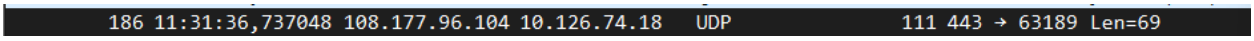
WIRESHARK LAB 6

Part 1: Basic

1. Select the first UDP segment sent by your computer via the `traceroute` command to gaia.cs.umass.edu. (Hint: this is 44th packet in the trace file in the *ip-wireshark-trace1-1.pcapng* file in footnote 2). Expand the Internet Protocol part of the packet in the packet details window. What is the IP address of your computer?

Answer
108.177.96.104

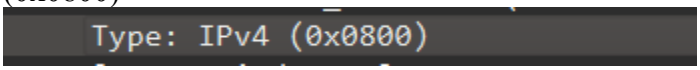


2. What is the value in the time-to-live (TTL) field in this IPv4 datagram's header?

Answer
56

3. What is the value in the upper layer protocol field in this IPv4 datagram's header? [Note: the answers for Linux/MacOS differ from Windows here].

Answer
(0x0800)

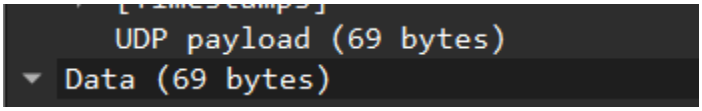


4. How many bytes are in the IP header?

Answer
20 Bytes

5. How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.

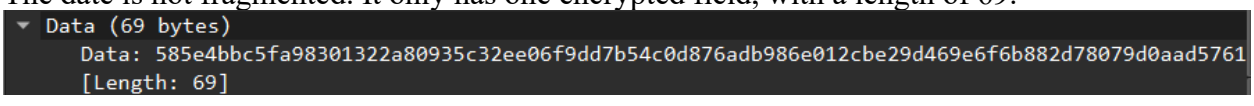
Answer



I check the Data Header, and see the length of the field. In this example the field is 69 in length, meaning it will take up 69 bytes

6. Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.

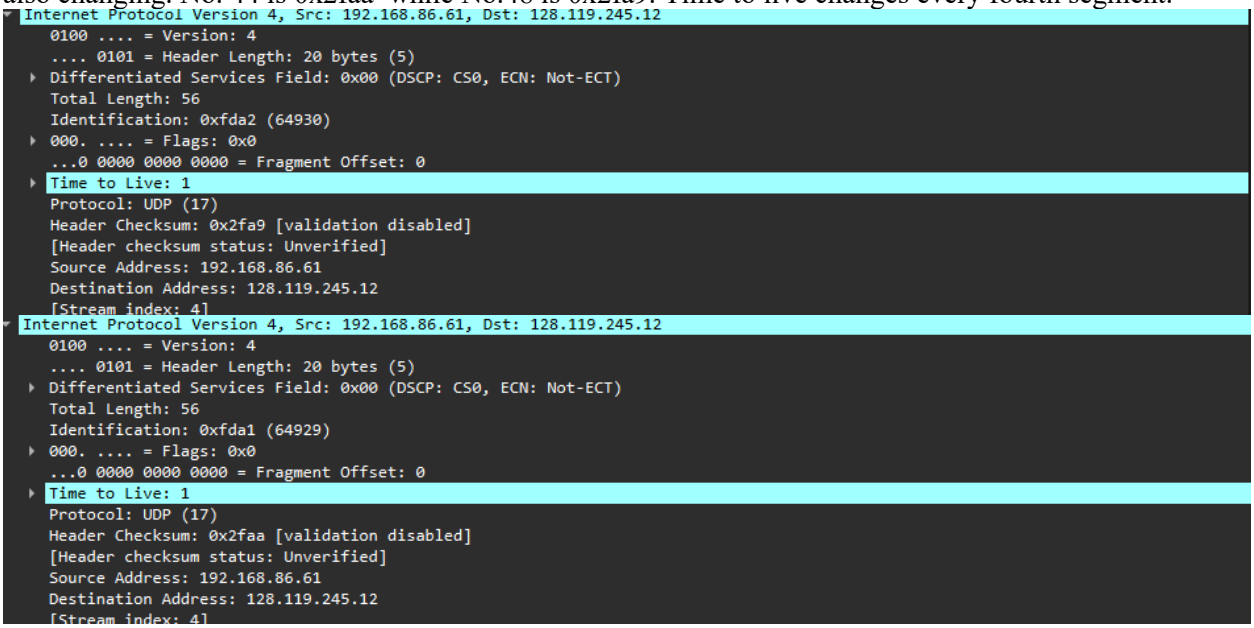
Answer
The date is not fragmented. It only has one encrypted field, with a length of 69.



7. Which fields in the IP datagram *always* change from one datagram to the next within this series of UDP segments sent by your computer destined to 128.119.245.12, via `traceroute`? Why?

Answer

The identification field is always changing between the series of UDP segments. No. 44 segment, has identification 0xfda1, while No.48 segment has identification 0xfda2. The header checksum is also changing. No. 44 is 0x2faa while No.48 is 0x2fa9. Time to live changes every fourth segment.



8. Which fields in this sequence of IP datagrams (containing UDP segments) stay constant? Why?

Answer

IP Version, Header Length, Type of Service, Protocol, Source IP Address, Destination IP Address, and Total Length and Flags/Fragment Offset

These fields describe the structure and endpoints of communication, which don't change from one probe to the next, only control-related values (TTL, ID, checksum, ports) vary.

9. Describe the pattern you see in the values in the Identification field of the IP datagrams being sent by your computer.

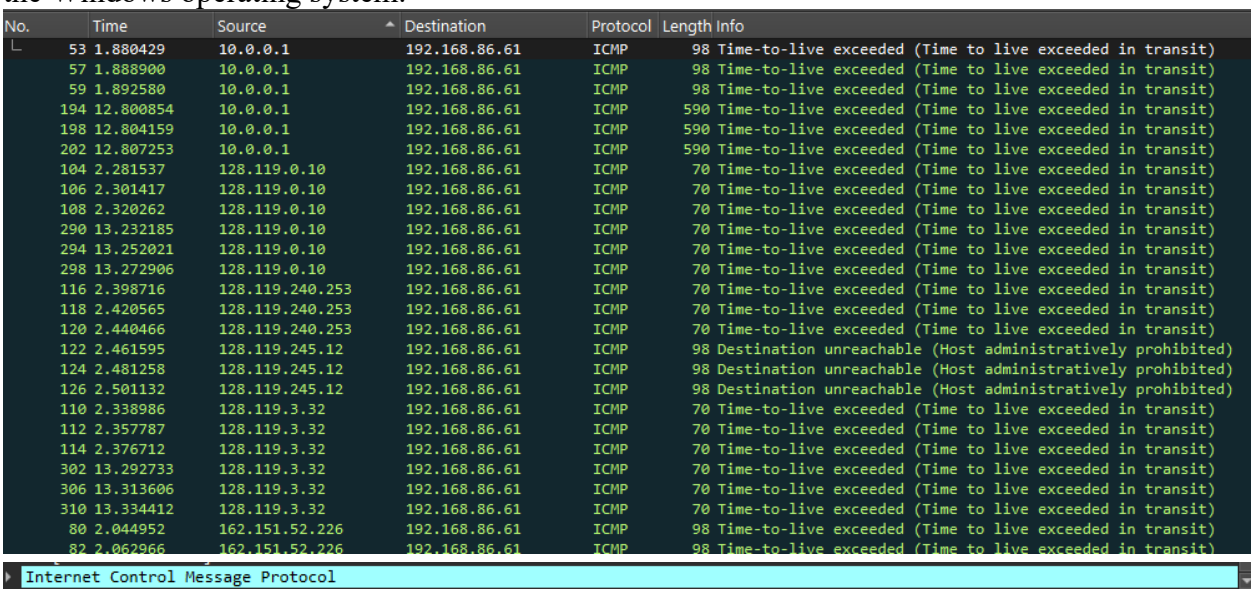
Answer

The identification field values in the IP datagrams increase in a regular pattern. Usually by one for each successive datagram. This reflects the host's method of assigning a unique ID to every packet it sends.

10. What is the upper layer protocol specified in the IP datagrams returned from the routers? [Note: the answers for Linux/MacOS differ from Windows here].

Answer

The upper-layer protocol is the Internet Control Message Protocol, because we are using the Windows operating system.

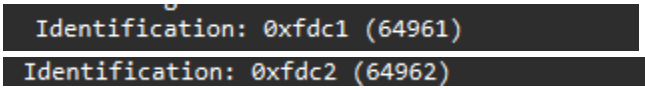


No.	Time	Source	Destination	Protocol	Length	Info
53	1.880429	10.0.0.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
57	1.888900	10.0.0.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
59	1.892580	10.0.0.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
194	12.800854	10.0.0.1	192.168.86.61	ICMP	590	Time-to-live exceeded (Time to live exceeded in transit)
198	12.804159	10.0.0.1	192.168.86.61	ICMP	590	Time-to-live exceeded (Time to live exceeded in transit)
202	12.807253	10.0.0.1	192.168.86.61	ICMP	590	Time-to-live exceeded (Time to live exceeded in transit)
104	2.281537	128.119.0.10	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
106	2.301417	128.119.0.10	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
108	2.320262	128.119.0.10	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
290	13.232185	128.119.0.10	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
294	13.252021	128.119.0.10	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
298	13.272906	128.119.0.10	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
116	2.398716	128.119.240.253	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
118	2.420565	128.119.240.253	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
120	2.440466	128.119.240.253	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
122	2.461595	128.119.245.12	192.168.86.61	ICMP	98	Destination unreachable (Host administratively prohibited)
124	2.481258	128.119.245.12	192.168.86.61	ICMP	98	Destination unreachable (Host administratively prohibited)
126	2.501132	128.119.245.12	192.168.86.61	ICMP	98	Destination unreachable (Host administratively prohibited)
110	2.338986	128.119.3.32	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
112	2.357787	128.119.3.32	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
114	2.376712	128.119.3.32	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
302	13.292733	128.119.3.32	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
306	13.313606	128.119.3.32	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
310	13.334412	128.119.3.32	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
80	2.044952	162.151.52.226	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
82	2.062966	162.151.52.226	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)

11. Are the values in the Identification fields (across the sequence of all of ICMP packets from all of the routers) similar in behavior to your answer to question 9 above?

Answer

Yes, it is very similar. The identification changes with the increment of 1, just like in question 9, and the header checksum. But the time to live does not change. It is always 1.



12. Are the values of the TTL fields similar, across all of ICMP packets from all of the routers?

Answer

Yes, the values of the TTL fields are similar across all of ICMP packets. Their value is 1.

Part 2: Fragmentation

13. Find the first IP datagram containing the first part of the segment sent to 128.119.245.12 sent by your computer via the `traceroute` command to gaia.cs.umass.edu, *after* you specified that the `traceroute` packet length should be 3000. (Hint: This is packet 179 in the *ip-wireshark-trace1-1.pcapng* trace file in footnote 2. Packets 179, 180, and 181 are three IP datagrams created by fragmenting the first single 3000-byte UDP segment sent to 128.119.145.12). Has that segment been fragmented across more than one IP datagram? (Hint: the answer is yes[4]!)

Answer

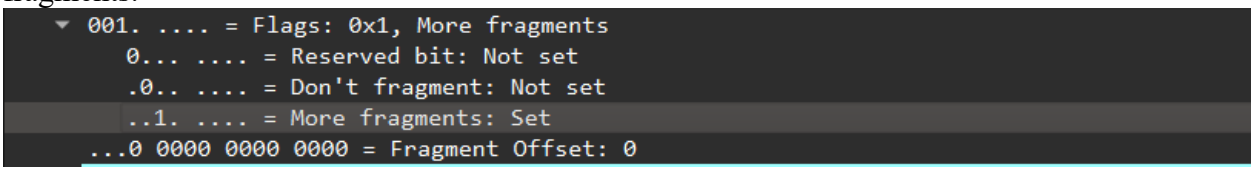
Yes, the 179 segments have been fragmented across more than one IP datagram.

The IP protocol tells us that it is fragmented.

14. What information in the IP header indicates that this datagram been fragmented?

Answer

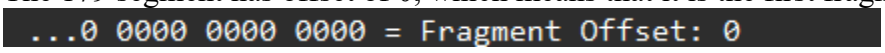
The ip.flags shows us that there are more fragments. In this case there are 3 other fragments.



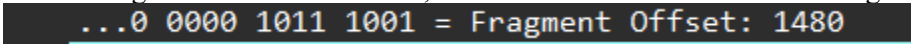
15. What information in the IP header for this packet indicates whether this is the first fragment versus a latter fragment?

Answer

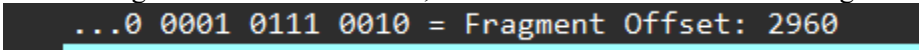
The 179 segment has offset of 0, which means that it is the first fragment



The 180 segment has offset of 1480, which means that it is the second fragment



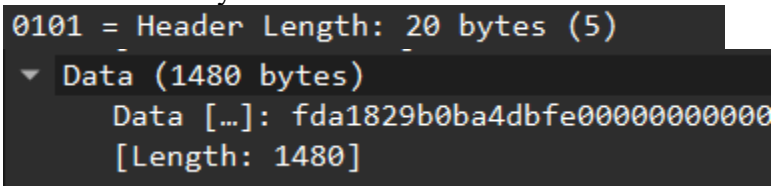
The 181 segment has offset of 2960, which means that it is the third fragment



16. How many bytes are there in is this IP datagram (header plus payload)?

Answer

The header length of the IP datagram is 20 bytes, and the payload is 1480 bytes. In total that would be 1500 bytes.



17. Now inspect the datagram containing the second fragment of the fragmented UDP segment. What information in the IP header indicates that this is *not* the first datagram fragment?

Answer

Same as the answer for 15, we can look for the Fragment Offset. When the fragment offset says 0, then that is the first fragment.

The 179 segment has offset of 0, which means that it is the first fragment

```
...0 0000 0000 0000 = Fragment Offset: 0
```

The 180 segment has offset of 1480, which means that it is the second fragment

```
...0 0000 1011 1001 = Fragment Offset: 1480
```

The 181 segment has offset of 2960, which means that it is the third fragment

```
...0 0001 0111 0010 = Fragment Offset: 2960
```

18. What fields change in the IP header between the first and second fragment?

Answer

Header checksum, ip.frag_offset, and ip.fragments is an extra field that appears at the last fragment.

19. Now find the IP datagram containing the third fragment of the original UDP segment. What information in the IP header indicates that this is the last fragment of that segment?

Answer

The last fragment has an extra header called ip.fragments. This shows all the fragments, and what their position is. The first being 179, then 180 and lastly 181.

```
[3 IPv4 Fragments (2980 bytes): #179(1480), #180(1480), #181(20)]
    [Frame: 179, payload: 0-1479 (1480 bytes)]
    [Frame: 180, payload: 1480-2959 (1480 bytes)]
    [Frame: 181, payload: 2960-2979 (20 bytes)]
```

Part 3: IPv6

20. What is the IPv6 address of the computer making the DNS AAAA request? This is the source address of the 20th packet in the trace. Give the IPv6 source address for this datagram in the exact same form as displayed in the Wireshark window[5].

Answer

21. What is the IPv6 destination address for this datagram? Give this IPv6 address in the exact same form as displayed in the Wireshark window.

The IPv6 destination address is: 2001:558:feed::1

```
Source Address: 2601:193:8302:4620:215c:f5ae:8b40:a27a
Destination Address: 2001:558:feed::1
[Stream index: 1]
```

Answer

22. What is the value of the flow label for this datagram?

Answer

The value of the flow label for the datagram is: 0x063ed0

```
.... 0110 0011 1110 1101 0000 = Flow Label: 0x63ed0
```

23. How much payload data is carried in this datagram?

Answer

The payload is carrying 37 payload data.

```
Payload Length: 37
```

24. What is the upper layer protocol to which this datagram's payload will be delivered at the destination?

Answer

The upper layer protocol is the UDP.

```
User Datagram Protocol
```

Lastly, find the IPv6 DNS response to the IPv6 DNS AAAA request made in the 20th packet in this trace. This DNS response contains IPv6 addresses for youtube.com.

25. How many IPv6 addresses are returned in the response to this AAAA request?

Answer

The response for 20th packet, is the 27th packet, with one answer.

```
Answers
youtube.com: type AAAA, class IN, addr 2607:f8b0:4006:815::200e
```

What is the first of the IPv6 addresses returned by the DNS for youtube.com (in the *ip-wireshark-trace2-1.pcapng* trace file, this is also the address that is numerically the smallest)? Give this IPv6 address in the exact same shorthand form as displayed in the Wireshark window.

Answer

```
Answers
youtube.com: type AAAA, class IN, addr 2607:f8b0:4006:815::200e
```

youtube.com: type AAAA, class IN, addr 2607:f8b0:4006:815::200e