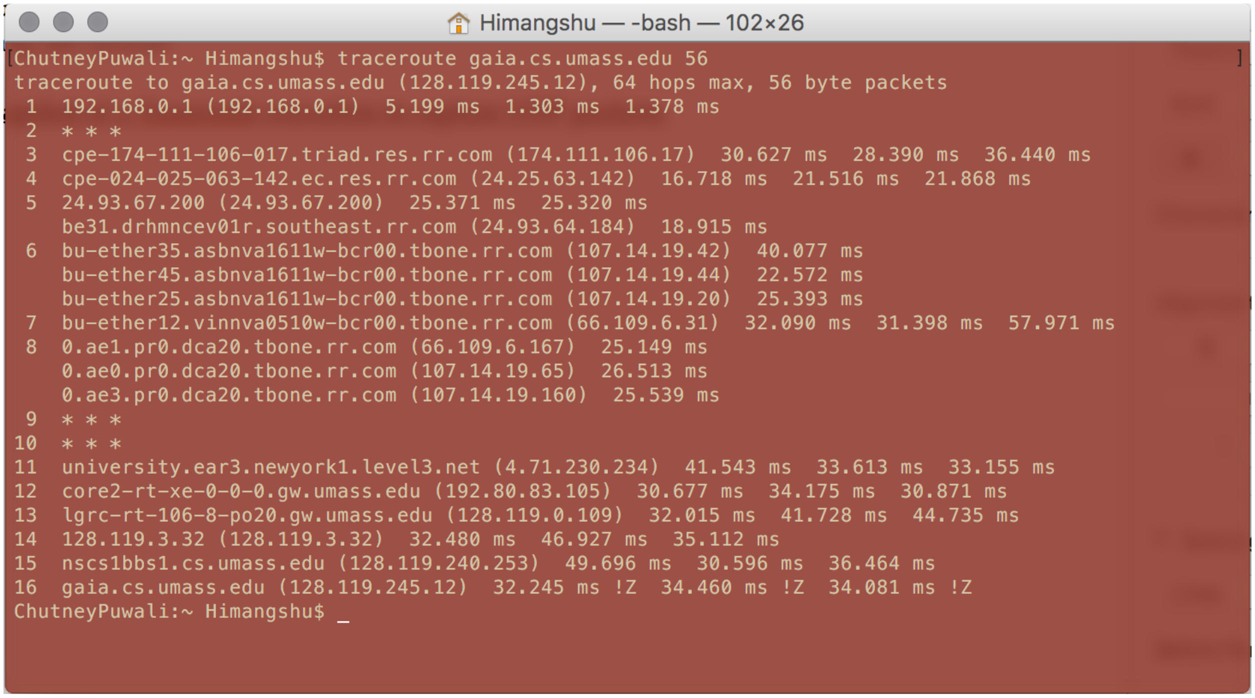
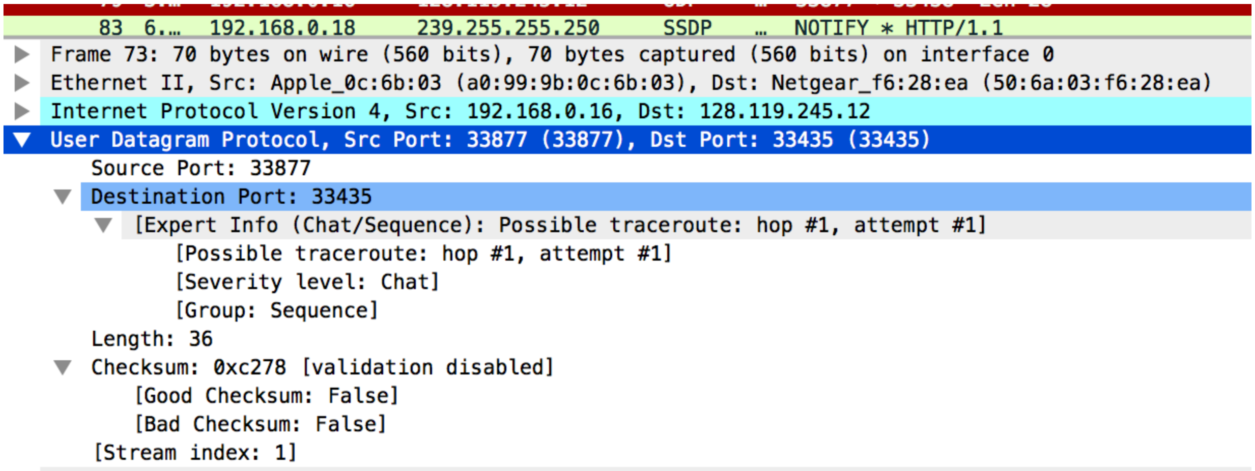
Course: Computer Networks(ECE/CSC 570)
Instructor: Mihail L. Sichitiu
Description: Spring 2016, Wireshark Assignment 5 Solutions.
Student Name: Himangshu Ranjan Borah
Student ID: 200105222
Unity ID: hborah

Snapshot of a Traceroute execution to capture UDP packets:



```
ChutneyPuwali:~ Himangshu$ traceroute gaia.cs.umass.edu 56
traceroute to gaia.cs.umass.edu (128.119.245.12), 64 hops max, 56 byte packets
 1  192.168.0.1 (192.168.0.1)  5.199 ms  1.303 ms  1.378 ms
 2  * * *
 3  cpe-174-111-106-017.triad.res.rr.com (174.111.106.17)  30.627 ms  28.390 ms  36.440 ms
 4  cpe-024-025-063-142.ec.res.rr.com (24.25.63.142)  16.718 ms  21.516 ms  21.868 ms
 5  24.93.67.200 (24.93.67.200)  25.371 ms  25.320 ms
    be31.drhmncev01r.southeast.rr.com (24.93.64.184)  18.915 ms
 6  bu-ether35.asbnva1611w-bcr00.tbone.rr.com (107.14.19.42)  40.077 ms
    bu-ether45.asbnva1611w-bcr00.tbone.rr.com (107.14.19.44)  22.572 ms
    bu-ether25.asbnva1611w-bcr00.tbone.rr.com (107.14.19.20)  25.393 ms
 7  bu-ether12.vinnva0510w-bcr00.tbone.rr.com (66.109.6.31)  32.090 ms  31.398 ms  57.971 ms
 8  0.ae1.pr0.dca20.tbone.rr.com (66.109.6.167)  25.149 ms
    0.ae0.pr0.dca20.tbone.rr.com (107.14.19.65)  26.513 ms
    0.ae3.pr0.dca20.tbone.rr.com (107.14.19.160)  25.539 ms
 9  * * *
10  * * *
11  university.ear3.newyork1.level3.net (4.71.230.234)  41.543 ms  33.613 ms  33.155 ms
12  core2-rt-xe-0-0-0.gw.umass.edu (192.80.83.105)  30.677 ms  34.175 ms  30.871 ms
13  lgrc-rt-106-8-po20.gw.umass.edu (128.119.0.109)  32.015 ms  41.728 ms  44.735 ms
14  128.119.3.32 (128.119.3.32)  32.480 ms  46.927 ms  35.112 ms
15  nscs1bbs1.cs.umass.edu (128.119.240.253)  49.696 ms  30.596 ms  36.464 ms
16  gaia.cs.umass.edu (128.119.245.12)  32.245 ms !Z  34.460 ms !Z  34.081 ms !Z
ChutneyPuwali:~ Himangshu$ _
```

Answer No 1:

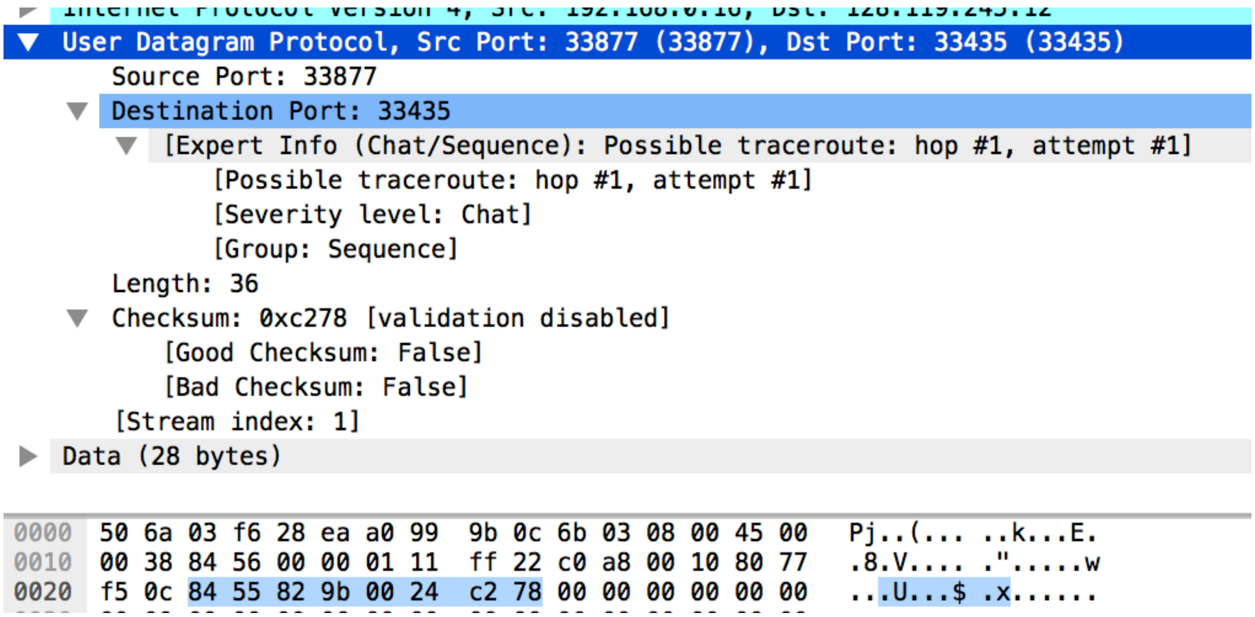


```
     83  6.…  192.168.0.18      239.255.255.250     SSDP    …  NOTIFY * HTTP/1.1
▶ Frame 73: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
▶ Ethernet II, Src: Apple_0c:6b:03 (a0:99:9b:0c:6b:03), Dst: Netgear_f6:28:ea (50:6a:03:f6:28:ea)
▶ Internet Protocol Version 4, Src: 192.168.0.16, Dst: 128.119.245.12
▼ User Datagram Protocol, Src Port: 33877 (33877), Dst Port: 33435 (33435)
     Source Port: 33877
   ▼ Destination Port: 33435
      ▼ [Expert Info (Chat/Sequence): Possible traceroute: hop #1, attempt #1]
           [Possible traceroute: hop #1, attempt #1]
           [Severity level: Chat]
           [Group: Sequence]
     Length: 36
   ▼ Checksum: 0xc278 [validation disabled]
        [Good Checksum: False]
        [Bad Checksum: False]
     [Stream index: 1]
```

From the snapshot above, we can see that the UDP packets has four fields namely

1. Source Port
2. Destination Port
3. Length
4. Checksum

Answer No 2:



```
▶ Internet Protocol Version 4, Src: 192.168.0.16, Dst: 128.119.245.12
▼ User Datagram Protocol, Src Port: 33877 (33877), Dst Port: 33435 (33435)
      Source Port: 33877
    ▼ Destination Port: 33435
        ▼ [Expert Info (Chat/Sequence): Possible traceroute: hop #1, attempt #1]
              [Possible traceroute: hop #1, attempt #1]
              [Severity level: Chat]
              [Group: Sequence]
      Length: 36
    ▼ Checksum: 0xc278 [validation disabled]
          [Good Checksum: False]
          [Bad Checksum: False]
      [Stream index: 1]
▶ Data (28 bytes)

0000  50 6a 03 f6 28 ea a0 99  9b 0c 6b 03 08 00 45 00   Pj..(... ..k...E.
0010  00 38 84 56 00 00 01 11  ff 22 c0 a8 00 10 80 77   .8.V.... .".....w
0020  f5 0c 84 55 82 9b 00 24  c2 78 00 00 00 00 00 00   ...U...$ .x......
```

From the packet contents field, we see that the UDP header is Total **8 Bytes** Long.

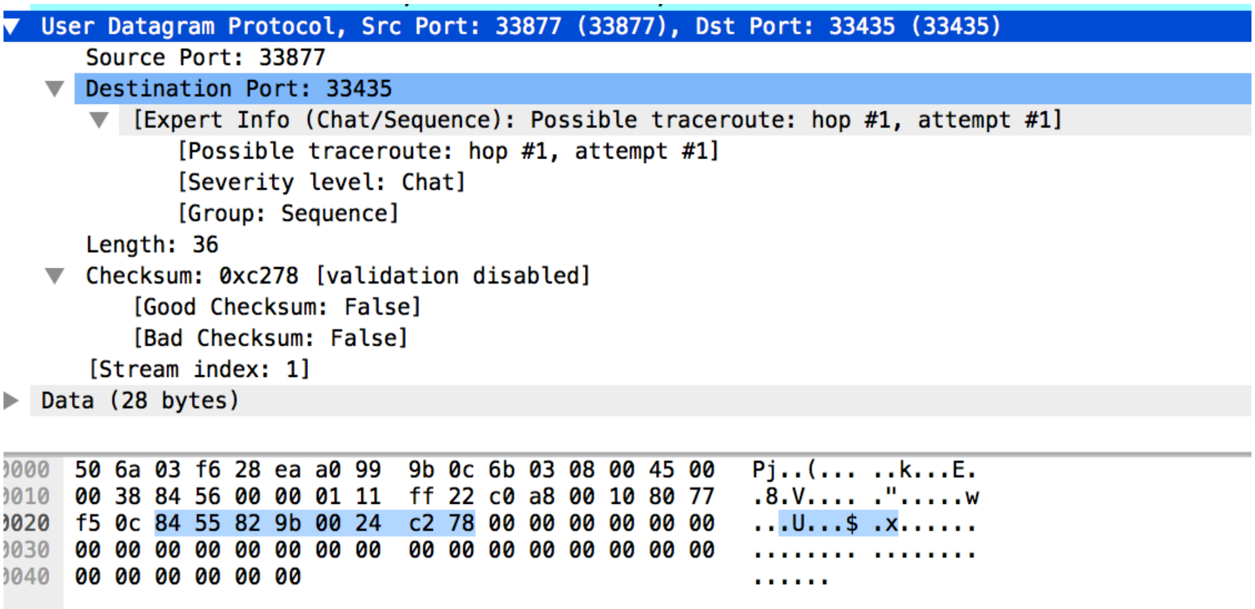
1. Source Port : **2 Bytes**
2. Destination Port : **2 Bytes**
3. Length : **2 Bytes**
4. Checksum : **2 Bytes**

Answer No 3:

The value of the Length Field in the UDP header in the Length of the Total packet at transport layer. That includes the Header size of UDP header and the Payload for the Application later, i.e. the data. In our case we have,

UDP Header : 8 bytes
Payload Data : 28 Bytes

Hence Total = 28 + 8 = **36 bytes** as seen in the image below.



```
▼ User Datagram Protocol, Src Port: 33877 (33877), Dst Port: 33435 (33435)
      Source Port: 33877
   ▼ Destination Port: 33435
      ▼ [Expert Info (Chat/Sequence): Possible traceroute: hop #1, attempt #1]
            [Possible traceroute: hop #1, attempt #1]
            [Severity level: Chat]
            [Group: Sequence]
      Length: 36
   ▼ Checksum: 0xc278 [validation disabled]
         [Good Checksum: False]
         [Bad Checksum: False]
      [Stream index: 1]
▶ Data (28 bytes)

0000  50 6a 03 f6 28 ea a0 99  9b 0c 6b 03 08 00 45 00   Pj..(... ..k...E.
0010  00 38 84 56 00 00 01 11  ff 22 c0 a8 00 10 80 77   .8.V.... .".....w
0020  f5 0c 84 55 82 9b 00 24  c2 78 00 00 00 00 00 00   ...U...$ .x......
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0040  00 00 00 00 00 00                                  ......
```

Answer No 4:

The maximum no. of length that can be represented by the 2 byte long Length Field in the header of UDP is = $2^{16}-1=65535$
In this Length, 8 bytes are always reserved for the UDP header, so the effective payload size is = 65535 - 8 = **65527 Bytes**

Answer No 5:

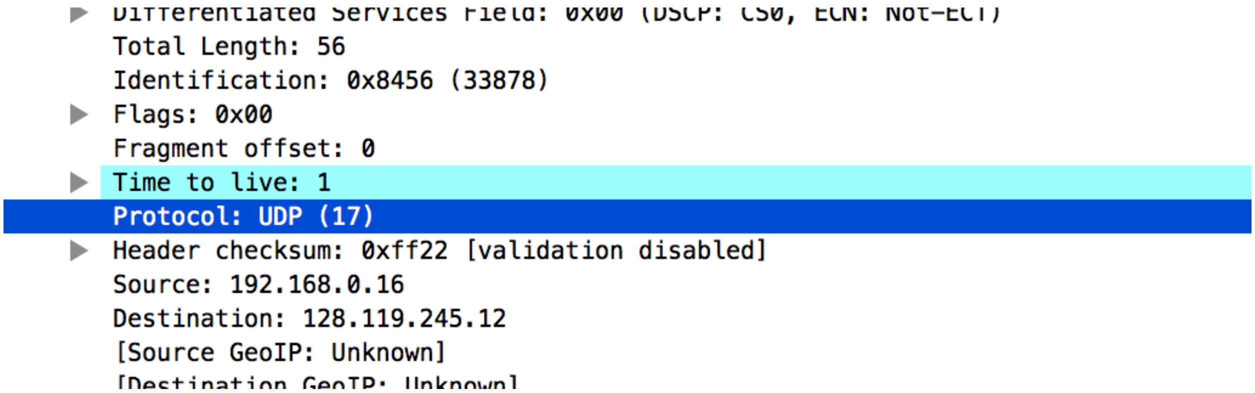
The Source Port Field is having 2 bytes received to be represented(Similar to the Length field above).
Hence, the total source ports that can be supported is $2^{16}-1=65535$
Thus, the largest possible no. for a source port = **65535**

Answer No 6:

The Value of the UDP Protocol is seen in the "Protocol" field of the corresponding IP Header.

The Decimal value of UDP protocol No: **17**
The Hex value of UDP protocol No: **0x11**

The Snapshot below shown:



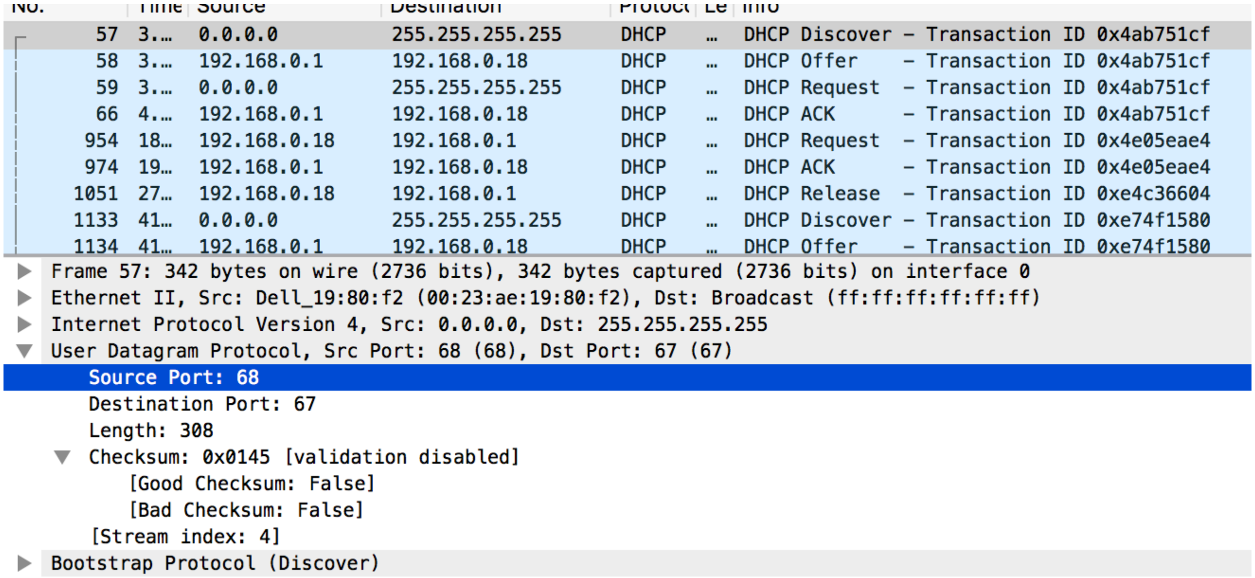
```
▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 56
  Identification: 0x8456 (33878)
▶ Flags: 0x00
  Fragment offset: 0
▶ Time to live: 1
  Protocol: UDP (17)
▶ Header checksum: 0xff22 [validation disabled]
  Source: 192.168.0.16
  Destination: 128.119.245.12
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
```

Answer No 7:

The Checksum field is calculated as the 16 bit one's complement of the one's complement sum of a *pseudo header of information from the IP header, the UDP header, and the data,* padded with zero octets at the end (if necessary) to make a multiple of two bytes. It's set to 0xffff if calculated to be zero.
[Source - wikipedia]

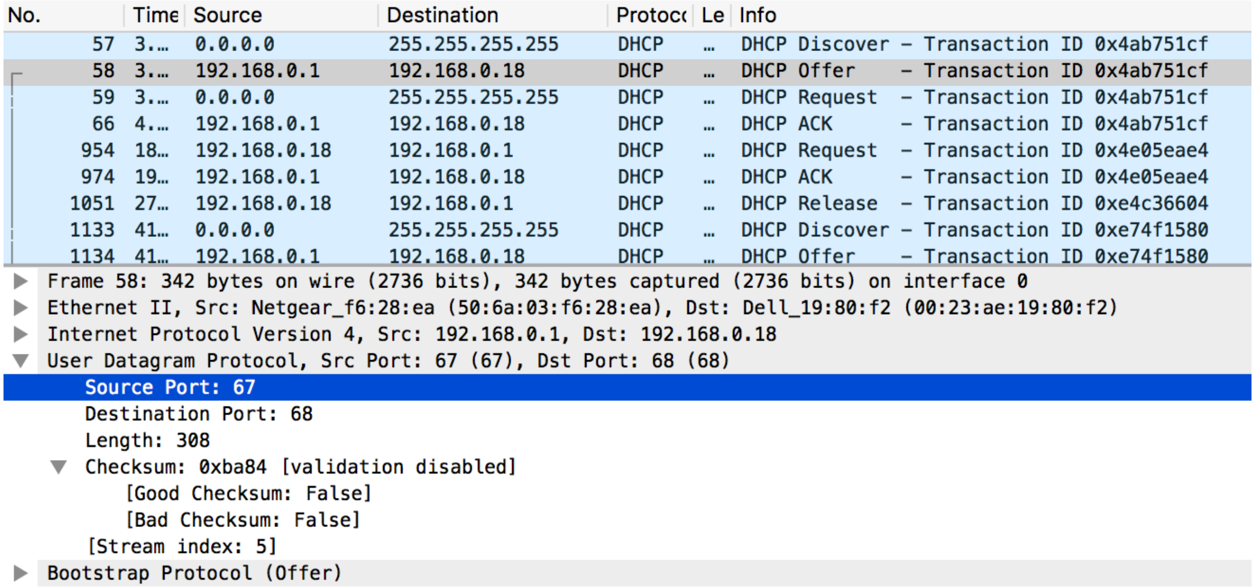
Answer No 8:

DHCP Discover Message Over UDP:



```
No.     Time Source         Destination      Protoco Le Info
    57  3.…  0.0.0.0        255.255.255.255  DHCP    …  DHCP Discover - Transaction ID 0x4ab751cf
    58  3.…  192.168.0.1    192.168.0.18     DHCP    …  DHCP Offer    - Transaction ID 0x4ab751cf
    59  3.…  0.0.0.0        255.255.255.255  DHCP    …  DHCP Request  - Transaction ID 0x4ab751cf
    66  4.…  192.168.0.1    192.168.0.18     DHCP    …  DHCP ACK      - Transaction ID 0x4ab751cf
   954  18…  192.168.0.18   192.168.0.1      DHCP    …  DHCP Request  - Transaction ID 0x4e05eae4
   974  19…  192.168.0.1    192.168.0.18     DHCP    …  DHCP ACK      - Transaction ID 0x4e05eae4
  1051  27…  192.168.0.18   192.168.0.1      DHCP    …  DHCP Release  - Transaction ID 0xe4c36604
  1133  41…  0.0.0.0        255.255.255.255  DHCP    …  DHCP Discover - Transaction ID 0xe74f1580
  1134  41…  192.168.0.1    192.168.0.18     DHCP    …  DHCP Offer    - Transaction ID 0xe74f1580
▶ Frame 57: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface 0
▶ Ethernet II, Src: Dell_19:80:f2 (00:23:ae:19:80:f2), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▶ Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
▼ User Datagram Protocol, Src Port: 68 (68), Dst Port: 67 (67)
     Source Port: 68
     Destination Port: 67
     Length: 308
   ▼ Checksum: 0x0145 [validation disabled]
        [Good Checksum: False]
        [Bad Checksum: False]
     [Stream index: 4]
▶ Bootstrap Protocol (Discover)
```

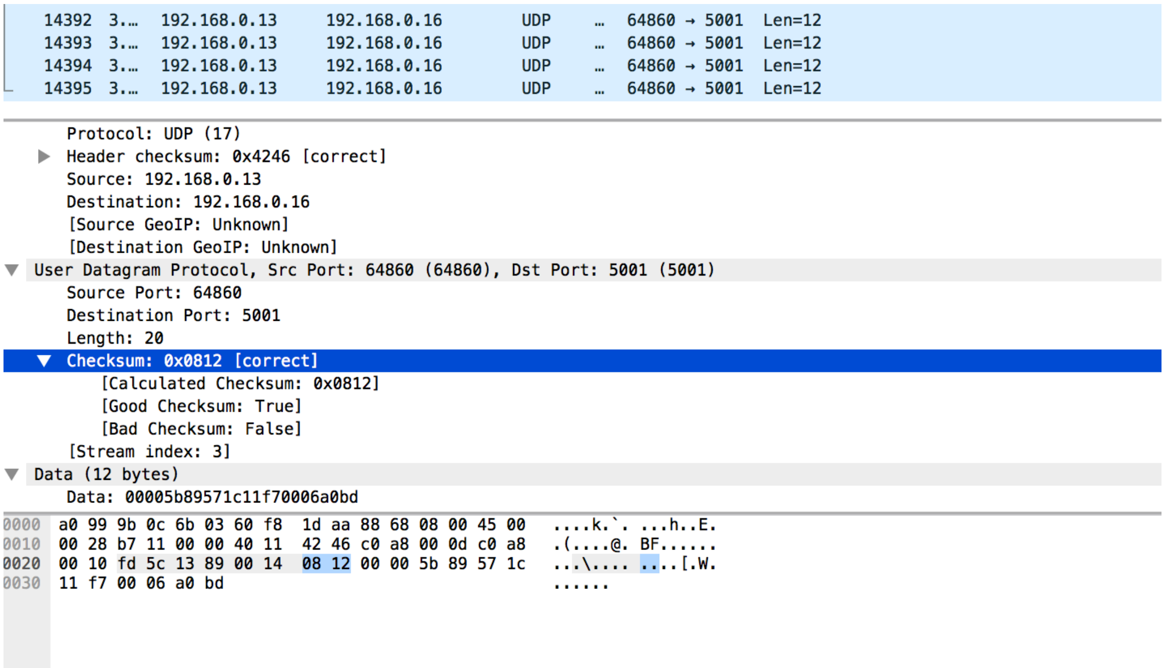
DHCP Offer to the above Discover Over UDP:



```
No.         Time  Source          Destination      Protocc Le  Info
       57   3.…   0.0.0.0         255.255.255.255  DHCP    …   DHCP Discover – Transaction ID 0x4ab751cf
       58   3.…   192.168.0.1     192.168.0.18     DHCP    …   DHCP Offer    – Transaction ID 0x4ab751cf
       59   3.…   0.0.0.0         255.255.255.255  DHCP    …   DHCP Request  – Transaction ID 0x4ab751cf
       66   4.…   192.168.0.1     192.168.0.18     DHCP    …   DHCP ACK      – Transaction ID 0x4ab751cf
      954   18…   192.168.0.18    192.168.0.1      DHCP    …   DHCP Request  – Transaction ID 0x4e05eae4
      974   19…   192.168.0.1     192.168.0.18     DHCP    …   DHCP ACK      – Transaction ID 0x4e05eae4
     1051   27…   192.168.0.18    192.168.0.1      DHCP    …   DHCP Release  – Transaction ID 0xe4c36604
     1133   41…   0.0.0.0         255.255.255.255  DHCP    …   DHCP Discover – Transaction ID 0xe74f1580
     1134   41…   192.168.0.1     192.168.0.18     DHCP    …   DHCP Offer    – Transaction ID 0xe74f1580
▶ Frame 58: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface 0
▶ Ethernet II, Src: Netgear_f6:28:ea (50:6a:03:f6:28:ea), Dst: Dell_19:80:f2 (00:23:ae:19:80:f2)
▶ Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.18
▼ User Datagram Protocol, Src Port: 67 (67), Dst Port: 68 (68)
      Source Port: 67
      Destination Port: 68
      Length: 308
   ▼ Checksum: 0xba84 [validation disabled]
         [Good Checksum: False]
         [Bad Checksum: False]
      [Stream index: 5]
▶ Bootstrap Protocol (Offer)
```

In the above pair of Discover and Offer packets of DHCP sent over UDP, we can see that the **Source Port of the sent packet is the Destination Port of the Reply packet and vice versa** in the UDP Headers. This shows how transport layer routes the packets to appropriate programs running on the application layer.

Answer To the Bonus Question:
The Packet Received(Experimented using IPERF in Mac):



```
   14392  3.…  192.168.0.13    192.168.0.16     UDP    …   64860 → 5001  Len=12
   14393  3.…  192.168.0.13    192.168.0.16     UDP    …   64860 → 5001  Len=12
   14394  3.…  192.168.0.13    192.168.0.16     UDP    …   64860 → 5001  Len=12
   14395  3.…  192.168.0.13    192.168.0.16     UDP    …   64860 → 5001  Len=12

      Protocol: UDP (17)
   ▶ Header checksum: 0x4246 [correct]
      Source: 192.168.0.13
      Destination: 192.168.0.16
      [Source GeoIP: Unknown]
      [Destination GeoIP: Unknown]
▼ User Datagram Protocol, Src Port: 64860 (64860), Dst Port: 5001 (5001)
      Source Port: 64860
      Destination Port: 5001
      Length: 20
   ▼ Checksum: 0x0812 [correct]
         [Calculated Checksum: 0x0812]
         [Good Checksum: True]
         [Bad Checksum: False]
      [Stream index: 3]
▼ Data (12 bytes)
      Data: 00005b89571c11f70006a0bd

0000  a0 99 9b 0c 6b 03 60 f8  1d aa 88 68 08 00 45 00   ....k.`. ...h..E.
0010  00 28 b7 11 00 00 40 11  42 46 c0 a8 00 0d c0 a8   .(....@. BF......
0020  00 10 fd 5c 13 89 00 14  08 12 00 00 5b 89 57 1c   ...\.... ....[.W.
0030  11 f7 00 06 a0 bd                                  ......
```

We Have the following fields for the checksum calculation(in 16 bit splits, IN HEX):

SOURCE IP PART 1 : C0 A8
SOURCE IP PART 2 : 00 0D
DESTINATION IP PART 1 : C0 A8
DESTINATION IP PART 2 : 00 10
UDP PACKET LENGTH : 00 14
UDP PROTOCL NO : 00 11
SOURCE PORT : FD 5C
DEST PORT : 13 89
LENGTH : 00 14
DATA 1 : 00 00
DATA 2 : 5B 89
DATA 3 : 57 1C
DATA 4 : 11 F7
DATA 5 : 00 06
DATA 6 : A0 BD

Adding up all of the above, we get = 3F7EA
Carry = 3
Adding 3 to F7EA WE GET = F7EA + 3 = F7ED
In binary, F7ED is = 1111 0111 1110 1101
Now, one's compliment of F7ED = 0000 1000 0001 0010
Which is = **0812 in Hex.**

In the snapshot above, we see that the value of the checksum field is = **0812**
So, we see that the calculation says that checksum is correct, so does Wireshark.