计算机网络实验报告

警

1.实验报告如有雷同，雷同各方当次实验成绩均以 0 分计。
2.当次小组成员成绩只计学号、姓名登录在下表中的。
3.在规定时间内未上交实验报告的，不得以其他方式补交，当次成绩按 0 分计。
4.实验报告文件以 PDF 格式提交。

院系	数据科学与计算机学院	班 级	信息与计算科学	组长	回煜淼
学号	16339021	16343065	16339049		
学生	回煜淼	桑娜	辛依繁		
实验分工					
回煜淼	学习实验内容，小组讨论，共同完成实验	桑娜	学习实验内容，小组讨论，共同完成实验		
辛依繁	学习实验内容，小组讨论，共同完成实验				

【实验题目】访问控制列表（ACL）实验。

【实验目的】

1. 掌握标准访问列表规则及配置。
2. 掌握扩展访问列表规则及配置。
3. 了解标准访问列表和扩展访问列表的区别。

【实验内容】

完成教材实例 8-4（P296），请写出步骤 1 安装与建立 FTP、WEB 的步骤，并完成 P297～P298 的测试要求。

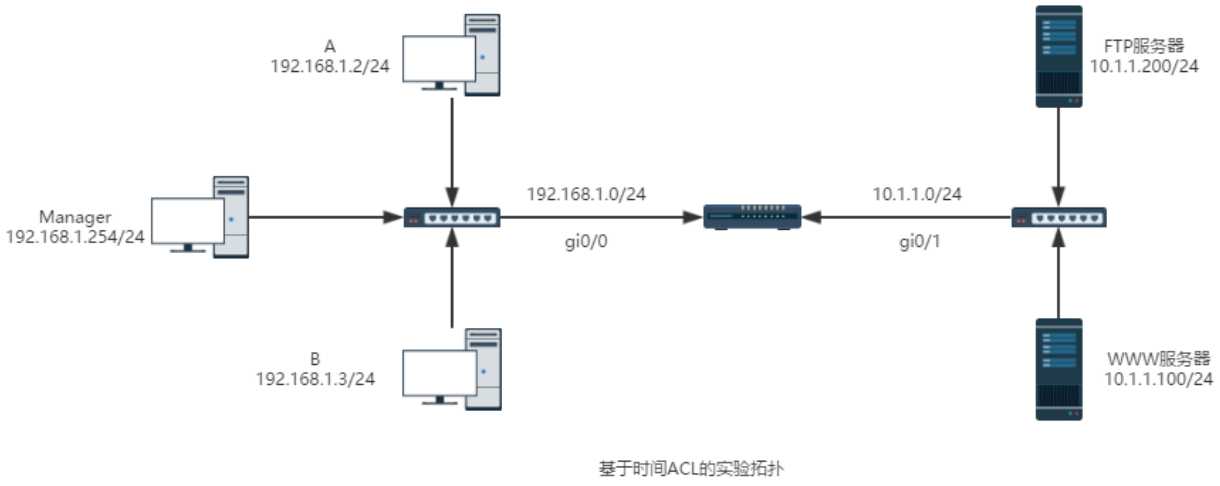
【实验要求】

重要信息信息需给出截图， 注意实验步骤的前后对比。

【实验记录】(如有实验拓扑请自行画出)

完成教材实例 8-4

实验拓扑如下：



基于时间ACL的实验拓扑

步骤 1：

（1） 配置 3 台计算机（A，B 和 Manager）的 IP 地址、子网掩码、网关。

（2） 检查计算机与服务器的连通性

三台主机与服务器的连通性如下所示：

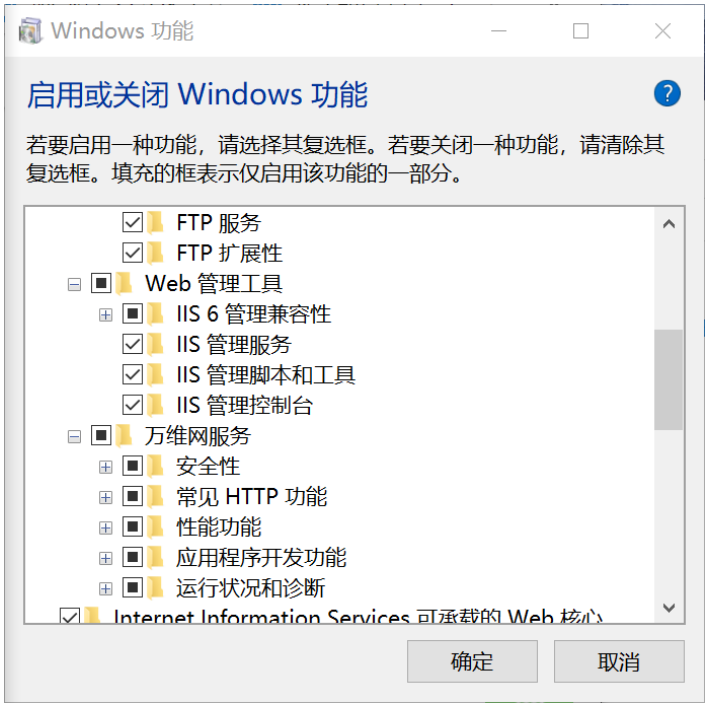
```
C:\Users\Administrator>ping 10.1.1.100

正在 Ping 10.1.1.100 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。

10.1.1.100 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),

C:\Users\Administrator>ping 10.1.1.200

正在 Ping 10.1.1.200 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。

10.1.1.200 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),
```

在配置路由器之前，三台主机与服务器之间是不连通的。

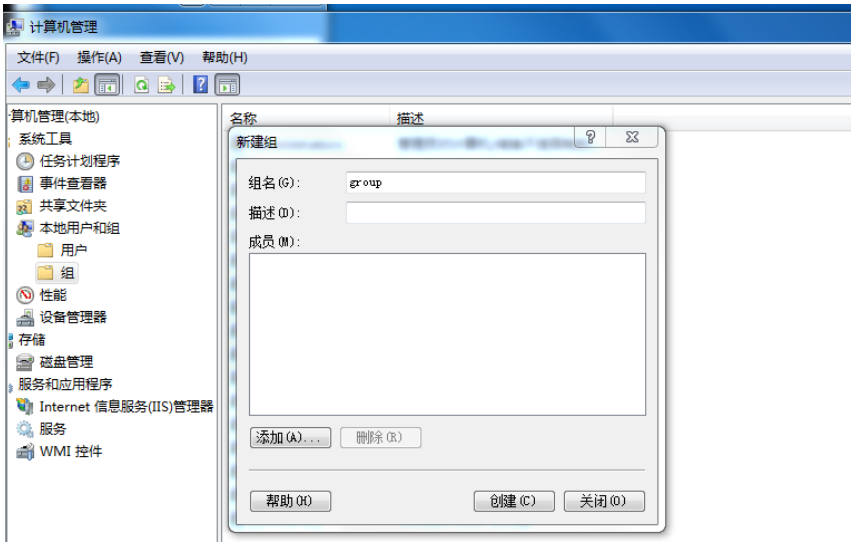
（3） 在服务器上安装 FTP 服务器和 WWW 服务器。FTP 服务器需至少创建一个用户名和口令。

FTP 服务器的建立：

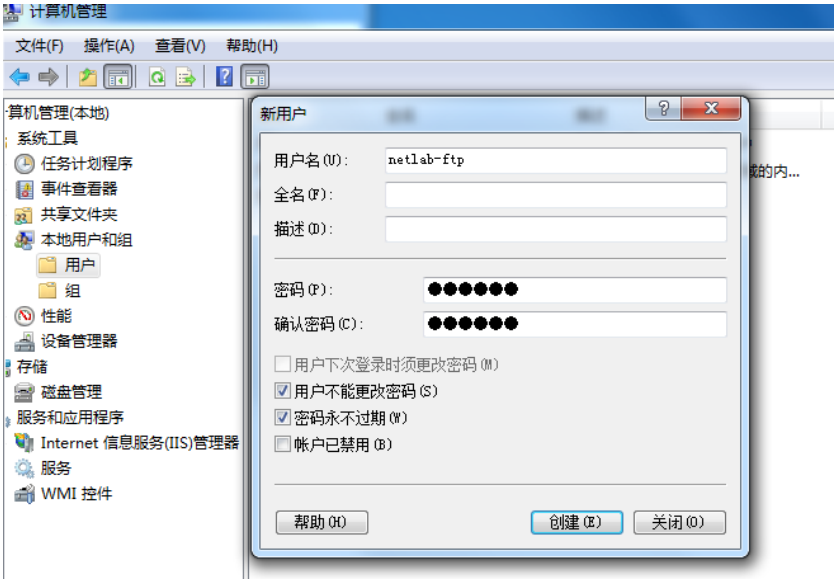
1. 打开控制面板——程序——打开或关闭 windows 功能——internet 信息服务——FTP 管理工具



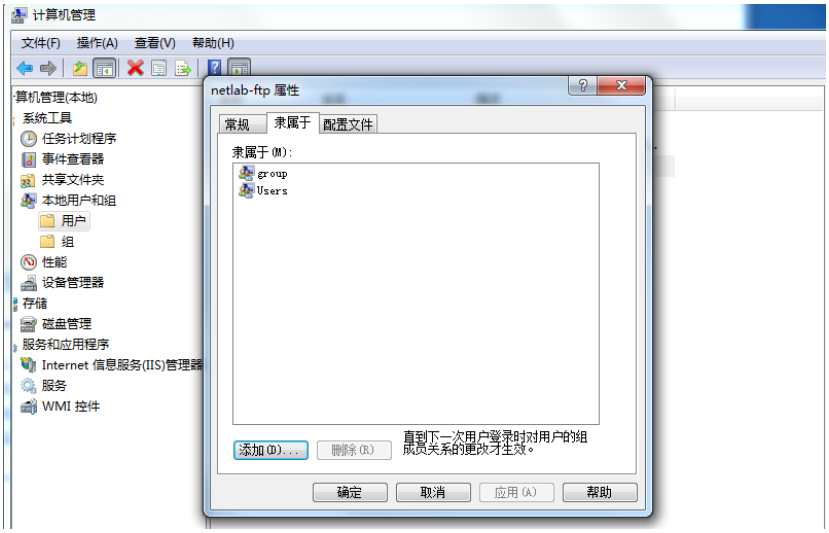
2. 右击“此电脑”——管理——本地用户和组——新建组——命名为“group”



3. 用户——新建用户——“netlab-ftp”——勾选“用户不能更改密码”、“密码永不过期”

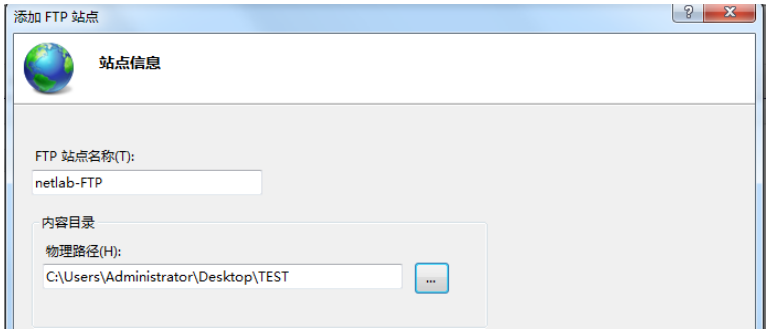


4. 在新建的用户“netlab-ftp”上右键——属性——隶属于，选择“group”



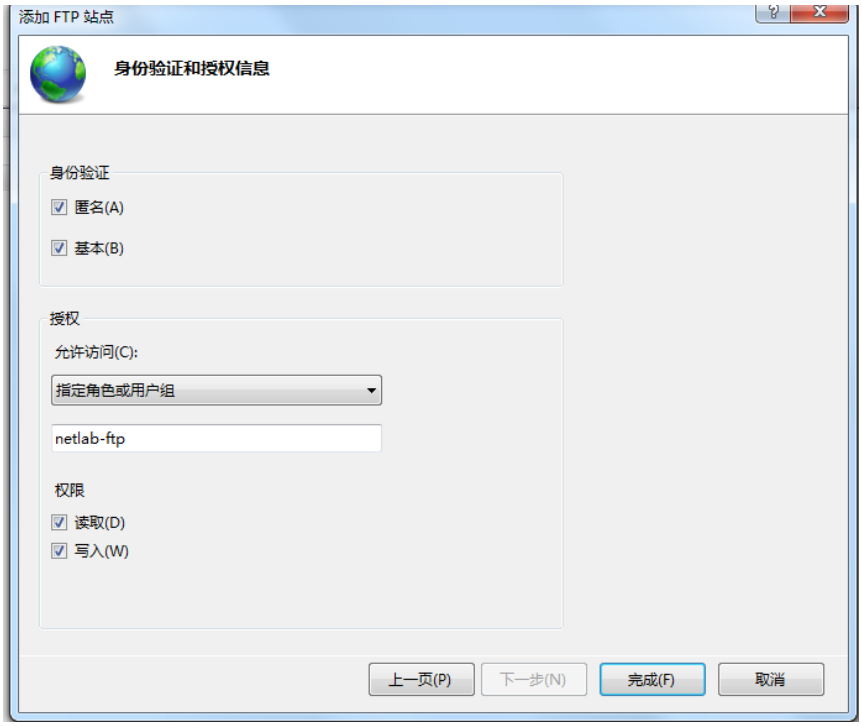
5. 搜索 iis——打开 iss——右键“网站”——添加 FTP 站点

6. 输入站点名称、物理路径：

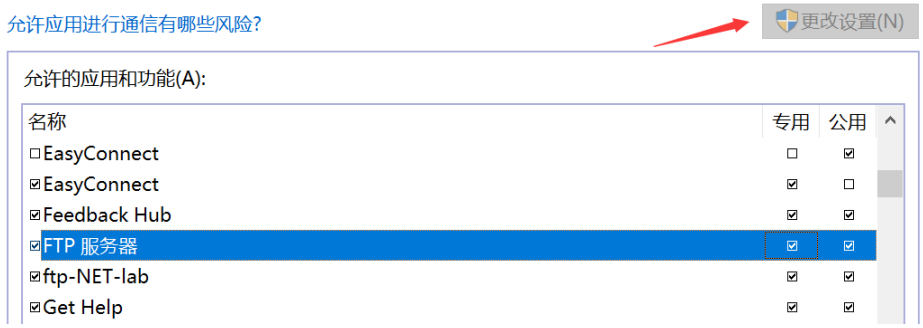


7. 绑定 ip 地址：10.1.1.200:21——无 SSL

8. 身份验证和授权信息——指定角色或用户组——netlab-ftp



9. 防火墙设置
   控制面板——系统和安全——防火墙——允许的应用——更改设置——勾选 FTP 服务器



10. 高级设置——入站规则——新建规则——程序

添加地址：C:\Windows\System32\svchost.exe

11. 在服务器上用浏览器登陆

计算机网络实验报告

/ 的索引

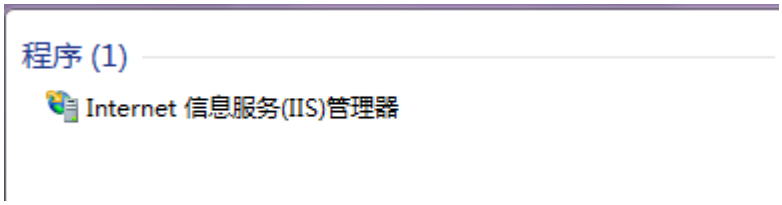
名称	大小	修改日期
test.docx	0 B	2018/6/3 上午12:24:00

成功~

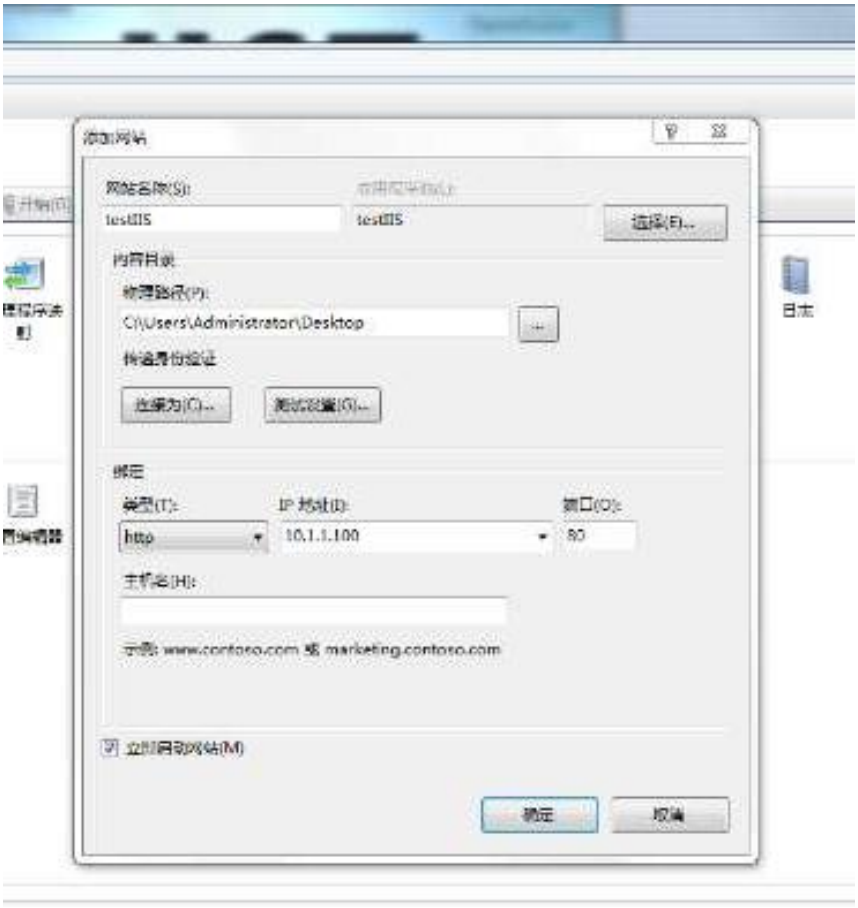
WWW 服务器的建立：前提保证连上实验网，已修改好 IP 地址。

1. 打开控制面板——程序——打开或关闭 windows 功能——internet 信息服务——web 管理工具

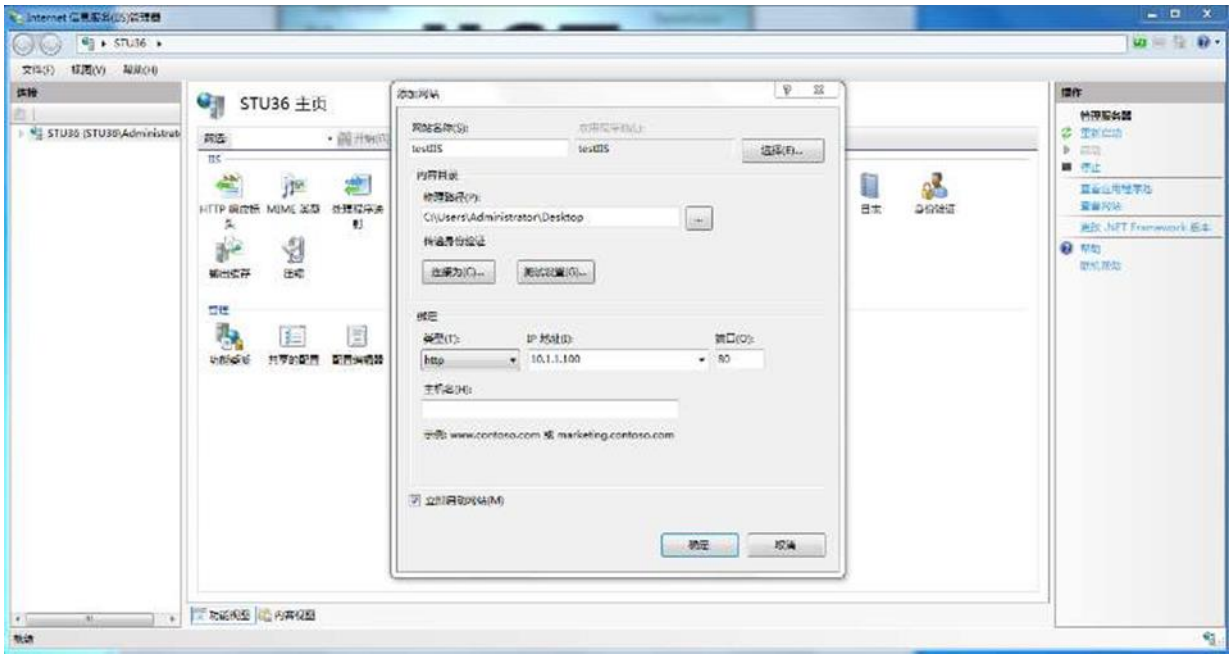
2. 在 “开始”栏里面寻找 IIS 管理器。单击进入界面

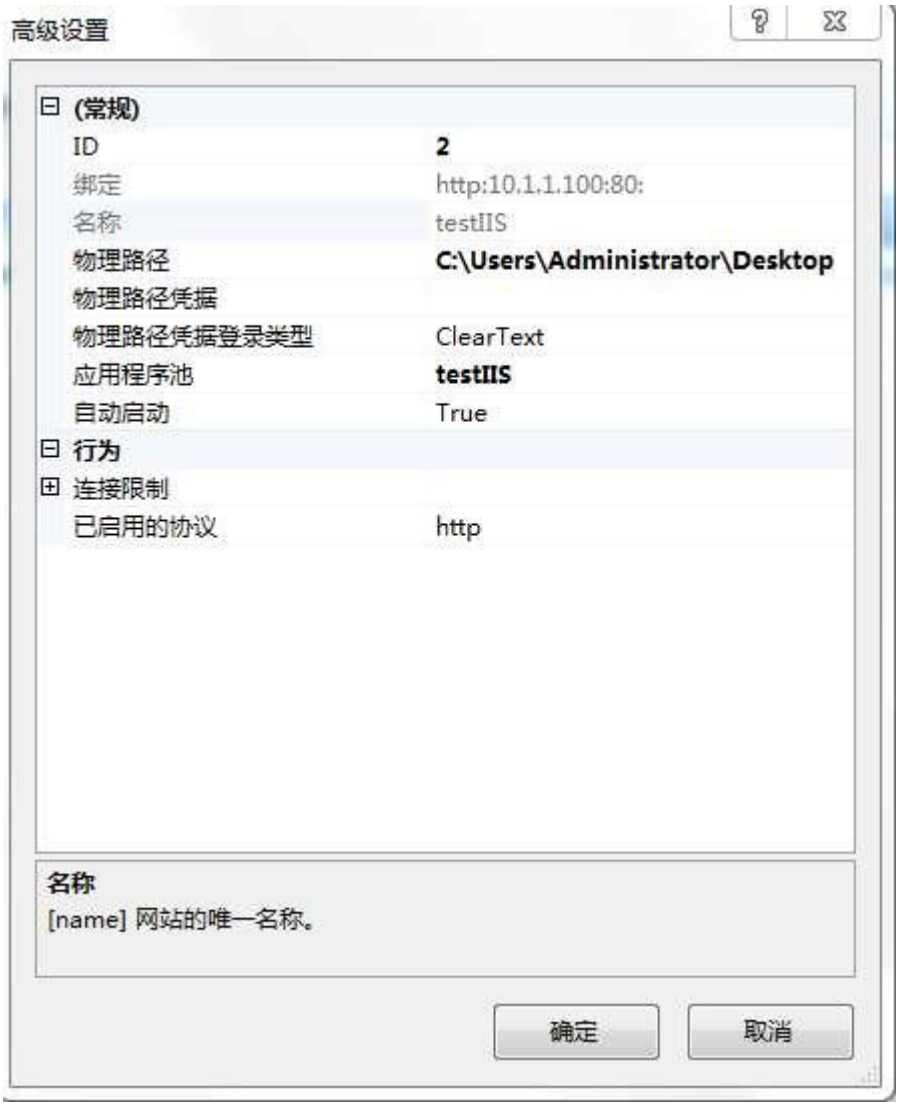


3. 在 default localhost 页面之后新增网站，此时定义网站名称为 testIIS，配置 IP 地址为 10.1.1.200，定义相应物理路径，注意物理路径的有效性。

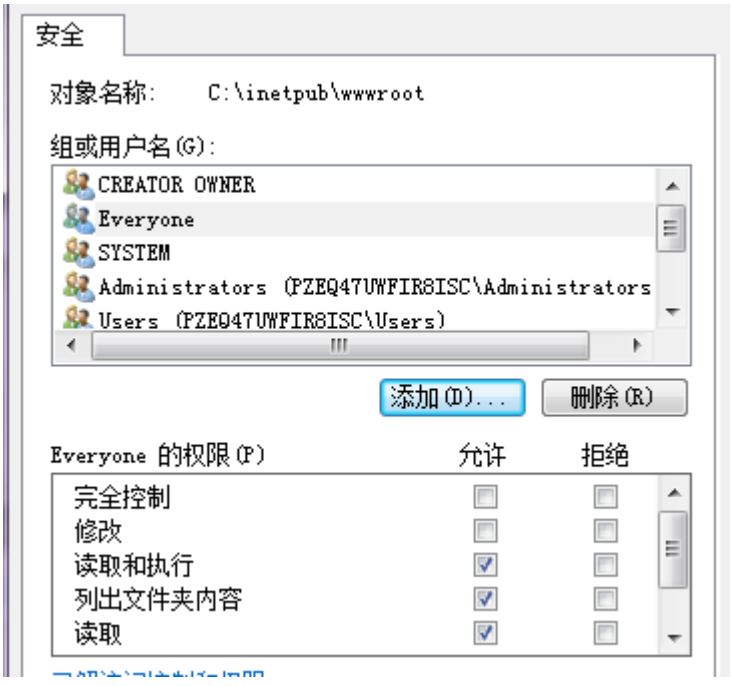


4. 设置好服务器的基本操作后，进入高级设置，检验相关参数是否正确。





5.之后右键进入编辑权限，此时增加“Everyone”，使得任何用户均可访问该 web 服务器。



6.最后本台 PC 上输入 http：//10.1.1.100，访问成功，可证 web 服务器搭建成功！

步骤 2：路由器基本配置。

首先，对路由器进行相关的基本配置，具体指令如下所示：

```
12-RSR20-1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
12-RSR20-1(config)#interface gigabitethernet 0/1
12-RSR20-1(config-if-GigabitEthernet 0/1)#ip address 10.1.1.1 255.255.255.0
12-RSR20-1(config-if-GigabitEthernet 0/1)#exit
12-RSR20-1(config)#interface gigabitethernet 0/0
12-RSR20-1(config-if-GigabitEthernet 0/0)#$2.168.1.1 255.255.255.0
12-RSR20-1(config-if-GigabitEthernet 0/0)#exit
12-RSR20-1(config)#
```

步骤 3：验证当前配置。

（1） 验证主机与服务器的连通性

如图所示，当我们对路由器进行配置之后，三台主机分别能够两个服务器连通，可以通过 ping 验证这一点，具体示意图如下：

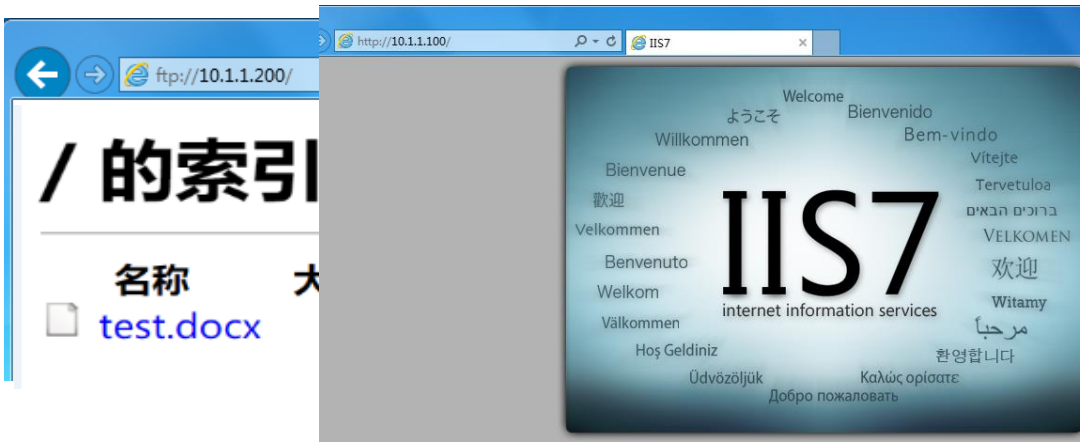
```
C:\Users\Administrator>ping 10.1.1.100

正在 Ping 10.1.1.100 具有 32 字节的数据:
来自 10.1.1.100 的回复: 字节=32 时间=7ms TTL=127
来自 10.1.1.100 的回复: 字节=32 时间<1ms TTL=127
来自 10.1.1.100 的回复: 字节=32 时间<1ms TTL=127
来自 10.1.1.100 的回复: 字节=32 时间=1ms TTL=127

10.1.1.100 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 7ms, 平均 = 2ms

C:\Users\Administrator>ping 10.1.1.200

正在 Ping 10.1.1.200 具有 32 字节的数据:
来自 10.1.1.200 的回复: 字节=32 时间<1ms TTL=127
来自 10.1.1.200 的回复: 字节=32 时间<1ms TTL=127
来自 10.1.1.200 的回复: 字节=32 时间<1ms TTL=127
来自 10.1.1.200 的回复: 字节=32 时间<1ms TTL=127

10.1.1.200 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 0ms, 平均 = 0ms
```

因为我们这个时候已经配置好路由器的目的网络地址，以及相应转发端口，因此，此时主机和服务器之间是可以连通的，连通性如上图所示。

（2） 经理机和员工机能否登录 FTP 服务器？通过 http://10.1.1.100 能否访问 WWW 服务器？判断目前结果是否达到预期目标，并说明原因。

首先观察经理机的状态，在浏览器的搜索栏里输入网址，敲击回车发现可以进入：

然后使用员工机进入，发现同样可以成功。



最后证明的结果是三台主机都可以在任何时间进入两个服务器。

目前的结果并没有达到预期目标，预期目标要实现基于时间段的访问控制，公司员工只有在正常上班时间才能访问 FTP 服务器，并且只有在下班时间才能访问 WWW 服务器，而经理机可以在任意时间访问这 2 台服务器，所以说并未达到预期目标。未能达到预期的原因是我们没有对时间进行设置，所以接下来我们要进行的步骤就是对访问时间进行控制。

步骤 4：配置时间段。

接下来我们定义了正常上班的时间段，输入相关指令如下：

```
12-RSR20-1(config)#time-range work-time
12-RSR20-1(config-time-range)#periodic weekdays 09:00 to 18:00
12-RSR20-1(config-time-range)#exit
```

步骤 5：配置 ACL。

配置 ACL 并应用时间段，以实现需求中基于时间段的访问控制。也就是要满足员工机只有特定的时间段才能进入，具体内容如下：

```
12-RSR20-1(config)#ip access-list extended accessctrl
12-RSR20-1(config-ext-nacl)#permit ip host 192.168.1.254 10.1.1.0.0.0.0.255
                                                          ^
% Invalid input detected at '^' marker.

12-RSR20-1(config-ext-nacl)#permit tcp 192.168.1.254 10.1.1.0.0.0.0.255
                                                          ^
% Invalid input detected at '^' marker.

12-RSR20-1(config-ext-nacl)#permit ip host 192.168.1.254 10.1.1.0 0.0.0.255
12-RSR20-1(config-ext-nacl)#$host 10.1.1.200 eq ftp time-range work-time
12-RSR20-1(config-ext-nacl)#$host 10.1.1.200 eq ftp-data time-range work-time
12-RSR20-1(config-ext-nacl)#$st 10.1.1.100 eq www time-range work-time
12-RSR20-1(config-ext-nacl)#$1.0 0.0.0.255 host 10.1.1.100 eq www
12-RSR20-1(config-ext-nacl)#exit
12-RSR20-1(config)#
```

步骤 6：应用 ACL。

将 ACL 应用到端口 0/0 的输入方向。

```
12-RSR20-1(config)#interface gigabitethernet 0/0
12-RSR20-1(config-if-GigabitEthernet 0/0)#ip access-group accessctrl in
12-RSR20-1(config-if-GigabitEthernet 0/0)#end
```

步骤 7：验证测试。

在使用基于时间的 ACL 时，要保证设备（路由器或交换机）的系统时间的准确性，因为

设备是根据自己的系统时间（而不是主机时间）判断当前时间是否在时间段范围内。可以在特权模式下使用 show clock 命令查看当前系统时间，并使用 clock set 命令调整系统时间。通过调整设备的系统时间实现在不同时间段测试 ACL 是否生效。

本实验分别做下列测试：

（1） 查看路由器的系统时间：使用 show clock 命令判断当前时间段。

```
12-RSR20-2#show clock
17:58:14 UTC Mon, Jun 4, 2018
12-RSR20-2#
```

这里我们查看了路由器的设备的时间，同时将设备时间设为 2018 年 6 月 4 日 17:58，是星期一，处于上班时间段。

（2） 经理的主机 Manager 使用步骤 1 建立的用户名登录 FTP 服务器，并通过 http://10.1.1.100 访问 WWW 服务器，在设定时间段内是否能登录和访问？

经理机访问情况如下：

可以看出经理机在上班时间(2018 年 6 月 4 日 17：58)两个服务器都可以正常访问

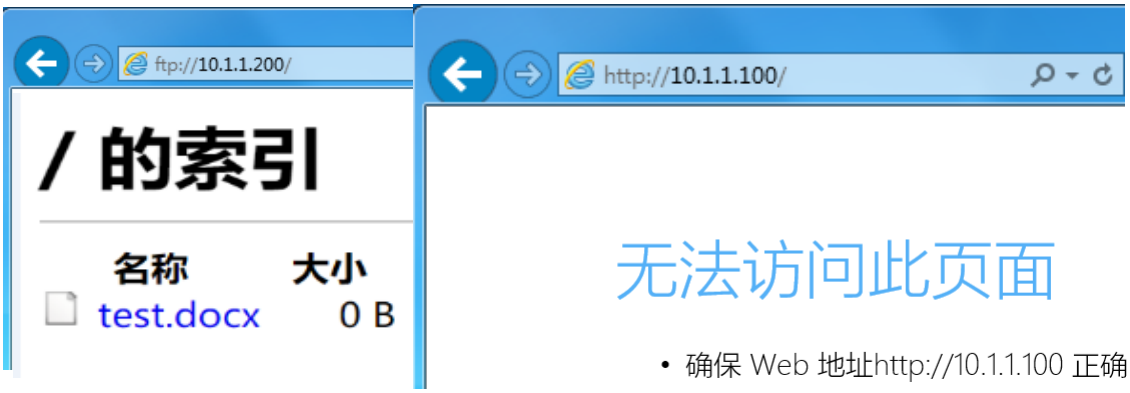
（3） 普通员工主机 A、B 分别使用步骤 1 建立的用户名登录 FTP 服务器，并通过 http://10.1.1.100 访问 WWW 服务器，在设定时间段内是否能登录和访问（登录 FTP 时分别通过 DOS 命令与浏览器方式，结合捕获报文分析）？

普通主机访问情况如下：

首先观察 DOS 的访问 FTP 情况：

```
C:\Users\Administrator>ftp 10.1.1.200
连接到 10.1.1.200.
220 Microsoft FTP Service
用户<10.1.1.200:<none>>:netlab-ftp
331 Password required
密码
230 User logged in.
ftp>
```

再以浏览器方式访问 ftp 和 WWW 服务器的情况：



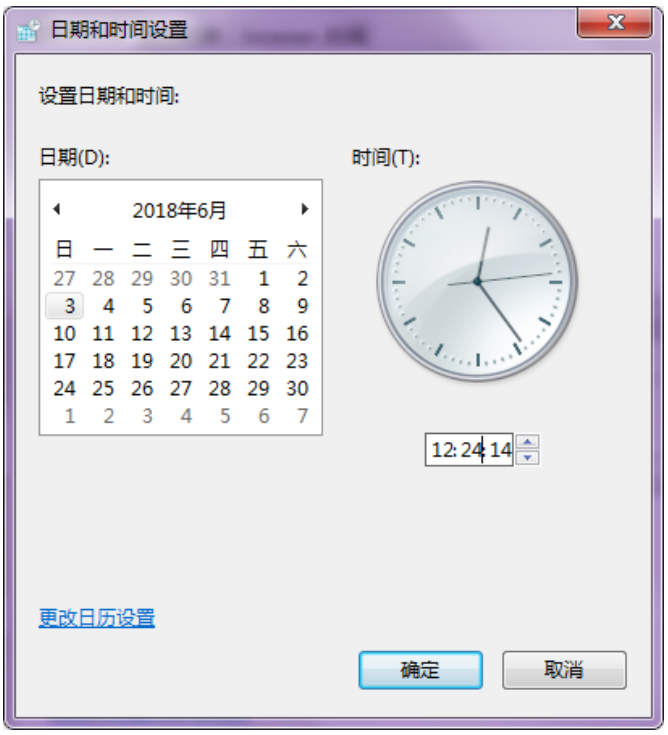
注：这个时候员工机可以访问 ftp 服务器，此时可以捕获到 ftp 报文，报文分析与（5）相同，此处不赘述～

可见普通主机在上班时间只能够访问 FTP 服务器，不能访问 WWW 服务器。

（4） 改变路由器系统时间段，在其他时间段执行（2）~（3）的测试。

修改路由器系统时间：

输入指令：clock set 12：24：00 6 3 2018



可见修改的时间为 2018 年 6 月 3 日 12:24，为非上班时间。

经理机连接情况：

对经理机进行刷新后，依旧没有任何变化，即经理机依旧可以顺畅的访问两个服务器。

普通机连接情况如下：

可见普通机在非上班时间是无法连接 FTP 服务器，却可以连接到 WWW 服务器中，结合前两步的验证可以看出，如今的设置已经符合我们的需求了。

（5） 捕获主机访问服务器时的数据包，并进行分析。

访问 FTP 服务器进行抓包如下：

员工机：

```
10.1.1.200      192.168.1.2 FTP      Response: 220 Microsoft FTP Service
192.168.1.2     10.1.1.200  FTP      Request: USER netlab-ftp
10.1.1.200      192.168.1.2 FTP      Response: 331 Password required for netlab-ftp.
192.168.1.2     10.1.1.200  FTP      Request: PASS 123456
10.1.1.200      192.168.1.2 FTP      Response: 230 User logged in.
```

可以看出，连接 FTP 服务器使用的是 ftp 协议，我们可以从抓包中直接看到访问 ftp 使用的用户名和密码。

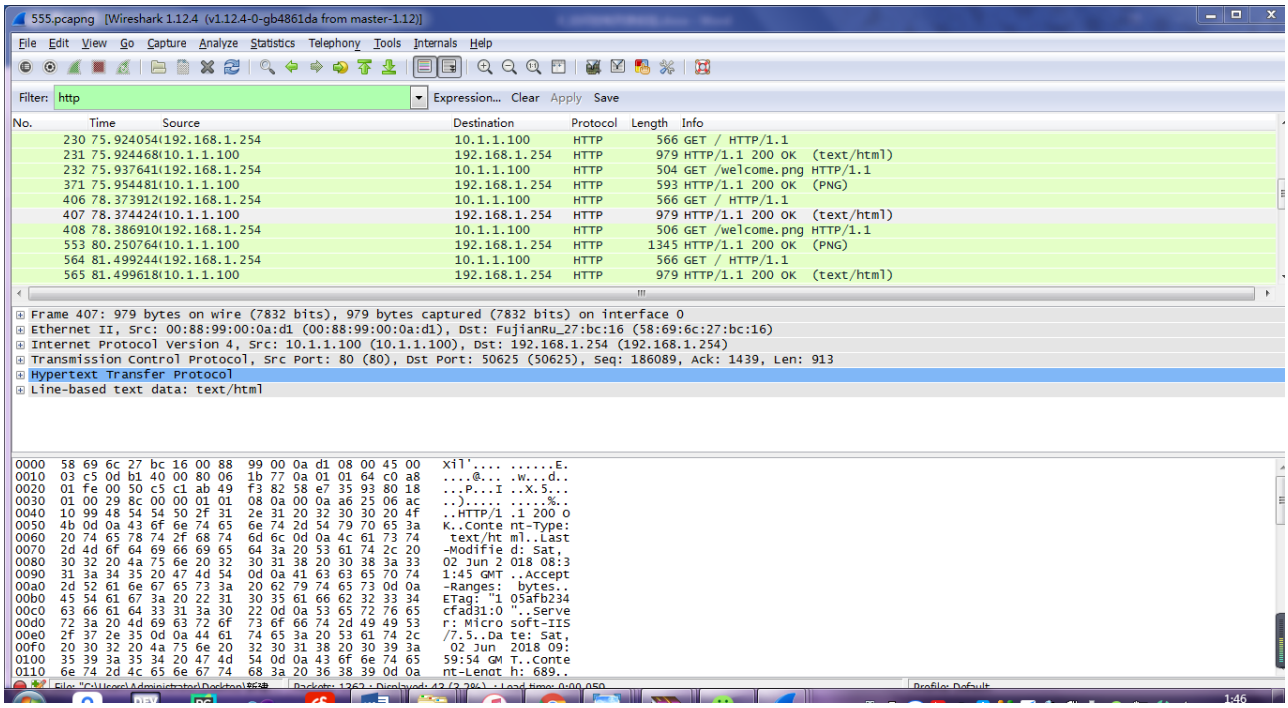
```
192.168.1.2     10.1.1.200  FTP      Request: SYST
10.1.1.200      192.168.1.2 FTP      Response: 215 Windows_NT
192.168.1.2     10.1.1.200  FTP      Request: PWD
10.1.1.200      192.168.1.2 FTP      Response: 257 "/" is current directory.
192.168.1.2     10.1.1.200  FTP      Request: TYPE I
10.1.1.200      192.168.1.2 FTP      Response: 200 Type set to I.
192.168.1.2     10.1.1.200  FTP      Request: SIZE /
10.1.1.200      192.168.1.2 FTP      Response: 550 Access is denied.
192.168.1.2     10.1.1.200  FTP      Request: CWD /
10.1.1.200      192.168.1.2 FTP      Response: 250 CWD command successful.
192.168.1.2     10.1.1.200  FTP      Request: PASV
10.1.1.200      192.168.1.2 FTP      Response: 227 Entering Passive Mode (10,1,1,200,5,253).
192.168.1.2     10.1.1.200  TCP      1118 → 21 [ACK] Seq=71 Ack=255 Win=65280 Len=0
192.168.1.2     10.1.1.200  FTP      Request: QUIT
10.1.1.200      192.168.1.2 FTP      Response: 221 Goodbye.
```

SYST：返回服务器使用的操作系统；215：系统类型为 Windows_NT；
PWD：显示当前工作目录；257：路径名为“/”；
TYPE Ⅰ：数据类型，2 进制；200：成功；
CWD：改变服务器上的工作目录；250：文件行为完成。

经理机：

```
10.1.1.200      192.168.1.254     FTP       93 Response: 220 Microsoft FTP Service
192.168.1.254   10.1.1.200        FTP       83 Request: USER netlab-ftp
10.1.1.200      192.168.1.254     FTP      105 Response: 331 Password required for netlab-ftp.
192.168.1.254   10.1.1.200        FTP       79 Request: PASS 123456
10.1.1.200      192.168.1.254     FTP       87 Response: 230 User logged in.
```

访问 WWW 服务器进行抓包如下：



此时表示的是经理机（192.168.1.254）和 web 服务器（10.1.1.100）之间的传输协议为 http。HTTP 协议即超文本传送协议(Hypertext Transfer Protocol)，是 Web 联网的基础，也是手机联网常用的协议之一，HTTP 协议是建立在 TCP 协议之上的一种应用。

HTTP 连接最显著的特点是客户端发送的每次请求都需要服务器回送响应，在请求结束后，会主动释放连接。从建立连接到关闭连接的过程称为"一次连接".

230	75.924054	192.168.1.254	10.1.1.100	HTTP 566 GET / HTTP/1.1
231	75.924468	10.1.1.100	192.168.1.254	HTTP 979 HTTP/1.1 200 OK (text/html)
232	75.937641	192.168.1.254	10.1.1.100	HTTP 504 GET /welcome.png HTTP/1.1
371	75.954481	10.1.1.100	192.168.1.254	HTTP 593 HTTP/1.1 200 OK (PNG)

230	75.924054	192.168.1.254	10.1.1.100	HTTP 566 GET / HTTP/1.1
231	75.924468	10.1.1.100	192.168.1.254	HTTP 979 HTTP/1.1 200 OK (text/html)
232	75.937641	192.168.1.254	10.1.1.100	HTTP 504 GET /welcome.png HTTP/1.1
371	75.954481	10.1.1.100	192.168.1.254	HTTP 593 HTTP/1.1 200 OK (PNG)

由于 HTTP 在每次请求结束后都会主动释放连接，因此 HTTP 连接是一种"短连接"，要保持客户端程序的在线状态，需要不断地向服务器发起连接请求。通常的 做法是即时不需要获得任何数据，客户端也保持每隔一段固定的时间向服务器发送一次"保持连接"的请求，服务器在收到该请求后对客户端进行回复，表明知道客户端"在线"。若服务器长时间无法收到客户端的请求，则认为客户端"下线"，若客户端长时间无法收到服务器的回复，则认为网络已经断开。

利用过滤器筛选出 http 协议，具体四个包为一分组一次请求网页，然后响应成功，再一次请求欢迎界面，然后再次响应成功：

```
Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
      [GET / HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: GET
    Request URI: /
    Request Version: HTTP/1.1
  Host: 10.1.1.100\r\n
```

```
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      [HTTP/1.1 200 OK\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Version: HTTP/1.1
    Status Code: 200
    Response Phrase: OK
  Content-Type: text/html\r\n
  Last-Modified: Sat, 02 Jun 2018 08:31:45 GMT\r\n
  Accept-Ranges: bytes\r\n
  ETag: "105afb234cfad31:0"\r\n
  Server: Microsoft-IIS/7.5\r\n
  Date: Sat, 02 Jun 2018 09:59:54 GMT\r\n
  Content-Length: 689\r\n
  \r\n
  [HTTP response 7/14]
  [Time since request: 0.000293000 seconds]
```

```
Hypertext Transfer Protocol
  GET /welcome.png HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /welcome.png HTTP/1.1\r\n]
    Request Method: GET
    Request URI: /welcome.png
    Request Version: HTTP/1.1
  Host: 10.1.1.100\r\n
```

```
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
       [HTTP/1.1 200 OK\r\n]
       [Severity level: Chat]
       [Group: Sequence]
     Request Version: HTTP/1.1
     Status Code: 200
     Response Phrase: OK
  Content-Type: image/png\r\n
  Last-Modified: Sat, 02 Jun 2018 08:31:45 GMT\r\n
  Accept-Ranges: bytes\r\n
  ETag: W/"70b5fe234cfad31:0"\r\n
  Server: Microsoft-IIS/7.5\r\n
  Date: Sat, 02 Jun 2018 09:59:54 GMT\r\n
```

其中 content-type 中 text/html 是指 text/html 的意思是将文件的 content-type 设置为 text/html 的形式，浏览器在获取到这种文件时会自动调用 html 的解析器对文件进行相应的处理。png 是指发送的是便携式网络图形。

我们先大致分析其中的一个包：

```
407 78.374424000 10.1.1.100 192.168.1.254 HTTP 979 HTTP/1.1 200 OK  (text/html)
Frame 407: 979 bytes on wire (7832 bits), 979 bytes captured (7832 bits) on interface 0
  Interface id: 0 (\Device\NPF_{C5167126-78C4-4B3C-BFDB-DD5EB87F177C})
  Encapsulation type: Ethernet (1)
  Arrival Time: May 30, 2018 10:09:00.828495000 [illegible]
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1527646140.828495000 seconds
  [Time delta from previous captured frame: 0.000512000 seconds]
  [Time delta from previous displayed frame: 0.000512000 seconds]
  [Time since reference or first frame: 78.374424000 seconds]
  Frame Number: 407
  Frame Length: 979 bytes (7832 bits)
  Capture Length: 979 bytes (7832 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:tcp:http:data-text-lines]
  [Number of per-protocol-data: 1]
  [Hypertext Transfer Protocol, key 0]
  [Coloring Rule Name: HTTP]
  [Coloring Rule String: http || tcp.port == 80 || http2]
Ethernet II, Src: 00:88:99:00:0a:d1 (00:88:99:00:0a:d1), Dst: FujianRu_27:bc:16 (58:69:6c:27:bc:16)
  Destination: FujianRu_27:bc:16 (58:69:6c:27:bc:16)
  Source: 00:88:99:00:0a:d1 (00:88:99:00:0a:d1)
  Type: IP (0x0800)
Internet Protocol Version 4, Src: 10.1.1.100 (10.1.1.100), Dst: 192.168.1.254 (192.168.1.254)
  Version: 4
  Header Length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
  Total Length: 965
  Identification: 0x0db1 (3505)
  Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 128
  Protocol: TCP (6)
  Header checksum: 0x1b77 [validation disabled]
  Source: 10.1.1.100 (10.1.1.100)
  Destination: 192.168.1.254 (192.168.1.254)
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
```

此时这个数据包的目的地址为 192.168.1.254,源地址为 10.1.1.100，数据帧号为 407，此时 http 协议采用 tcp 端口 80.

```
[Destination GeoIP: Unknown]
Transmission Control Protocol, Src Port: 80 (80), Dst Port: 50625 (50625), Seq: 186089, Ack: 1439, Len: 913
    Source Port: 80 (80)
    Destination Port: 50625 (50625)
    [Stream index: 1]
    [TCP Segment Len: 913]
    Sequence number: 186089    (relative sequence number)
    [Next sequence number: 187002    (relative sequence number)]
    Acknowledgment number: 1439    (relative ack number)
    Header Length: 32 bytes
  .... 0000 0001 1000 = Flags: 0x018 (PSH, ACK)
    Window size value: 256
    [Calculated window size: 65536]
    [Window size scaling factor: 256]
  Checksum: 0x298c [validation disabled]
    Urgent pointer: 0
  Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  [SEQ/ACK analysis]
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    Content-Type: text/html\r\n
    Last-Modified: Sat, 02 Jun 2018 08:31:45 GMT\r\n
    Accept-Ranges: bytes\r\n
    ETag: "105afb234cfad31:0"\r\n
    Server: Microsoft-IIS/7.5\r\n
    Date: Sat, 02 Jun 2018 09:59:54 GMT\r\n
  Content-Length: 689\r\n
    \r\n
    [HTTP response 3/14]
    [Time since request: 0.000512000 seconds]
    [Prev request in frame: 232]
    [Prev response in frame: 371]
    [Request in frame: 406]
    [Next request in frame: 408]
    [Next response in frame: 553]
Line-based text data: text/html
```

我们发现：HTTP 是应用层协议，TCP 是传输层协议！数据包在网络传输过程中，HTTP 被封装在 TCP 包内，也就是说 http 是基于 tcp 传输而实现的。看 http 数据包，里面有修改数据的时间以及 web 服务器的信息---Microsoft-IIS，内容类型，长度等参数。我们把 http 协议分为两类，具体如下：

请求 GET:

```
[SEQ/ACK analysis]
Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
      [GET / HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: GET
    Request URI: /
    Request Version: HTTP/1.1
  Host: 10.1.1.100\r\n
  Connection: keep-alive\r\n
  Cache-Control: max-age=0\r\n
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36\r\n
  Upgrade-Insecure-Requests: 1\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: en,zh-CN;q=0.9,zh;q=0.8\r\n
  If-None-Match: "105afb234cfad31:0"\r\n
  If-Modified-Since: Sat, 02 Jun 2018 08:31:45 GMT\r\n
  \r\n
  [Full request URI: http://10.1.1.100/]
  [HTTP request 7/14]
  [Prev request in frame: 566]
  [Response in frame: 718]
  [Next request in frame: 719]
```

GET/HTTP/1.1 协议及版本为：HTTP/1.1

Host 请求的主机名为 10.1.1.100

User-Agent: Mozilla/5.0 (Windows NT 10.0; WIN 64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/36.0.198：与浏览器和操作系统有关的信息，有些网站会显示用户的系统版本和浏览器的版本信息，这都是通过获取该头部得到的。

Accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8:告诉服务器当前客户端可以接收的文档的类型。其实这里包含了*/*，就表示什么都可以接收；

Accept-Language: en，q=0.9；zh-CN,zh;q=0.8:当前客户端可以支持的语言，在浏览器的工具->选项中可以得到相关信息

Accept-Encoding: gzip,deflate,sdch:客户端支持的编码
Connection: keep-alive:客户端支持的连接方式，保持一段连接，默认为 3000ms

响应 Response：

```
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
        [HTTP/1.1 200 OK\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Version: HTTP/1.1
      Status Code: 200
      Response Phrase: OK
    Content-Type: image/png\r\n
    Last-Modified: Sat, 02 Jun 2018 08:31:45 GMT\r\n
    Accept-Ranges: bytes\r\n
    ETag: W/"70b5fe234cfad31:0"\r\n
    Server: Microsoft-IIS/7.5\r\n
    Date: Sat, 02 Jun 2018 09:59:54 GMT\r\n
  + Content-Length: 184946\r\n
    \r\n
    [HTTP response 6/14]
    [Time since request: 0.003251000 seconds]
    [Prev request in frame: 564]
    [Prev response in frame: 565]
    [Request in frame: 566]
    [Next request in frame: 717]
    [Next response in frame: 718]
```

HTTP/1.1 200 OK：响应协议为 HTTP1.1，状态码为 200，表示请求成功，OK 是对状态码的解释；
Server: 服务器的版本信息；
Content-Type: image/png 便携式网络图形

```
Portable Network Graphics
    PNG Signature: 89504e470d0a1a0a
  + Image Header (IHDR)
  + Image data chunk (IDAT)
  + Image Trailer (IEND)
```

Date:响应的时间，这可能会有 8 小时的时区差。

本次实验完成后，请根据组员在实验中的贡献，请实事求是，自评在实验中应得的分数。（按百分制）

学号	学生	自评分
16339021	回煜淼	100
16343065	桑娜	100
16339049	辛依繁	100