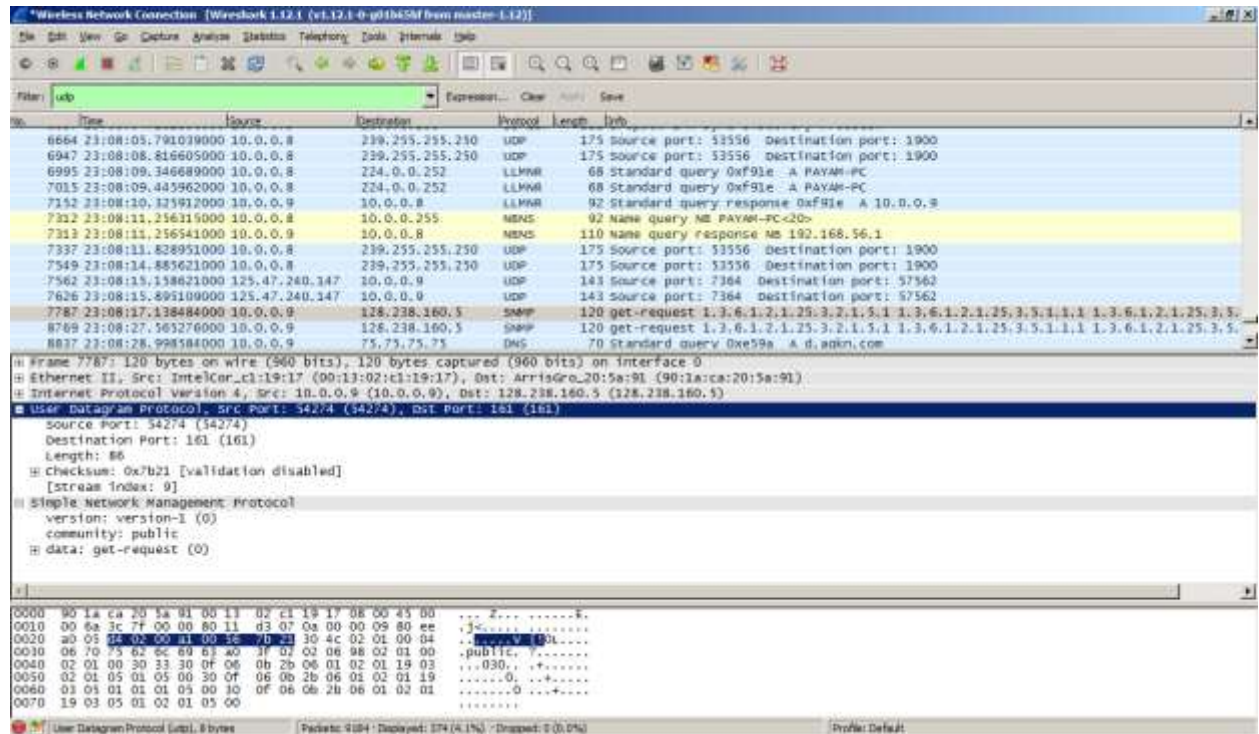


1. Select one UDP packet from your trace. From this packet, determine how many fields there are in the UDP header. Name these fields.



Answer 1: There are 4 fields in UDP header.

- a. Source Port
- b. Destination Port
- c. Length
- d. Checksum

2. By consulting the displayed information in Wireshark's packet content field for this packet, determine the length (in bytes) of each of the UDP header fields.

Answer 2: Total header length 8 bytes (each field is 2 bytes)

- Source port : 16 bit or 2 bytes

```
Source Port: 54274 (54274)
Destination Port: 161 (161)
Length: 86
Checksum: 0x7b21 [validation disabled]
[Good Checksum: False]
[Bad Checksum: False]
[Stream index: 9]
Simple Network Management Protocol
version: version-1 (0)
community: public
data: get-request (0)
```

0 90 1a ca 20 5a 91 00 13 02 c1 19 08 00 45 00

0 00 6a 3c 7f 00 00 80 11 d3 07 0a 00 00 09 80 ee

0 a0 05 d4 02 00 a1 00 1e 70 21 30 4c 02 01 00 04

0 06 70 75 62 6c 69 63 a0 3f 02 06 98 02 01 00

0 02 01 00 30 12 0f 06 0b 2b 06 01 02 01 19

0 05 01 05 01 05 00 30 0f 06 0b 2b 06 01 02 01 19

0 01 01 01 01 01 05 00 30 0f 06 0b 2b 06 01 02 01 19

0 19 03 05 01 02 01 05 00 0f 06 0b 2b 06 01 02 01 19

- Destination port : 16 bit or 2 bytes

```

Destination Port: 161 (161)
Length: 86
[+] Checksum: 0x7b21 [validation disabled]
    [Good Checksum: False]
    [Bad Checksum: False]
    [Stream index: 9]
Simple Network Management Protocol
  version: version-1 (0)
  community: public
[+] data: get-request (0)

```

00	90	1a	ca	20	5a	91	00	13	02	c1	19	17
10	00	6a	3c	7f	00	00	80	11	d3	07	0a	00
20	a0	05	d4	02	00	a1	00	56	7b	21	30	4c

- Length: 16 bit or 2 bytes

```

Destination Port: 161 (161)
Length: 86
[+] Checksum: 0x7b21 [validation disabled]
    [Good Checksum: False]
    [Bad Checksum: False]
    [Stream index: 9]
Simple Network Management Protocol
  version: version-1 (0)
  community: public
[+] data: get-request (0)

```

00	90	1a	ca	20	5a	91	00	13	02	c1	19	17
10	00	6a	3c	7f	00	00	80	11	d3	07	0a	00
20	a0	05	d4	02	00	a1	00	56	7b	21	30	4c
30	06	70	75	62	6c	69	63	a0	3f	02	02	06

- Checksum: 16 bit or 2 bytes

```

Length: 86
[+] Checksum: 0x7b21 [validation disabled]
    [Good Checksum: False]
    [Bad Checksum: False]
    [Stream index: 9]
Simple Network Management Protocol
  version: version-1 (0)
  community: public
[+] data: get-request (0)

```

00	90	1a	ca	20	5a	91	00	13	02	c1	19	17
10	00	6a	3c	7f	00	00	80	11	d3	07	0a	00
20	a0	05	d4	02	00	a1	00	56	7b	21	30	4c
30	06	70	75	62	6c	69	63	a0	3f	02	02	06

3. The value in the Length field is the length of what? (You can consult the text for this answer).
Verify your claim with your captured UDP packet.

```

Length: 86
Checksum: 0x7b21 [validation disabled]
  [Good Checksum: False]
  [Bad Checksum: False]
  [Stream index: 9]
Simple Network Management Protocol

```

Answer 3: A length specifies the length in bytes of the UDP header and UDP data

4. What is the maximum number of bytes that can be included in a UDP payload? (Hint: the answer to this question can be determined by your answer to 2. above)

```

Length: 86
Checksum: 0x7b21 [validation disabled]
  [Good Checksum: False]
  [Bad Checksum: False]
  [Stream index: 9]
Simple Network Management Protocol

```

Answer 4: The field size sets a theoretical limit of 65,535 bytes (8 byte header + 65,527 bytes of data) for a UDP datagram (as length of "length field is 16 bit hence total 2^{16} bytes can be identified). Hence the theoretical UDP payload limit is 65527 bytes.

5. What is the largest possible source port number? (Hint: see the hint in 4.)

```

Source Port: 54274 (54274)
Destination Port: 161 (161)
Length: 86
Checksum: 0x7b21 [validation disabled]
  [Good Checksum: False]
  [Bad Checksum: False]
  [Stream index: 9]
Simple Network Management Protocol
  version: version-1 (0)
  community: public
  data: get-request (0)

```

00	90	1a	ca	20	5a	91	00	13	02	c1	19	17	08	00	45	00
10	00	6a	3c	7f	00	00	80	11	d3	07	0a	00	00	09	80	ee
20	a0	05	d4	02	00	a1	00	56	7b	21	30	4c	02	01	00	04

Answer 5: a 16 bit integer value, allowing for port numbers between 0 and 65535
Hence the largest port number is 65535.

WireShark Lab – UDP

Payam Rastogi
(pr1228@nyu.edu)

6. What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notation. To answer this question, you'll need to look into the Protocol field of the IP datagram containing this UDP segment (see Figure 4.13 in the text, and the discussion of IP header fields).

7787	23:08:17.138484000	10.0.0.9	128.238.160.5	SNMP	120	get
8769	23:08:27.565276000	10.0.0.9	128.238.160.5	SNMP	120	get
8837	23:08:28.998584000	10.0.0.9	75.75.75.75	DNS	70	Sta

Frame 7787: 120 bytes on wire (960 bits), 120 bytes captured (960 bits) on interface 0
Ethernet II, Src: IntelCor_c1:19:17 (00:13:02:c1:19:17), Dst: ArrisGro_20:5a:91 (90:1a:ca:20:5a:91)
Internet Protocol Version 4, Src: 10.0.0.9 (10.0.0.9), Dst: 128.238.160.5 (128.238.160.5)
Version: 4
Header Length: 20 bytes
+ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT)
Total Length: 106
Identification: 0x3c7f (15487)
+ Flags: 0x00
Fragment offset: 0
Time to live: 128
Protocol: UDP (17)
+ Header checksum: 0xd307 [validation disabled]
Source: 10.0.0.9 (10.0.0.9)
Destination: 128.238.160.5 (128.238.160.5)

00	90	1a	ca	20	5a	91	00	13	02	c1	19	17	08	00	45	00	...	Z...E.
10	00	6a	3c	7f	00	00	80	11	d3	07	0a	00	00	09	80	ee	.	j<....
20	a0	05	d4	02	00	a1	00	56	7b	21	30	4c	02	01	00	04	V	!0L....

Answer 6: Protocol number for UDP is 17 (hexadecimal: 11)

7. Examine a pair of UDP packets in which your host sends the first UDP packet and the second UDP packet is a reply to this first UDP packet. (Hint: for a second packet to be sent in response to a first packet, the sender of the first packet should be the destination of the second packet). Describe the relationship between the port numbers in the two packets.

2	00:50:38.381287000	10.0.0.9	75.75.75.75	DNS	81	Standard query 0xc4ea AAAA beacon-6.newrelic.com
4	00:50:38.400189000	75.75.75.75	10.0.0.9	DNS	189	Standard query response 0xc4ea CNAME beacon-6.newrelic.com
23	00:50:38.616684000	10.0.0.9	128.238.160.5	SNMP	120	get-request 1.3.6.1.2.1.25.3.2.1.5.1 1.3.6.1.2.1.25.3.2.1.5.1
28	00:50:39.879942000	10.0.0.2	239.255.255.250	UDP	175	Source port: 49564 Destination port: 1900
51	00:50:42.867464000	10.0.0.2	239.255.255.250	UDP	175	Source port: 49564 Destination port: 1900
86	00:50:45.879957000	10.0.0.2	239.255.255.250	UDP	175	Source port: 49564 Destination port: 1900

Frame 2: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0
Ethernet II, Src: IntelCor_c1:19:17 (00:13:02:c1:19:17), Dst: ArrisGro_20:5a:91 (90:1a:ca:20:5a:91)
Internet Protocol Version 4, Src: 10.0.0.9 (10.0.0.9), Dst: 75.75.75.75 (75.75.75.75)
User Datagram Protocol, Src Port: 59036 (59036), Dst Port: 53 (53)
Source Port: 59036 (59036)
Destination Port: 53 (53)

Answer 7: In UDP packet from 10.0.0.9 -> 75.75.75.75

Source port: 59036

Destination port: 53

WireShark Lab – UDP

Payam Rastogi
(pr1228@nyu.edu)

2	00:50:38.381287000	10.0.0.9	75.75.75.75	DNS	81 Standard que
4	00:50:38.400169000	75.75.75.75	10.0.0.9	DNS	189 Standard que
23	00:50:38.616684000	10.0.0.9	128.238.160.5	SNMP	120 get-request
28	00:50:39.879942000	10.0.0.2	239.255.255.250	UDP	175 Source port:
51	00:50:42.867464000	10.0.0.2	239.255.255.250	UDP	175 Source port:
86	00:50:45.879957000	10.0.0.2	239.255.255.250	UDP	175 Source port:

⊕	Frame 4: 189 bytes on wire (1512 bits), 189 bytes captured (1512 bits) on interface 0
⊕	Ethernet II, Src: ArrisGro_20:5a:91 (90:1a:ca:20:5a:91), Dst: IntelCor_c1:19:17 (00:13
⊕	Internet Protocol Version 4, Src: 75.75.75.75 (75.75.75.75), Dst: 10.0.0.9 (10.0.0.9)
⊖	User Datagram Protocol, Src Port: 53 (53), Dst Port: 59036 (59036)
	Source Port: 53 (53)
	Destination Port: 59036 (59036)
	. . .

In UDP Packet from 75.75.75.75 -> 10.0.0.9

Source Port: 53

Destination Port: 59036

The source port no. in the UDP packet from the client to server becomes destination port no. in the UDP packet from server to client.