

华东师范大学计算机科学技术系上机实践报告

课程名称：计算机网络	年级：2022	上机实践成绩：
指导教师：洪道诚	姓名：朱宇笑	创新实践成绩：
实验名称：超文本传输协议(HTTP)	学号：10225001410	上机实践日期：2023.11.10
座位编号：F	组号：6	上机实践时间：2 学时

一、 实验目的

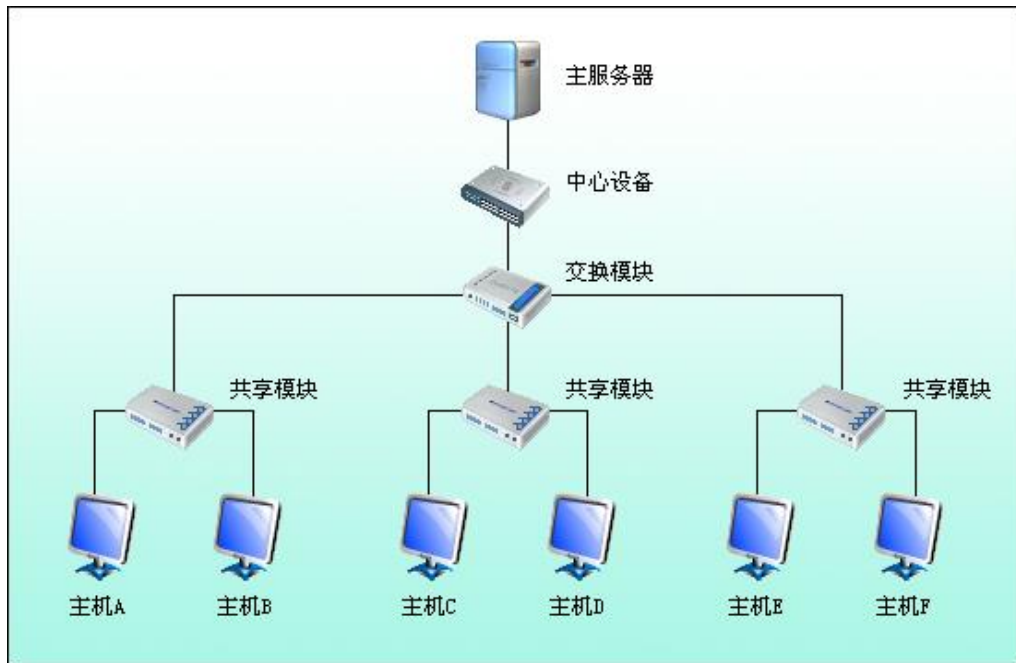
1. 掌握HTTP的报文格式
2. 掌握HTTP的工作原理
3. 掌握HTTP常用方法

二、 实验设备

1. PC机
2. 仿真编辑器和协议分析器

三、 实验原理

实验采用网络结构一



(一) . HTTP 体系结构

万维网（WWW）服务是分布式的客户/服务器服务，客户使用浏览器能够得到服务器提供的服务，这种服务的提供是分布在许多网站中的，如下图所示：

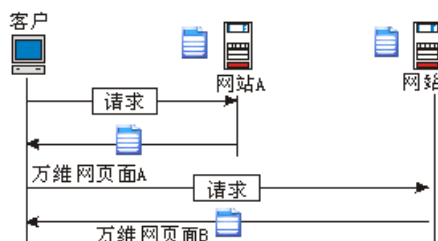


图 2-1 分布式的服务

每一个网站保存有一个或多个文档，叫做万维网页面。在图 13-1 所示的例子中，客户需要查看网站 A 的某些信息。浏览器用来读取万维网上的文档。它向网站 A 发送一个请求，这个请求包含了网站 A 和网站中万维网页面的地址，这个地址叫做统一资源定位符 (URL)。网站 A 收到请求后，将指定的文档发送给这个客户。它用同样的方法与网站 B 通信。

(1) . 客户 (浏览器)

许多厂商提供商用浏览器，可以解释和显示万维网页面。每一个浏览器通常由 3 个部分组成：控制程序、客户程序及解释程序。控制程序从键盘或鼠标接收输入，使用客户程序访问要浏览的文档。在文档找到后，控制程序就使用解释程序 (可以是 HTML、Java 或 JavaScript 等) 中的一个，把文档显示在屏幕上。

(2) . 服务器

万维网页面存储在服务器上。每当有客户请求到达时，对应的文档就发送给客户。为了提高效率，服务器通常在其高速缓存中存储被请求过的文档；对高速缓存进行访问要比磁盘快得多。通过多线程或多进程可使服务器的效率更加提高，服务器在同一时间可回答多个请求。

(3) . 统一资源定位符 (URL)

客户要访问万维网页面就需要这个页面地址。为了方便地访问文档，HTTP 协议使用统一资源定位符 (URL)。URL 是 Internet 上指定信息的标准。URL 由 4 部分组成：协议、主机、端口和路径，如下图所示：



图 2-2URL

●协议：协议指定了用这个 URL 的客户/服务器程序。例如，HTTP 协议、FTP 协议和 TELNET 协议等。

●主机：主机指明了信息所存放的地址，可以是逻辑地址也可以是相应的域名。存放万维网页面的计算机通常使用以字符“WWW”开始的域名，但这不是强制性的。

●端口：端口指定了使用主机的某个端口，端口是可选的。如果包含了端口，那么端口就插入在主机和路径之间，和主机用冒号分隔开。

●路径：路径指定了文件存放的位置。路径本身可以包含斜线，用于将目录与子目录和文件分隔开。

(二) . 万维网文档

万维网文档可分为 3 类：静态文档、动态文档和活动文档。这种分类是基于文档内容被确定的时间。

(1) . 静态文档

静态文档是固定内容的文档，它由服务器创建，并存储在服务器中。客户只能得到文档的一个副本。换言之，文件的内容是在文件被创建时就确定的，而不是在它被使用时。当然，在服务器中的内容是可以改变的，但用户不能改变它。当客户访问文档时，文档的一个副本就被发送出去。用户可以使用浏览程序显示这个文档，如下图所示：

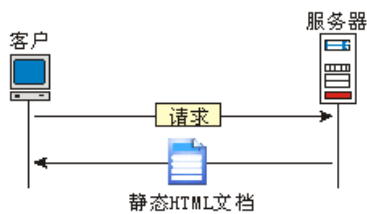


图 2-3 静态文档

(2) . 动态文档

动态文档是在浏览器请求该文档时才由万维网服务器创建出来。当请求到达时，万维网服务器就运行创建动态文档的应用程序。服务器返回这个程序或脚本的输出，把它作为对请求该文档的浏览器的响应。因为对每一个请求都创建出新的文档，因此每一个请求得到的动态文档的内容就会不同。

(3) . 活动文档

对于许多应用，需要程序能够在客户端运行，这就叫做活动文档。当浏览器请求活动文档时，服务器就发送这个文档的一个副本或脚本。然后这个文档就在客户（浏览器）端运行。

(三) . HTTP 协议简介

HTTP（超文本传输协议）主要用于访问万维网上的数据。协议以普通文本、超文本、音频、视频等格式传输数据。之所以称为超文本协议，原因是在应用环境中，它可以快速的在文档之间跳转。HTTP 在熟知端口 80 上使用 TCP 服务。

(四) . HTTP 报文格式

HTTP 报文有两种一般的类型：请求和响应。这两种类型的报文格式几乎是相同的。报文格式如下图所示：

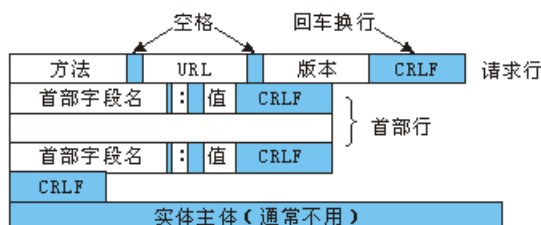


图 2-4HTTP 报文格式

(五) . HTTP 方法

HTTP 报文中的方法是客户端向服务器端发出的实际命令和请求。常用 HTTP 方法如下表所示：

表 2-1HTTP 方法

方法	动作
GET	从服务器请求一个文档
HEAD	请求关于一个文档的信息，但不是这个文档本身
POST	从客户向服务器发送一些信息
PUT	从服务器向客户发送一些信息
TRACE	把到达的请求回送
CONNECT	用于使用代理连接
OPTION	询问关于可用的选项

(六). HTTP 状态码

在响应报文中，请求行被替换为状态行，由 3 位数字组成，表示请求是否被理解或被满足。HTTP 协议的状态码如下表所示：

表 2-2HTTP 状态码

代码	短语	说明
提供信息		
100	Continue	请求的开始部分已经收到，客户可以继续它的请求
101	Switching	服务器同意客户请求，切换到在更新首部中定义的协议
成功		
200	OK	请求成功
201	Created	新的 URL 被创建
202	Accepted	请求被接受，但还没有马上起作用
204	Nocontent	主体中没有内容
重新定向		
301	Movedpermanently	服务器已经不再使用所请求的 URL
302	Movedtemporarily	所请求的 URL 已暂时地删除了
304	Notmodified	文档还没有被修改
客户差错		
400	Badrequest	在请求中有语法差错
401	Unauthorized	请求缺少适当地授权
403	Forbidden	服务被拒绝
404	Notfound	文档未找到
405	Methodnotallowed	这个方法不被这个 URL 所支持
406	Notacceptable	所请求的格式不可接受
服务器差错		
500	Internalservererror	在服务器端有差错，如崩溃
501	Notimplemented	所请求的动作不能完成
503	Serviceunavailable	服务暂时不可用，但可能在以后能够被请求

(七). 持久与非持久连接

在版本 1.1 以前的 HTTP 协议默认是非持久连接的，而在版本 1.1 中，持久连接是默认的连接。

(1). 非持久连接

在非持久连接中，对每一个请求/响应都要建立一次 TCP 连接。非持久连接的步骤如下：

- 客户打开 TCP 连接，并发送请求。
- 服务器发送响应，并关闭连接。
- 客户读取数据，直到它遇到文件结束标记；然后它关闭连接。

使用非持久连接时，对于在不同文件中的 N 个不同图片的请求，连接必须打开和关闭 N 次。非持久连接策略给服务器造成了很大的开销，因为服务器需要 N 个不同的缓存，而每次打开连接时都要使用慢开始过程。

(2). 持久连接

HTTP 版本 1.1 指明持久连接是默认的策略。在使用持久连接时，服务器在发送响应后，让连接继续为一些请求打开。服务器可以在客户发送关闭请求时等待或关闭这个连接。发送端通常在每一个响应中都发送数据长度。但是，有时发送端并不知道数据的长度（例如动态文档或活动文档），这时服务器就把长度不知道这件事通知客户，并在发送数据后关闭这个连接，这样客户就知道数据结束的地方已经到了。

(八). HTTP 代理服务器

HTTP 支持代理服务器（proxyserver）。代理服务器保留对最近请求的响应的副本。在有代理服务器的情况下，HTTP 客户把请求发送给代理服务器。代理服务器检查它的高速缓存。如果在高速缓存中有这个响应，代理服务器就直接应答客户的请求；反之，如果在高速缓存中没有这个响应，代理服务器就把请求发送给相应的 HTTP 服务器，当 HTTP 服务器的响应到达代理服务器时，代理服务器将该响应转发给客户，同时将该响应存储到高速缓存中，以便以后为其它客户的请求使用。

代理服务器减少了原始服务器的负荷，减小了通信量，也减小了延时。但是，由于使用了代理服务器，客户必须配置成接入到代理服务器而不是那个目标服务器。

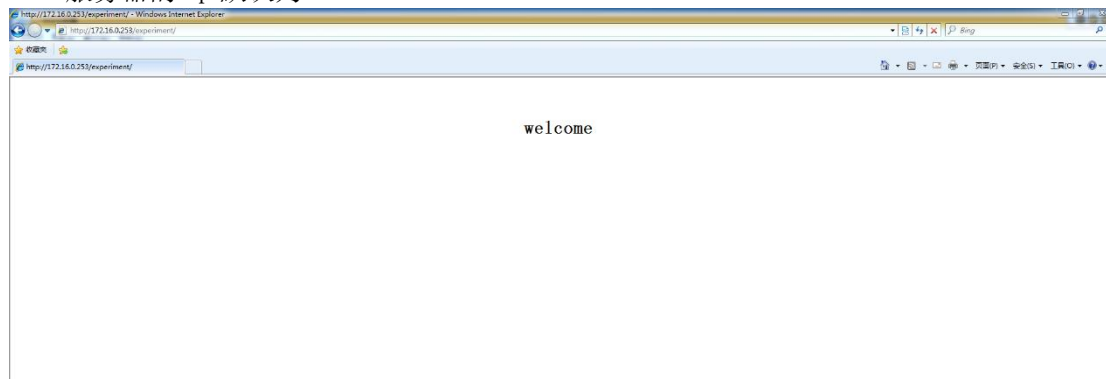
四、 实验步骤

练习一 页面访问

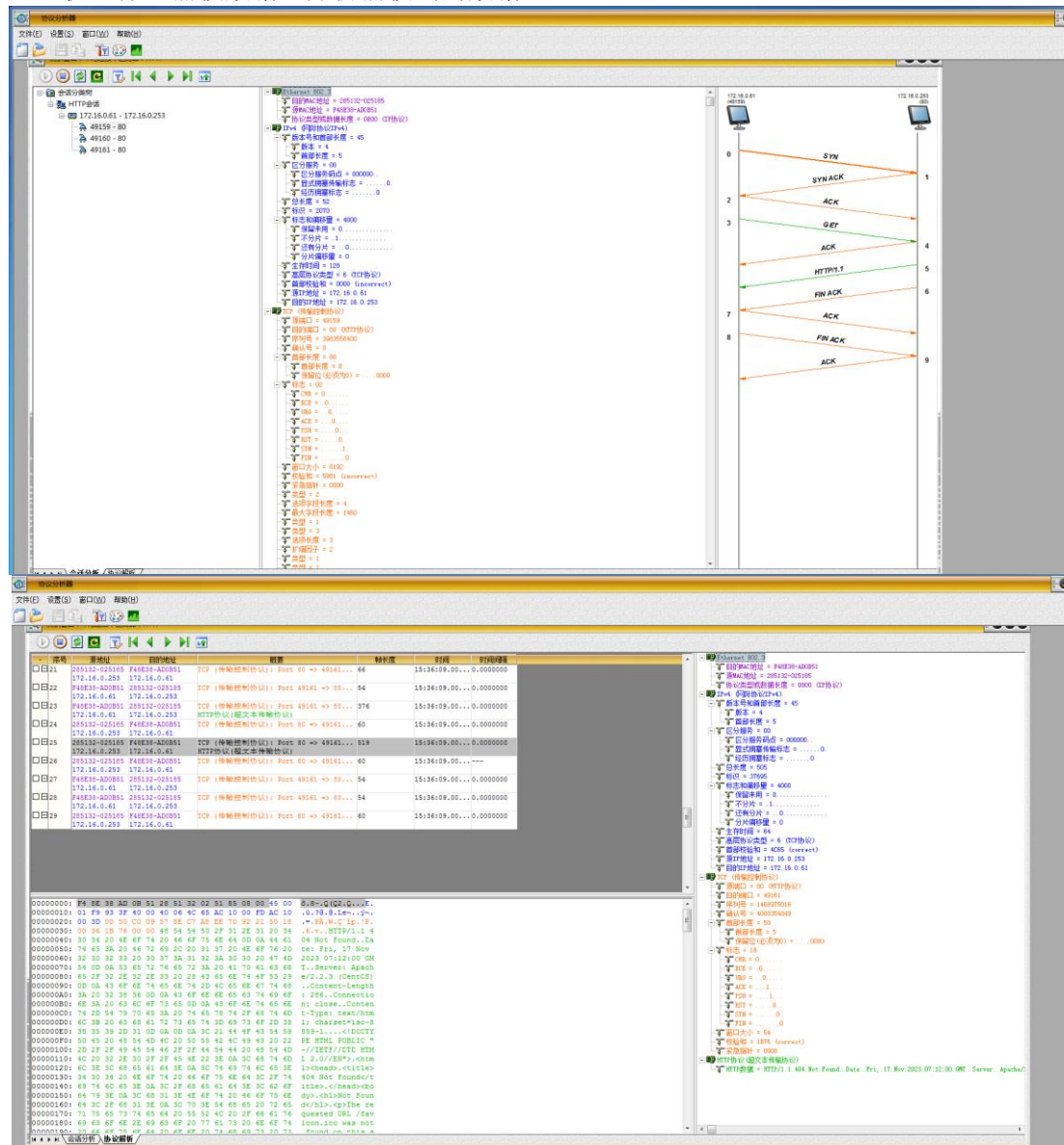
各主机打开工具区的“拓扑验证工具”，选择相应的网络结构，配置网卡后，进行拓扑验证，如果通过拓扑验证，关闭工具继续进行实验，如果没有通过，请检查网络连接。

本练习一人一组，现仅以主机 A 为例，其它主机参考主机 A 的操作。

1. 主机 A 清空 IE 缓存。
2. 主机 A 启动协议分析器开始捕获数据，并设置过滤条件（提取 HTTP 协议）。
3. 主机 A 启动 IE 浏览器，在“地址”框中输入 `http://服务器的 ip/experiment`，并连接，服务器的 ip 默认为 172.16.0.253。



4. 主机 A 停止捕获数据，分析捕获到的数据。



回答以下问题：

- 本练习使用 HTTP 协议的哪种方法？简述这种方法的作用。

回答：Get 方法。作用是从指定的资源请求数据。

- 根据本练习的报文内容，填写下表。

表 2-3 实验结果

主机名	JServer.NetLab
URL	http:// JServer.NetLab / experiment
服务器类型	Apache/2.2.3
传输文本类型	Text/html
访问时间	64

- 参考“会话分析”视图显示结果，绘制此次访问过程的报文交互图（包括 TCP 协议）。

回答：见上方截图。

- 简述 TCP 协议和 HTTP 协议之间的关系。

回答：HTTP 协议是建立在 TCP 连接基础上的应用层协议，它使用 TCP 协议提供的可靠传输来确保数据的正确性和完整性。

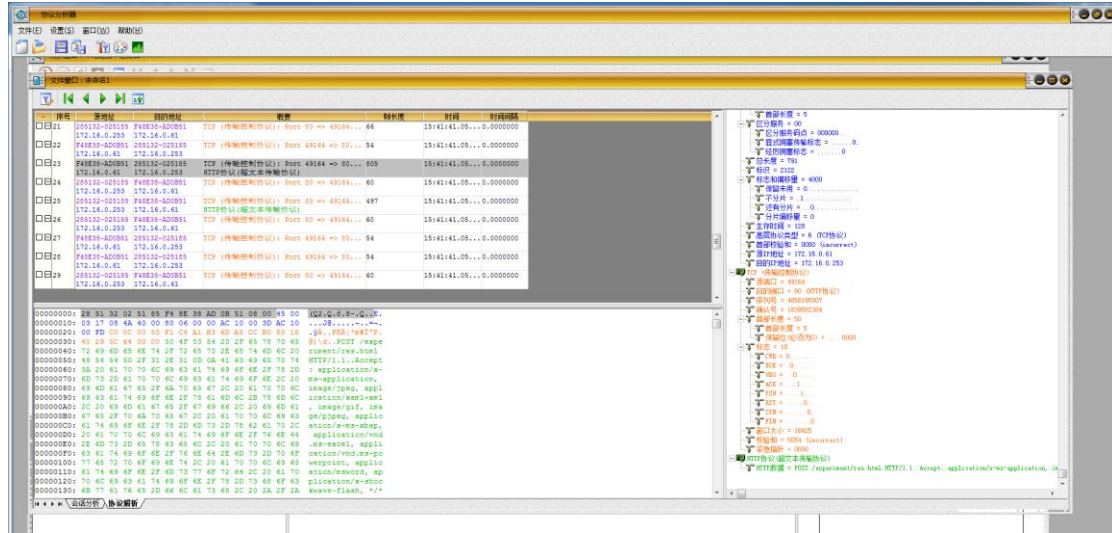
练习二 页面提交

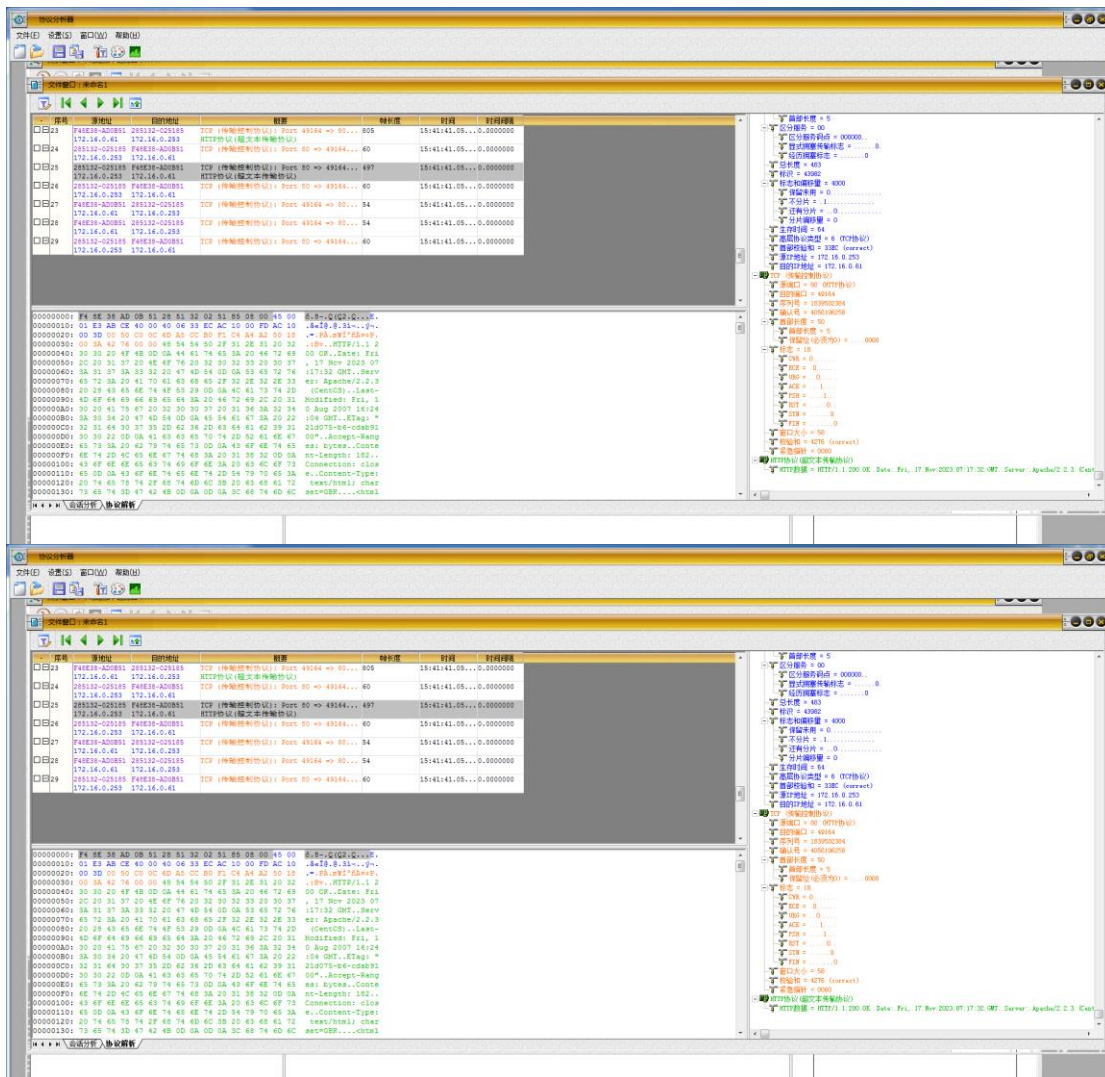
本练习一人一组，现仅以主机 A 为例，其它主机参考主机 A 的操作。

1. 主机 B 启动协议分析器开始捕获数据，并设置过滤条件（提取 HTTP 协议）。
2. 主机 A 启动 IE 浏览器，在“地址”框中输入“http://服务器的 ip/experiment/post.html”，并连接，服务器的 ip 默认为 172.16.0.253。在返回页面中，填写“用户名”和“密码”，点击[确定]按钮。



3. 主机 A 停止捕获数据，分析捕获到的数据。





回答以下问题：

- 本练习的提交过程使用 HTTP 协议的哪种方法？简述这种方法的作用。
 回答：Post 方法。作用是向目的服务器发送数据，并请求服务器接收并处理该数据。
- 此次通信分几个阶段？每个阶段完成什么工作？
 回答：2 个阶段。第一阶段访问页面，第二阶段提交页面。
- 参考“会话分析”视图显示结果，绘制此次提交过程的报文交互图(包括 TCP 协议)。

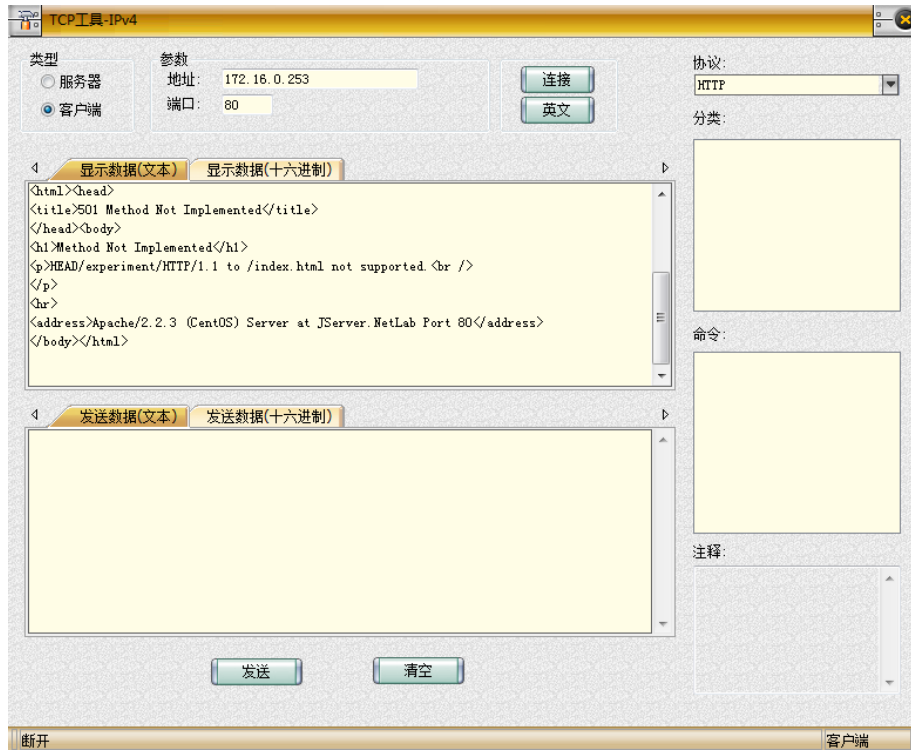
练习三 获取页面信息

本练习一人一组，现仅以主机 A 为例，其它主机参考主机 A 的操作。

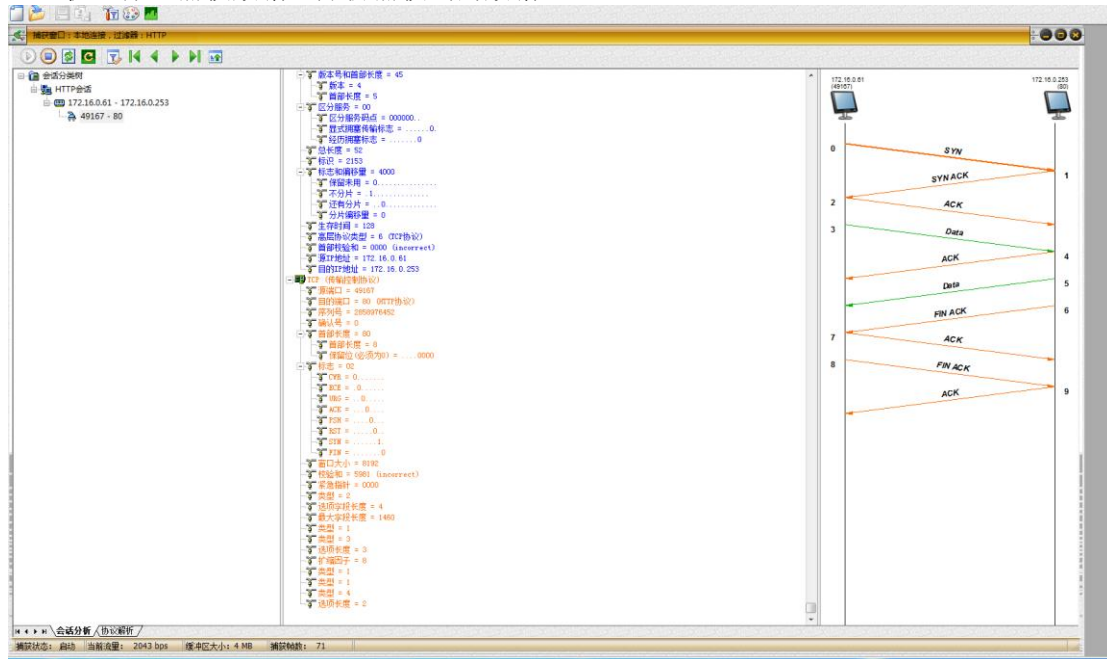
1. 主机 A 启动实验平台工具栏中的“TCP 工具”。
2. 主机 A 启动协议分析器开始捕获数据，并设置过滤条件（提取 HTTP 协议）。
3. 主机 A 在“TCP 工具”上，选中“客户端”单选框，设置“IP 地址”为服务器 IP（默认为 172.16.0.253）；设置“端口”为 80；单击[连接]按钮来和服务器建立连接。
4. 主机 A 在“TCP 工具”上，设置“发送数据（文本）”为以下内容：
 HEAD/experiment/HTTP/1.1<CRLF>
 Host:172.16.0.253<CRLF>
 <CRLF>
 点击[发送]按钮。（注：<CRLF>是回车换行）

点击[断开]按钮，断开 TCP 连接（由于不同 http 版本所遵循的规范不同，有些 HTTP 服务器不需要断开操作）。

5. 主机 A 在“TCP 工具”上的“显示数据（文本）”中察看服务器返回信息。



6. 主机 A 停止捕获数据，分析捕获到的数据。



第 12 次实验 超文本传输协议(HTTP)

The image displays two screenshots of the Wireshark network traffic analysis tool. The top screenshot shows a list of captured packets, with packet 4 selected. The packet list table is as follows:

序号	源地址	目的地址	概要	帧长度	时间	时间间隔
0	F48E38-AD0B51	285132-025185	TCP (传输控制协议): Port 49167 -> 80...	66	15:48:33.07...	
1	172.16.0.61	172.16.0.253	TCP (传输控制协议): Port 80 -> 49167...	66	15:48:33.07...	0.0000000
2	F48E38-AD0B51	285132-025185	TCP (传输控制协议): Port 49167 -> 80...	54	15:48:33.07...	0.0000000
3	F48E38-AD0B51	285132-025185	TCP (传输控制协议): Port 80 -> 49167...	60	15:48:35.07...	2.0000000
4	172.16.0.253	172.16.0.61	HTTP协议 (超文本传输协议)	369	15:48:35.07...	0.0000000
5	285132-025185	F48E38-AD0B51	TCP (传输控制协议): Port 80 -> 49167...	60	15:48:35.07...	0.0000000
6	172.16.0.61	172.16.0.253	TCP (传输控制协议): Port 80 -> 49167...	60	15:48:35.07...	0.0000000
7	F48E38-AD0B51	285132-025185	TCP (传输控制协议): Port 49167 -> 80...	54	15:48:35.07...	0.0000000
8	F48E38-AD0B51	285132-025185	TCP (传输控制协议): Port 49167 -> 80...	54	15:48:35.07...	0.0000000
9	172.16.0.61	172.16.0.253	TCP (传输控制协议): Port 80 -> 49167...	60	15:48:35.07...	0.0000000

The packet details pane for packet 4 shows the following structure:

- 应用层: 应用层数据 = HEAD/experiment/HTTP/1.1...Host:172.16.0.253
- 传输层: 传输层数据 = 标志 = 18, 窗口大小 = 256, 校验和 = 59AD (incorrect), 紧急指针 = 0000
- 网络层: IP 协议 (传输控制协议), 源地址 = 172.16.0.61, 目的地址 = 172.16.0.253
- 链路层: Ethernet II (以太网 II), 源 MAC 地址 = F48E38-AD0B51, 目的 MAC 地址 = 285132-025185

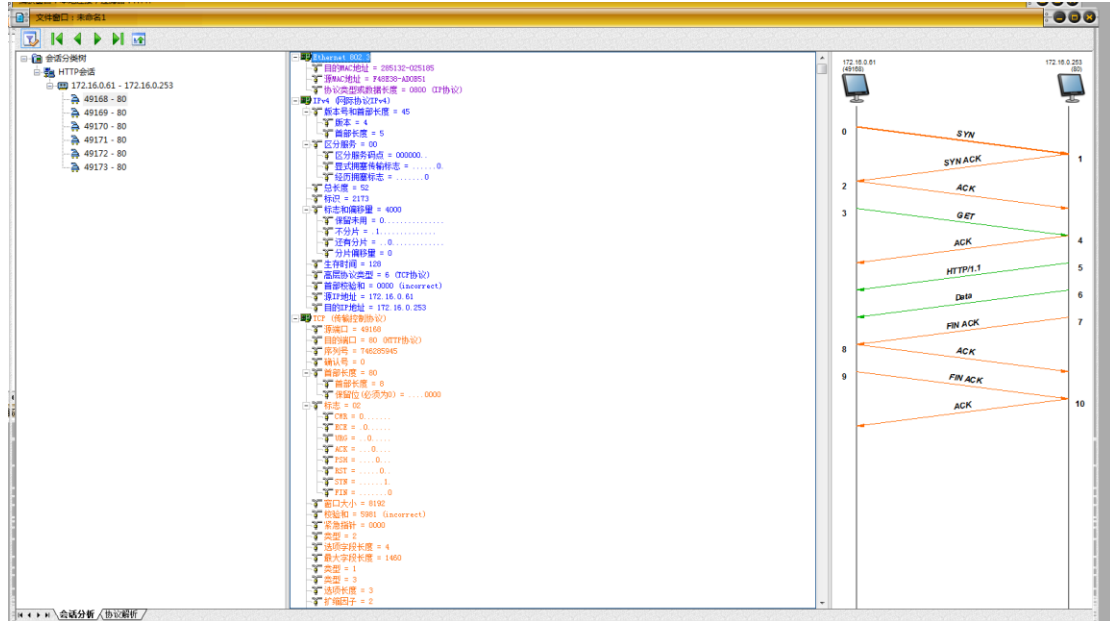
The bottom screenshot shows the same traffic analysis, but with packet 4 selected. The packet details pane shows the following structure:

- 应用层: 应用层数据 = HEAD/experiment/HTTP/1.1...Host:172.16.0.253
- 传输层: 传输层数据 = 标志 = 18, 窗口大小 = 256, 校验和 = 59AD (incorrect), 紧急指针 = 0000
- 网络层: IP 协议 (传输控制协议), 源地址 = 172.16.0.61, 目的地址 = 172.16.0.253
- 链路层: Ethernet II (以太网 II), 源 MAC 地址 = F48E38-AD0B51, 目的 MAC 地址 = 285132-025185

练习四 较复杂的页面访问

本练习一人一组，现仅以主机 A 为例，其它主机参考主机 A 的操作。

1. 本练习中要求各主机设置 DNS 服务器地址，（DNS 服务器的 IP 地址即主控中心平台的 IP 地址）其 IP 地址以 172.16.0.253 为例。
2. 主机 A 使用 “ipconfig/flushdns” 命令清空 DNS 高速缓存。
3. 主机 A 启动协议分析器开始捕获数据并设置过滤条件（提取 DNS、HTTP 协议）。
4. 主机 A 启动 IE 浏览器，在地址框中输入 http://JServer.NetLab/complexpage.htm。
5. 主机 A 停止捕获数据，察看相关会话，分析捕获到的数据。



回答以下问题：

- 结合本次实验结果，简述浏览器是如何处理一个访问请求的。

回答：浏览器处理一个访问请求的流程如下：

- 1.用户在浏览器中输入 URL，直接输入 IP 或者输入域名。
 - 2.如果输入的是域名，浏览器需要使用 DNS 解析将域名解析成 IP 地址，然后通过 IP 地址确认要访问的服务器。
 - 3.浏览器构建请求行信息，包括请求方法、请求 URI 和 HTTP 协议版本。常见的请求方法有 GET 和 POST。
 - 4.浏览器会查找缓存，包括 DNS 缓存和浏览器缓存，以确定是否已经请求过该资源。如果缓存中有匹配的资源，则直接返回缓存的响应，避免重新向服务器发送请求。
 - 5.如果需要向服务器发送请求，浏览器会准备 IP 地址和端口，并等待 TCP 队列轮训执行。
 - 6.建立 TCP 连接，即三次握手，为后续的数据传输做好准备。
 - 7.发送 HTTP 请求，包括请求行、请求头和请求体。
 - 8.服务器处理请求，并将结果返回给浏览器。浏览器根据返回的结果进行页面渲染。
 - 9.最后断开 TCP 连接，即四次挥手。
- 6.恢复网络环境，将“首选DNS服务器”清空。

五、 实验结果总结

思考问题：

- 1.一个主页是否只有一个连接？

回答：一个主页不一定只有一个连接。一个主页可以有多个连接，这些连接可以指向网站的其他页面或者外部网站。连接的数量和类型取决于网站的设计和內容。一些网站的主页可能有大量的链接，而另一些网站的主页可能只有少数几个链接。

2.同时打开多个浏览器窗口并访问一个 WEB 站点的不同页面时，系统是根据什么把返回的页面正确地显示到相应窗口的？

回答：当使用多个浏览器窗口访问一个 WEB 站点的不同页面时，每个窗口实际上是由浏览器为每个独立的标签或选项卡创建的。每个窗口都有其独立的地址栏和状态栏。

每一个浏览器窗口可以有多个连接同时打开，这取决于用户在该窗口中打开的标签或选项卡的数量。每个连接对应一个 TCP/IP 连接，并使用一个特定的端口号来标识。

当用户在某个窗口中点击一个链接或提交一个表单时，浏览器会使用该窗口当前打开的连接（或创建一个新的连接）来发送 HTTP 请求到服务器。服务器处理该请求后，将响应数据通过相同的连接返回给浏览器。

3.为什么 HTTP 不保持与客户端的 TCP 连接？

回答：HTTP 协议不保持与客户端的 TCP 连接，主要是出于以下几个原因：

提高效率：HTTP 是一种无状态的协议，这意味着每次请求都是独立的，服务器不会为每个请求保持状态。通过不保持连接，服务器可以更快地处理请求并返回响应，从而提高整体效率。

节省资源：如果 HTTP 保持与客户端的 TCP 连接，那么服务器需要为每个连接维护额外的资源，如内存和文件描述符。随着连接数量的增加，服务器资源会迅速耗尽，影响服务器的性能和稳定性。通过不保持连接，服务器可以更有效地利用其资源来处理更多的请求。

支持并发请求：由于 HTTP 不保持连接，每个请求都可以独立发送，这使得服务器可以同时处理多个请求。这种并发处理能力是 HTTP 协议的重要特点，使得 Web 服务器能够支持大量的并发用户访问。

简化客户端实现：不保持连接简化了客户端的实现，因为客户端不需要管理连接的建立和释放。客户端只需发送请求并接收响应，而不需要处理连接的细节。

灵活性和可扩展性：不保持连接使得 HTTP 协议更加灵活和可扩展。例如，通过使用 HTTP 代理、负载均衡器等中间件，可以轻松地扩展和修改 HTTP 通信的行为。此外，不保持连接也使得 HTTP 能够与其他传输层协议（如 HTTPS 使用的 TLS/SSL）无缝集成。