Mininet 实验环境报告

陈彦帆 2018K8009918002

（一）互联网协议实验

1、实验内容

（1）通过 mininet 实验环境，在节点 h1 上开启 wireshark 抓包，用 wget 下载 www.baidu.com 页面。

（2）调研说明 wireshark 抓到包涉及的几种协议：ARP, DNS, TCP, HTTP。

（3）调研解释 h1 下载 baidu 页面的整个过程。

2、实验流程

（1）启动并配置环境

```
sudo mn --nat
h1 echo "nameserver 8.8.8.8" > /etc/resolv.conf
```

（2）启动 wireshark

```
h1 wireshark &
```

（3）get

```
h1 wget www.baidu.com
```

3、实验结果

获取到的页面 html 文件:

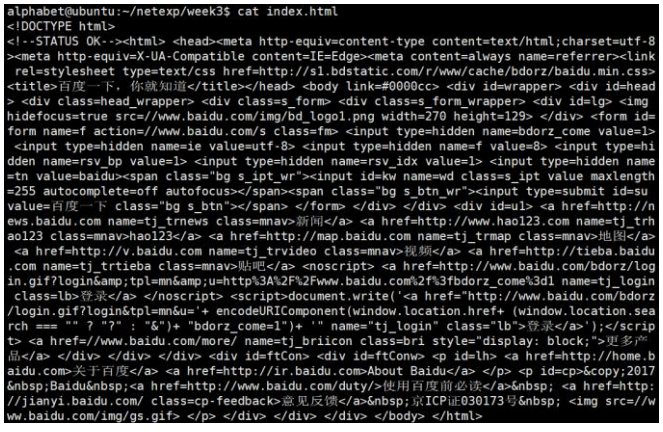


图 1 获取的 index.html 内容

抓取到的网络包（No.19-No.35）：

19	98.669453374	e2:6c:75:26:51:26	Broadcast	ARP	42 Who has 10.0.0.3? Tell 10.0.0.1
20	98.669830942	26:99:69:30:b9:dc	e2:6c:75:26:51:26	ARP	42 10.0.0.3 is at 26:99:69:30:b9:dc
21	98.669839750	10.0.0.1	8.8.8.8	DNS	73 Standard query 0x30bc A www.baidu.com
22	98.669841496	10.0.0.1	8.8.8.8	DNS	73 Standard query 0xadc5 AAAA www.baidu.com
23	98.948488275	8.8.8.8	10.0.0.1	DNS	158 Standard query response 0x30bc A www.baidu.com CNAME www.a.s
24	98.961968192	8.8.8.8	10.0.0.1	DNS	183 Standard query response 0xadc5 AAAA www.baidu.com CNAME www.
25	98.962602374	10.0.0.1	104.193.88.77	TCP	74 60816 → 80 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK_PERM=1
26	99.454550208	104.193.88.77	10.0.0.1	TCP	58 80 → 60816 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
27	99.454652329	10.0.0.1	104.193.88.77	TCP	54 60816 → 80 [ACK] Seq=1 Ack=1 Win=42340 Len=0
28	99.454853134	10.0.0.1	104.193.88.77	HTTP	194 GET / HTTP/1.1
29	99.455360185	104.193.88.77	10.0.0.1	TCP	54 80 → 60816 [ACK] Seq=1 Ack=141 Win=64240 Len=0
30	100.077182962	104.193.88.77	10.0.0.1	HTTP	2551 HTTP/1.1 200 OK (text/html)
31	100.077208017	10.0.0.1	104.193.88.77	TCP	54 60816 → 80 [ACK] Seq=141 Ack=2498 Win=40880 Len=0
32	100.078341656	10.0.0.1	104.193.88.77	TCP	54 60816 → 80 [FIN, ACK] Seq=141 Ack=2498 Win=40880 Len=0
33	100.078744491	104.193.88.77	10.0.0.1	TCP	54 80 → 60816 [ACK] Seq=2498 Ack=142 Win=64239 Len=0
34	100.553146185	104.193.88.77	10.0.0.1	TCP	54 80 → 60816 [FIN, PSH, ACK] Seq=2498 Ack=142 Win=64239 Len=0
35	100.553270215	10.0.0.1	104.193.88.77	TCP	54 60816 → 80 [ACK] Seq=142 Ack=2499 Win=40880 Len=0

图 2 过程中 wireshark 抓取的网络包

4、分析与讨论

调研说明涉及的几种协议及其运行机制

①ARP

ARP(Address Resolution Protocol 地址解析协议)是一个通过解析网络层地址(IP)来找寻数据链路层地址(MAC)的网络传输协议。

在以太网协议中规定，同一局域网中的一台主机要和另一台主机进行直接通信，必须要知道目标主机的 MAC 地址。而在 TCP/IP 协议中，网络层和传输层只关心目标主机的 IP 地址。这就导致在以太网中使用 IP 协议时，数据链路层的以太网协议接到上层 IP 协议提供的数据中，只包含目的主机的 IP 地址。于是需要一种方法，根据目的主机的 IP 地址，获得其 MAC 地址。这就是 ARP 协议要做的事情。

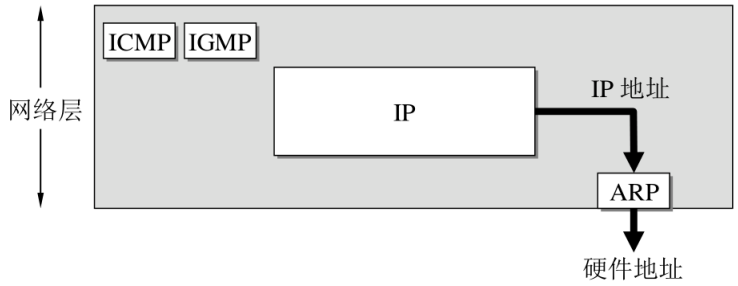


图 3 ARP 协议的作用示意图

运行机制：

I.若发送方 H1（H1 可为主机或路由器）与接收方 H2 在同一局域网中。这时 H1 在局域网中广播 ARP 请求 H2 地址，等待接收方 H2 响应，从而找到 H2 的硬件地址。

II.若发送方 H1 与接收方 H2 不在同一局域网中。这时 H1 在局域网中广播 ARP 请求当前网络的一个路由器 R 的硬件地址。此后发往接收方的所有帧，都发往路由器 R，通过它向外发送。

在本实验中，发送方 h1 的 IP 地址为 10.0.0.1，接收方不在同一局域网中，对应情况 II。于是 h1 通过 ARP 协议寻找一个路由器 10.0.0.3 的物理地址。

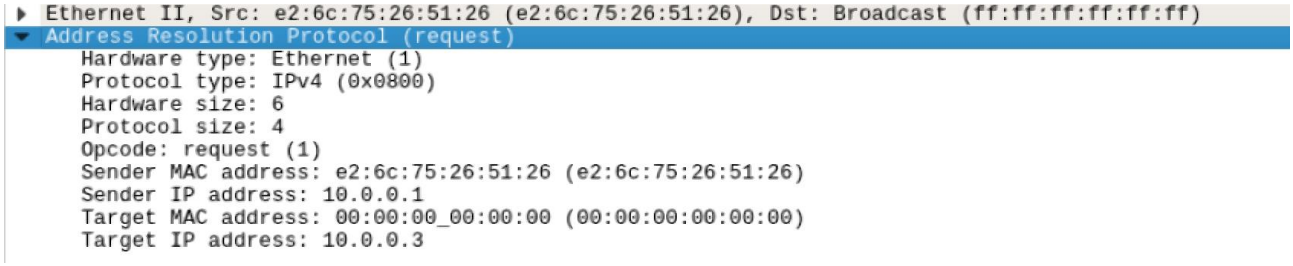


图 4 No.19 网络包部分内容

接收方返回自己的物理地址:26:99:69:30:b9:dc。发送方 h1 将其缓存到 ARP 高速缓存中，在自动失效之前无须再次查询。

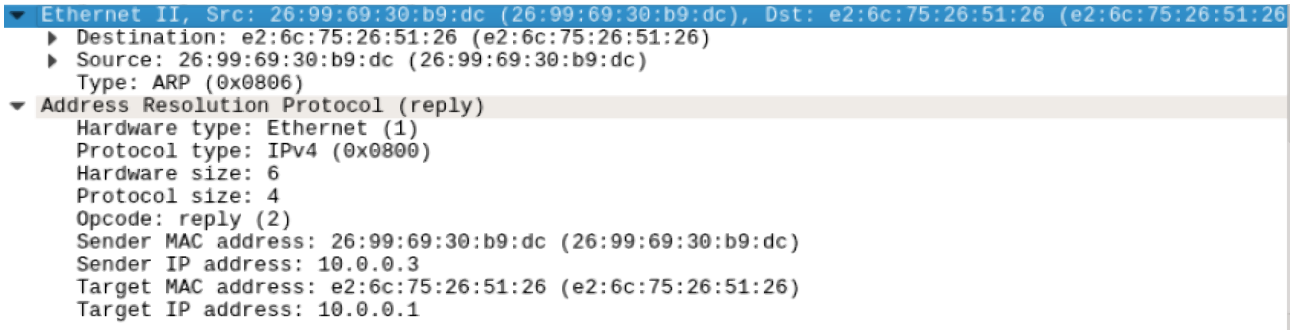


图 5 No.20 网络包部分内容

②DNS

DNS (Domain Name System 域名系统)是一种网络服务，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。

运行机制：

当某一个应用进程需要把主机名解析为 IP 地址时，该应用进程就调用解析程序(resolver)，并成为 DNS 的一个客户，把待解析的域名放在 DNS 请求报文中，以 UDP 用户数据报方式发给指定的域名服务器。域名服务器在查找域名后，把对应的 IP 地址放在回答报文中返回。应用进程获得目的主机的 IP 地址后即可进行通信。

如图 6，在本实验中，返回的网络包显示 www.baidu.com 是一个别名，指向 CNAME(Canonical Name)www.a.shifen.com，而后者指向 www.wshifen.com，并返回了两个 IP 地址。

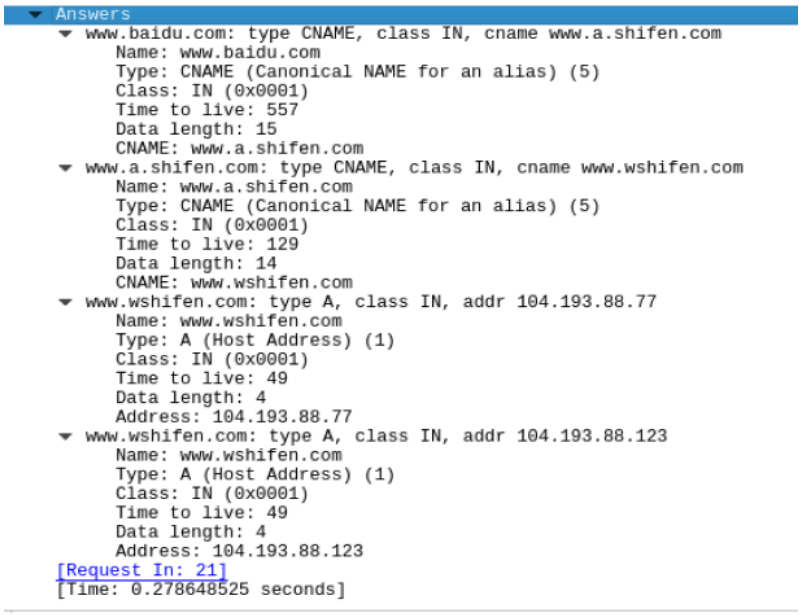


图 6 No.23 网络包部分内容

③TCP

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，完成传输层所指定的功能。该层的协议为应用进程提供端到端的通信服务。

TCP 的主要特点：

I. TCP 是面向连接的运输层协议。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。

II. 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。

III. TCP 提供可靠交付的服务。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。

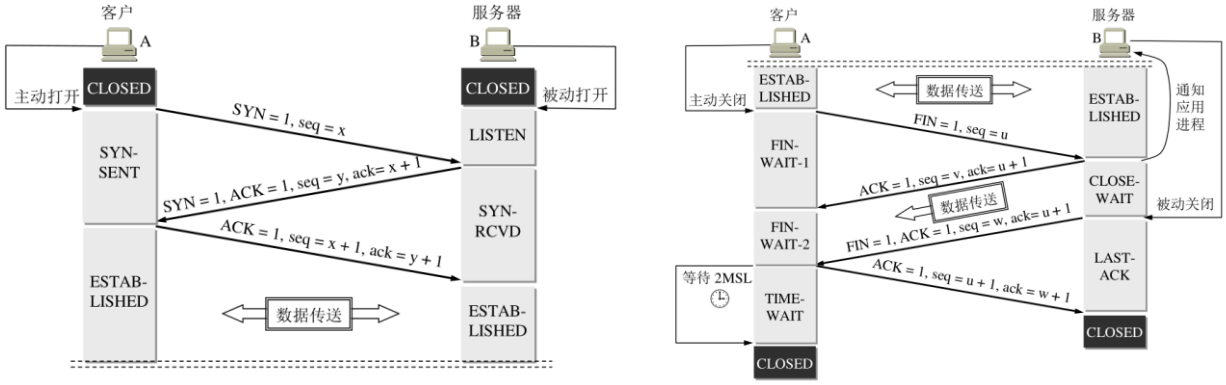
IV. TCP 提供全双工通信。TCP 允许通信双方的应用进程在任何时候都能发送数据。TCP 连接的两端都设有发送缓存和接收缓存，用来临时存放双向通信的数据。在发送时，应用程序在把数据传送给 TCP 的缓存后，就可以做自己的事，而 TCP 在合适的时候把数据发送出去。在接收时，TCP 把收到的数据放入缓存，上层的应用进程在合适的时候读取缓存中的数据。

V.面向字节流。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。

运行机制：

I.建立连接

TCP 建立连接的过程叫做握手，握手需要在客户和服务器之间交换三个 TCP 报文段，称为三次握手。服务器要确认客户的连接请求，然后客户要对服务器的确认进行确认。



（a）　　(b)

图 7 TCP 建立连接(a)、释放连接(b)过程示意图

本实验中，图 2 的 No.25-No.27 网络包显示了 TCP 建立连接三次握手的过程。

II.可靠传输

停止等待协议能够在不可靠的传输网络上实现可靠的通信。每发送完一个窗口中的分组就停止发送，等待对方的确认。在收到确认后再发送下一个窗口。分组需要进行编号。

超时重传是指只要超过了一段时间仍然没有收到确认，就重传前面发送过的分组。

发送方维持一个发送窗口，凡位于发送窗口内的分组都可连续发送出去，而不需要等待对方的确认。接收方一般采用累积确认，对按序到达的最后一个分组发送确认，表明到这个分组为止的所有分组都已正确收到了。

流量控制就是让发送方的发送速率不要太快，要让接收方来得及接收。拥塞控制就是防止过多的数据注入到网络中，这样可以使网络中的路由器或链路不致过载。

发送窗口取为拥塞窗口和接收方的接收窗口中较小的一个。

在本实验中，图 2 的 No.29 报文是对 No.28 HTTP 报文的确认。No.31 报文是对 No.30 HTTP 报文的确认。

III. 连接释放

TCP 的连接释放采用四报文握手机制。任何一方都可以在数据传送结束后发出连接释放的通知，待对方确认后就进入半关闭状态。当另一方也没有数据再发送时，则发送连接释放通知，对方确认后就完全关闭了 TCP 连接。(见图 7(b))

本实验中，图 2 的 No.32-No.35 网络包显示了连接释放四次握手的过程。

④ HTTP

HTTP（HyperText Transfer Protocol 超文本传输协议）定义了浏览器（即万维网客户进程）怎样向万维网服务器请求万维网文档，以及服务器怎样把文档传送给浏览器。HTTP 使用了面向连接的 TCP 作为运输层协议，保证了数据的可靠传输。

HTTP 有两类报文：

(1) 请求报文—从客户向服务器发送请求报文，见图 8(a)。

(2) 响应报文—从服务器到客户的回答，见图 8(b)。

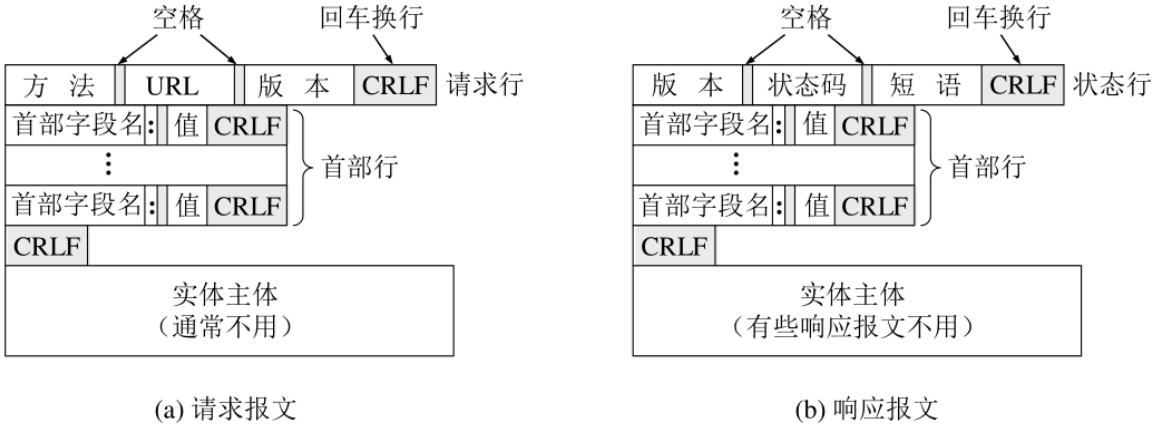


图 8 HTTP 请求报文和响应报文结构示意图

运行机制：

服务器进程不断地监听 TCP 的端口 80，以便发现是否有浏览器向它发出连接建立请求。一旦监听到连接建立请求并建立了 TCP 连接之后，浏览器就向万维网服务器发出浏览某个页面的 HTTP 请求，服务器接着就通过 HTTP 返回所请求的页面作为响应。最后，TCP 连接被释放。

本实验中，图 2 的 No.28 是 h1 发往 www.baidu.com 的 HTTP get 请求报文，No.30 是服务器返回的 HTTP 响应报文。

总结：

① h1 通过 ARP 协议获取当前网络的一个路由器的物理地址。此后发往局域网外的接收方的所有帧，都发往该物理地址。

② h1 通过 DNS 协议向 8.8.8.8 获取 www.baidu.com 的 IP 地址 104.193.88.77。

③ h1 与 104.193.88.77 建立 TCP 连接。h1 发送 HTTP get 请求，104.193.88.77 返回 HTTP 响应。h1 与 104.193.88.77 释放 TCP 连接。

（二）流完成时间实验

1、实验内容

（1）利用提供的 fct_exp.py 脚本，测量以下各个条件的流完成时间并绘制流完成时间改进对数图。每个数据点做 5 次实验，取均值。

文件大小：1MB，10MB，100MB

带宽：10Mbps，50Mbps，100Mbps，500Mbps，1Gbps

延迟：100ms

（2）调研解释图中的现象

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，可变参数为文件大小和带宽，返回值为流完成时间（取 5 次测量的均值）。在脚本中更改参数进行测试。

（2）在脚本中利用 matplotlib 包进行绘图。

（3）运行命令： sudo python fct_exp.py

3、实验结果

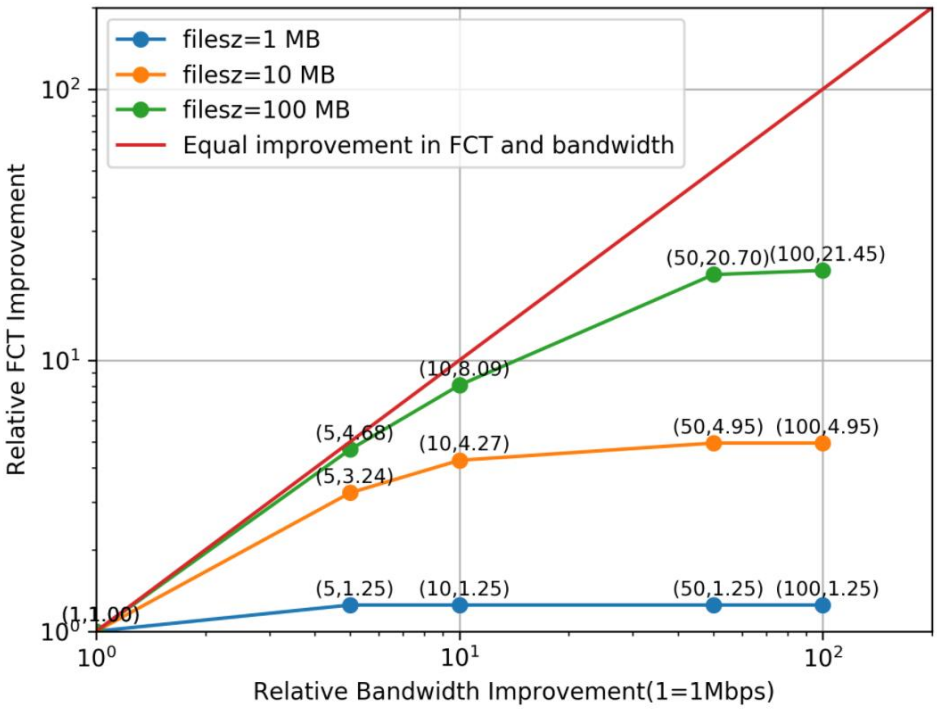


图 9 不同带宽、文件大小下流完成时间(FCT)改进变化图(delay=100ms)

4、分析与讨论

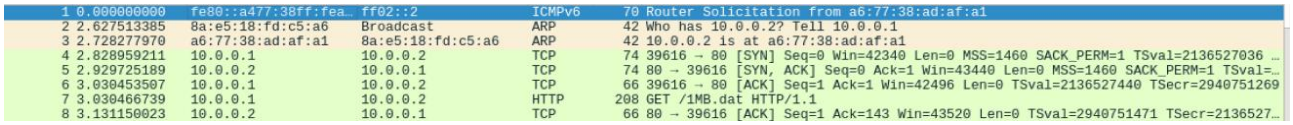
（1）结果分析

① 当延迟相同，相对带宽改进(RBI)相同时，文件大小越大，相对流完成时间改进(RFCTI)就越大。RBI/RFCTI 均小于 1。

② 当延迟相同，文件大小相同时，RBI/RFCTI 随着 RBI 的增大而减小。当带宽增大到一定大小时，再提升带宽，FCT 几乎不再减少。

（2）解释

将流完成时间 T 划分为与文件大小无关的时间 T1 和传输数据包的时间 T2。T1 包括 ARP 请求物理地址的时间，TCP 建立连接和释放连接的时间等。图 10 中的 No.2-No.8 报文（包括 ARP 请求，TCP 建立连接，HTTP get 请求及其响应）与带宽、文件大小无关，消耗的时间属于 T1。T1 不因带宽增大而减小，因此 RBI/RFCTI 均小于 1。



1	0.000000000	fe80::a477:38ff:fea…	ff02::2	ICMPv6	70 Router Solicitation from a6:77:38:ad:af:a1
2	2.627513385	8a:e5:18:fd:c5:a6	Broadcast	ARP	42 Who has 10.0.0.2? Tell 10.0.0.1
3	2.728277970	a6:77:38:ad:af:a1	8a:e5:18:fd:c5:a6	ARP	42 10.0.0.2 is at a6:77:38:ad:af:a1
4	2.828959211	10.0.0.1	10.0.0.2	TCP	74 39616 → 80 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK_PERM=1 TSval=2136527036 …
5	2.929725189	10.0.0.2	10.0.0.1	TCP	74 80 → 39616 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM=1 TSval=…
6	3.030453507	10.0.0.1	10.0.0.2	TCP	66 39616 → 80 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=2136527440 TSecr=2940751269
7	3.030466739	10.0.0.1	10.0.0.2	HTTP	208 GET /1MB.dat HTTP/1.1
8	3.131150023	10.0.0.2	10.0.0.1	TCP	66 80 → 39616 [ACK] Seq=1 Ack=143 Win=43520 Len=0 TSval=2940751471 TSecr=2136527…

图 10 本实验 wireshark 抓取到的部分网络包

①文件大小相同时，带宽的增大可以增大拥塞窗口的大小，当拥塞窗口小于接收窗口时，一次能发出的分组数更多，T2 减小。但 T1 是不变的，当文件较小或带宽较大时，T 受 T1 的影响相对较大，因此 FCT 相对改进较小。

②为了得到合适的发送窗口，TCP 采用慢启动机制，一开始的发送窗口较小，然后根据情况迅速增大。因此在数据传输的开始阶段，吞吐量远达不到带宽，这部分时间不因带宽增大或文件大小的减小而减小。当带宽较大或文件较小时，这部分时间占总时间的比重较大，因此相对改进较小。