

2.答：包含有浏览器信息，在文件中的值为：

```
GET / HTTP/1.1
Host: cs.nju.edu.cn
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Date: Tue, 07 Mar 2017 11:29:44 GMT
Server: Apache/2.2.22 (Unix) mod_jk/1.2.23
Accept-Ranges: bytes
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8
```

可以看出我用的是 linux 系统下的 Firefox 浏览器

还包含有主机网址，接受文件类型，语言，编码方式，连接状态，请求序列号等信息；

3.传输层协议是 TCP 协议，报文在实验报告的目录下名为 TCP 的文件中；他是 HTTP 的上层协议，由于 OSI 协议架构，通过不同层的协议完成不同的任务来是的网络功能完善；所以浏览器要同时接受这两个协议；

4.

27	2.508489608	192.168.1.104	117.18.237.29	OCSP	505 Request
31	2.639525839	117.18.237.29	192.168.1.104	OCSP	862 Response
33	2.644397756	192.168.1.104	117.18.237.29	OCSP	505 Request
48	3.249194056	192.168.1.104	202.119.32.7	HTTP	387 GET / HTTP/1.1
106	3.269322187	202.119.32.7	192.168.1.104	HTTP	1123 HTTP/1.1 200 OK (text/html)
110	3.434818739	192.168.1.104	202.119.32.7	HTTP	371 GET /_visitcount?siteId=66&type=1&columnId=1564 HTTP/1.1

上述图片中的 http 报文中就包含了 IP 地址，其中 192.168.1.104 为主机 PC 的 IP 地址，117.18.237.29 为网页服务器端的 IP 地址；

```
Hypertext Transfer Protocol
> POST / HTTP/1.1\r\n
Host: ocsp.digicert.com\r\n
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:50.0) Gecko/20100101 Firefox/50.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
> Content-Length: 83\r\n
Content-Type: application/ocsp-request\r\n
Connection: keep-alive\r\n
\r\n
[Full request URI: http://ocsp.digicert.com/]
[HTTP request 1/2]
[Response in frame: 31]
[Next request in frame: 33]
File Data: 83 bytes
```

HTTP 报文内容如下：

报文类型为 HTTP Request 报文

主机地址：ocsp.digicert.com\r\n

主机浏览器：Firefox

接受端语言: en-US  
接受端编码: gzip  
内容长度: 83bytes  
类型: request  
连接状态: keep-alive (保持连接)

5.与 IP 地址相关联的协议是网络层,因为每个 IP 地址对应于一个局域网,而在网络层是将不同的 IP 包传到不同的局域网当中,所以 IP 地址与网络层相关联(应用层用端口来区分,链路层以 mac 地址区分,传输层以套接字来区分)

6.

```
▼ Ethernet II, Src: Azurewav_9e:ed:78 (40:e2:30:9e:ed:78), Dst: Tp-LinkT_ea:af:40 (fc:d7:33:ea:af:40)
  ▼ Destination: Tp-LinkT_ea:af:40 (fc:d7:33:ea:af:40)
    Address: Tp-LinkT_ea:af:40 (fc:d7:33:ea:af:40)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
  ▼ Source: Azurewav_9e:ed:78 (40:e2:30:9e:ed:78)
    Address: Azurewav_9e:ed:78 (40:e2:30:9e:ed:78)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
  Type: IPv4 (0x0800)
▼ Internet Protocol Version 4, Src: 192.168.1.104, Dst: 202.119.32.7
  0100 .... = Version: 4
  ....0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 52
  Identification: 0x5ddc (24028)
  > Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 64
  Protocol: TCP (6)
  Header checksum: 0x3059 [validation disabled]
  [Header checksum status: Unverified]
  Source: 192.168.1.104
  Destination: 202.119.32.7
```

可以看出,在以太网帧中:source 为本机 PC (mac 地址为: 40:e2:30:9e:ed:78)

由于 mac 地址与主机一一对应,而 IP 则表示一整片局域网的地址,所以主机 PC 想接受到数据必须通过 mac 地址加以判断;

8.在计算机网络中,把用于规定数据的交换规则、报文格式,以及如何发送和接受数据的一套规则成为网络协议或通信协议;

这些协议用来规定数据如何通过不同网络和介质进行传播的种种细节,使得一方送给另一方的数据包对方可以理解,使得双方能协调地,高效地收发数据包。

协议需要标准化来使得双方能够互相理解,从而使得数据传输变得可靠和有效;

协议的实现不需要标准化,不同条件可以通过不同的方法实现相同功能的协议,只要保证协议对上层提供的服务与原有的协议一致即可;

为了降低网络设计的复杂性,绝大多数网络都是按分层或分级的方式设计的。分层可以将庞大而复杂的问题,转化为若干较小的局部问题来解决。每一层的目的是向上一层提供特定的服务,而把实现的细节加以屏蔽。

9.TCP 协议设计的理念都是首次提出的，并且定义了当前 TCP 协议的绝大部分特性，比如数据报，分层等；

实现 TCP 协议经历了许多，最初他们想实现一些基础的传输功能，并且取得了成功，后来想要进一步完善：

1. 网络连接必须要持续即使是网络或网关丢失；
2. 网络必须支持不同种类的社交服务；
3. 网络架构必须适用于不同的网络环境；
4. 网络架构必须允许资源的分发管理；
5. 网络架构必须高效；
6. 网络架构必须简单；
7. 网络上的资源数据必须可靠（有序，完整）

然而实现这些目标（在同一层网络架构上）是很难的，他们最终必须在分层的方式上实现不同的网络功能，并且能够完成不同目标；

所以我们在设计某些负责功能的架构时，不妨也可以试一试分治的原则，从不同的小问题出发，逐一解决，往往比直接解决一个大的问题要方便的多；