Mininet 实验设计报告

中国科学院大学
页
2022 年 3 月 29 日

一、互联网协议实验

1. 实验内容

（1） 搭建 mininet 实验环境，安装 wireshark，
（2） 在节点 h1 上开启 wireshark 抓包，用 wget 下载 www. ucas.ac.com 页面。+
（3） 观察 wireshark 的输出，说明 wireshark 抓到包涉及的：ARP, DNS, TCP, HTTP 和 TLS 等协议的运行机制。

2. 实验流程

（1） 安装 wireshark，并搭建 mininet 实验环境
sudo apt install mininet
sudo apt install wireshark
h1 # echo "nameserver 1.2.4.8" > /etc/resolv.conf
（2） 启动 wireshark，在 xterm 中通过 wget www. ucas.ac.cn 实现抓包
（3） 观察 wireshark 了解相关信息

3. 实验结果

抓包得到的页面 html 页面如图 1 所示：

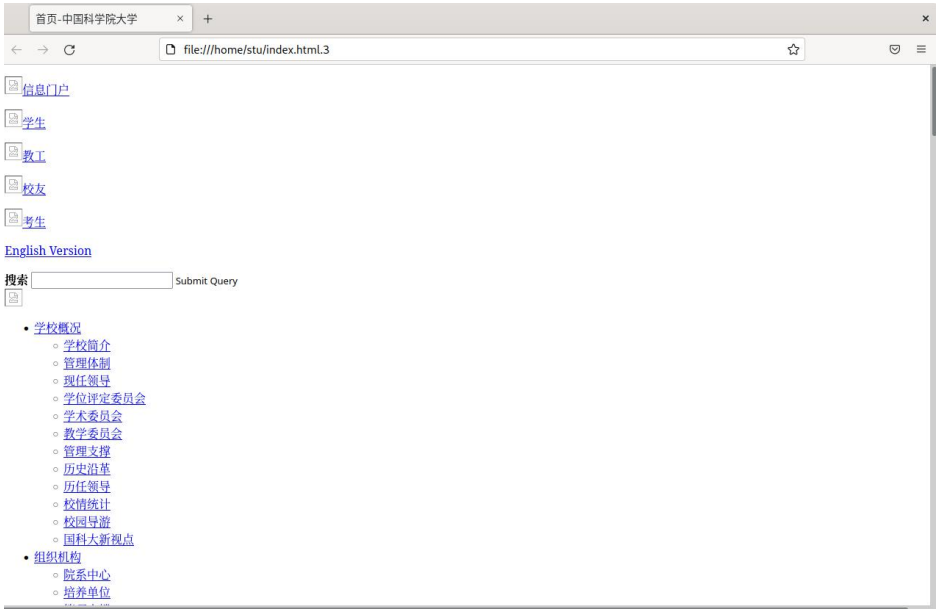
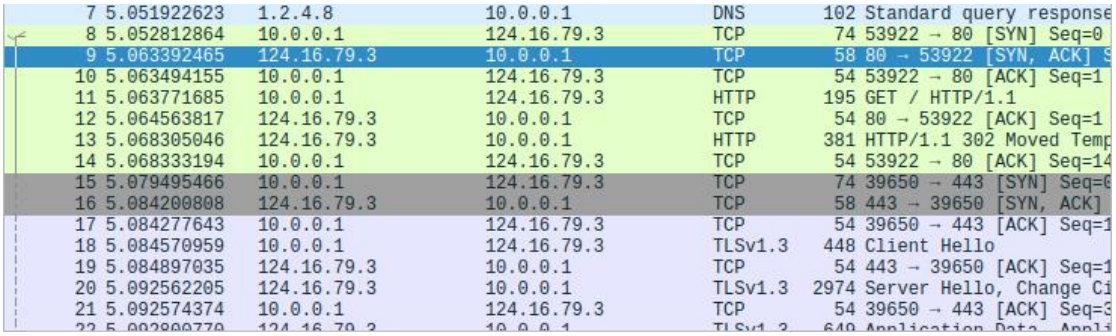


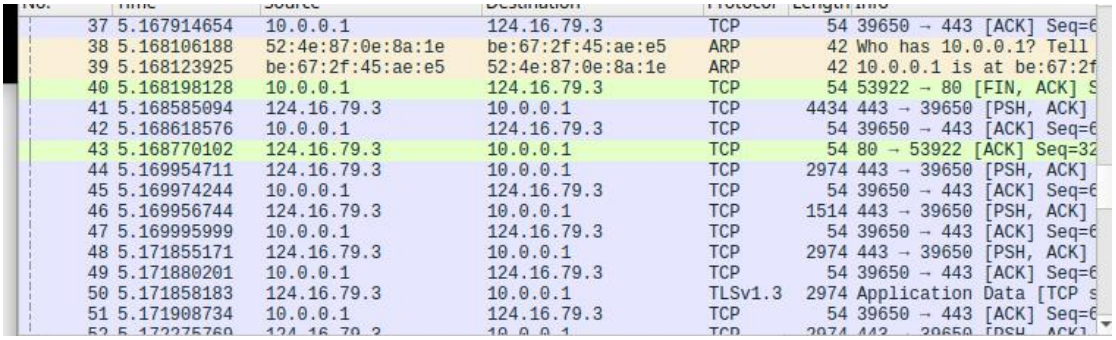
图 1：抓包获取的 ucas 页面

wireshark 抓取的网络包如图 2—4 所示：



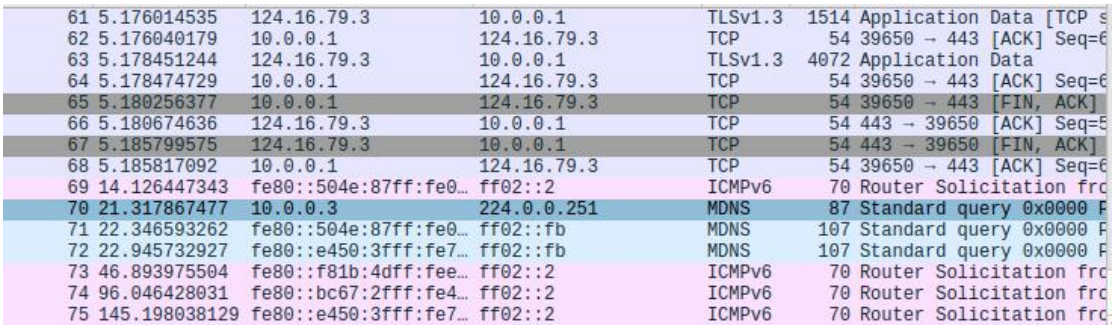
7	5.051922623	1.2.4.8	10.0.0.1	DNS	102 Standard query response
8	5.052812864	10.0.0.1	124.16.79.3	TCP	74 53922 → 80 [SYN] Seq=0
9	5.063392465	124.16.79.3	10.0.0.1	TCP	58 80 → 53922 [SYN, ACK] S
10	5.063494155	10.0.0.1	124.16.79.3	TCP	54 53922 → 80 [ACK] Seq=1
11	5.063771685	10.0.0.1	124.16.79.3	HTTP	195 GET / HTTP/1.1
12	5.064563817	124.16.79.3	10.0.0.1	TCP	54 80 → 53922 [ACK] Seq=1
13	5.068305046	124.16.79.3	10.0.0.1	HTTP	381 HTTP/1.1 302 Moved Temp
14	5.068333194	10.0.0.1	124.16.79.3	TCP	54 53922 → 80 [ACK] Seq=14
15	5.079495466	10.0.0.1	124.16.79.3	TCP	74 39650 → 443 [SYN] Seq=0
16	5.084200808	124.16.79.3	10.0.0.1	TCP	58 443 → 39650 [SYN, ACK]
17	5.084277643	10.0.0.1	124.16.79.3	TCP	54 39650 → 443 [ACK] Seq=1
18	5.084570959	10.0.0.1	124.16.79.3	TLSv1.3	448 Client Hello
19	5.084897035	124.16.79.3	10.0.0.1	TCP	54 443 → 39650 [ACK] Seq=1
20	5.092562205	124.16.79.3	10.0.0.1	TLSv1.3	2974 Server Hello, Change Ci
21	5.092574374	10.0.0.1	124.16.79.3	TCP	54 39650 → 443 [ACK] Seq=3

图 2



37	5.167914654	10.0.0.1	124.16.79.3	TCP	54 39650 → 443 [ACK] Seq=6
38	5.168106188	52:4e:87:0e:8a:1e	be:67:2f:45:ae:e5	ARP	42 Who has 10.0.0.1? Tell
39	5.168123925	be:67:2f:45:ae:e5	52:4e:87:0e:8a:1e	ARP	42 10.0.0.1 is at be:67:2f
40	5.168198128	10.0.0.1	124.16.79.3	TCP	54 53922 → 80 [FIN, ACK] S
41	5.168585094	124.16.79.3	10.0.0.1	TCP	4434 443 → 39650 [PSH, ACK]
42	5.168618576	10.0.0.1	124.16.79.3	TCP	54 39650 → 443 [ACK] Seq=6
43	5.168770102	124.16.79.3	10.0.0.1	TCP	54 80 → 53922 [ACK] Seq=32
44	5.169954711	124.16.79.3	10.0.0.1	TCP	2974 443 → 39650 [PSH, ACK]
45	5.169974244	10.0.0.1	124.16.79.3	TCP	54 39650 → 443 [ACK] Seq=6
46	5.169956744	124.16.79.3	10.0.0.1	TCP	1514 443 → 39650 [PSH, ACK]
47	5.169995999	10.0.0.1	124.16.79.3	TCP	54 39650 → 443 [ACK] Seq=6
48	5.171855171	124.16.79.3	10.0.0.1	TCP	2974 443 → 39650 [PSH, ACK]
49	5.171880201	10.0.0.1	124.16.79.3	TCP	54 39650 → 443 [ACK] Seq=6
50	5.171858183	124.16.79.3	10.0.0.1	TLSv1.3	2974 Application Data [TCP s
51	5.171908734	10.0.0.1	124.16.79.3	TCP	54 39650 → 443 [ACK] Seq=6

图 3



61	5.176014535	124.16.79.3	10.0.0.1	TLSv1.3	1514 Application Data [TCP s
62	5.176040179	10.0.0.1	124.16.79.3	TCP	54 39650 → 443 [ACK] Seq=6
63	5.178451244	124.16.79.3	10.0.0.1	TLSv1.3	4072 Application Data
64	5.178474729	10.0.0.1	124.16.79.3	TCP	54 39650 → 443 [ACK] Seq=6
65	5.180256377	10.0.0.1	124.16.79.3	TCP	54 39650 → 443 [FIN, ACK]
66	5.180674636	124.16.79.3	10.0.0.1	TCP	54 443 → 39650 [ACK] Seq=5
67	5.185799575	124.16.79.3	10.0.0.1	TCP	54 443 → 39650 [FIN, ACK]
68	5.185817092	10.0.0.1	124.16.79.3	TCP	54 39650 → 443 [ACK] Seq=6
69	14.126447343	fe80::504e:87ff:fe0…	ff02::2	ICMPv6	70 Router Solicitation fro
70	21.317867477	10.0.0.3	224.0.0.251	MDNS	87 Standard query 0x0000 P
71	22.346593262	fe80::504e:87ff:fe0…	ff02::fb	MDNS	107 Standard query 0x0000 P
72	22.945732927	fe80::e450:3fff:fe7…	ff02::fb	MDNS	107 Standard query 0x0000 P
73	46.893975504	fe80::f81b:4dff:fee…	ff02::2	ICMPv6	70 Router Solicitation fro
74	96.046428031	fe80::bc67:2fff:fe4…	ff02::2	ICMPv6	70 Router Solicitation fro
75	145.198038129	fe80::e450:3fff:fe7…	ff02::2	ICMPv6	70 Router Solicitation fro

图 4

抓包过程主要出现了整个过程主要依次出现了 DNS、TCP、HTTP 、ARP、TLSv1.3 和 ICMPv6 等多种协议

4. 分析和讨论

1）DNS 协议

DNS (Domain Name System 域名系统)协议是一种用来将域名转化为 IP 地址的网络服务，请求方发送待解析的主机名，服务方返回对应的 IP 地址。对应的协议属于应用层协议。用户主机上运行着 DNS 的客户端，浏览器将接收到的 url 中抽取出域名字段，就是访问的主机名，并将这个主机名传送给 DNS 应用的客户端，DNS 客户机端向 DNS 服务器端发送一份查询报文，报文中包含着要访问的主机名字段（中间包括一些列缓存查询以及分布式 DNS 集群的工作），该 DNS 客户机最终会收到一份回答报文，其中包含有该主机名对应的 IP 地址，一旦该浏览器收到来自 DNS 的 IP 地址，就可以向该 IP 地址定位的 HTTP 服务器发起 TCP 连接。在上图 2 中，由本地 IP 地址 10.0.0.1 作为 source ，向目的地址 1.2.4.8

（DNS 服务器 IP）发送查询的请求，然后 1.2.4.8 返回给本地 IP 其解析结果。从图 2 中可以得到解析的结果为 124.16.79.3。

2）TCP 协议

传输控制协议（TCP，Transmission Control Protocol）是一种面向连接的、可靠的、基于字节流的传输层通信协议。TCP 是因特网中的传输层协议，使用三次握手协议建立连接。当主动方发出 SYN 连接请求后，等待对方回答 SYN+ACK，并最终对对方的 SYN 执行 ACK 确认。这种建立连接的方法可以防止产生错误的连接，TCP 使用的流量控制协议是可变大小的滑动窗口协议。

TCP 三次握手的过程如下图 5：

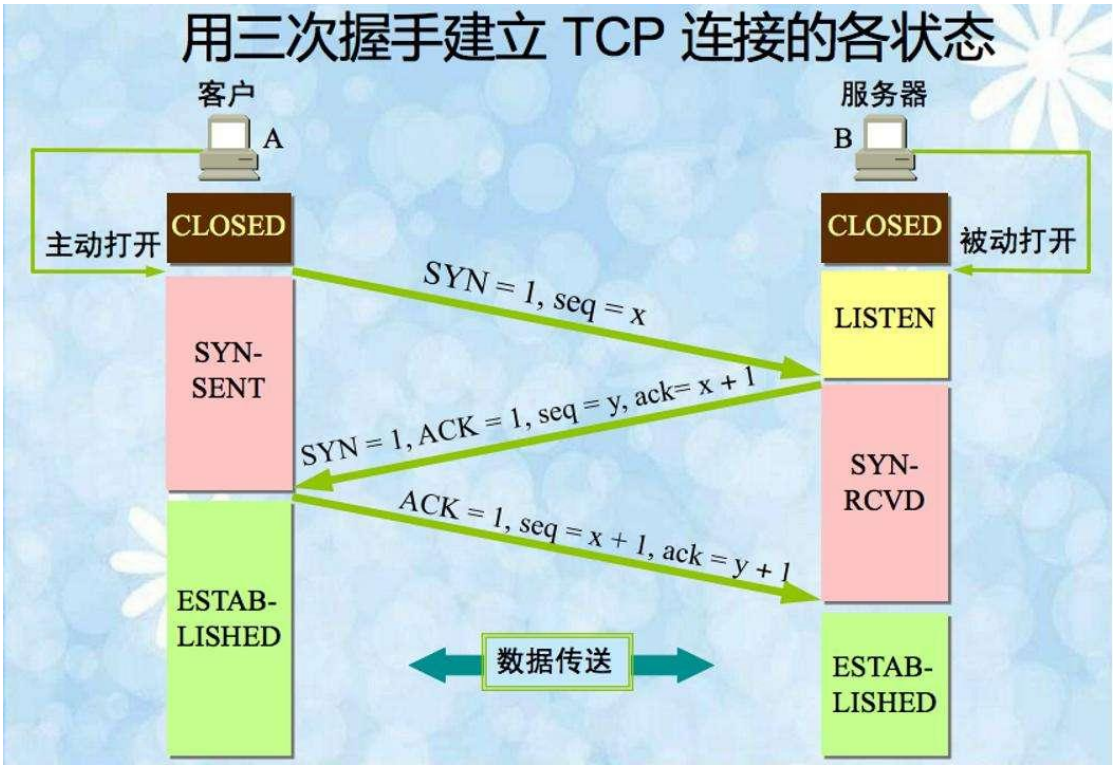


图５：TCP 协议３次握手过程

1.客户端发送 SYN（SEQ=x）报文给服务器端，进入 SYN_SEND 状态。

2.服务器端收到 SYN 报文，回应一个 SYN （SEQ=y）ACK（ACK=x+1）报文，进入 SYN_RECV 状态。

3.客户端收到服务器端的 SYN 报文，回应一个 ACK（ACK=y+1）报文，进入 Established 状态。

TCP 的释放和连接过程是相似的，终止一个连接要经过四次握手，这是由 TCP 的半关闭（half-close）造成的。

（1） 某个应用进程首先调用 close，称该端执行“主动关闭”（active close）。该端的 TCP 于是发送一个 FIN 分节，表示数据发送完毕。

（2） 接收到这个 FIN 的对端执行 “被动关闭”（passive close），这个 FIN 由 TCP 确认。

（3） 一段时间后，接收到这个文件结束符的应用进程将调用 close 关闭它的套接字。这导致它的 TCP 也发送一个 FIN。

（4） 接收这个最终 FIN 的原发送端 TCP（即执行主动关闭的那一端）确认这个 FIN

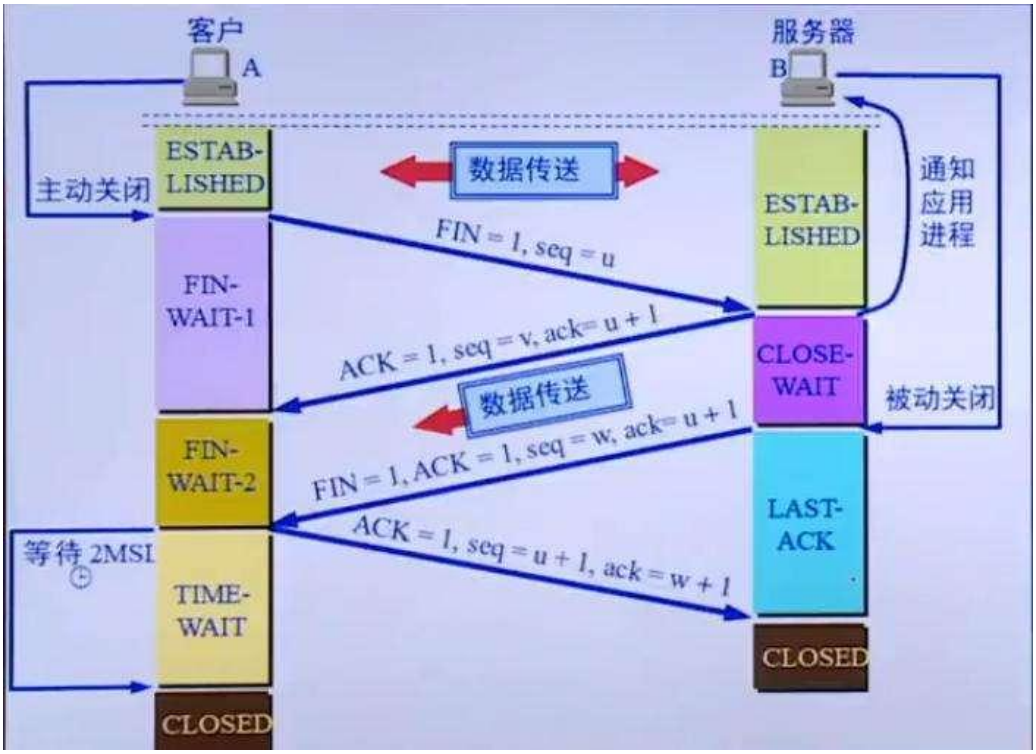


图 6：TCP 的 4 次挥手过程

TCP 是一种面向广域网的通信协议，目的是在跨越多个网络通信时，为两个通信端点之间提供一条具有下列特点的通信方式：

（1）基于流的方式；。TCP 中的“流”(stream)指的是流入到进程或从进程流出的字节序列。“面向字节流”的含义是：虽然应用程序和 TCP 的交互是一次一个数据块（大小不等），但 TCP 把应用程序交下来的数据仅仅看成是一连串的无结构的字节流。TCP 并不知道所传送的字节流的含义。

（2）面向连接；应用程序在使用 TCP 协议之前，必须先建立 TCP 连接。在传送数据完毕后，必须释放已经建立的 TCP 连接。

（3）可靠通信方式；通过 TCP 连接传送的数据，无差错、不丢失、不重复，
并且按序到达。

（4）在网络状况不佳的时候尽量降低系统由于重传带来的带宽开销；

（5）通信连接维护是面向通信的两个端点的，而不考虑中间网段和节点；每一条 TCP 连接只能有两个端点(endpoint)，每一条 TCP 连接只能是点对点的。

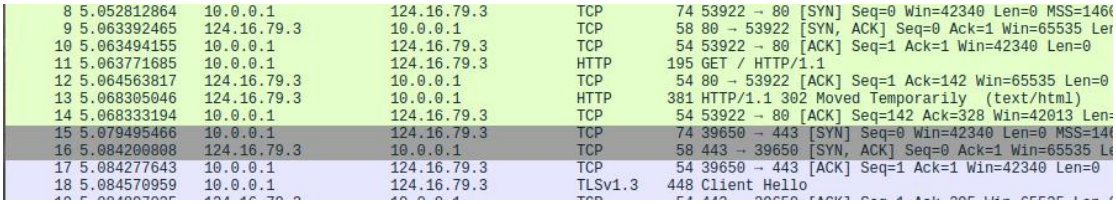


图 7：实验中的握手和挥手过程

本次实验中的 3 握手过程如图 6 的 No.8-10 所示，4 次挥手如 No.14-17 所示

3）HTTP 协议

HTTP（HyperText Transfer Protocol 超文本传输协议）是一个简单的请求-响应协议，它通常运行在 TCP 之上。它指定了客户端可能发送给服务器什么样的消息以及得到什么样的

响应。基于 HTTP 的客户/服务器模式的信息交换过程分为四个过程：建立连接、发送请求信息、发送响应信息、关闭连接。HTTP 有两类报文如图 8 所示：(1) 请求报文—从客户向服务器发送请求报文。(2) 响应报文—从服务器到客户的回答。

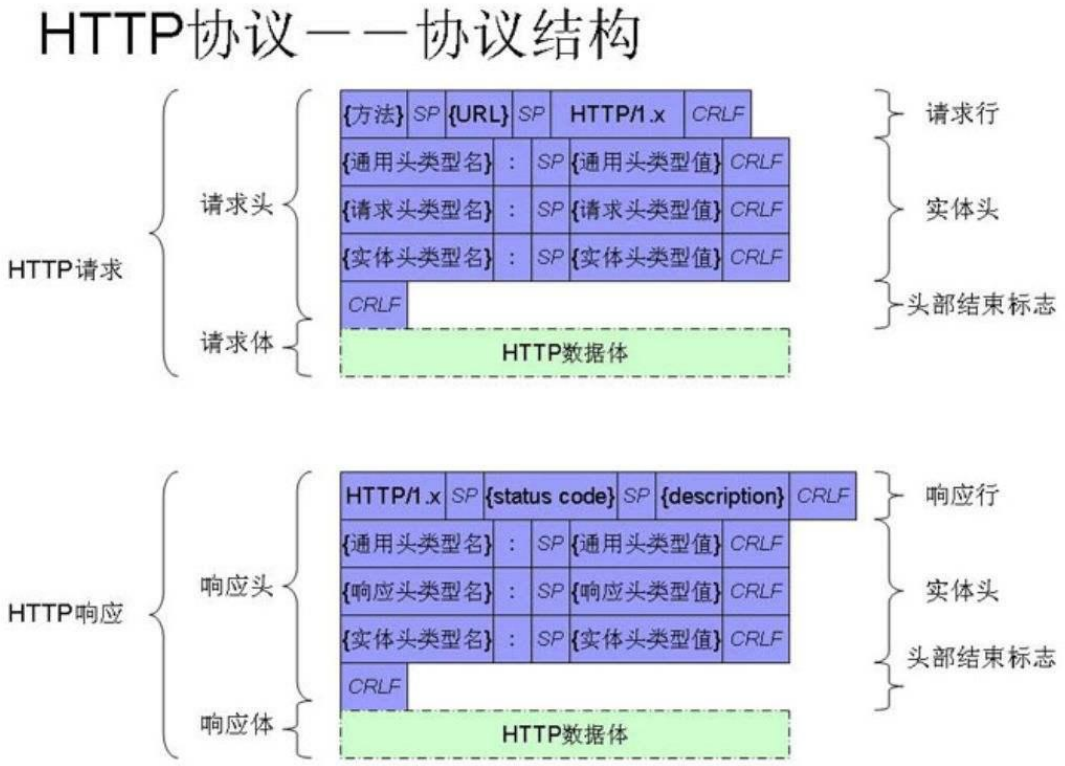


图 8：HTTP 的报文结构

HTTP/1.1 协议中共定义了八种方法（也叫“动作”）来以不同方式操作指定的资源，因为我们进行了 wget 操作，所以如图 9 中出现了 GET 请求，从指定资源请求数据。

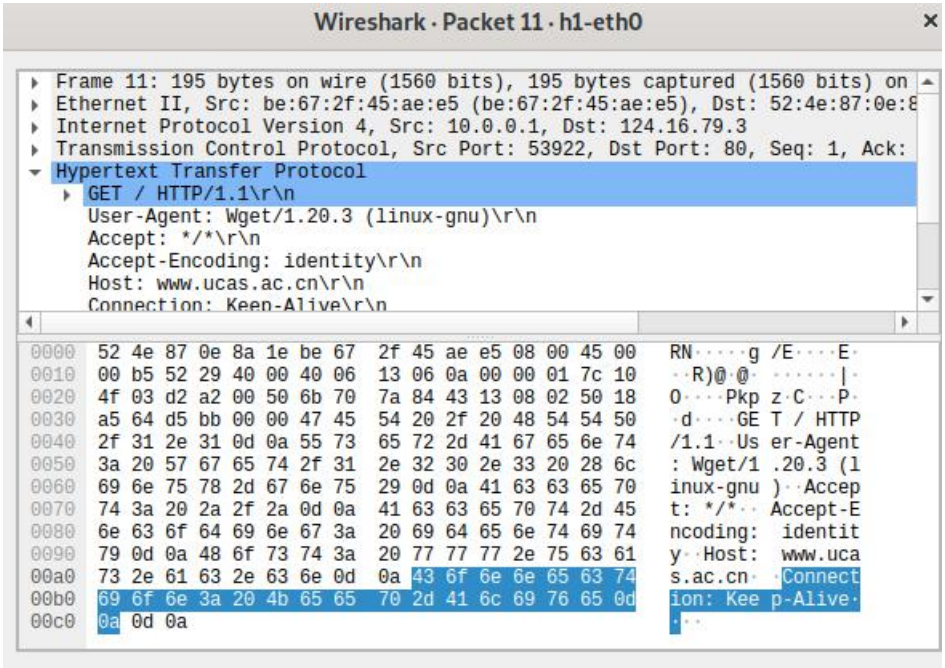


图 9：HTTP 的 GET 操作

4）ARP 协议

图 10：ARP 协议工作层次

ARP(Address Resolution Protocol 地址解析协议)是根据 IP 地址获取物理地址的一个 TCP/IP 协议。主机发送信息时将包含目标 IP 地址的 ARP 请求广播到局域网络上的所有主机，并接收返回消息，以此确定目标的物理地址。

一台主机有 IP 数据报文发送给另一台主机，它都要知道接收方的逻辑（IP）地址。但是 IP 地址必须封装成帧才能通过物理网络。这就意味着发送方必须有接收方的物理（MAC）地址，因此需要完成逻辑地址到物理地址的映射。而 ARP 协议可以接收来自 IP 协议的逻辑地址，将其映射为相应的物理地址，然后把物理地址递交给数据链路层，当主机需要找出这个网络中的另一个主机的物理地址时，它就可以发送一个 ARP 请求报文，这个报文包好了发送方的 MAC 地址和 IP 地址以及接收方的 IP 地址。因为发送方不知道接收方的物理地址，所以这个查询分组会在网络层中进行广播。局域网中的每一台主机都会接受并处理这个 ARP 请求报文，然后进行验证，查看接收方的 IP 地址是不是自己的地址，只有验证成功的主机才会返回一个 ARP 响应报文，这个响应报文包含接收方的 IP 地址和物理地址。这个报文利用收到的 ARP 请求报文中的请求方物理地址以单播的方式直接发送给 ARP 请求报文的请求方。

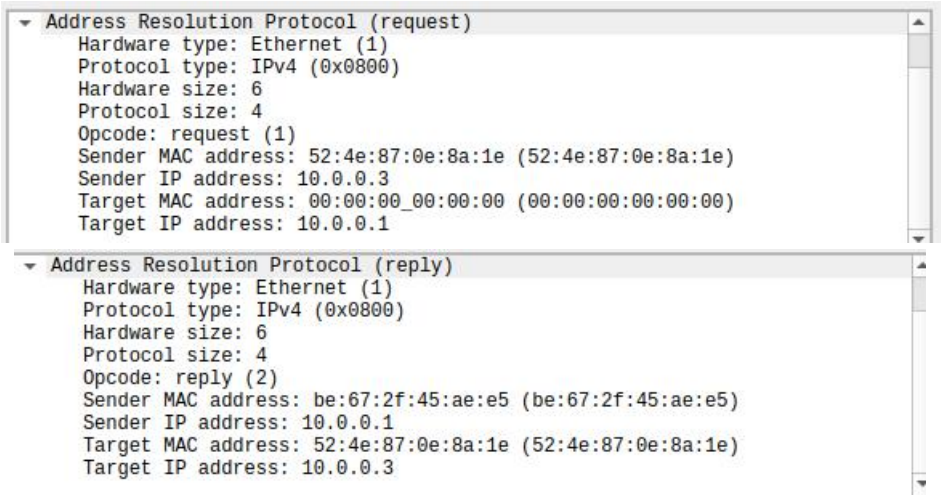


图 11：ARP 解析地址

在本实验中，发送方 的 IP 地址为 10.0.0.1，接收方不在同一局域网中，于是发送方通过 ARP 协议寻找一个路由器 10.0.0.1 的物理地址，然后 10.0.0.1 返回自己的 MAC 地址 be:67:2f:45:ae:e5。

5）整体流程：

1.浏览器向 DNS 服务器请求解析该 URL 中的域名所对应的 IP 地址;

2.解析出 IP 地址后，根据该 IP 地址和默认端口 80，和服务器建立 TCP 连接;

3. 浏览器发出读取文件(URL 中域名后面部分对应的文件)的 HTTP 请求，该请求报文作为 TCP 三次握手的第三个报文的数据发送给服务器;
4. 服务器对浏览器请求作出响应，并把对应的 html 文本发送给浏览器;
5. 释放 TCP 连接;
6. 浏览器将该 html 文本并显示内容。

二、流完成时间实验

1. 实验内容

利用 fct_exp.py 脚本绘制流完成时间图，分析传输不同大小的数据包所需要的时间与带宽之间的关系

2. 实验流程

（1）修改 fct_exp.py 脚本，将提供的测试语句包装成测试函数 test，修改数据包大小、延迟和带宽等参数，在脚本中更改参数进行测试。

文件大小: 10MB, 100MB

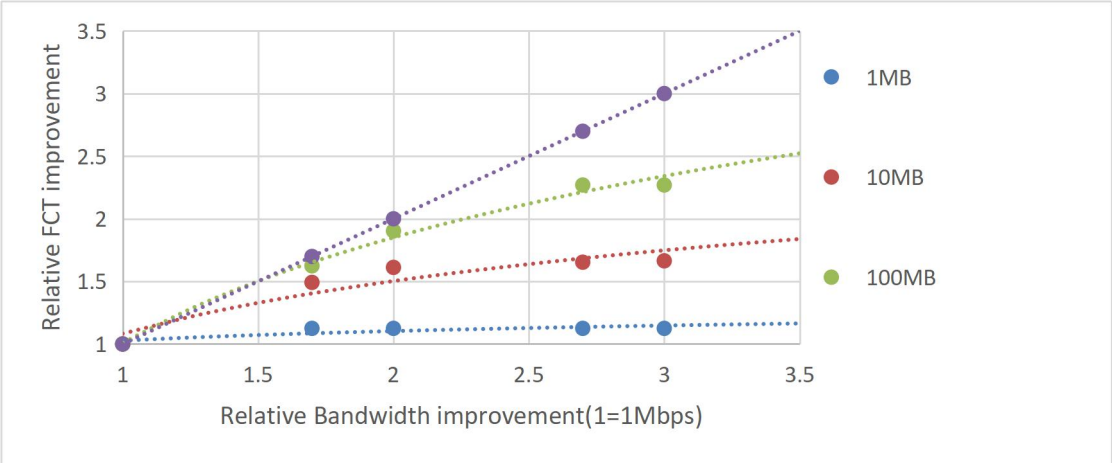
带宽：10Mbps, 50Mbps, 100Mbps, 1Gbps

延迟：10ms，100ms

运行命令： sudo python2 fct_exp.py

（2）根据结果绘图分析，每个数据点做 5 次实验，取均值。

3. 实验结果



4. 分析和讨论

带宽较小时，拥塞窗口很容易就到达慢开始门限，因此平均传输速率不大，FCT 提升有限;当带宽增大到一定大小时，再提升带宽，FCT 提升稳定趋于平缓。

TCP 协议利用慢启动和拥塞避免机制来避免传输拥塞，慢启动算法在主机刚开始发送数据报的时先探测网络的状况。如果网络状况良好，发送方每发送一次文段都能正确的接受确认报文段，发送窗口的大小会在需求大时增加。文件传输分为建立连接和文件传输俩步，

数据传输开始的时候，网络使用慢启动算法找到合适的发送窗口大小需要一定时间，这段时间无法被忽略，因此带宽在增长到一定程度后对 FCT 的影响就变得十分有限，这也是 FCT 提升趋于平缓的原因。