Mozilla/5.0’

HTTP请求信息中需要浏览器类型，因为网络服务器可以利用这一信息根据客户的能力来定制响应。

P5. The text below shows the reply sent from the server in response to the HTTP

GET message in the question above. Answer the following questions, indicating where in the message below you find the answer.



a. Was the server able to successfully find the document or not? What time

was the document reply provided?

Yes. The server able to successfully find the document, and the time of the document reply provided is: Tue, 07 Mar 2008 12:39:45 GMT

We can find it all in the first line of the response to the HTTP: (200 OK) and (Date: Tue, 07 Mar 2008 12:39:45 GMT)

b. When was the document last modified?

Sat, 10 Dec2005 18:27:46 GMT

From: ‘Last-Modified: Sat, 10 Dec2005 18:27:46 GMT’

c. How many bytes are there in the document being returned?

There are 3874 Bytes in the document being returned

From: ‘Content-Length: 3874’xxx

d. What are the first 5 bytes of the document being returned? Did the server

agree to a persistent connection

The first 5 bytes of the document being returned is ‘<!doc’, and the server agree to a persistent connection.

From: ‘Connection: Keep-Alive’

P7. Suppose within your Web browser you click on a link to obtain a Web page.

The IP address for the associated URL is not cached in your local host, so a

DNS lookup is necessary to obtain the IP address. Suppose that n DNS

servers are visited before your host receives the IP address from DNS; the

successive visits incur an RTT of RTT1, . . ., RTTn. Further suppose that the

Web page associated with the link contains exactly one object, consisting of a

small amount of HTML text. Let RTT0 denote the RTT between the local host

and the server containing the object. **Assuming zero transmission time of the**

**object, how much time elapses from when the client clicks on the link until**

**the client receives the object?**

The total time that elapses from when the client clicks on the link until the client receives the object is:

Total time = RTT1 + RTT2 + RTT3 + RTT4 + RTT0 + 2 \* RTT0

= RTT1 + RTT2 + RTT3 + RTT4 + 3 \* RTT0

P8. Referring to Problem P7, suppose the HTML file references eight very small

objects on the same server. Neglecting transmission times, how much time

elapses with

1. Non-persistent HTTP with no parallel TCP connections?
2. Non-persistent HTTP with the browser configured for 5 parallel connections?

c. Persistent HTTP?

a. Non-persistent HTTP with no parallel TCP connections:

total\_time = RTT0 + 8 \* (RTT1 + RTT2 + ... + RTTn + 2 \* RTT\_TCP)

b. Non-persistent HTTP with the browser configured for 5 parallel connections:

total\_time = RTT0 + (8/5) \* (RTT1 + RTT2 + ... + RTTn + 2 \* RTT\_TCP)

1. Persistent HTTP:

total\_time = RTT0 + (RTT1 + RTT2 + ... + RTTn + 2 \* RTT\_TCP)