第一次实验

1 互联网协议实验

1.1 实验内容

使用 wireshark 抓包工具，用 wget 下载 www.ucas.ac.con，观察输出。根据输出分析 ARP、DNS、TCP、HTTP 协议的工作过程。

1.2 实验步骤

在 ubuntu 虚拟机终端中依次输入以下命令：

```
sudo mn --nat # 启动mininet
xterm h1 # 打开h1终端
echo "nameserver 1.2.4.8" > /etc/resolv.conf # 设置DNS服务器
wireshark & # 打开wireshark，然后选择h1-eth0开始抓包
wget www.ucas.ac.cn # h1中输入该指令，下载网页，开始调研
```

1.3 实验结果

如下图所示：

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	96:db:c9:a8:6c:1e	Broadcast	ARP	42	Who has 10.0.0.3? Tell 10.0.0.1
2	0.000196508	32:7b:f8:78:af:0e	96:db:c9:a8:6c:1e	ARP	42	10.0.0.3 is at 32:7b:f8:78:af:0e
3	0.000198247	10.0.0.1	1.2.4.8	DNS	74	Standard query 0x9773 A www.ucas.ac.cn
4	0.000198994	10.0.0.1	1.2.4.8	DNS	74	Standard query 0xdb7d AAAA www.ucas.ac.cn
5	0.093098375	1.2.4.8	10.0.0.1	DNS	90	Standard query response 0x9773 A www.ucas.ac.cn A 124.16.77.5
6	0.110865487	1.2.4.8	10.0.0.1	DNS	132	Standard query response 0xdb7d AAAA www.ucas.ac.cn SOA gsns.gscas.ac.cn
7	0.110979655	10.0.0.1	124.16.77.5	TCP	74	37744 → 80 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK_PERM TSval=2451964676 TSecr=0 WS=512
8	0.150074917	124.16.77.5	10.0.0.1	TCP	58	80 → 37744 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
9	0.150098825	10.0.0.1	124.16.77.5	TCP	54	37744 → 80 [ACK] Seq=1 Ack=1 Win=42340 Len=0
10	0.150663978	10.0.0.1	124.16.77.5	HTTP	183	GET / HTTP/1.1
11	0.151377422	124.16.77.5	10.0.0.1	TCP	54	80 → 37744 [ACK] Seq=1 Ack=130 Win=65535 Len=0
12	0.182157419	124.16.77.5	10.0.0.1	HTTP	392	HTTP/1.1 301 Moved Permanently (text/html)
13	0.182185094	10.0.0.1	124.16.77.5	TCP	54	37744 → 80 [ACK] Seq=130 Ack=339 Win=42002 Len=0
14	0.210877957	10.0.0.1	124.16.77.5	TCP	74	33772 → 443 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK_PERM TSval=2451964776 TSecr=0 WS=512
15	0.310139680	124.16.77.5	10.0.0.1	TCP	58	443 → 33772 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
16	0.310169524	10.0.0.1	124.16.77.5	TCP	54	33772 → 443 [ACK] Seq=1 Ack=1 Win=42340 Len=0
17	0.331911272	10.0.0.1	124.16.77.5	TLSv1.2	462	Client Hello (SNI=www.ucas.ac.cn)
18	0.332817940	124.16.77.5	10.0.0.1	TCP	54	443 → 33772 [ACK] Seq=1 Ack=409 Win=65535 Len=0
19	0.372323042	124.16.77.5	10.0.0.1	TLSv1.2	2774	Server Hello
20	0.372348837	10.0.0.1	124.16.77.5	TCP	54	33772 → 443 [ACK] Seq=409 Ack=2721 Win=40880 Len=0
21	0.406692690	124.16.77.5	10.0.0.1	TLSv1.2	684	Certificate, Server Key Exchange, Server Hello Done
22	0.406711098	10.0.0.1	124.16.77.5	TCP	54	33772 → 443 [ACK] Seq=409 Ack=3351 Win=40880 Len=0
23	0.417590970	10.0.0.1	124.16.77.5	TLSv1.2	147	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
24	0.418205667	124.16.77.5	10.0.0.1	TCP	54	443 → 33772 [ACK] Seq=3351 Ack=502 Win=65535 Len=0
25	0.453125187	124.16.77.5	10.0.0.1	TLSv1.2	312	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
26	0.453289664	10.0.0.1	124.16.77.5	TLSv1.2	212	Application Data
27	0.453896212	124.16.77.5	10.0.0.1	TCP	54	443 → 33772 [ACK] Seq=3609 Ack=660 Win=65535 Len=0
28	0.480359640	124.16.77.5	10.0.0.1	TCP	1414	443 → 33772 [PSH, ACK] Seq=3609 Ack=660 Win=65535 Len=1360 [TCP segment of a reassembled PDU]

图 1. wireshark 抓包结果

然后进入各个协议中查看详细信息，如下几张图所示：

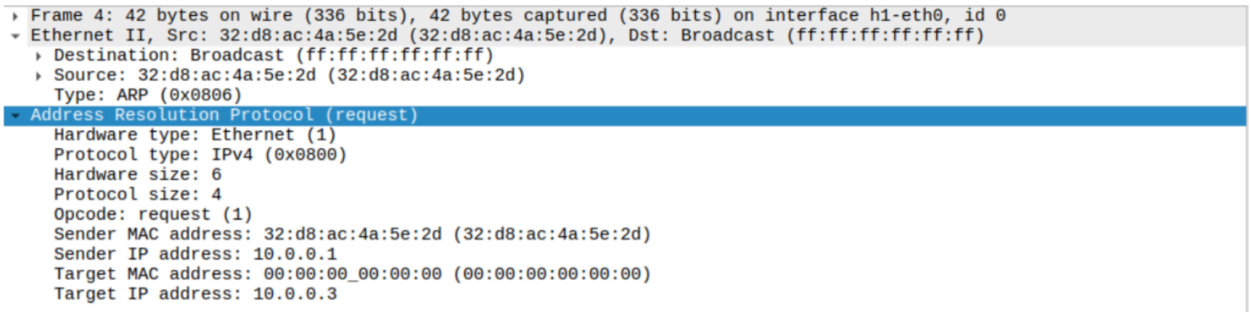

图 **2.** ARP 协议

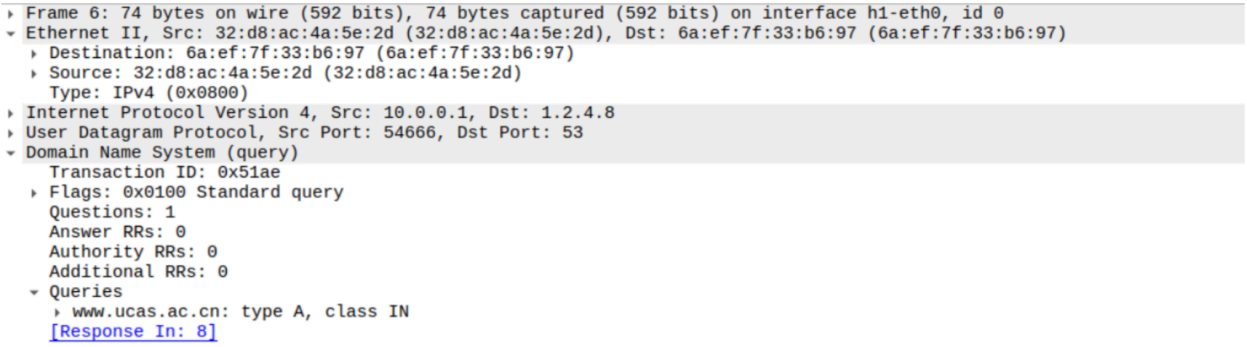

图 **3.** DNS 协议

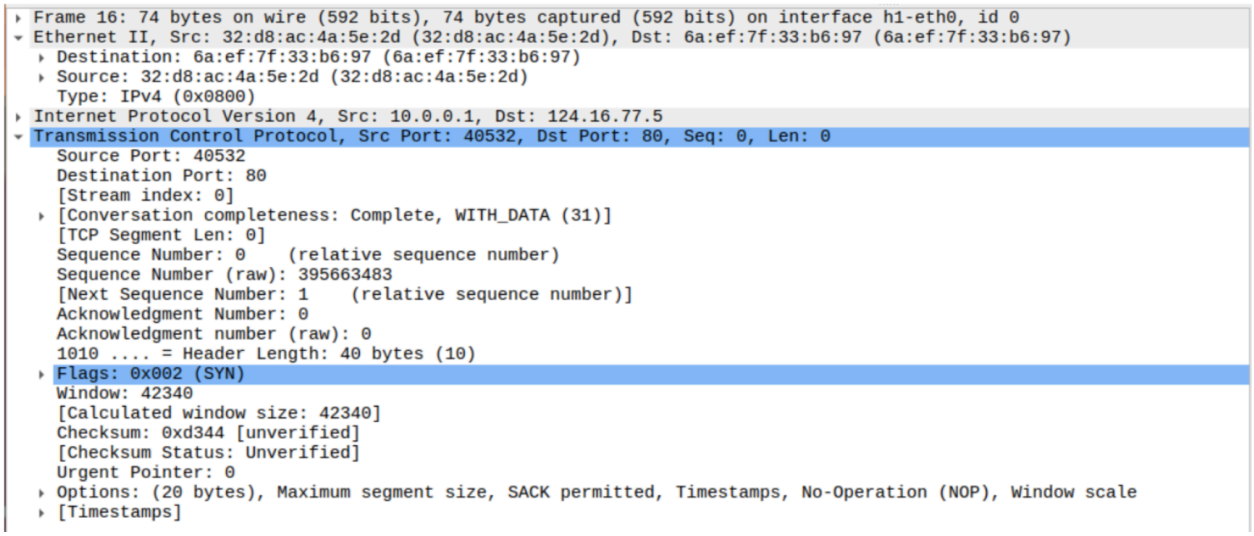

图 **4.** TCP 协议

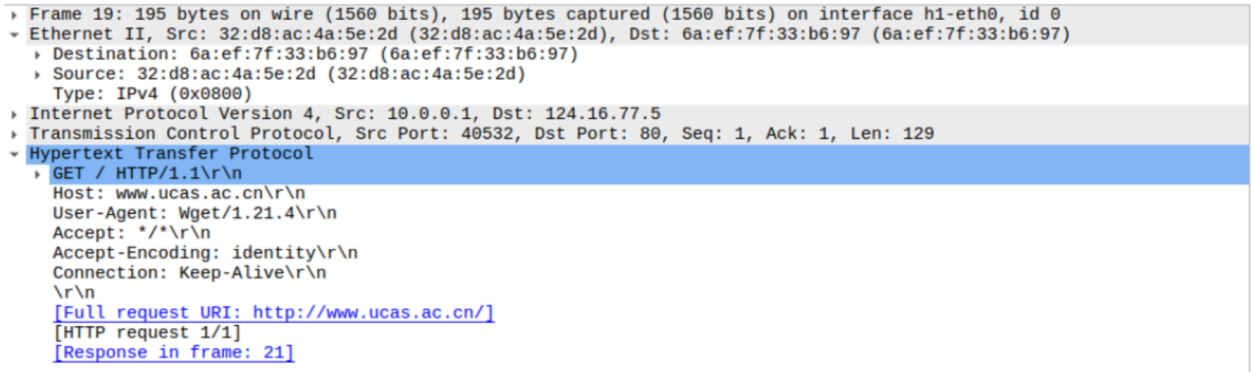

图 **5.** HTTP 协议

还得到了 TCP 流的详细信息：

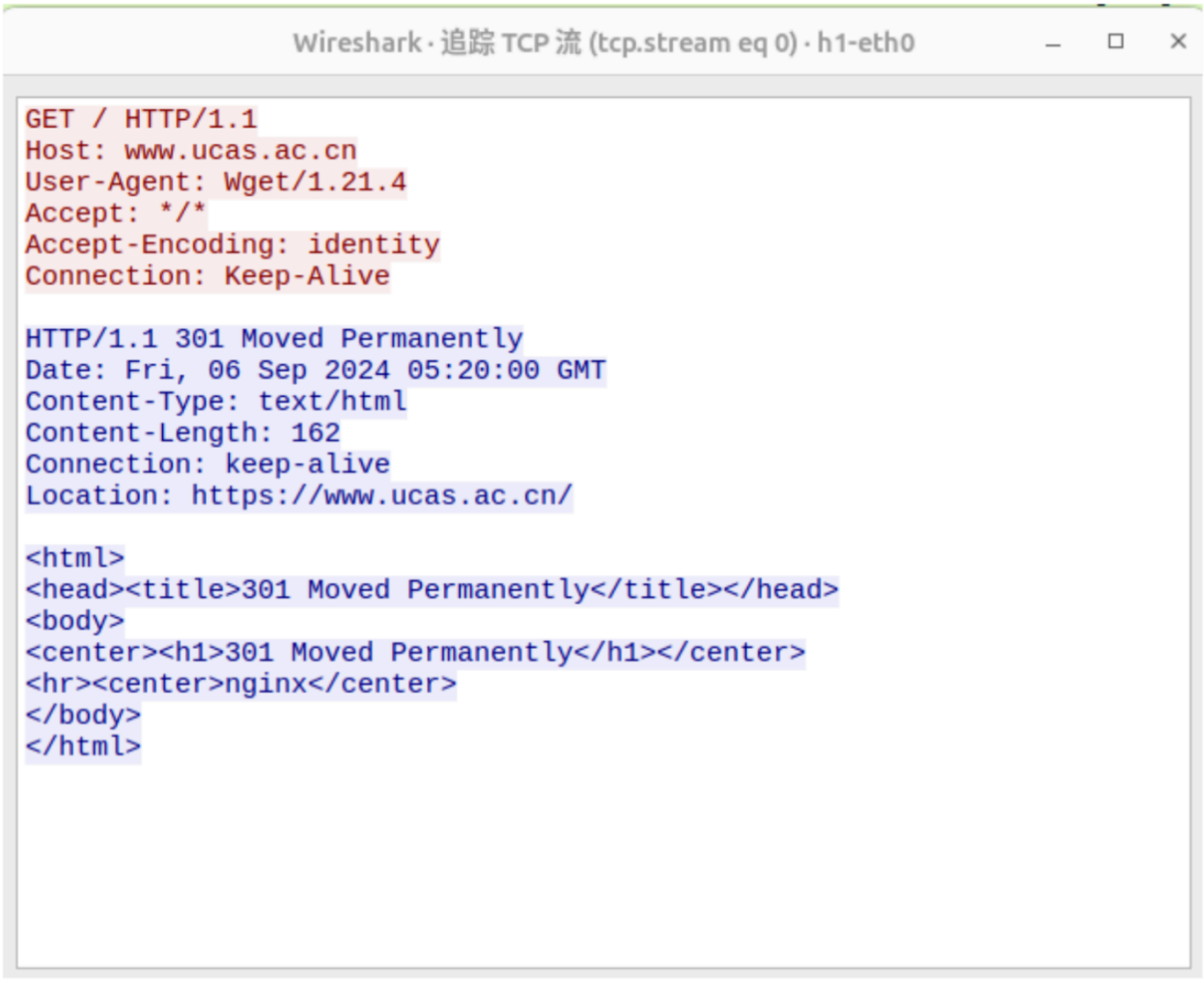

图 **6.** TCP 流

1.4 实验分析

在抓包过程中，我们可以看到获取目标网页的过程中，总共使用了四种协议，即 ARP、DNS、TCP 和 HTTP 协议。封装层次总结如下：

1. ARP 协议：Ethernet < ARP;
2. DNS 协议：Ethernet < IPv4 < UDP < DNS;
3. TCP 协议：Ethernet < IPv4 < TCP;
4. HTTP 协议：Ethernet < IPv4 < TCP < HTTP;

1.5 实验解释

ARP（地址解析协议，Address Resolution Protocol）是一种用于将网络层地址（如 IP 地址）解析为链路层地址（如 MAC 地址）的协议。它在局域网（LAN）中非常重要，尤其是在以太网中。当一台主机需要发送数据到另一台主机时，它首先需要知道目标主机的 MAC 地址。如果它不知道目标主机的 MAC 地址，它会广播一个 ARP 请求包到网络中，网络中拥有该 IP 地址的主机会回复一个 ARP 响应包，告知其 MAC 地址，随后主机会将 IP 地址和对应的 MAC 地址缓存起来，以便下次通信时不需要再次发送 ARP 请求。

DNS（域名系统，Domain Name System）协议用于将人类可读的域名解析为机器可读的 IP 地址。它是互联网的重要组成部分，允许用户通过域名访问网站，而不需要记住复杂的 IP 地址。当用户在浏览器中输入一个域名时，浏览器会首先检查本地缓存中是否有该域名的 IP 地址。如果没有，它会向本地 DNS 服务器发送一个 DNS 查询请求。本地 DNS 服务器会检查自己的缓存，如果没有找到对应的 IP 地址，它会向根 DNS 服务器发送查询请求。根 DNS 服务器会返回一个顶级域（如.com）的 DNS 服务器地址。如果还是找不到，本地 DNS 服务器会继续向顶级域 DNS 服务器发送查询请求，直到找到负责该域名的权威 DNS 服务器，并获取最终的 IP 地址，本地 DNS 服务器将 IP 地址返回给用户的浏览器，浏览器使用该 IP 地址与目标服务器建立连接。

TCP（传输控制协议，Transmission Control Protocol）是一种面向连接的、可靠的传输层协议。它提供了可靠的数据传输服务，确保数据包按顺序到达，并且没有丢失或重复。TCP 使用三次握手来建立连接：客户端发送一个 SYN（同步）包到服务器；服务器收到 SYN 包后，回复一个 SYN-ACK（同步-确认）包；客户端收到 SYN-ACK 包后，回复一个 ACK（确认）包，连接建立完成。在连接建立后，TCP 使用序列号和确认号来确保数据包按顺序到达，并且没有丢失或重复。每个数据包都有一个序列号，接收方会发送一个确认号来确认已收到的数据。TCP 使用四次挥手来终止连接：一方发送一个 FIN（终止）包，表示不再发送数据；另一方收到 FIN 包后，回复一个 ACK 包；另一方也发送一个 FIN 包，表示不再发送数据；最后一方收到 FIN 包后，回复一个 ACK 包，连接终止完成。

HTTP（超文本传输协议，HyperText Transfer Protocol）是一种用于在客户端和服务器之间传输超文本的应用层协议。它是万维网（WWW）的基础，定义了客户端（通常是浏览器）如何向服务器请求资源以及服务器如何响应这些请求。HTTP 是无状态协议，每个请求都是独立的，与之前的请求没有关联。服务器不会保留任何关于客户端的状态信息。HTTP 使用请求-响应模型。客户端发送一个 HTTP 请求，服务器处理请求并返回一个 HTTP 响应。请求和响应都包含头部信息和可选的主体内容。HTTP 响应包含一个状态码，用于表示请求的结果。

2 流完成时间实验

2.1 实验内容

在给定带宽、延迟和文件大小前提下，查看流完成时间，变化文件大小 (10MB, 100MB)、带宽 (10Mbps, 100Mbps, 1Gbps)、延迟 (100ms)，查看不同条件下的流完成时间。利用 fct_exp.py 脚本，重现 PPT 中的实验结果。

2.2 实验步骤

在 fct_exp.py 文件中设置好带宽和延迟参数值之后，依次输入以下命令：

```
sudo python3 fct_exp.py # 运行fct_exp.py脚本
xterm h1 h2 # 启动两个host
dd if=/dev/zero of=1MB.dat bs=1M count=1 # 在h2中生成1MB的文件
wget http://10.0.0.2/1MB.dat # 在h1中下载1MB的文件，记录时间和速度
```