

实验课程：计算机网络实践	姓名：李彤	学号：10235101500
实验名称：Lab03 IPV4	实验日期：2024.12.08	指导老师：王廷

实验目的

- 学会通过Wireshark分析ip协议
- 了解IP数据报的组成
- 了解IP各部分的含义

实验内容与实验步骤

- 捕获数据
- 数据分析
- 作业题
- 画出IP报文
- 画出网络路径
- 计算报文的校验和

实验环境

- Wireshark v2.0.2
- wget
- tracert (windows)

实验过程

前期调试

之前两个实验已经用过了wireshark和wget，这里就只展示一下tracert的使用。



IPV4 结构分析

Wireshark · 分组 1 · WLAN (tcp port 80)

> Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{CED
> Ethernet II, Src: d4:d8:53:f7:c4:70 (d4:d8:53:f7:c4:70), Dst: 54:c6:ff:7b:38:02 (54:c6:ff:7b:38:02):
✓ Internet Protocol Version 4, Src: 172.31.186.183, Dst: 182.61.200.6

0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 52
Identification: 0x6083 (24707)
> Flags: 0x4000, Don't fragment
Fragment offset: 0
Time to live: 128

54 c6 ff 7b 38 02 d4 d8 53 f7 c4 70 08 00 45 00
00 34 60 83 00 00 06 00 00 00 ac 1f ba b7 06 3d
c8 06 d1 d4 00 50 4b 94 2a c9 00 00 00 00 80 02
fa f0 e5 41 00 00 02 04 05 b4 01 03 03 08 01 01
04 02

TTL Protocol version DSCP
total length id Flags offset checksum src dst
T..{8... S..p..E..
..4..@.....=
...PK..*.....
...A.....
..

Time	Source	Destination	Protocol	Length	Info
1 0.000000	172.31.186.183	182.61.200.6	TCP	66	5371

无线网络适配器 WLAN:

```

连接特定的 DNS 后缀 . . . . . : fe80::c707:2591:7306:e8d2%22
本地连接 IPv6 地址 . . . . . : fe80::c707:2591:7306:e8d2%22
IPv4 地址 . . . . . : 172.31.186.183
子网掩码 . . . . . : 255.255.0.0
默认网关 . . . . . : 172.31.0.1
  
```

总长为52字节，IP报文头显示为20字节，我们点击TCP包发现其长度恰好为32字节，因此可以推断这个Total length应该是 **IP头和有效载荷** 的总和。

Internet Protocol Version 4, Src: 172.31.186.183, Dst: 182.61.200.6

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 52

Transmission Control Protocol, Src Port: 53716, Dst Port: 80, Seq: 0, Len: 0

Source Port: 53716

0000	54 c6 ff 7b 38 02 d4 d8 53 f7 c4 70 08 00 45 00	T··{8··· S··p··E·
0010	00 34 60 83 40 00 80 06 00 00 ac 1f ba b7 b6 3d	·4`·@··· ······=
0020	c8 06 d1 d4 00 50 4b 94 2a c9 00 00 00 00 80 02	····PK· *······
0030	fa f0 e5 41 00 00 02 04 05 b4 01 03 03 08 01 01	···A···· ······
0040	04 02	··

3. How does the value of the Identification field change or stay the same for different packets? For instance, does it hold the same value for all packets in a TCP connection or does it differ for each packet? Is it the same in both directions? Can you see any pattern if the value does change?

ans:

根据IPV4结构可以看出，Flag字段和Fragment Offset共用同一字段。

因此结合实验手册推测：当上层数据包过大就会被拆分，拆分得到的小包共享相同的Flag值，然后设置Offset来确定每个小包的偏移量。

所以不同的数据包的标识字段（Flag值）是不一样的，用于区分，且因为没有被拆分，所以其Offset值均为0；如果一个数据包被拆分了，那么它拆分得到的小数据包的标识字段（Flag）就是相同的，表明它们来源于相同的数据包，然后Offset值则确定了每个小数据包的相对位置，这样可以确保按照原来的顺序接受。

4. What is the initial value of the TTL field for packets sent from your computer? Is it the maximum possible value, or some lower value?

ans:

TTL的初始值是128，根据其位数可知其最大值应当为255（0xff），所以这只是一个较低值。

Time to live: 128

Protocol: TCP (6)

54 c6 ff 7b 38 02 d4 d8 53 f7 c4 70 08 00 45 00	T··{8··· S··p··E·
00 34 60 83 40 00 80 06 00 00 ac 1f ba b7 b6 3d	·4`·@··· ······=

5. How can you tell from looking at a packet that it has not been fragmented? Most often IP packets in normal operation are not fragmented. But the receiver must have a way to be sure. Hint: you may need to read your text to confirm a guess.

ans:

观察Flag和Fragment Offset字段我们可以发现一个 **Don't Fragment** 提示信息，这说明这个数据包没有被拆分。因为Flag肯定是用来区分不同数据包的，所以应该是由Fragment Offset字段来标志数据包有没有被拆分，

如果Fragment Offset的值为0，那就说明没有被拆分，否则就是被拆分了。

```
> Flags: 0x4000, Don't fragment
   Fragment offset: 0
```

6. What is the length of the IP Header and how is this encoded in the header length field? Hint: notice that only 4 bits are used for this field, as the version takes up the other 4 bits of the byte. You may guess and check your text.

ans:
wireshark显示version和Header Length共用同一个字节。这里我们可以看到Header Length的值为5（0101），但是报文头的长度却显示为20（???），思考后猜测这里应该是version占据该字节的低4位，Header Length占据高4位，所以真正的报文头长度应该是Header Length的位值再乘以4，即 $5 * 4 = 20$ ，同时我们也可以推出报文头的最大取值为60字节（ $0xf = 15$ ）。

网络路径

前面tracert的跳数太多了，所以换了一个少一点的
根据实验手册可知，在IP数据包的源IP地址和目的IP地址之间的网络路径上还有很多的IP路由器。数据包从源IP地址出发，不断在对应的网络路径上的IP路由器上跳跃，最终跳到目的IP地址上。并且每次跳跃的时候，数据包都会向目的地址发出响应。
这里我们借助tracert工具输出跳跃路径上路由器的IP地址，从而画出IP数据包的网络路径。

```
C:\Users\24916>tracert www.ecnu.edu.cn

通过最多 30 个跃点跟踪
到 www.ecnu.edu.cn [202.120.92.60] 的路由:

 1      *           *           *           请求超时。
 2      2 ms        2 ms        2 ms        10.100.4.1
 3      6 ms        2 ms        5 ms        10.10.9.2
 4     46 ms       10 ms       41 ms        10.200.102.3
 5      5 ms        8 ms        2 ms        10.200.104.1
 6      2 ms        2 ms        2 ms        202.120.92.60

跟踪完成。

C:\Users\24916>_
```

172.31.186.183（源IP地址，即本机地址）-> 10.100.4.1 -> 10.10.9.2 -> 10.200.102.3 -> 10.200.104.1 -> 202.120.92.60（目的IP地址，即www.ecnu.edu.cn的IP）

IP头校验和

因为校验的是远程服务器发送给本机地址的数据包，所以这里换了一个包

type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 182.61.200.6, Dst: 172.31.186.183

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

0000	d4 d8 53 f7 c4 70 54 c6 ff 7b 38 02 08 00 45 00	..S..pT..{8...E.
0010	00 34 60 83 40 00 2e 06 07 26 b6 3d c8 06 ac 1f	·4`·@·.··&·=....
0020	ba b7 00 50 d1 d4 c3 44 e4 cb 4b 94 2a ca 80 12	..P...D ..K*...
0030	20 00 79 a4 00 00 02 04 05 64 01 03 03 05 01 01	·y.....·d.....
0040	04 02	..

1. 将IP头两两字节一组分为10组，得到每组值分别为： 0x4500、0x0034、0x6083、0x4000、0x2e06、0x0726(checksum)、0xb63d、0xc806、0xac1f、0xbab7
2. 相加得到和为0x3fffc，再将3加到低位得到0xffff

4500 + 34 + 6083 + 4000 + 2E06 + 726 + B63D + C806 + AC1F + BAB7 +

3 FFFC

3. 因为前面求和已经加上了校验码（0x0726），所以这里只需要对0xffff取反即可，得到结果为0，校验和为0，所以校验通过，可以接受该数据包。
4. 经wireshark验证，checksum是正确的，分析正确。