Mininet 实验环境

学号：2016K8009908007
姓名：薛峰

（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.baidu.com 页面；

(2) 调研说明 wireshark 抓到的几种协议：ARP, DNS, TCP, HTTP；

(3) 调研解释 h1 下载 baidu 页面的整个过程。

二、实验流程

1、搭建实验环境

- $ sudo apt install wireshark
- $ sudo mn --nat # allows hosts to connect with the Internet
- mininet> xterm h1
- h1 # echo "nameserver 1.2.4.8" > /etc/resolv.conf
- h1 # wireshark &

2、实验步骤

- h1 # wget www.baidu.com

三、实验结果及分析

1、实验结果

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	a2:46:45:23:92:be	Broadcast	ARP	42	Who has 10.0.0.3? Tell 10.0.0.1
2	0.002389084	0e:5a:15:c0:d6:0b	a2:46:45:23:92:…	ARP	42	10.0.0.3 is at 0e:5a:15:c0:d6:0b
3	0.002395238	10.0.0.1	1.2.4.8	DNS	73	Standard query 0x0d6c A www.baidu.com
4	0.004258118	10.0.0.1	1.2.4.8	DNS	73	Standard query 0xdad9 AAAA www.baidu.com
5	5.009943194	10.0.0.1	1.2.4.8	DNS	73	Standard query 0x0d6c A www.baidu.com
6	5.009998793	10.0.0.1	1.2.4.8	DNS	73	Standard query 0xdad9 AAAA www.baidu.com
7	10.012966359	10.0.0.1	1.2.4.8	DNS	73	Standard query 0x62af A www.baidu.com
8	10.013004144	10.0.0.1	1.2.4.8	DNS	73	Standard query 0x3698 AAAA www.baidu.com
9	15.018647650	10.0.0.1	1.2.4.8	DNS	73	Standard query 0x62af A www.baidu.com
10	15.018717318	10.0.0.1	1.2.4.8	DNS	73	Standard query 0x3698 AAAA www.baidu.com
11	15.035544396	1.2.4.8	10.0.0.1	DNS	302	Standard query response 0x62af A www.baidu.com CNAME www.a.shifen.com A 61…
12	15.041359564	1.2.4.8	10.0.0.1	DNS	157	Standard query response 0x3698 AAAA www.baidu.com CNAME www.a.shifen.com S…
13	15.042483500	10.0.0.1	61.135.169.125	TCP	74	56328 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=35688224…
14	15.085097663	61.135.169.125	10.0.0.1	TCP	58	80 → 56328 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
15	15.085145317	10.0.0.1	61.135.169.125	TCP	54	56328 → 80 [ACK] Seq=1 Ack=1 Win=29200 Len=0
16	15.085278318	10.0.0.1	61.135.169.125	HTTP	194	GET / HTTP/1.1
17	15.089456215	61.135.169.125	10.0.0.1	TCP	54	80 → 56328 [ACK] Seq=1 Ack=141 Win=65535 Len=0
18	15.328851500	61.135.169.125	10.0.0.1	TCP	1474	80 → 56328 [ACK] Seq=1 Ack=141 Win=65535 Len=1420 [TCP segment of a reasse…
19	15.328870922	10.0.0.1	61.135.169.125	TCP	54	56328 → 80 [ACK] Seq=141 Ack=1421 Win=31240 Len=0
20	15.329064515	61.135.169.125	10.0.0.1	TCP	86	80 → 56328 [PSH, ACK] Seq=1421 Ack=141 Win=65535 Len=32 [TCP segment of a …
21	15.329194051	10.0.0.1	61.135.169.125	TCP	54	56328 → 80 [ACK] Seq=141 Ack=1453 Win=31240 Len=0
22	15.329069770	61.135.169.125	10.0.0.1	HTTP	1383	HTTP/1.1 200 OK (text/html)
23	15.329203681	10.0.0.1	61.135.169.125	TCP	54	56328 → 80 [ACK] Seq=141 Ack=2782 Win=34080 Len=0
24	15.330361986	10.0.0.1	61.135.169.125	TCP	54	56328 → 80 [FIN, ACK] Seq=141 Ack=2782 Win=34080 Len=0
25	15.330811253	61.135.169.125	10.0.0.1	TCP	54	80 → 56328 [ACK] Seq=2782 Ack=142 Win=65535 Len=0
26	15.346463099	61.135.169.125	10.0.0.1	TCP	54	80 → 56328 [FIN, ACK] Seq=2782 Ack=142 Win=65535 Len=0
27	15.346488977	10.0.0.1	61.135.169.125	TCP	54	56328 → 80 [ACK] Seq=142 Ack=2783 Win=34080 Len=0

2、结果分析

ARP 协议：即地址解析协议， 用于实现从 IP 地址到 MAC 地址的映射，即询问目标 IP 对应的 MAC 地址。例如，目前有两台机器 PC1 和 PC2，在 PC1 中运行指令"ping ip2"，此时 PC1 便有了通信需要的源目 IP 地址，但是 PC1 仍然没有通信需要的目的 MAC 地址。ARP 协议的作用就是获取 PC2 的 MAC 地址。PC1 会将包含目标 IP 地址的 ARP 请求广播到网络上的所有主机（该网络下只有 PC1 和 PC2），并接收返回消息，以此确定 PC2 的物理地址。之后，PC1 还会将 PC2 的 MAC 信息放入本地的“ARP 缓存表”中，表内存放了 IP 和 MAC 地址的映射，下次请求时可直接查询 ARP 缓存表。

结果图中的两个 ARP 包，第一个是 ARP 请求包，第二个是 ARP 回应包。

ARP 请求包：

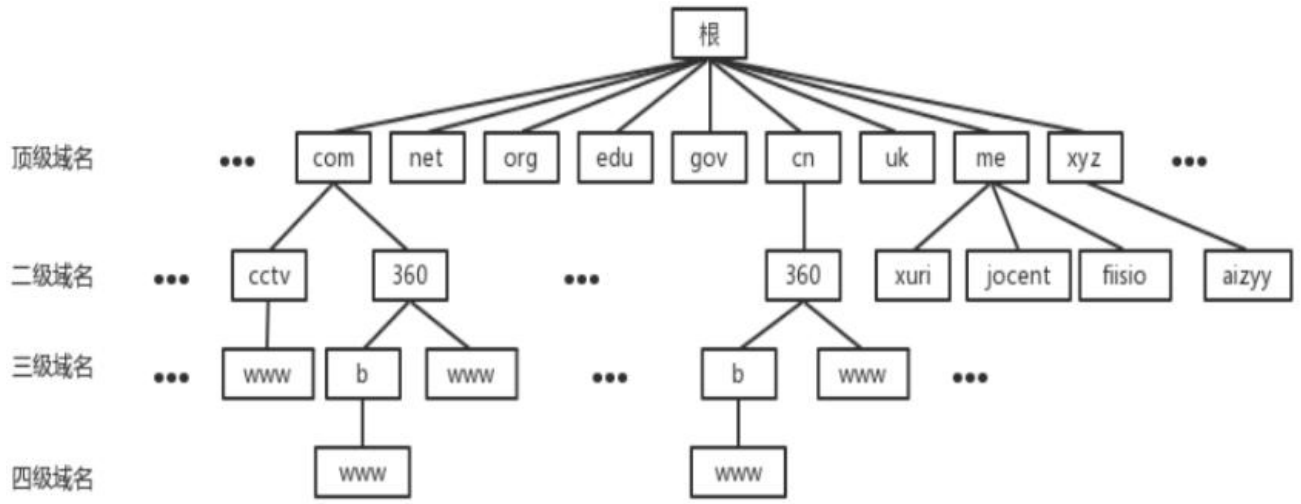
```
▸ Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▸ Ethernet II, Src: a2:46:45:23:92:be (a2:46:45:23:92:be), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▾ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: a2:46:45:23:92:be (a2:46:45:23:92:be)
    Sender IP address: 10.0.0.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 10.0.0.3
```

可以看出其传输方式为广播，操作代码为请求，并且发送的 IP 为 10.0.0.。

ARP 回应包：

```
▸ Frame 2: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▸ Ethernet II, Src: 0e:5a:15:c0:d6:0b (0e:5a:15:c0:d6:0b), Dst: a2:46:45:23:92:be (a2:46:45:23:92:be)
▾ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 0e:5a:15:c0:d6:0b (0e:5a:15:c0:d6:0b)
    Sender IP address: 10.0.0.3
    Target MAC address: a2:46:45:23:92:be (a2:46:45:23:92:be)
    Target IP address: 10.0.0.1
```

DNS 协议：即域名解析协议，用于将域名转换为 IP 地址，是一个应用层协议。DNS（Domain Name System),，即域名系统，用于维护系统内的每个主机的 IP 和主机名的对应关系。当用户输入域名的时，会自动查询 DNS 服务器，由 DNS 服务器检索数据库，得到对应的 IP 地址。DNS 域名结构为层次结构，如下图[1]：



例如域名 www.baidu.com，其中 com 为一级域名，表示这是一个企业域名； baidu 为二级域名，指公司名；www 只是一种习惯用法。

同样地，域名服务器也有层次结构，

分类	作用
根域名服务器	最高层次的域名服务器，本地域名服务器解析不了的域名就会向其求助
顶级域名服务器	负责管理在该顶级域名服务器下注册的二级域名
权限域名服务器	负责一个区的域名解析工作
本地域名服务器	当一个主机发出 DNS 查询请求时，这个查询请求首先发给本地域名服务器

域名解析的过程为：

(1) 输入域名后，先查找自己主机对应的域名服务器，域名服务器先查找自己的数据库中的数据；

(2) 如果没有， 就向上级域名服务器进行查找，依次类推，最多回溯到根域名服务器；

(3) 根域名服务器告诉本地域名服务器，下一次应该查询的顶级域名服务器的 IP 地址 ；

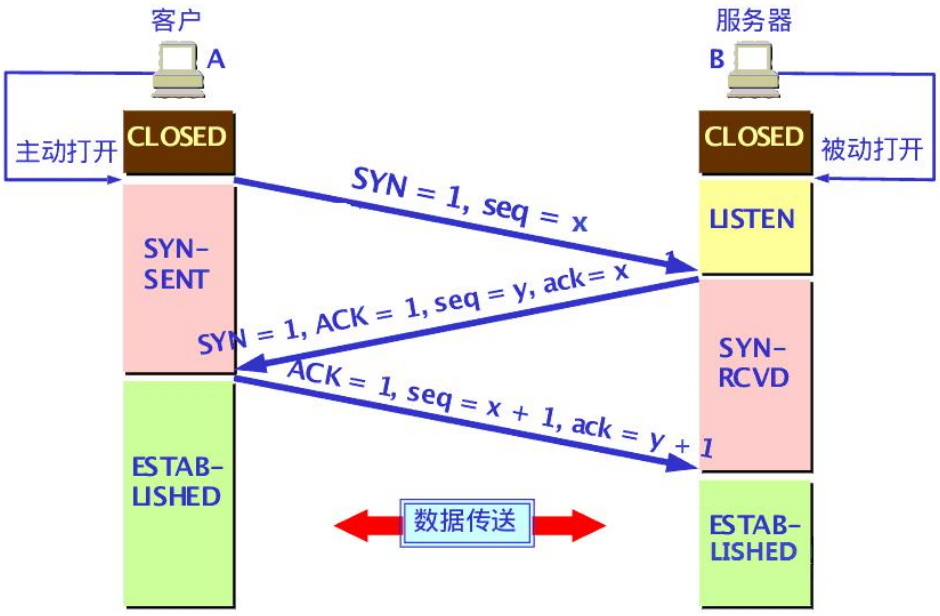
(4) 本地域名服务器向顶级域名服务器进行查询 ；

(5) 以此类推，直到查询到要查询的主机的 IP 地址 ；

(6) 本地域名服务器最后把查询结果告诉主机。

TCP 协议：TCP 和 IP 在一起协同工作，其中 TCP 在运输层，IP 在网际层。TCP 负责应用软件和网络软件之间的通信，IP 负责计算机之间的通信；TCP 负责将数据分割并装入 IP 包，然后在它们到达的时候重新组合它们，IP 负责将包发送至接受者。

TCP 三次握手：指建立一个 TCP 连接时，需要客户端和服务端总共发送 3 个包以确认连接的建立。流程如下：



(1)建立连接时，客户端发送 SYN 包（SYN=i）到服务器，并进入到 SYN-SEND 状态，等待服务器确认

(2)服务器收到 SYN 包，必须确认客户的 SYN（ack=i+1），同时自己也发送一个 SYN 包（SYN=k），即 SYN+ACK 包，此时服务器进入 SYN-RECV 状态

(3)客户端收到服务器的 SYN+ACK 包，向服务器发送确认报 ACK（ack=k+1），此包发送完毕，客户端和服务器进入 ESTABLISHED 状态，完成三次握手，客户端与服务器开始传送数据。

TCP 四次挥手：指断开一个 TCP 连接时，需要客户端和服务端总共发送 4 个包以确认连接的断开。流程如下：

(1)第一次挥手：Client 发送一个 FIN，用来关闭 Client 到 Server 的数据传送，Client 进入 FIN_WAIT_1 状态。

(2)第二次挥手：Server 收到 FIN 后，发送一个 ACK 给 Client，确认序号为收到序号+1（与 SYN 相同，一个 FIN 占用一个序号），Server 进入 CLOSE_WAIT 状态。

(3)第三次挥手：Server 发送一个 FIN，用来关闭 Server 到 Client 的数据传送，Server 进入 LAST_ACK 状态。

(4)第四次挥手：Client 收到 FIN 后，Client 进入 TIME_WAIT 状态，接着发送一个 ACK 给 Server，确认序号为收到序号+1，Server 进入 CLOSED 状态，完成四次挥手[2]。

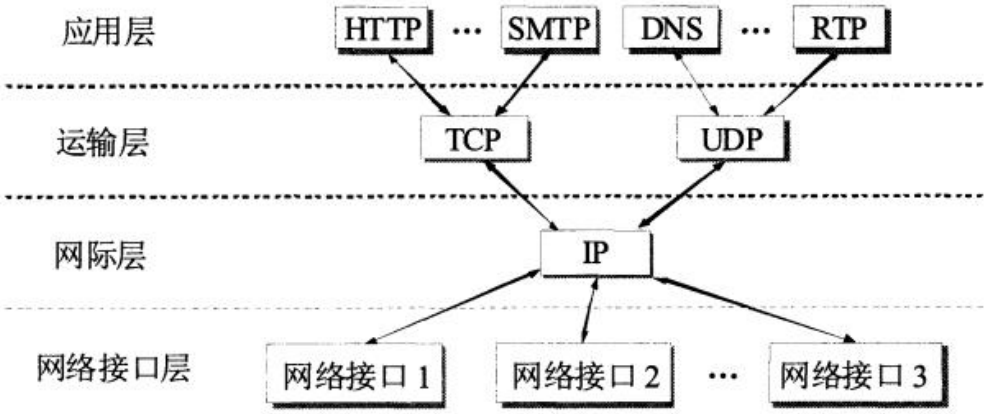
HTTP 协议：Hyper Text Transfer Protocol，即超文本传输协议，是用于从 WWW 服务器传输超文本到本地浏览器的传送协议。HTTP 是一个应用层协议，由请求和响应构成。其特点有：

(1) 基于 TCP 协议：HTTP 协议规定客户端和服务端数据传输的格式和数据交互行为，并不负责数据传输的细节。底层是基于 TCP 实现的。现在使用的版本当中是默认持久连接的，也就是多次 HTTP 请求使用一个 TCP 连接。

(2) 多次请求：在客户端请求网页时多数情况下并不是一次请求就能成功的，服务端首先是响应 HTML 页面，然后浏览器收到响应之后发现 HTML 页面还引用了其他的资源，还会自动发送 HTTP 请求这些需要的资源。

(3) 无状态：就是说每次 HTTP 请求都是独立的，任何两个请求之间没有什么必然的联系。

各协议间的关系：



下载 baidu 页面的过程：

(1) h1 通过 ARP 协议获取交换机的 mac 地址。h1 广播的信号（第一个 ARP 包）被交换机接收之后，交换机将其 mac 地址回复给 h1（第二条 ARP 包）；

(2) h1 通过 DNS 协议将获取了域名 www.baidu.com 的 IP 地址；

(3) h1 通过 TCP 协议与 baidu 服务器建立一个连接；

(4) h1 通过 HTTP 协议进行数据的传输。

（二）流完成时间实验

一、实验内容

(1) 利用 fct_exp.py 脚本复现课件中的图；

(2) 调研解释图中的现象。

二、实验流程

1、搭建实验环境

- $ sudo python fct_exp.py
- mininet> xterm h1 h2

2、实验步骤

- h2 # dd if=/dev/zero of=1MB.dat bs=1M count=1
- h1 # wget http://10.0.0.2/1MB.dat

三、实验结果及分析

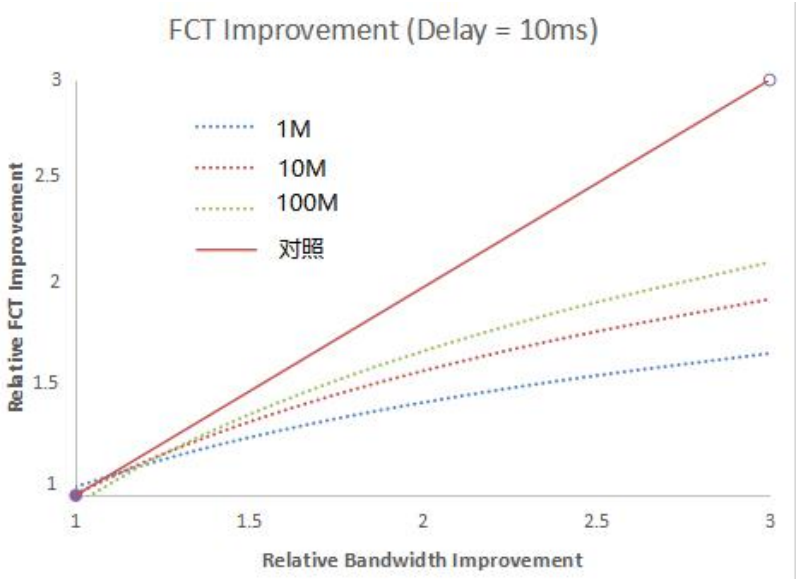
1、实验结果

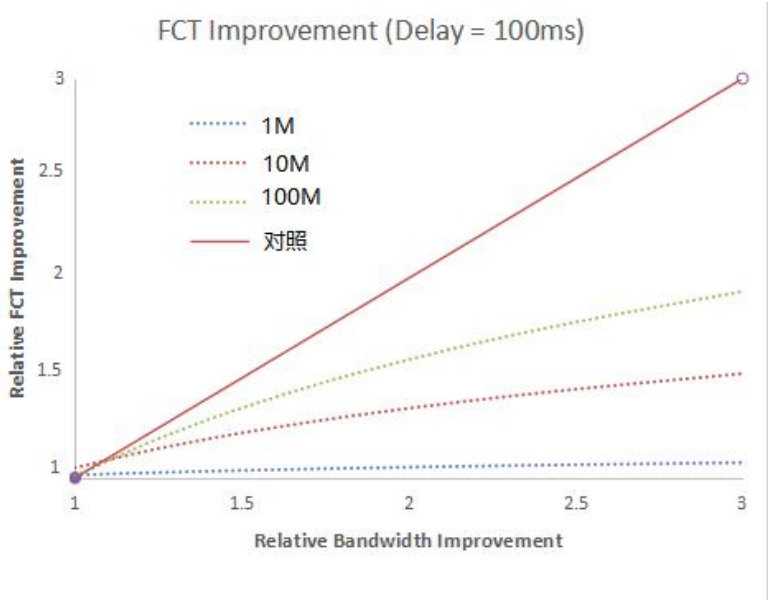
Dealy = 10ms：

文件大小\带宽	10Mbps	50Mbps	100Mbps	500Mbps	1GMbps
1M	0.9s	0.2s	0.2s	0.2s	0.002s
10M	8.8s	1.9s	1.0s	0.8s	0.04s
100M	90s	18s	9.7s	2.9s	0.7s

Dealy = 100ms：

文件大小\带宽	10Mbps	50Mbps	100Mbps	500Mbps	1GMbps
1M	1.5s	1.2s	1.2s	1.2s	1.2s
10M	9.5s	3s	2.3s	1.9s	1.9s
100M	90s	19s	11s	4.2s	4.2s





2、结果分析

现象：当文件大小固定时，随着带宽的增加，传输的平均速率也会增加，但最终趋于稳定。

解释：TPC 传输存在慢启动机制，即开始传输的一段时间传输的速度非常慢。这是因为在连接建立的初期，如

果传输速率大，发送方可能会突然发送大量数据，导致网络瘫痪。因此，在通信一开始时，TCP 会通过慢启动算法，对发送数据量进行控制。所以随着带宽的增加，慢启动机制逐渐成为限制平均传输效率的瓶颈，因此当带宽增加到一定程度时，传输效率会趋于稳定。

四、实验总结

通过本次实验，我对 mininet 环境有了一个大致的了解，为日后的实验打下了基础。并且通过自主调研的形式，我学习了各种各样的网络协议，这让我对计算机网络有了初步的认识。并且在和同学交流的过程中，相互分享了搜集到的资料，并且共同弄懂了不理解的地方，这让我觉得这样的学习方式更易于掌握知识。所以还是希望老师以后能发布一些分组查阅资料的任务。

参考文献：

【1】DNS（域名解析协议）详解. https://blog.csdn.net/baidu_37964071/article/details/80500825

【2】TCP 协议详解. https://www.cnblogs.com/buxiangxin/p/8336022.html