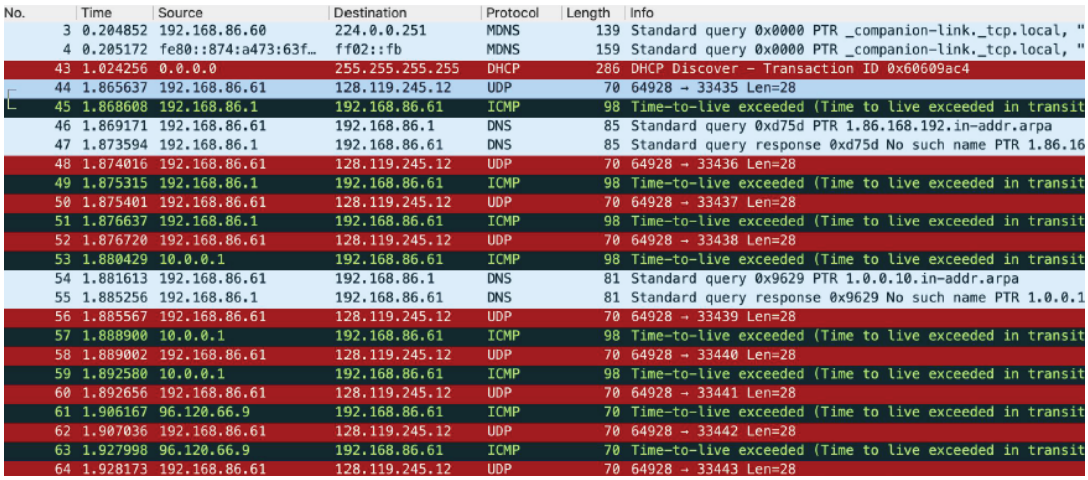
计算机学院 计算机网络 课程实验报告

实验题目：IP		学号：202200400053
日期：2024-05-17	班级： 2 班	姓名： 王宇涵
Email：1941497679@qq.com		

实验方法介绍：

使用 traceroute 进行抓包，通过 wireShark 分析数据包来理解 IP 协议的组成部分，包括基础 IPv4 协议，片段化，IPv6 三个部分.

实验过程描述：

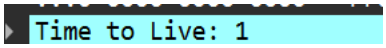


No.	Time	Source	Destination	Protocol	Length	Info
3	0.204852	192.168.86.60	224.0.0.251	MDNS	139	Standard query 0x0000 PTR _companion-link._tcp.local, "
4	0.205172	fe80::874:a473:63f…	ff02::fb	MDNS	159	Standard query 0x0000 PTR _companion-link._tcp.local, "
43	1.024256	0.0.0.0	255.255.255.255	DHCP	286	DHCP Discover - Transaction ID 0x60609ac4
44	1.865637	192.168.86.61	128.119.245.12	UDP	70	64928 → 33435 Len=28
45	1.868608	192.168.86.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit
46	1.869171	192.168.86.61	192.168.86.1	DNS	85	Standard query 0xd75d PTR 1.86.168.192.in-addr.arpa
47	1.873594	192.168.86.1	192.168.86.61	DNS	85	Standard query response 0xd75d No such name PTR 1.86.16
48	1.874016	192.168.86.61	128.119.245.12	UDP	70	64928 → 33436 Len=28
49	1.875315	192.168.86.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit
50	1.875401	192.168.86.61	128.119.245.12	UDP	70	64928 → 33437 Len=28
51	1.876637	192.168.86.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit
52	1.876720	192.168.86.61	128.119.245.12	UDP	70	64928 → 33438 Len=28
53	1.880429	10.0.0.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit
54	1.881613	192.168.86.61	192.168.86.1	DNS	81	Standard query 0x9629 PTR 1.0.0.10.in-addr.arpa
55	1.885256	192.168.86.1	192.168.86.61	DNS	81	Standard query response 0x9629 No such name PTR 1.0.0.1
56	1.885567	192.168.86.61	128.119.245.12	UDP	70	64928 → 33439 Len=28
57	1.888900	10.0.0.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit
58	1.889002	192.168.86.61	128.119.245.12	UDP	70	64928 → 33440 Len=28
59	1.892580	10.0.0.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit
60	1.892656	192.168.86.61	128.119.245.12	UDP	70	64928 → 33441 Len=28
61	1.906167	96.120.66.9	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit
62	1.907036	192.168.86.61	128.119.245.12	UDP	70	64928 → 33442 Len=28
63	1.927998	96.120.66.9	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit
64	1.928173	192.168.86.61	128.119.245.12	UDP	70	64928 → 33443 Len=28

1. 选择通过 traceroute 命令发送到 gaia.cs.umass.edu 的计算机发送的第一个 UDP 段（提示：这是 ipwireshark-trace1-1.pcapng 文件中的跟踪文件中的第 44 个数据包）。展开数据包详细信息窗口中的 Internet 协议部分。你的计算机的 IP 地址是什么？

答：192.168.86.61

2. IPv4 数据报头中 TTL 字段的值是多少？

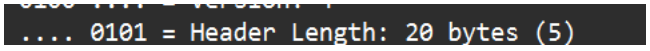


答：1

3. IPv4 数据报头中的上层协议字段的值是多少？

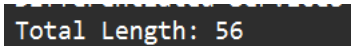
答：为 UDP(17)

4. IP 头中有多少字节？



答：20bytes

5. IP 数据报的有效载荷中有多少字节？解释一下你是如何确定有效载荷字节数的。



答：36bytes : 56 – 20 = 36bytes

6. 这个 IP 数据报是否被分片？解释一下你是如何确定数据报是否已被分片的。

```
000. .... = Flags: 0x0
   0... .... = Reserved bit: Not set
   .0.. .... = Don't fragment: Not set
   ..0. .... = More fragments: Not set
```

答：没有被分片，可以看出 Not Set

44	1.865637	192.168.86.61	128.119.245.12	UDP	70 64928 → 33435 Len=28
48	1.874016	192.168.86.61	128.119.245.12	UDP	70 64928 → 33436 Len=28
50	1.875401	192.168.86.61	128.119.245.12	UDP	70 64928 → 33437 Len=28
52	1.876720	192.168.86.61	128.119.245.12	UDP	70 64928 → 33438 Len=28
56	1.885567	192.168.86.61	128.119.245.12	UDP	70 64928 → 33439 Len=28
58	1.889002	192.168.86.61	128.119.245.12	UDP	70 64928 → 33440 Len=28
60	1.892656	192.168.86.61	128.119.245.12	UDP	70 64928 → 33441 Len=28
62	1.907036	192.168.86.61	128.119.245.12	UDP	70 64928 → 33442 Len=28
64	1.928173	192.168.86.61	128.119.245.12	UDP	70 64928 → 33443 Len=28
67	1.940279	192.168.86.61	128.119.245.12	UDP	70 64928 → 33444 Len=28
69	1.951481	192.168.86.61	128.119.245.12	UDP	70 64928 → 33445 Len=28
71	1.965335	192.168.86.61	128.119.245.12	UDP	70 64928 → 33446 Len=28

7. 在由你的计算机发送到 128.119.245.12 的 traceroute 路由器序列中，IP 数据报中的哪些字段会随着每个数据报的发送而不断变化？为什么会变化？

```
Identification: 0xfda5 (64933)
Header Checksum: 0x2ea6 [validation disabled]
Time to Live: 1
```

答：标识位，校验和，生命周期一直在变化，因为目的地端口一直在改变，网络路由器对数据报进行处理时所做的更改引起.

8. 在这个包含 UDP 段的 IP 数据报序列中，哪些字段保持不变？为什么会保持不变？

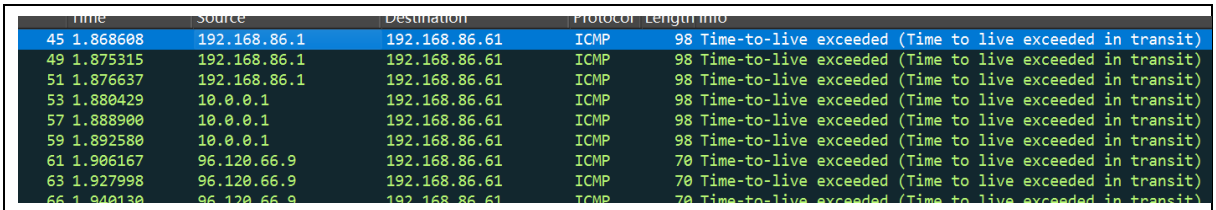
```
Total Length: 56
Version: 4
Header Length: 20 bytes (5)
Differentiated Services Field:
000. .... = Flags: 0x0
...0 0000 0000 0000 = Fragment Offset: 0
Protocol: UDP (17)
[Header checksum status: Unverified]
Source Address: 192.168.86.61
Destination Address: 128.119.245.12
```

答：总长，版本号，头文件长度，服务类型，标志位，分片偏移量，上层协议类型，源地址和目的地址均保持不变因为这些数据报具有相同的 IP 头部信息，它们在网络中的路由和处理方式也应该是相同的.

9. 描述一下你在你的计算机发送的 IP 数据报的标识字段中看到的值的模式。

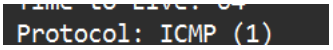
```
Identification: 0xfda1 (64929)
Identification: 0xfda2 (64930)
Identification: 0xfda3 (64931)
```

答：发现是连续且每次加 1 的.



Time	Source	Destination	Protocol	Length	Info
45 1.868608	192.168.86.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
49 1.875315	192.168.86.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
51 1.876637	192.168.86.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
53 1.880429	10.0.0.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
57 1.888900	10.0.0.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
59 1.892580	10.0.0.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
61 1.906167	96.120.66.9	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
63 1.927998	96.120.66.9	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
66 1.940130	96.120.66.9	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)

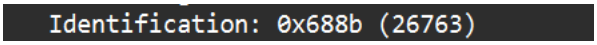
10. 从路由器返回的 IP 数据报中指定的上层协议是什么？[注意：对于 Linux/MacOS，答案与 Windows 不同]。



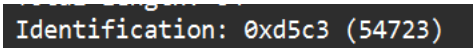
```
Protocol: ICMP (1)
```

答 ：ICMP

11. ICMP 数据包序列中的标识字段的值（跨所有路由器的所有 ICMP 数据包）在行为上是否类似于您上面对问题 9 的回答？



```
Identification: 0x688b (26763)
```

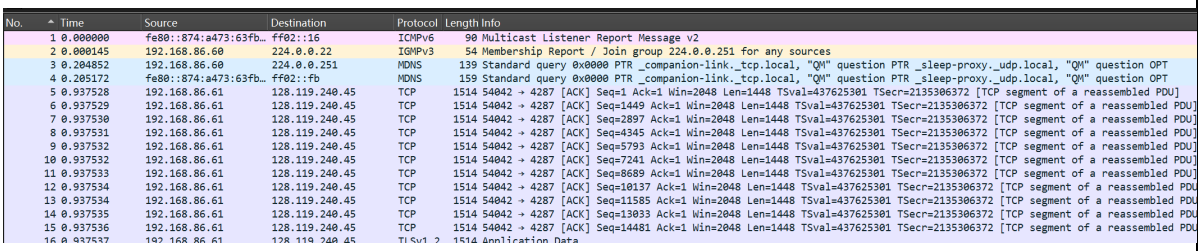


```
Identification: 0xd5c3 (54723)
```

答 ： 不类似, 对于不同路由器发送的数据包, 无法做到连续性, 对于相同路由器且数据包编号相差 1 的仍然能做到连续性.

12. 在所有路由器的所有 ICMP 数据包中，TTL 字段的值是否相似？

答 ： 不同路由器的 TTL 不同, 相同路由器的 TTL 相同.



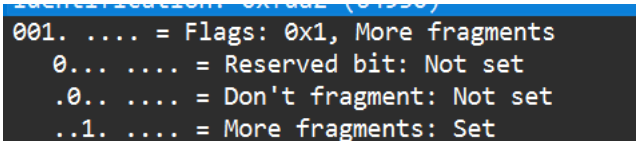
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::874:a473:63fb…	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
2	0.000145	192.168.86.60	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.251 for any sources
3	0.204852	192.168.86.60	224.0.0.251	MDNS	139	Standard query 0x0000 PTR _companion-link._tcp.local, "QM" question PTR _sleep-proxy._udp.local, "QM" question OPT
4	0.205172	fe80::874:a473:63fb…	ff02::fb	MDNS	159	Standard query 0x0000 PTR _companion-link._tcp.local, "QM" question PTR _sleep-proxy._udp.local, "QM" question OPT
5	0.937528	192.168.86.61	128.119.240.45	TCP	1514	54042 → 4287 [ACK] Seq=1 Ack=1 Win=2048 Len=1448 TSval=437625301 TSecr=2135306372 [TCP segment of a reassembled PDU]
6	0.937529	192.168.86.61	128.119.240.45	TCP	1514	54042 → 4287 [ACK] Seq=1449 Ack=1 Win=2048 Len=1448 TSval=437625301 TSecr=2135306372 [TCP segment of a reassembled PDU]
7	0.937530	192.168.86.61	128.119.240.45	TCP	1514	54042 → 4287 [ACK] Seq=2897 Ack=1 Win=2048 Len=1448 TSval=437625301 TSecr=2135306372 [TCP segment of a reassembled PDU]
8	0.937531	192.168.86.61	128.119.240.45	TCP	1514	54042 → 4287 [ACK] Seq=4345 Ack=1 Win=2048 Len=1448 TSval=437625301 TSecr=2135306372 [TCP segment of a reassembled PDU]
9	0.937532	192.168.86.61	128.119.240.45	TCP	1514	54042 → 4287 [ACK] Seq=5793 Ack=1 Win=2048 Len=1448 TSval=437625301 TSecr=2135306372 [TCP segment of a reassembled PDU]
10	0.937532	192.168.86.61	128.119.240.45	TCP	1514	54042 → 4287 [ACK] Seq=7241 Ack=1 Win=2048 Len=1448 TSval=437625301 TSecr=2135306372 [TCP segment of a reassembled PDU]
11	0.937533	192.168.86.61	128.119.240.45	TCP	1514	54042 → 4287 [ACK] Seq=8689 Ack=1 Win=2048 Len=1448 TSval=437625301 TSecr=2135306372 [TCP segment of a reassembled PDU]
12	0.937534	192.168.86.61	128.119.240.45	TCP	1514	54042 → 4287 [ACK] Seq=10137 Ack=1 Win=2048 Len=1448 TSval=437625301 TSecr=2135306372 [TCP segment of a reassembled PDU]
13	0.937534	192.168.86.61	128.119.240.45	TCP	1514	54042 → 4287 [ACK] Seq=11585 Ack=1 Win=2048 Len=1448 TSval=437625301 TSecr=2135306372 [TCP segment of a reassembled PDU]
14	0.937535	192.168.86.61	128.119.240.45	TCP	1514	54042 → 4287 [ACK] Seq=13033 Ack=1 Win=2048 Len=1448 TSval=437625301 TSecr=2135306372 [TCP segment of a reassembled PDU]
15	0.937536	192.168.86.61	128.119.240.45	TCP	1514	54042 → 4287 [ACK] Seq=14481 Ack=1 Win=2048 Len=1448 TSval=437625301 TSecr=2135306372 [TCP segment of a reassembled PDU]
16	0.937537	192.168.86.61	128.119.240.45	TLSv1.2	1514	Application Data

13. 找到由你的计算机通过 traceroute 命令发送到 gaia.cs.umass.edu 的第一个 UDP 段的第一个部分所在的第一个 IP 数据报，之后你指定 traceroute 数据包长度为 3000。（提示：这是位于脚注 2 中的 ip-wireshark-trace1-1.pcapng 跟踪文件中的第 179 个数据包。数据包 179、180 和 181 是对第一个发送到 128.119.145.12 的 3000 字节 UDP 段进行分片得到的三个 IP 数据报）。该段是否已被分片成多个 IP 数据报？（提示：答案是 yes）

179 12.788154	192.168.86.61	128.119.245.12	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=0, ID=fda2) [Reassembled in #181]

答 ：yes

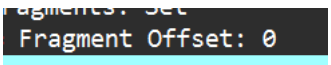
14. IP 头部中的哪些信息表明该数据报已被分片？



```
001. .... = Flags: 0x1, More fragments
   0... .... = Reserved bit: Not set
   .0.. .... = Don't fragment: Not set
   ..1. .... = More fragments: Set
```

答 ：more fragments : set.

15. IP 头部中的哪些信息表明这是第一个片段还是后续片段？



```
Fragment Offset: 0
```

```
.0 0000 1011 1001 = Fragment Offset: 1480
```

答 ：Fragment Offset 为 0 代表是第一个片段, 否则为后续片段

16. 这个 IP 数据报中有多少字节（头部加有效载荷）？

```
Total Length: 1500
```

答：1500 字节

17. 现在检查包含分片 UDP 段的第二个片段的数据报。IP 头部中的哪些信息表明这不是第一个数据报片段？

```
.0 0000 1011 1001 = Fragment Offset: 1480
```

答：此时有偏移量代表不是第一个数据包片段.

18. 在第一个和第二个片段之间 IP 头部中有哪些字段发生变化？

```
0 0000 1011 1001 = Fragment Offset: 1480
```

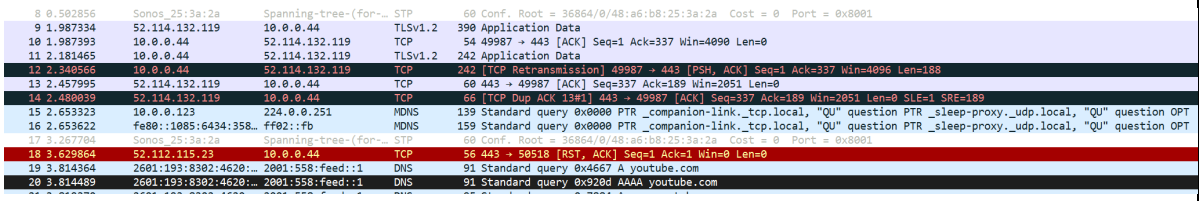
```
Header Checksum: 0x094c [validation disabled]
```

答：片段偏移量和校验和

19. 现在找到原始 UDP 段的第三个分片所在的 IP 数据报。IP 头部中的哪些信息表明这是该段的最后一个片段？

```
0... .... = Reserved bit: Not set
.0.. .... = Don't fragment: Not set
..0. .... = More fragments: Not set
```

答：More Fragments : Not set 表示是最后一个片段



20. 进行 DNS AAAA 请求的计算机的 IPv6 地址是什么？这是跟踪中的第 20 个数据包。给出该数据报的 IPv6 源地址，其格式与 Wireshark 窗口中显示的完全相同。

```
Source Address: 2601:193:8302:4620:215c:f5ae:8b40:a27a
```

答：2601:193:8302:4620:215c:f5ae:8b40:a27a

21. 该数据报的 IPv6 目的地址是什么？以与 Wireshark 窗口中显示的完全相同的形式给出该 IPv6 地址。

```
Destination Address: 2001:558:feed::1
```

答：2001:558:feed::1

22. 该数据报的流标签（Flow Label）的值是多少？

```
Flow Label: 0x63ed0
```

答：0x63ed0

23. 此数据报中携带了多少有效载荷数据？

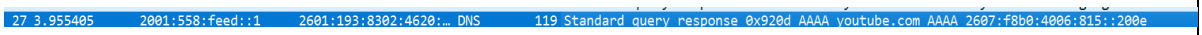
```
Payload Length: 37
```

答：37bytes

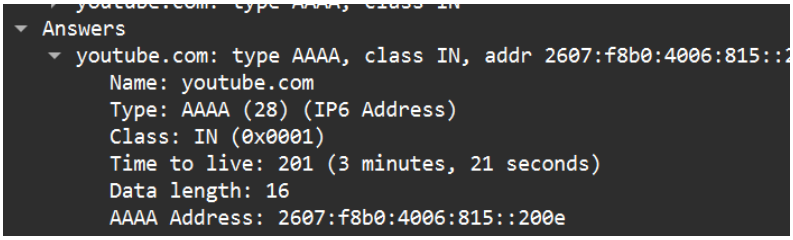
24. 此数据报的有效载荷将在目的地被交付给哪个上层协议？

```
Next Header: UDP (17)
```

答：UDP

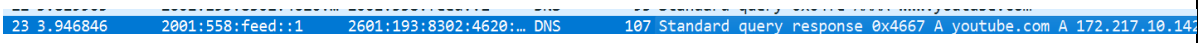


25. 对于此 AAAA 请求，DNS 响应中返回了多少个 IPv6 地址？



答：返回了 1 个 IPv6 地址

26. 对于 youtube.com 的 DNS 返回的第一个 IPv6 地址是什么（在 ip-wireshark-trace2-1.pcapng 跟踪文件中，这也是数字上最小的地址）？以与 Wireshark 窗口中显示的完全相同的简写形式给出此 IPv6 地址。



答：172.217.10.142

分析：

IPv4（Internet Protocol version 4）是互联网上广泛使用的第四版 Internet 协议。它定义了互联网上每个设备的唯一标识符，称为 IP 地址。IPv4 地址由 32 位二进制数字组成，通常被分成四个八位的数字，以点分十进制表示，如 192.0.2.1。

片段化是指在网络传输过程中，将原始数据包分割成更小的片段以适应网络链路的最大传输单元（MTU）。这种分割通常发生在发送端的路由器上，而在接收端的路由器上再次组装。片段化允许大的数据包在不同网络中传输，因为不同网络链路可能有不同的最大传输单元。

IPv6（Internet Protocol version 6）是 IPv4 的继任者，设计用来解决 IPv4 地址空间有限的问题。IPv6 使用了 128 位的地址空间，相比 IPv4 的 32 位，地址数量大大增加，几乎可以满足任何未来的互联网需求。

结论：

在本次实验中，我学到了如何使用 Wireshark 工具和 traceroute 来抓取分析网络数据包。

通过分析网络数据包，我能够了解到在网络通信过程中发生的各种事件和细节，包括数据报的分片、IP 地址的解析、上层协议的使用等。

我还学会了查看和理解 IP 头部中的各个字段，以及如何根据这些字段的值来判断数据报的特性，比如是否分片、是否是第一个分片等。

此外，我还学会了如何解析 DNS 请求和响应，以及如何查找特定的网络活动，如 DNS 查询和 ICMP 响应。

通过这些分析，我对网络通信的工作原理有了更深入的理解，也提升了我的网络分析技能。