

网络层数据分组的捕获和解析

实验报告

于海鑫 | 2017211305 班 | 2017211240

1 实验介绍

1.1 实验内容和实验目的

本次实验内容：

- (1). 捕获在连接Internet过程中产生的网络层分组：DHCP分组，ARP分组，IP数据分组，ICMP分组。
- (2). 分析各种分组的格式，说明各种分组在建立网络连接过程中的作用。
- (3). 分析IP数据分组分片的结构。通过本次实验了解计算机上网的工作过程，学习各种网络层分组的格式及其作用，理解长度大于1500字节IP数据组分片传输的结构。

1.2 实验重点

重点分析网络层分组的格式，掌握各种分组在网络通信中的应用，了解整个上网的工作过程。发送ICMP分组，并分析其结构和功能。制作长度大于1500字节的IP数据分组，发送并分析其分片传输的过程。

1.3 实验环境

- 操作系统 Windows 10
- 网卡型号 Realtek RTL8822BE

使用的 Wireshark 软件版本等在此不表。

2 分析网络层分组结构

2.1 DHCP 分组

通过断网重连的方式，捕获如图 1 所示的 DHCP 分组。

Wireshark 对该分组的解析见图 2。

由此可见，计算机以广播的形式发送了一个 DHCP request 信息，并在该信息中以拓展的形式(Option code: 50)请求 IP 地址 10.122.246.65。此时如果该地址未被分配，服务器会直接将该地址分配给我们，否则会重新分配新的 IP 地址。

0000	ff ff ff ff ff ff f8 da 0c 59 9f 81 08 00 45 00Y....E.
0010	01 5e ba 2b 00 00 80 11 7f 64 00 00 00 00 ff ff	..^+....d.....
0020	ff ff 00 44 00 43 01 4a 1c ba 01 01 06 00 4d aa	...D.C.J.....M.
0030	7f f3 00 00 80 00 00 00 00 00 00 00 00 00 00
0040	00 00 00 00 00 00 f8 da 0c 59 9f 81 00 00 00 00Y.....
0050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00a0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00b0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00c0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00d0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00e0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00f0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0110	00 00 00 00 00 00 63 82 53 63 35 01 03 3d 07 01c Sc5...=..
0120	f8 da 0c 59 9f 81 32 04 0a 7a f6 41 0c 0f 44 45	...Y...2. z.A...DE
0130	53 4b 54 4f 50 2d 44 38 42 49 51 33 55 51 12 00	SKTOP-D8 BIQ3UQ..
0140	00 00 44 45 53 4b 54 4f 50 2d 44 38 42 49 51 33	..DESKTO P-D8BIQ3
0150	55 3c 08 4d 53 46 54 20 35 2e 30 37 0e 01 03 06	U<MSFT 5.07....
0160	0f 1f 21 2b 2c 2e 2f 77 79 f9 fc ff	..!+,.w y...

图 1: DHCP 分组

```

Dynamic Host Configuration Protocol (Request)
  Message type: Boot Request (1)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0x4daa7ff3
  Seconds elapsed: 0
  Bootp flags: 0x8000, Broadcast flag (Broadcast)
    1... .... = Broadcast flag: Broadcast
    .000 0000 0000 0000 = Reserved flags: 0x0000
  Client IP address: 0.0.0.0
  Your (client) IP address: 0.0.0.0
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address: HonHaiPr_59:9f:81 (f8:da:0c:59:9f:81)
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  > Option: (53) DHCP Message Type (Request)
  > Option: (61) Client identifier
  > Option: (50) Requested IP Address (10.122.246.65)
  > Option: (12) Host Name
  > Option: (81) Client Fully Qualified Domain Name
  > Option: (60) Vendor class identifier
  > Option: (55) Parameter Request List
  > Option: (255) End

```

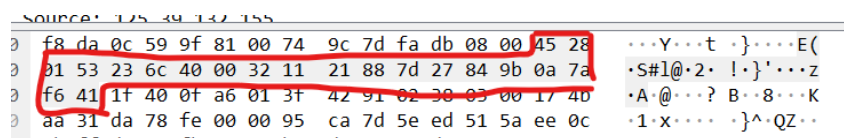
图 2: Wireshark 对于 DHCP 分组的解析

2.1.1 上网流程

当我们的计算机连接上 **BUPT-portal** 热点时，首先会广播 **DHCP request** 包，当服务器回应 IP 地址后，即可正常上网。

2.2 IP 数据分组

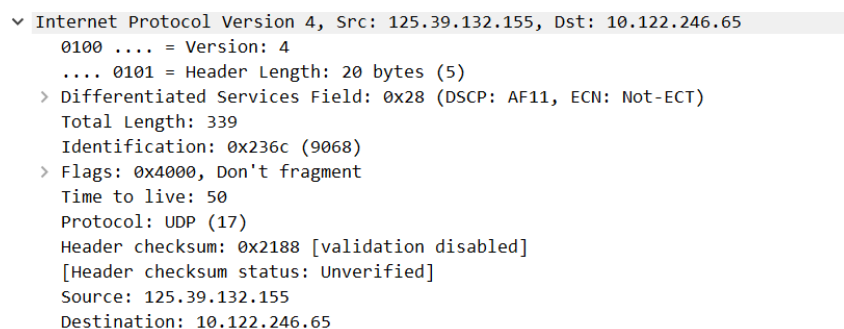
任选一包，其 IP 包头数据见图 3。



Offset	Hex	ASCII
0	f8 da 0c 59 9f 81 00 74	...
8	9c 7d fa db 08 00 45 28	...Y...t...}...E(
16	01 53 23 6c 40 00 32 11	...S#l@.2.!.}...'...z
24	f6 41 1f 40 0f a6 01 3f	...A.@...? B...8...K
32	aa 31 da 78 fe 00 00 95	...1.x...}^..QZ..

图 3: IP 包头

Wireshark 对其分析见图 4。

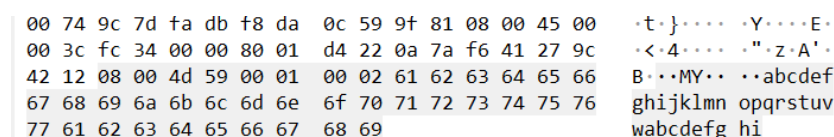


Field	Value
Internet Protocol Version 4, Src: 125.39.132.155, Dst: 10.122.246.65	
Version: 4	0100 = Version: 4
Header Length: 20 bytes (5) 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x28 (DSCP: AF11, ECN: Not-ECT)	> Differentiated Services Field: 0x28 (DSCP: AF11, ECN: Not-ECT)
Total Length: 339	Total Length: 339
Identification: 0x236c (9068)	Identification: 0x236c (9068)
Flags: 0x4000, Don't fragment	> Flags: 0x4000, Don't fragment
Time to live: 50	Time to live: 50
Protocol: UDP (17)	Protocol: UDP (17)
Header checksum: 0x2188 [validation disabled]	Header checksum: 0x2188 [validation disabled]
[Header checksum status: Unverified]	[Header checksum status: Unverified]
Source: 125.39.132.155	Source: 125.39.132.155
Destination: 10.122.246.65	Destination: 10.122.246.65

图 4: IP 包头的解析结果

2.3 ICMP 数据分组

使用 **ping** 指令发送 ICMP 分组，数据见图 5。



Offset	Hex	ASCII
0	00 74 9c 7d fa db f8 da	...t...}...Y...E..
8	0c 59 9f 81 08 00 45 00	...<.4... ..".z.A'..
16	00 3c fc 34 00 00 80 01	B...MY... ..abcdef
24	42 12 08 00 4d 59 00 01	ghijklmn opqrstuv
32	00 02 61 62 63 64 65 66	wabcedfg hi

图 5: ICMP 分组

Wireshark 对其的解码结果见图 6。

```

Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0x4d59 [correct]
  [Checksum Status: Good]
  Identifier (BE): 1 (0x0001)
  Identifier (LE): 256 (0x0100)
  Sequence number (BE): 2 (0x0002)
  Sequence number (LE): 512 (0x0200)
  [Response frame: 72]
Data (32 bytes)
  Data: 6162636465666768696a6b6c6d6e6f707172737475767761...
  [Length: 32]

```

图 6: ICMP 包的解析结果

2.4 分片的 IP 数据分组

通过“ping -l 8000 10.3.9.5”指令造出一个长度为 8000 字节的分组并使用 Wireshark 捕获，共捕获到如图 7 所示的分组。

1306	47.387976	10.122.246.65	10.3.9.5	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=613d) [Reassembled in #1311]
1307	47.387977	10.122.246.65	10.3.9.5	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=613d) [Reassembled in #1311]
1308	47.387978	10.122.246.65	10.3.9.5	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=613d) [Reassembled in #1311]
1309	47.387978	10.122.246.65	10.3.9.5	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=4440, ID=613d) [Reassembled in #1311]
1310	47.387978	10.122.246.65	10.3.9.5	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=5920, ID=613d) [Reassembled in #1311]
1311	47.387979	10.122.246.65	10.3.9.5	ICMP	642	Echo (ping) request ID=000001, seq=5/1280, ttl=128 (reply in 1312)

图 7: 分片的 IP 包

现在给出其中一个分片的解析结果，见图片 8。

```

Internet Protocol Version 4, Src: 10.122.246.65, Dst: 10.3.9.5
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 1500
  Identification: 0x613d (24893)
  Flags: 0x222b, More fragments
    0... .. = Reserved bit: Not set
    .0.. .. = Don't fragment: Not set
    .1. .... = More fragments: Set
    ...0 0010 0010 1011 = Fragment offset: 555
  Time to live: 128
  Protocol: ICMP (1)
  Header checksum: 0x9df5 [validation disabled]
  [Header checksum status: Unverified]
  Source: 10.122.246.65
  Destination: 10.3.9.5
  Reassembled IPv4 in frame: 1311

```

图 8: 解析结果

几个分组的 IP 头部分只有校验和和偏移量部分不同，其余部分完全一致。在最后一个分组内，“More fregments”部分被置为 0，表示不再有新的分片。全套分组大小为 8000 个字节的 IP 数据再加上 8 字节的 ICMP 部分，共 8008 字节。

3 结论与心得

该实验共花费了大约两个小时的时间来完成。并未发现任何问题。

通过该实验，我对各种数据包的格式有了更深的理解，并对 IP 的分片机制有了新的认知。