

Principles of Computer Networks

Homework 4*

Chenghua Liu[†]

liuch18@mails.tsinghua.edu.cn

Department of Computer Science

Tsinghua University

目录

1	problems in chapter 4	1
1.1	problem 2	1
1.2	problem 13	2
1.3	problem 14	2
1.4	problem 15	2
1.5	problem 18	2
1.6	problem 25	2
1.7	problem 27	3
2	IEEE 802.3 协议实验	3
2.1	IPv4 部分 (ARP 协议)	3
2.2	IPv6 部分 (ND 协议)	5

1 problems in chapter 4

1.1 problem 2

纯 ALOHA 协议的信道利用率最高为 18.4%。所以有

$$N \frac{1000 \text{ bit}}{100 \text{ s}} \leq 56 \text{ kbps} \times 18.4\%$$

得到 $N \leq 1030$ ，因此最多同时支持 1030 个站点。

*problems from Computer Networks, 5th Edition

[†]2018011687

1.2 problem 13

Ethernet 使用曼彻斯特编码，每发送一位数据需要信号变化两次，因此需要的波特率是比特率的两倍，即 20MHz。

1.3 problem 14

对于曼彻斯特编码，LH 表示 0，HL 表示 1，输出为

LH LH LH HL HL HL LH HL LH HL

1.4 problem 15

本题增加以下两点补充条件：

1. 本题采用了确认帧来检测可能的发送错误，因此发送方不需要使发送时间达到二倍信道传输延迟来检测冲突。
2. 若信道空闲且有数据要发送，则立即发送数据。

发送方数据发送时间

$$\frac{256 \text{ b}}{10 \text{ Mbps}} = 25.6 \mu s$$

数据帧传输时间

$$\frac{1 \text{ km}}{200 \text{ m/ms}} = 5 \mu s$$

确认帧发送时间

$$\frac{32 \text{ b}}{10 \text{ Mbps}} = 3.2 \mu s$$

确认帧传输时间 $5 \mu s$ ，所以总传输时间为 $38.8 \mu s$ 。

所以数据传输速率为

$$\frac{256 - 32 \text{ b}}{38.8 \mu s} = 5.77 \text{ Mbps}$$

1.5 problem 18

Fast Ethernet 所要求的线路延迟是 Ethernet 的 1/10。因此只需 1/10 的时间就能获知线路发生冲突。

1.6 problem 25

每帧数据的错误率

$$p = 1 - (1 - 10^{-7})^{64 \times 8} = 5.12 \times 10^{-5}$$

因此平均每秒损坏帧数

$$\frac{11 \text{ Mbps}}{64 \times 8 \text{ b}} \times p = 1.1$$

1.7 problem 27

一种可能的情况是, 如果应用环境要求很高的实时性, 即一个损坏的帧没有时间来重传, 我们可以用纠错码来修复。

另一种情况, 如果链路的误码率较高, 使得重传成本过大, 那么可以用纠错码来尝试修复一部分损坏的帧。

2 IEEE 802.3 协议实验

2.1 IPv4 部分 (ARP 协议)

- 1) 依次查看捕获的各数据帧, 目的地为实验主机的数据帧中长度最小的是多大? 查看这种帧的各个域, 看看先导域是否包含在记录的数据中; 记录的数据是从哪个字段开始, 至哪个字段结束? 这是否验证了 IEEE 802.3 标准中规定的最小帧长为 64 字节?

答:

前同步码	目的地址	源地址	数据长度	数据	校验
8	2	2	2	46 ~ 1500	4

最小帧长 54: 有些数据没有填入。验证了 IEEE 802.3 标准中规定的最小帧长为 64 字节。

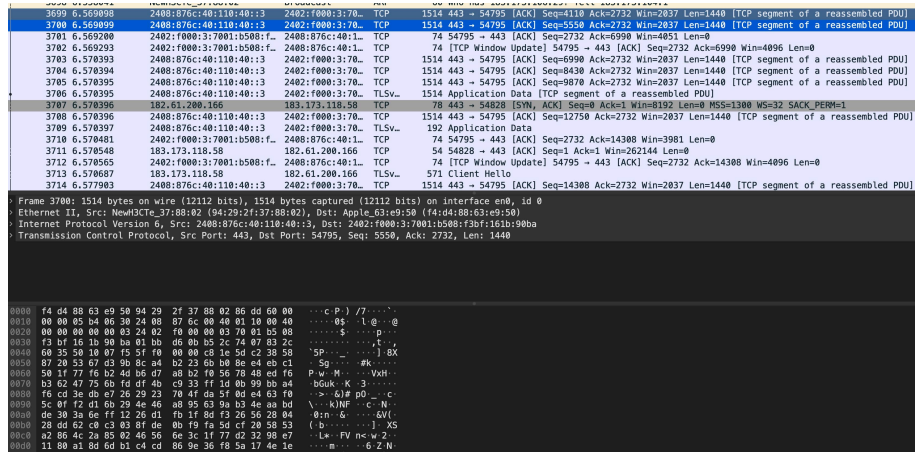
3721	6.582096	183.173.118.58	58.285.228.41	TCP	1294	54796 → 443 [ACK] Seq=1 Ack=1 Win=4896 Len=1240 [TCP segment of a reassembled PDU]
3722	6.582223	183.173.118.58	182.61.200.166	TCP	54	54828 → 443 [ACK] Seq=318 Ack=159 Win=261952 Len=0
3723	6.582179	183.173.118.58	58.285.228.36	TCP	54	54812 → 443 [ACK] Seq=39 Ack=2 Win=4896 Len=0
3724	6.582190	183.173.118.58	58.285.228.41	TLSv.	317	Application Data
3725	6.582210	183.173.118.58	58.285.228.41	TLSv.	100	Application Data
3726	6.582313	183.173.118.58	182.61.200.166	TLSv.	105	Change Cipher Spec, Encrypted Handshake Message
3727	6.582398	183.173.118.58	182.61.200.166	TCP	1354	54828 → 443 [ACK] Seq=569 Ack=159 Win=262144 Len=1380 [TCP segment of a reassembled PDU]

```
Frame 2711: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface end, id 0
  Interface id: 0 (en0)
  Encapsulation type: Ethernet (1)
  Arrival Time: Nov 24, 2021 17:55:52.484080000 CST
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 163747752.484080000 seconds
  [Time delta from previous captured frame: 0.000067000 seconds]
  [Time delta from previous displayed frame: 0.000067000 seconds]
  [Time since reference or first frame: 6.578548800 seconds]
  Frame Number: 2711
  Frame Length: 54 bytes (432 bits)
  Capture Length: 54 bytes (432 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
000  9a 29 2f 37 8a 02 f4 d4 88 63 e9 50 00 00 45 00  3728  .C.P.E.
010  00 20 00 00 00 00 00 00 8e 84 d7 ad 7a 00 3d    (.000  .Vine
020  c3 06 06 0c 41 5b 8f f2 5f 48 00 c1 de 71 5b 18  .M.Nap
030  10 00 53 ab 00 00                                .S..
```

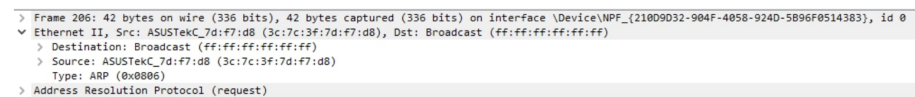
- 2) 查找捕获帧中长度最长的帧。可以多访问一些网页以捕获更多的帧, 看看这些帧的长度最大是多大? 为什么?

答:

帧的格式: 7 个字节的前导码 +6 个字节的源 mac 地址 +6 个字节的目的地 mac 地址 +2 个字节的类型 +46 - 1500 个字节的的数据 +4 个字节的奇偶验证, 在实际操作中帧最长为 1514, 最短为 54。

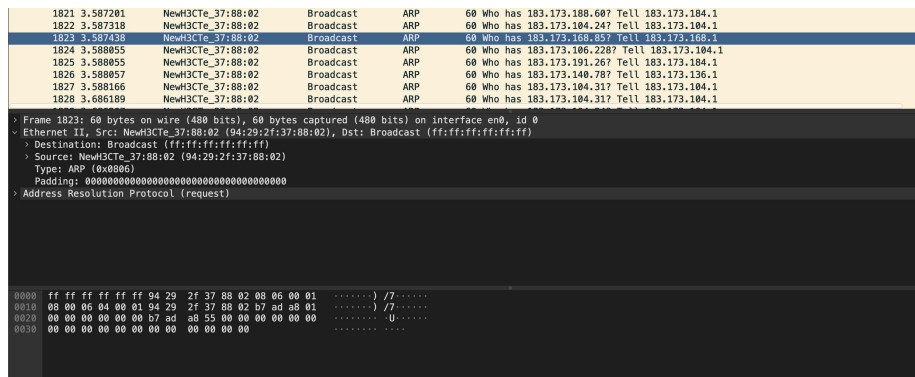


3) 找到捕获帧中由实验主机发出的 ARP 请求 (Request) 帧，辨认其目的地址域和源地址域，参照下图。看看它的目的 MAC 地址是多少？



答：

目的 MAC 地址：94: 29: 2f: 37: 88: 02



4) 对比一下封装 ARP 分组的帧和其他帧 (封装 IP 分组的帧)，看看它们的类型字段分别是多少？

封装 ARP 分组的帧的类型字段占 2 字节，对应的值如下：

- ARP 请求：1
- ARP 响应：2
- RARP 请求：3

- RARP 响应: 4

封装 IP 分组的帧的类型字段占 8 字节, 具体为

- 过程字段: 3 位, 设置了数据包的重要性, 取值越大数据越重要, 取值范围为: 0 (正常) 7 (网络控制)
- 延迟字段: 1 位, 取值: 0 (正常)、1 (期特低的延迟)
- 流量字段: 1 位, 取值: 0 (正常)、1 (期特高的流量)
- 可靠性字段: 1 位, 取值: 0 (正常)、1 (期特高的可靠性)
- 成本字段: 1 位, 取值: 0 (正常)、1 (期特最小成本)
- 未使用: 1 位

答:

- 5) 在验证最小帧长的时候, 选择的数据帧是目的地为实验主机的数据帧。如果选择由实验主机发出的数据帧则会发现, 帧长度可能会比 60 字节还小, 例如上图中的帧就只有 42 字节。试分析这种帧的各个域, 并解释这一现象。

答: 该帧由 14 bytes 的以太网首部 (目的 MAC、源 MAC、类型) 和 28 bytes 的 ARP 数据包组成。与收到的 60 bytes 长的 ARP 包相比, 这个包少了 18 bytes 的填充字段, 这是因为 Wireshark 捕获的发出帧是尚未封装完全的。

- 6) 上网查找资料, 看看除了 IP 和 ARP 之外, 还有哪些 IEEE 802.3 协议支持的网络层分组类型, 编码分别是什么? 列举一个。

答: 还有 PPP (0x880B), GSMP (0x880C), PPPoE (0x8863, 0x8864) 等等。

2.2 IPv6 部分 (ND 协议)

- 1) 你观察到的 ND 报文有几种类型 (Type)? 这些报文的长度是多少?

答: 报文类型的长度为 1 字节。通过观察 (见下图), ND 协议定义了 5 种 ICMPv6 报文类型, 如下表所示:

ICMPv6 类型	消息名称
Type = 133	RS(Router Solicitation, 路由器请求)
Type = 134	RA(Router Advertisement, 路由器公告)
Type = 135	NS(Neighbor Solicitation, 邻居请求)
Type = 136	RA(Neighbor Advertisement, 邻居公告)
Type = 137	Redirect(重定向报文)

```
7 0.475123 2402:f000:2:b801:cc81:8... ff02::1:ffb9:7... ICMPv6 86 Neighbor Solicitation for fe80::92
8 0.483197 fe80::9203:25ff:feb9:7f... 2402:f000:2:b8... ICMPv6 86 Neighbor Advertisement fe80::9203:
150 5.016505 fe80::9203:25ff:feb9:7f... ff02::1 ICMPv6 166 Router Advertisement from 90:03:25
187 10.034164 fe80::9203:25ff:feb9:7f... ff02::1 ICMPv6 166 Router Advertisement from 90:03:25
280 11.991095 fe80::9203:25ff:feb9:7f... ff02::1 ICMPv6 166 Router Advertisement from 90:03:25
391 15.051656 fe80::9203:25ff:feb9:7f... ff02::1 ICMPv6 166 Router Advertisement from 90:03:25
505 23.142058 fe80::9203:25ff:feb9:7f... ff02::1 ICMPv6 166 Router Advertisement from 90:03:25
518 25.087126 fe80::9203:25ff:feb9:7f... ff02::1 ICMPv6 166 Router Advertisement from 90:03:25
626 32.992038 fe80::9203:25ff:feb9:7f... ff02::1 ICMPv6 166 Router Advertisement from 90:03:25
644 37.067882 fe80::9203:25ff:feb9:7f... ff02::1 ICMPv6 166 Router Advertisement from 90:03:25
657 39.015853 fe80::9203:25ff:feb9:7f... ff02::1 ICMPv6 166 Router Advertisement from 90:03:25
722 39.581853 2402:f000:2:b801:cc81:8... 2402:f000:1:80... ICMPv6 237 Destination Unreachable (Port unreacha
825 42.487712 2402:f000:2:b801:cc81:8... 2402:f000:1:80... ICMPv6 242 Destination Unreachable (Port unreacha

> Internet Protocol Version 6, Src: fe80::9203:25ff:feb9:7f0a, Dst: ff02::1
v Internet Control Message Protocol v6
  Type: Router Advertisement (134)
  Code: 0
  Checksum: 0x19ef [correct]
  [Checksum Status: Good]
  Cur hop limit: 64
  > Flags: 0x00, Prf (Default Router Preference): Medium
  Router lifetime (s): 1800
  Reachable time (ms): 0
  Retrans timer (ms): 0
  > ICMPv6 Option (Source link-layer address : 90:03:25:b9:7f:0a)
  > ICMPv6 Option (MTU : 1500)
  > ICMPv6 Option (Prefix information : 2402:f000:2:b801::/64)
  > ICMPv6 Option (Recursive DNS Server 2402:f000:1:80::1:80::28)
```

2) 你所在的网络中, 路由器是否会周期性地发送其 IPv6 和 MAC 地址? 如果是, 发送周期大约是多少?

答: 会周期性发送, 发送周期大约为 3-5 秒左右。

3) Router Solicitation (路由器请求, RS) 和 Router Advertisement (路由器通告, RA) 报文的 IPv6 目标地址分别是多少, 它们代表什么含义? RS 和 RA 的以太网帧的 MAC 目标地址分别是多少, 它们又代表什么含义?

答:

RS:

Destination: HuaweiTe_b9:7f:0a (90:03:25:b9:7f:0a)

MAC: Apple_63:e9:50 (f4:d4:88:63:e9:50)

RA:

Destination: Apple_63:e9:50 (f4:d4:88:63:e9:50)

MAC: HuaweiTe_b9:7f:0a (90:03:25:b9:7f:0a)

4) 观察并总结 Neighbor Solicitation (邻居请求, NS) 报文的 IPv6 目标地址的特点, 查阅资料并解释 IPv6 Solicited-Node Multicast Address 如何计算。观察并总结 RS、RA 以及 NS 以太网帧的 MAC 目标地址的特点, 查阅并解释 IPv6 Multicast MAC Address 如何计算。

答: 因为所有的 Solicited-Node 多播地址的前缀都是相同的, 只有最后 24 位不同, 而最后 24 位取自单播或者任播地址的最后 24 位。并且, 按照规定, 单播地址或任播地址的后 64 位除特殊情况必须是该接口的接口标识符 (Interface Identifier, 即 Interface ID)。所以, 这 24 位实际就是取自于 64 位接口标识符的后 24 位, 也就是说, 一般情况下, 特别强调是一般情况下, 一个接口, 无论配置了多少 IPv6 地址, 这些地址对应的被请求-节点多播地址是相同的且只有一个, 所以仅仅需要加入一个多播地址。

Solicited-Node 多播地址 (被请求的节点多播地址) 是作为节点的单播地址和任播地址的函数, 通过计算得出的。Solicited-Node 多播地址按如下方法形成: 取地址 (单播或任播) 的低阶 24 位, 把这些位挂到前缀 FF02:0:0:0:1:FF00::/104 上, 产生从 FF02:0:0:0:1:FF00:0000 到 FF02:0:0:0:1:FFFF:FFFF 范围内的多播地址。例如, 对应 IPv6 地址 4037::01:800:200E:8C6C 的 Solicited-Node 多播地址是 FF02::1:FF0E:8C6C。仅高阶比特不同的 IPv6 地址 (例如, 由于与不同聚合关联的多个高阶前缀) 将映射到相同的 Solicited-Node 地址, 由此减少了节点必须加入的多播地址数目。

对于 IPv6 Multicast MAC Address, 取 IPv6 组播地址 (Multicast Address) 的最后 32bit, 再在前面加上 16bit 的 33-33, 就是 MAC 组播地址了。例如: 请求节点组播地址为: FF02:0:0:0:1:FF1E:8329 取 IPv6 组播地址的最后 32bit: FF-1E-83-29 再在前面加上 16bit 的 33-33, 就是 MAC 组播地址: 33-33-FF-1E-83-29。