# 中国科学院大学
# 《计算机网络(研讨课)》实验报告

姓名 姚永舟 学号 2022K8009926016 专业 计算机科学与技术

实验项目编号 2 实验名称 互联网协议实验、流完成时间实验

# 实验一:互联网协议实验

## 一、 实验内容

1. 在节点 h1 上开启 wireshark 抓包,用 wget 下载 www.ucas.ac.cn 页面
2. 调研说明 wireshark 抓到的几种协议 ARP, DNS, TCP, HTTP, HTTPS
3. 调研解释 h1 下载 ucas 页面的整个过程, 几种协议的运行机制

## 二、 实验流程

1. 在终端执行 sudo mn –nat,将 host 连接至 internet,并启动 mininet
2. 在 mininet 中输入 xterm h1,打开控制 h1 的终端
3. 在 h1 中输入 echo "nameserver 1.2.4.8" > /etc/resolv.conf,设置 DNS 服务器
4. 在 h1 中输入 wireshark & ,打开 wireshark
5. 在 wireshark 中选择 h1-eth0,开始抓包
6. 在 h1 中输入 wget www.ucas.ac.cn,下载网页
7. 观察 wireshark 抓包结果,并调研分析获得的几种互联网协议

## 三、 实验结果

**ARP 协议**

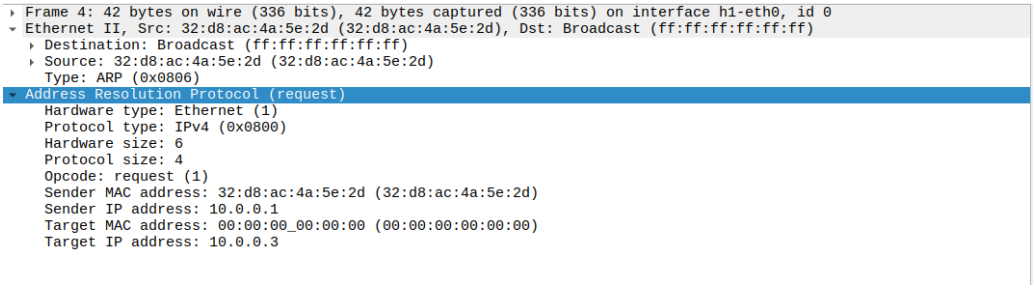


图 1: ARP 协议

### DNS 协议

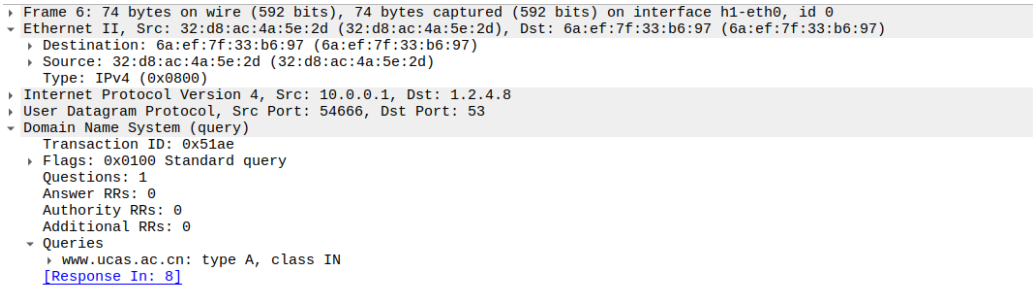


图 2: DNS 协议

### TCP 协议

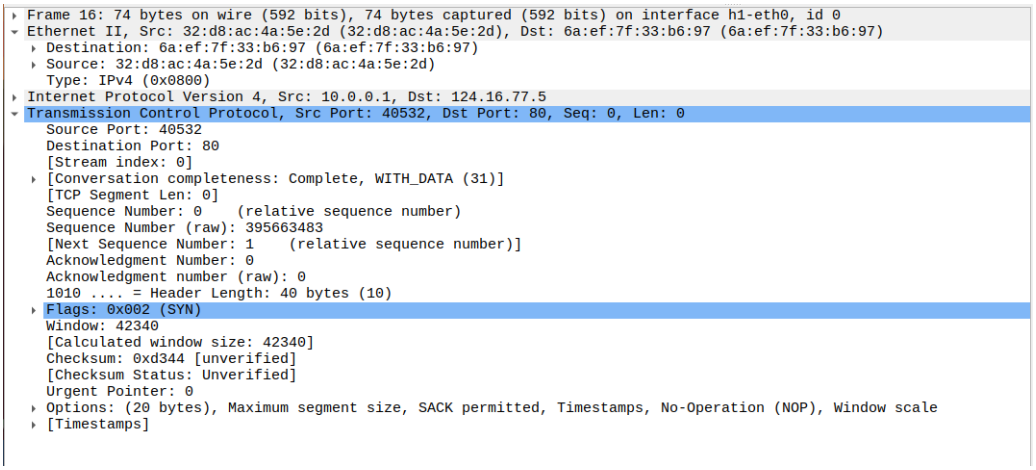


图 3: TCP 协议

### HTTP 协议

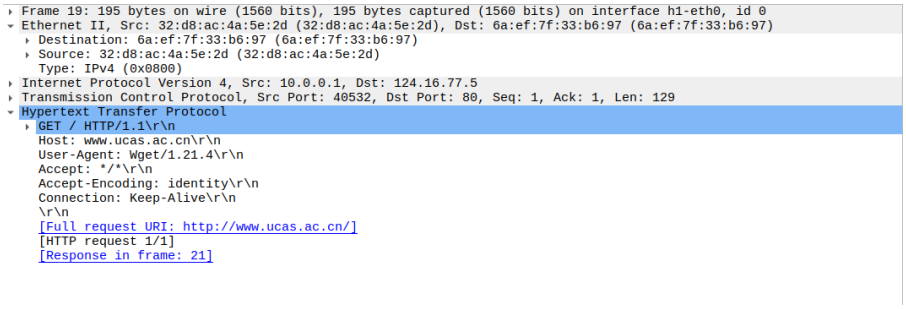


图 4: HTTP 协议

**TCP flow**

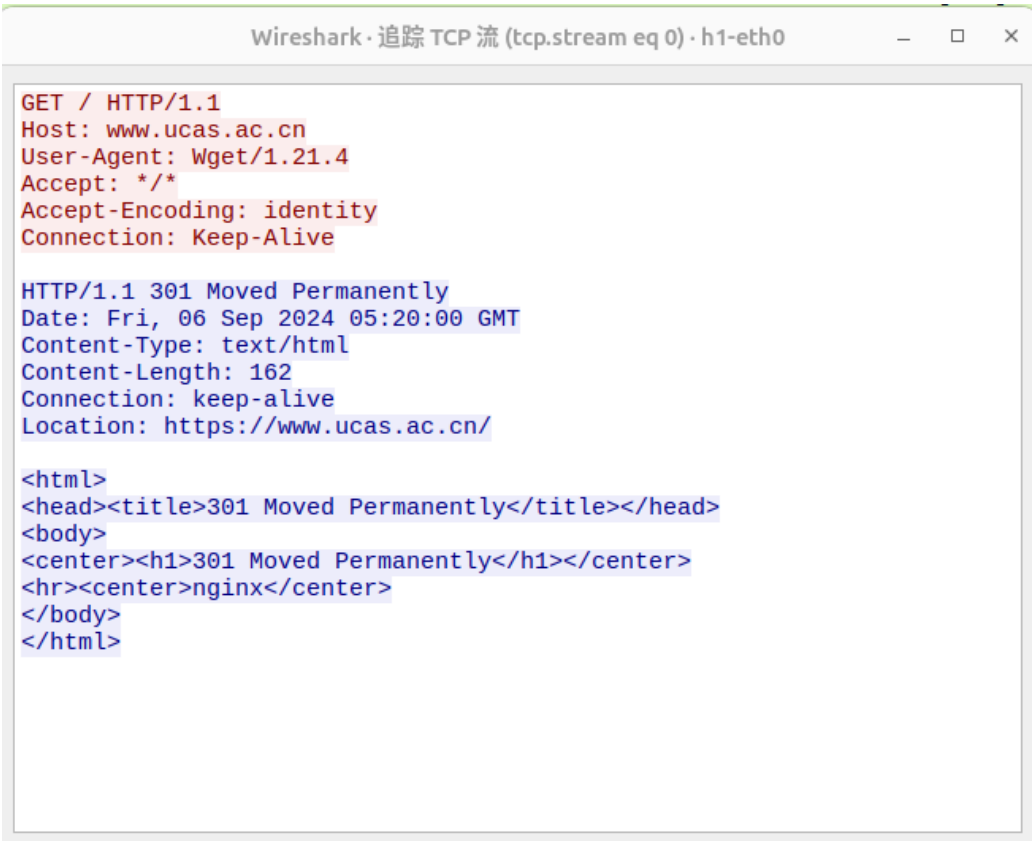


图 5: TCP flow

# 四、 实验分析

在获取 UCAS 主页的传输过程中使用了以下几种协议：ARP 协议、DNS 协议、TCP 协议、HTTP 协议。并且各个协议的封装层次如下:

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

# 五、 调研结果

**1. ARP（地址解析）协议**

ARP 协议“Address Resolution Protocol”(地址解析协议) 的缩写。其作用是在以太网环境中, 数据的传输所依懒的是 MAC 地址而非 IP 地址, 而将已知 IP 地址转换为 MAC 地址的工作是由 ARP 协议来完成的。

在局域网中, 网络中实际传输的是“帧”, 帧里面是有目标主机的 MAC 地址的。在以太网中, 一个主机和另一个主机进行直接通信, 必须要知道目标主机的 MAC 地址。目标 MAC 地址是通过地址解析协议获得的。所谓“地址解析”就是主机在发送帧前将目标 IP 地址转换成目标 MAC 地址的过程。ARP 协议的基本功能就是通过目标设备的 IP 地址, 查询目标设备的 MAC 地址, 以保证通信的顺利进行。ARP 通过发送一个 ARP 请求帧到

局域网中的所有设备来查找目标设备的 MAC 地址。这个请求包含源设备的 IP 地址和 MAC 地址。目标设备收到请求后，会回复一个包含其 IP 地址和 MAC 地址的 ARP 响应帧。

**2. DNS(域名系统) 协议**

DNS (Domain Name System) 是一个应用层协议，域名系统 (DNS) 的作用是将人类可读的域名 (如 www.example.com) 转换为机器可读的 IP 地址 (如 192.0.2.44)。DNS 系统使用树状层次结构，包括多个 DNS 服务器，它们负责不同的域名解析。当用户输入一个域名时，客户端的 DNS 解析器将向根 DNS 服务器发送查询，然后逐级查询更低级别的 DNS 服务器，直到找到与域名相关的 IP 地址。

DNS 协议建立在 UDP 或 TCP 协议之上, 默认使用 53 号端口。客户端默认通过 UDP 协议进行通讯, 但是由于广域网中不适合传输过大的 UDP 数据包, 因此规定当报文长度超过了 512 字节时, 应转换为使用 TCP 协议进行数据传输。DNS 是一种可以将域名和 IP 地址相互映射的以层次结构分布的数据库系统。

**3. TCP(传输控制)协议**

TCP (Transmission Control Protocol 传输控制协议) 是一种面向连接的、可靠的、基于字节流的传输层通信协议, 由 IETF 的 RFC 793 定义。在简化的计算机网络 OSI 模型中, 它完成第四层传输层所指定的功能。

应用层向 TCP 层发送用于网间传输的、用 8 位字节表示的数据流, 然后 TCP 把数据流分区成适当长度的报文段 (通常受该计算机连接的网络的数据链路层的最大传输单元 (MTU) 的限制) 。之后 TCP 把结果包传给 IP 层, 由它来通过网络将包传送给接收端实体的 TCP 层。TCP 将用户数据打包构成报文段, 它发送数据时启动一个定时器, 另一端收到数据进行确认, 对失序的数据重新排序, 丢弃重复的数据。简单说, TCP 协议的作用是, 保证数据通信的完整性和可靠性，防止丢包。

**4. HTTP 协议**

HTTP 协议 (超文本传输协议 HyperText Transfer Protocol), 它是基于 TCP 协议的应用层传输协议, 用于从 WWW 服务器传输超文本到本地浏览器的传输协议, HTTP 是一个应用层协议, 由请求和响应构成, 是一个标准的客户端和服务器模型, 简单来说就是客户端和服务端进行数据传输的一种规则。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。客户端发送 HTTP 请求到服务器，请求特定资源（如网页或图像）。服务器收到请求后，会发送 HTTP 响应，包含请求的资源以及相关信息。HTTP 通信通常是无状态的，每个请求和响应都独立于之前的请求和响应。

**5. h1 下载 ucas 页面的整个过程以及几种协议的运行机制**

1. DNS 解析域名 www.ucas.ac.cn，获取 IP 地址

2. 三次握手建立 TCP 连接

第一次握手: 客户端发送 SYN 包至服务器，并进入 SYN_SENT 状态，等待服务器确认

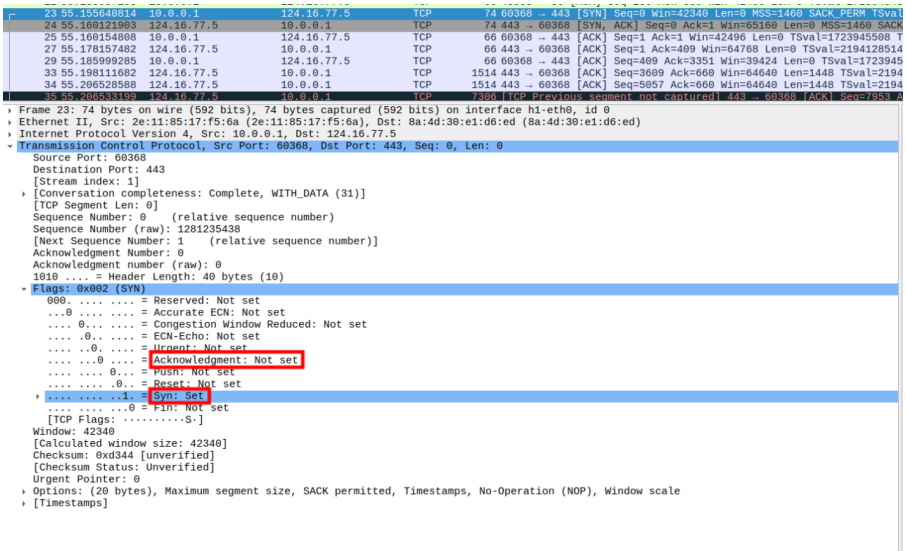


图 6: 第一次握手

第二次握手: 服务器收到客户端的 SYN 包，发送一个 ACK，同时发送自己的 SYN，此时服务器进入 SYN_RCVD 状态

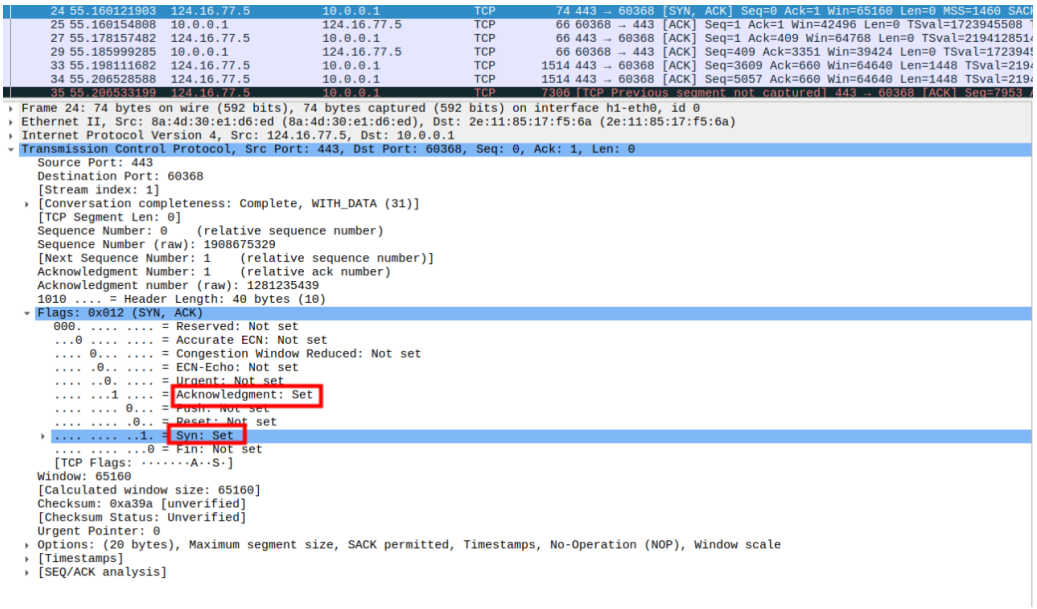


图 7: 第二次握手

第三次握手: 客户端接收到服务器发送的 SYN+ACK 后，进入 ESTABLISHED 状态，并发送服务器 SYN 包的确认 ACK，服务器接收到客户端 ACK 后，进入 ESTABLISHED 状态

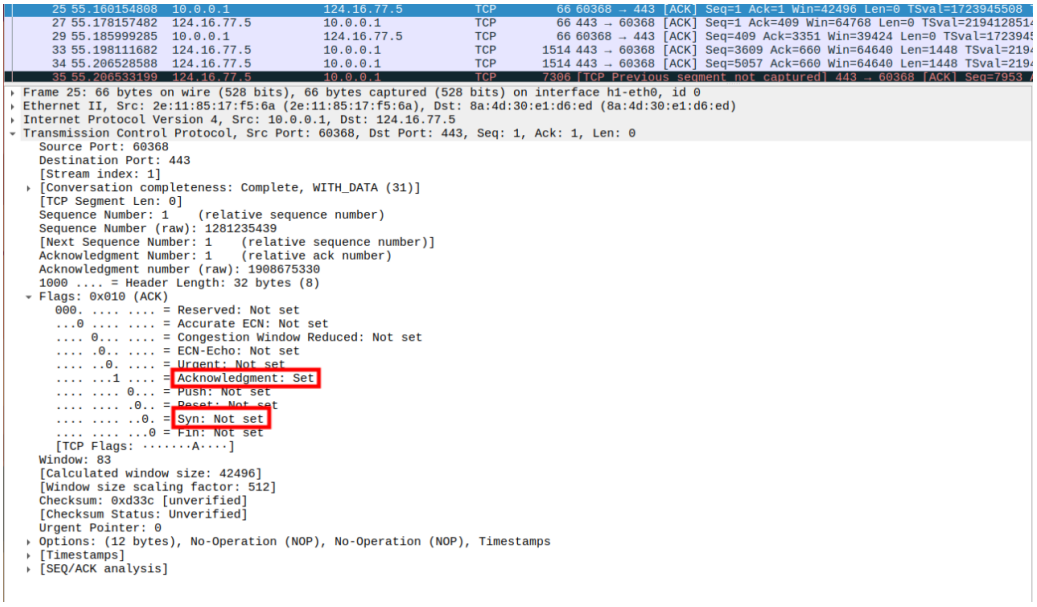


图 8: 第三次握手

经过上述的三次握手，TCP 连接正式建立，双方都置为 ACK flag, 交换并确认了对方的初始序列号。

3. 发送和收取数据 (本地主机和目的主机开始 HTTP 访问)

1）本地主机向域名发出 GET 方法报文 (HTTP 请求);

2）该 GET 方法报文通过 TCP->IP(DNS) ->MAC (ARP)-> 网关-> 目的主机;

3）目的主机收到数据帧，通过 IP->TCP->HTTP，HTTP 协议单元会回应 HTTP 协议格式封装好的 HTML 形式数据 (HTTP 响应)；从请求信息中获得客户机想访问的主机名。从请求信息中获取客户机想要访问的 web

资源，用读取到的 web 资源数据，创建一个 HTTP 响应。

4）该 HTML 数据通过 TCP->IP(DNS) ->MAC(ARP)-> 网关-> 我的主机;

5）我的主机收到数据帧，通过 IP->TCP->HTTP-> 浏览器，浏览器以网页形式显示 HTML 内容。

4. 四次挥手断开 TCP 连接

## 六、 实验总结

本次实验主要是简单的动手操作，通过抓包工具 wireshark 抓取 h1 下载网页的过程，分析了 ARP、DNS、TCP、HTTP 等协议的运行机制。实验本身难度不大，但涉及到各种协议，需要花时间调研。通过本次实验，我对互联网协议有了更深入的了解，对网络通信的过程有了更清晰的认识。

# 实验二：流完成时间实验

## 一、 实验内容

1. 利用 fct_exp.py 脚本复现幻灯片中的图
2. 调研解释图中的现象 (TCP 传输、慢启动机制等)

## 二、 实验流程

在终端中输入在终端中输入中启动、两个在终端输入其中分别设置为 1，10，进行不同大小的实验在终端中输入获取主机上对应大小的文件记录每次完成传输的时间和速度，每个数据点做五次实验，取均值根据结果绘图，复现讲义上的图

1. 在 fct_exp.py 文件中设置好带宽和延迟参数值
2. 在终端中输入 sudo python3 fct_exp.py
3. 在终端中输入 xterm h1 h2 启动两个 host
4. 在 h2 终端中输入 dd if=/dev/zero of=1MB.dat bs=1M count=1，其中 bs 可以分别设置为 1,10,100，进行不同大小的实验
5. 在 h1 终端中输入 wget http://10.0.0.2/1MB.dat 获取主机 h2 上对应大小的文件
6. 记录每次完成传输的时间和速度，每个数据点做五次实验，取均值
7. 根据结果绘图，复现讲义上的图

## 三、 实验流程

1. 对于查看不同条件下的流完成时间而言，首先在 fctexp.py 文件中设置好带宽和延迟参数值，然后对于不同文件大小，每个数据点重复测试 5 次，取平均值并记录
2. 依然是根据给定的参数值，在每个数据点重复测试 5 次，取均值并绘图。

## 四、 实验数据

**10ms 延迟下,带宽为 10Mbps 下, 不同文件大小的流完成时间**

1. 文件大小为 10MB

表 1: 10MB, 10ms, 10Mbps

| 测试次数 | 流完成时间 (单位 s) | 传输速率 (单位 MB/s) |
|---|---|---|
| 1 | 8.8 | 1.14 |
| 2 | 8.8 | 1.14 |
| 3 | 8.8 | 1.14 |
| 4 | 8.8 | 1.14 |
| 5 | 8.8 | 1.14 |
| 均值 | 8.8 | 1.14 |

2. 文件大小为 100MB

表 2: 100MB, 10ms, 10Mbps

| 序号 | 流完成时间 (单位 s) | 传输速率 (单位 MB/s) |
|---|---|---|
| 1 | 88 | 1.14 |
| 2 | 88 | 1.14 |
| 3 | 88 | 1.14 |
| 4 | 88 | 1.14 |
| 5 | 88 | 1.14 |
| 均值 | 88 | 1.14 |

**10ms 延迟下,带宽为 100Mbps 下, 不同文件大小的流完成时间**

1. 文件大小为 10MB

表 3: 10MB, 10ms, 100Mbps

| 序号 | 流完成时间 (单位 s) | 传输速率 (单位 MB/s) |
|---|---|---|
| 1 | 0.9 | 10.6 |
| 2 | 0.9 | 10.6 |
| 3 | 0.9 | 10.6 |
| 4 | 0.9 | 10.6 |
| 5 | 0.9 | 10.6 |
| 均值 | 0.9 | 10.6 |

2. 文件大小为 100MB

表 4: 100MB, 10ms, 100Mbps

| 序号 | 流完成时间 (单位 s) | 传输速率 (单位 MB/s) |
|---|---|---|
| 1 | 8.8 | 11.3 |
| 2 | 8.8 | 11.3 |
| 3 | 8.8 | 11.3 |
| 4 | 8.8 | 11.3 |
| 5 | 8.8 | 11.3 |
| 均值 | 8.8 | 11.3 |

**10ms 延迟下,带宽为 1Gbps 下, 不同文件大小的流完成时间**

1. 文件大小为 10MB

表 5: 10MB, 10ms, 1Gbps

| 序号 | 流完成时间 (单位 s) | 传输速率 (单位 MB/s) |
|---|---|---|
| 1 | 0.2 | 46 |
| 2 | 0.2 | 45.9 |
| 3 | 0.2 | 46.1 |
| 4 | 0.2 | 46 |
| 5 | 0.2 | 46 |
| 均值 | 0.2 | 46 |

2. 文件大小为 100MB

表 6: 100MB, 10ms, 1Gbps

| 序号 | 流完成时间 (单位 s) | 传输速率 (单位 MB/s) |
|---|---|---|
| 1 | 1 | 98.7 |
| 2 | 1 | 98.9 |
| 3 | 1 | 99.3 |
| 4 | 1 | 99.2 |
| 5 | 1 | 99.3 |
| 均值 | 1 | 99.08 |

**100ms 延迟下,带宽为 10Mbps 下, 不同文件大小的流完成时间**

1. 文件大小为 10MB

表 7: 10MB, 100ms, 10Mbps

| 序号 | 流完成时间 (单位 s) | 传输速率 (单位 MB/s) |
|---|---|---|
| 1 | 9.4 | 1.14 |
| 2 | 9.4 | 1.14 |
| 3 | 9.4 | 1.14 |
| 4 | 9.4 | 1.14 |
| 5 | 9.4 | 1.14 |
| 均值 | 9.4 | 1.14 |

2. 文件大小为 100MB

表 8: 100MB, 100ms, 10Mbps

| 序号 | 流完成时间 (单位 s) | 传输速率 (单位 MB/s) |
|---|---|---|
| 1 | 88 | 1.14 |
| 2 | 88 | 1.14 |
| 3 | 88 | 1.14 |
| 4 | 88 | 1.14 |
| 5 | 88 | 1.14 |
| 均值 | 88 | 1.14 |

**100ms 延迟下,带宽为 100Mbps 下, 不同文件大小的流完成时间**

1. 文件大小为 10MB

表 9: 10MB, 100ms, 100Mbps

| 序号 | 流完成时间 (单位 s) | 传输速率 (单位 MB/s) |
|---|---|---|
| 1 | 2.3 | 4.28 |
| 2 | 2.3 | 4.28 |
| 3 | 2.3 | 4.28 |
| 4 | 2.3 | 4.28 |
| 5 | 2.3 | 4.28 |
| 均值 | 2.3 | 4.28 |

2. 文件大小为 100MB

表 10: 100MB, 100ms, 100Mbps

| 序号 | 流完成时间 (单位 s) | 传输速率 (单位 MB/s) |
|---|---|---|
| 1 | 10 | 11.4 |
| 2 | 10 | 11.4 |
| 3 | 10 | 11.4 |
| 4 | 10 | 11.4 |
| 5 | 10 | 11.4 |
| 均值 | 10 | 11.4 |

**100ms 延迟下,带宽为 1Gbps 下, 不同文件大小的流完成时间**

1. 文件大小为 10MB

表 11: 10MB, 100ms, 1Gbps

| 序号 | 流完成时间 (单位 s) | 传输速率 (单位 MB/s) |
|---|---|---|
| 1 | 1.8 | 5.47 |
| 2 | 2 | 4.99 |
| 3 | 1.8 | 5.45 |
| 4 | 1.8 | 5.47 |
| 5 | 1.8 | 5.47 |
| 均值 | 1.84 | 5.37 |

2. 文件大小为 100MB

表 12: 100MB, 100ms, 1Gbps

| 序号 | 流完成时间 (单位 s) | 传输速率 (单位 MB/s) |
|---|---|---|
| 1 | 3.5 | 28.5 |
| 2 | 3.5 | 28.5 |
| 3 | 3.5 | 28.5 |
| 4 | 3.5 | 28.5 |
| 5 | 3.5 | 28.5 |
| 均值 | 3.5 | 28.5 |

**下面是复现讲义图而做的不同带宽和数据包大小的实验,其中流完成时间单位为 s, 传输速率单位为 MB/s**

表 13: 复现讲义图过程测量数据

| | | 时间 | 传输速率 | 时间 | 传输速率 | 时间 | 传输速率 | 时间 | 传输速率 | 时间 | 传输速率 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 序号 | 10Mbps | | 50Mbps | | 100Mbps | | 500Mbps | | 1Gbps | |
| 1MB | 1 | 1.5 | 675 | 1.2 | 837 | 1.2 | 845 | 1.2 | 851 | 1.2 | 851 |
| | 2 | 1.5 | 675 | 1.2 | 837 | 1.2 | 845 | 1.2 | 849 | 1.2 | 851 |
| | 3 | 1.5 | 675 | 1.2 | 837 | 1.2 | 845 | 1.2 | 852 | 1.2 | 849 |
| | 4 | 1.5 | 675 | 1.2 | 837 | 1.2 | 845 | 1.2 | 851 | 1.2 | 851 |
| | 5 | 1.5 | 675 | 1.2 | 837 | 1.2 | 845 | 1.2 | 848 | 1.2 | 852 |
| | 均值 | 1.5 | 675 | 1.2 | 837 | 1.2 | 845 | 1.2 | 850.2 | 1.2 | 850.8 |
| 10MB | 1 | 9.4 | 1.14 | 3.1 | 3.22 | 2.3 | 4.28 | 1.9 | 5.4 | 1.8 | 5.47 |
| | 2 | 9.4 | 1.14 | 3.1 | 3.22 | 2.3 | 4.28 | 1.9 | 5.4 | 2 | 4.99 |
| | 3 | 9.4 | 1.14 | 3.1 | 3.22 | 2.3 | 4.28 | 1.9 | 5.4 | 1.8 | 5.45 |
| | 4 | 9.4 | 1.14 | 3.1 | 3.22 | 2.3 | 4.28 | 1.9 | 5.4 | 1.8 | 5.47 |
| | 5 | 9.4 | 1.14 | 3.1 | 3.22 | 2.3 | 4.28 | 1.9 | 5.4 | 1.8 | 5.47 |
| | 均值 | 9.4 | 1.14 | 3.1 | 3.22 | 2.3 | 4.28 | 1.9 | 5.4 | 1.84 | 5.37 |
| 100MB | 1 | 88 | 1.14 | 19 | 5.7 | 10 | 11.4 | 4.1 | 25.7 | 3.5 | 28.5 |
| | 2 | 88 | 1.14 | 19 | 5.7 | 10 | 11.4 | 4.2 | 26.6 | 3.5 | 28.5 |
| | 3 | 88 | 1.14 | 19 | 5.7 | 10 | 11.4 | 3.7 | 27.4 | 3.5 | 28.5 |
| | 4 | 88 | 1.14 | 19 | 5.7 | 10 | 11.4 | 4.1 | 26.9 | 3.5 | 28.5 |
| | 5 | 88 | 1.14 | 19 | 5.7 | 10 | 11.4 | 4.1 | 25.7 | 3.5 | 28.5 |
| | 均值 | 88 | 1.14 | 19 | 5.7 | 10 | 11.4 | 4.04 | 26.46 | 3.5 | 28.5 |

通过软件绘图后，得到如下图像：

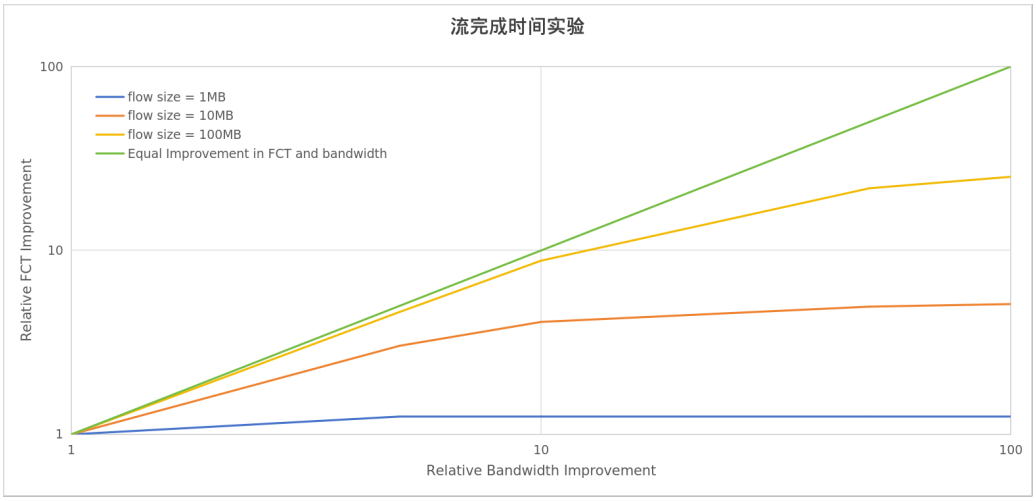


图 9: 流完成时间

可以看到绘制图像与讲义中的图像基本一致，复现效果良好。

对图中现象及实验结果的现象进行总结:

1. 在带宽一定时，传输速率与数据包大小成正比
2. 在数据包大小一定时，传输速率并不会随着带宽线性增加（传输速率的增加速度随带宽增加变缓）。对于各个大小的数据包而言，当带宽达到 500Mbps 后，传输速率基本不再增加
3. 当带宽达到 50Mbps 后，1MB 数据包的传输速率不再因为带宽改变而发生改变
4. 在改变延迟的实验中发现，减小延迟可以显著增加高带宽、大数据包的传输速率。同时在实验过程中也可以较为明显的观察到，在数据包开始传输的一段时间内速度是较慢的，有一个逐步增加的过程。这段时间的长短与延迟成正相关，100ms 延迟下时间较长，10ms 延迟下时间较短。

# 五、 拓展调研

**TCP 传输**

TCP 协议会将应用层的数据流分割成适当长度的报文段，最大传输段大小（MSS）通常受该计算机连接的网络的数据链路层的最大传送单元（MTU）限制。而 TCP 的传输速率是由其阻塞算法决定的，TCP 拥塞算法缓慢地探测网络的可用带宽增加传输速率直到检测到分组丢失，然后指数地降低传输速率。

当数据包大小不变带宽增加时该算法会增加传输速率直至分组丢失而降低传输速率时指数级的因此速率并不会随着带宽的增加而线性增加。

同时对数据进行分组也会造成丢包、排队、阻塞等问题这也会影响到传输速率的增长。

**慢启动机制**

慢启动是 TCP 使用的一种阻塞控制机制。慢启动也叫做指数增长期。慢启动是指每次 TCP 接收窗口收到确认时都会增长。增加的大小就是已确认段的数目。这种情况一直保持到要么没有收到一些段要么窗口大小到达预先定义的阈值。如果发生丢失事件 TCP 就认为这是网络阻塞就会采取措施减轻网络拥挤。一旦发生丢失事件或者到达阈值 TCP 就会进入线性增长阶段。这时，每经过一个 RTT 窗口增长一个段。

由于 TCP 连接会随着时间进行自我调谐，起初会限制连接的最大速度，如果数据传输成功，会随着时间的推移提高传输速度。这就是 TCP 的慢启动机制。

这样就解释了在带宽较高时，小数据包没有达到期待的网速的问题。在慢启动阶段 TCP 预的窗口大小会随着每接受到一个段而指数级增长，对于数据包大小较小的包，在窗口还没有达到带宽的阈值时可能传输就已经结束了, 因此此时测得的传输速率会明显小于对应带宽的最大速率。

# 六、 实验总结

1. 本次实验通过对不同带宽、延迟、数据包大小的传输过程进行定量测量，给我带来了更加直观的印象，可以更好地理解网络传输的机制，对于网络的性能优化有一定的帮助。
2. 此外，通过调研，也加深了我对 TCP 协议的认识，对于网络传输的机制有了更深入的了解。
3. 通过复现讲义的效果图并分析，我也对影响网速的因素有了进一步的认识