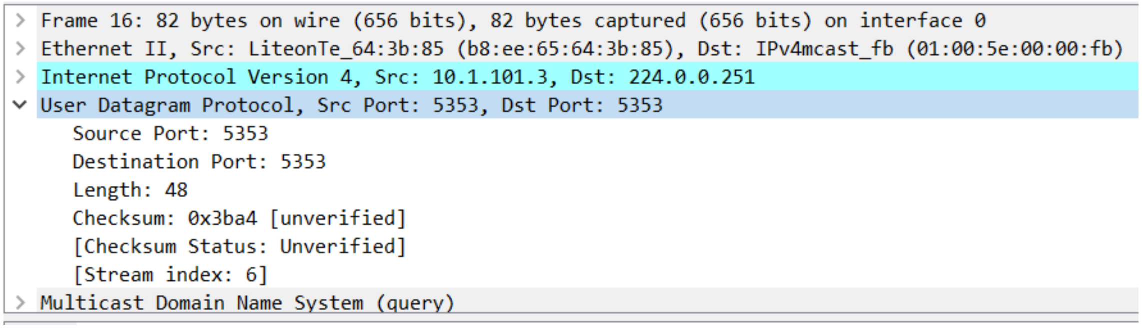
Wireshark Lab #2A: UDP

Suyi Liu, Yuan Jing Vincent Yan

February 22, 2017

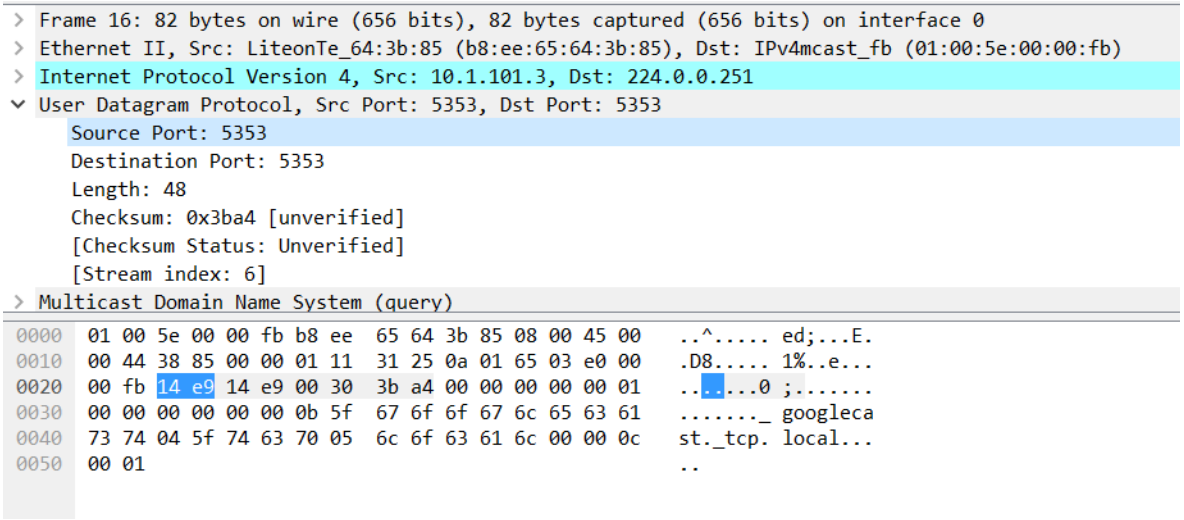
Problem 1



```
> Frame 16: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0
> Ethernet II, Src: LiteonTe_64:3b:85 (b8:ee:65:64:3b:85), Dst: IPv4mcast_fb (01:00:5e:00:00:fb)
> Internet Protocol Version 4, Src: 10.1.101.3, Dst: 224.0.0.251
v User Datagram Protocol, Src Port: 5353, Dst Port: 5353
     Source Port: 5353
     Destination Port: 5353
     Length: 48
     Checksum: 0x3ba4 [unverified]
     [Checksum Status: Unverified]
     [Stream index: 6]
> Multicast Domain Name System (query)
```

There are 6 fields. These fields are Source Port, Destination Port, Length, CheckSum, CheckSum Status, and Stream Index. Even though Checksum Status and Stream Index have length 0, we will count them here for now.

Problem 2

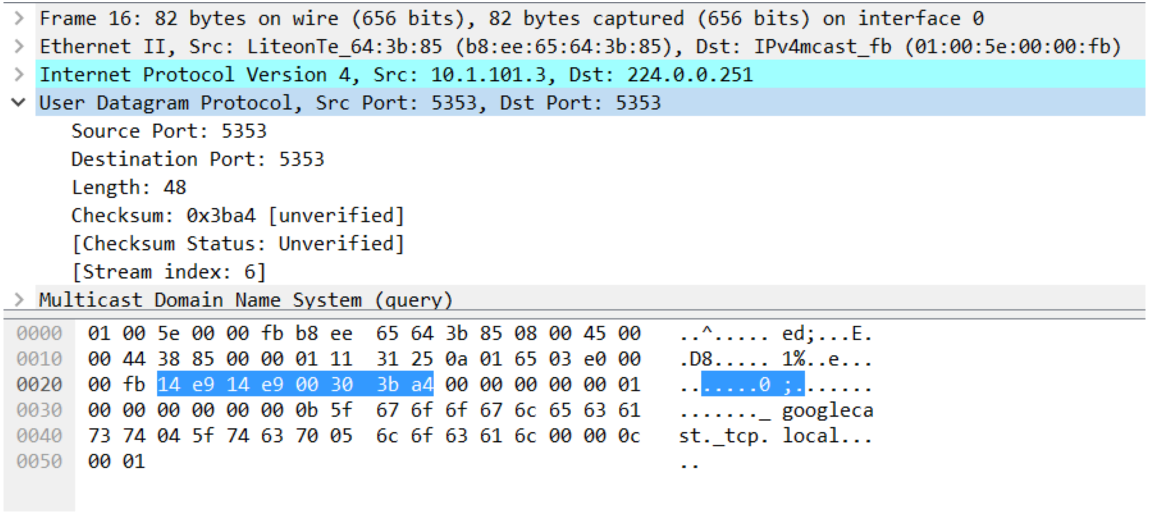


```
> Frame 16: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0
> Ethernet II, Src: LiteonTe_64:3b:85 (b8:ee:65:64:3b:85), Dst: IPv4mcast_fb (01:00:5e:00:00:fb)
> Internet Protocol Version 4, Src: 10.1.101.3, Dst: 224.0.0.251
v User Datagram Protocol, Src Port: 5353, Dst Port: 5353
     Source Port: 5353
     Destination Port: 5353
     Length: 48
     Checksum: 0x3ba4 [unverified]
     [Checksum Status: Unverified]
     [Stream index: 6]
> Multicast Domain Name System (query)

0000  01 00 5e 00 00 fb b8 ee  65 64 3b 85 08 00 45 00   ..^..... ed;...E.
0010  00 44 38 85 00 00 01 11  31 25 0a 01 65 03 e0 00   .D8..... 1%..e...
0020  00 fb 14 e9 14 e9 00 30  3b a4 00 00 00 00 00 01   .......0 ;.......
0030  00 00 00 00 00 00 0b 5f  67 6f 6f 67 6c 65 63 61   ......._ googleca
0040  73 74 04 5f 74 63 70 05  6c 6f 63 61 6c 00 00 0c   st._tcp. local...
0050  00 01                                              ..
```

The length of each UDP header field is 2 bytes (except for Checksum Status

and Stream Index).
For example, the Source Port header is 2 bytes in the screenshot shown above.

Problem 3



```
> Frame 16: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0
> Ethernet II, Src: LiteonTe_64:3b:85 (b8:ee:65:64:3b:85), Dst: IPv4mcast_fb (01:00:5e:00:00:fb)
> Internet Protocol Version 4, Src: 10.1.101.3, Dst: 224.0.0.251
v User Datagram Protocol, Src Port: 5353, Dst Port: 5353
     Source Port: 5353
     Destination Port: 5353
     Length: 48
     Checksum: 0x3ba4 [unverified]
     [Checksum Status: Unverified]
     [Stream index: 6]
> Multicast Domain Name System (query)

0000  01 00 5e 00 00 fb b8 ee  65 64 3b 85 08 00 45 00   ..^..... ed;...E.
0010  00 44 38 85 00 00 01 11  31 25 0a 01 65 03 e0 00   .D8..... 1%..e...
0020  00 fb 14 e9 14 e9 00 30  3b a4 00 00 00 00 00 01   .......0 ;.......
0030  00 00 00 00 00 00 0b 5f  67 6f 6f 67 6c 65 63 61   ......._ googleca
0040  73 74 04 5f 74 63 70 05  6c 6f 63 61 6c 00 00 0c   st._tcp. local...
0050  00 01                                              ..
```

The length field specifies the number of bytes in the UDP segment(header plus data).
As shown in the screenshot, the number of bytes in the UDP segment is indeed 48 (starting from ”14 e9 14 e9. . . ” as highlighted above).

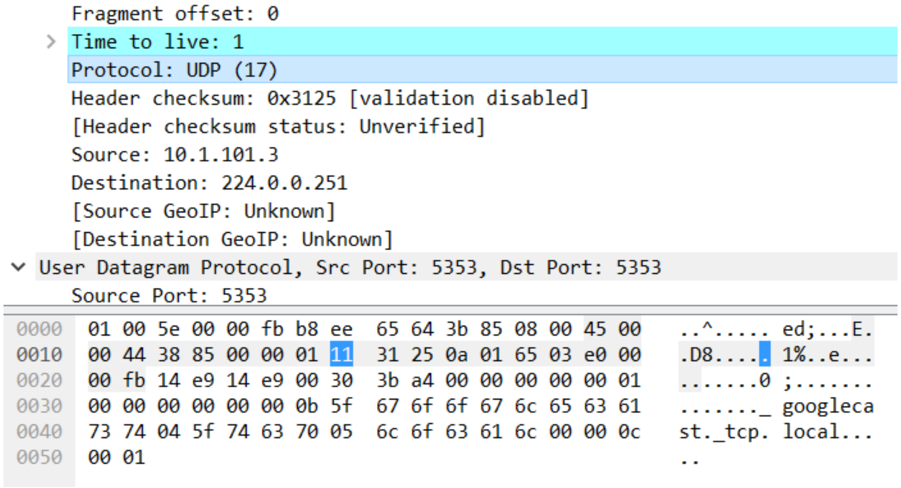
Problem 4

The maximum number of bytes that can be included in a UDP payload is 65527 bytes, because there are four fields in the UDP header, each with length of 2 bits, $65535 - 2 * 4 = 65527$ bytes.

Problem 5

65535. Since the length of Source Port field is 2 bytes, which equals 16 bits, the largest number represented using 16 bits is $2^{16} - 1 = 65535$.

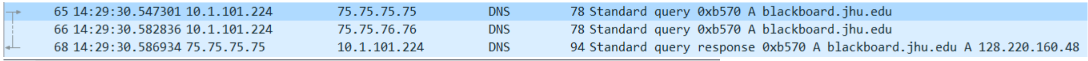
Problem 6



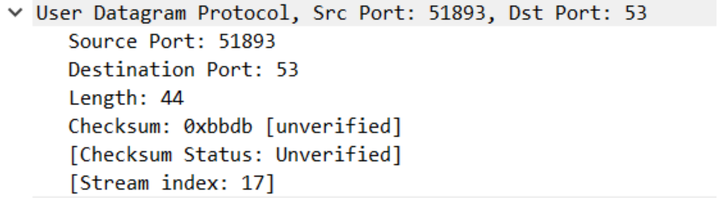
```
    Fragment offset: 0
  > Time to live: 1
    Protocol: UDP (17)
    Header checksum: 0x3125 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.1.101.3
    Destination: 224.0.0.251
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
v User Datagram Protocol, Src Port: 5353, Dst Port: 5353
    Source Port: 5353

0000  01 00 5e 00 00 fb b8 ee  65 64 3b 85 08 00 45 00   ..^..... ed;...E.
0010  00 44 38 85 00 00 01 11  31 25 0a 01 65 03 e0 00   .D8..... 1%..e...
0020  00 fb 14 e9 14 e9 00 30  3b a4 00 00 00 00 00 01   .......0 ;.......
0030  00 00 00 00 00 00 0b 5f  67 6f 6f 67 6c 65 63 61   ......._ googleca
0040  73 74 04 5f 74 63 70 05  6c 6f 63 61 6c 00 00 0c   st._tcp. local...
0050  00 01                                              ..
```

The protocol number for UDP is 0x11 in hexadecimal notation, and 17 in decimal notation.

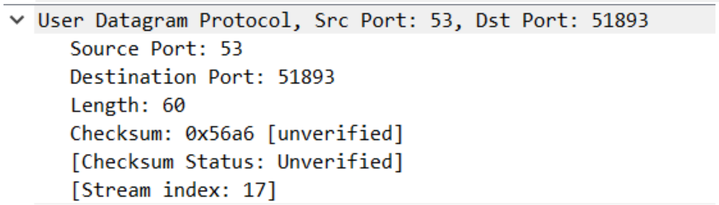
Problem 7



```
65 14:29:30.547301 10.1.101.224    75.75.75.75     DNS    78 Standard query 0xb570 A blackboard.jhu.edu
66 14:29:30.582836 10.1.101.224    75.75.76.76     DNS    78 Standard query 0xb570 A blackboard.jhu.edu
68 14:29:30.586934 75.75.75.75     10.1.101.224    DNS    94 Standard query response 0xb570 A blackboard.jhu.edu A 128.220.160.48
```



```
v User Datagram Protocol, Src Port: 51893, Dst Port: 53
    Source Port: 51893
    Destination Port: 53
    Length: 44
    Checksum: 0xbbdb [unverified]
    [Checksum Status: Unverified]
    [Stream index: 17]
```



```
v User Datagram Protocol, Src Port: 53, Dst Port: 51893
    Source Port: 53
    Destination Port: 51893
    Length: 60
    Checksum: 0x56a6 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 17]
```

As shown in the screenshot, the Source Port Number(51893) of the first packet becomes the Destination Port Number of the second packet. And the Destination Port Number(53) of the first packet becomes the Source Port Number of the second packet.