

1. What is the IP address of your computer?

```
▼ Internet Protocol Version 4, Src: 192.168.0.16, Dst: 151.101.65.140
```

My computer's IP address is 192.168.0.16.

2. Within the IP packet header, what is the value in the upper layer protocol field?

```
Fragment Offset: 0
> Time to live: 1
  Protocol: ICMP (1)
  Header checksum: 0x0000 [validation disabled]
  [Header checksum status: Unverified]
  Source: 192.168.0.16
  Destination: 151.101.65.140
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
Internet Control Message Protocol
```

The value in the upper layer protocol field is ICMP (1).

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

```
▼ Internet Protocol Version 4, Src: 192.168.0.16, Dst: 151.101.65.140
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▼ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    0000 00.. = Differentiated Services Codepoint: Default (0)
    .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
  Total Length: 56
```

There are 20 bytes in the IP header and 56 total length. Total length minus the bytes in the header equals the payload of the IP datagram which is 36.

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

```
▼ Flags: 0x00
  0... .... = Reserved bit: Not set
  .0.. .... = Don't fragment: Not set
  ..0. .... = More fragments: Not set
  Fragment offset: 0
```

No, the IP datagram has not been fragmented. You can see this because the "More fragments" flag has not been set.

5. Which fields in the IP datagram always change from one datagram to the next within the series of ICMP messages sent by your computer?

The fields in the IP datagram that always change from one datagram to the next within the series of ICMP messages sent by your computer are Identification, Time-to-live and Header checksum.

6. Which fields stay constant? Which of the fields must stay constant? Which fields must change? Why?

The fields that stay constant are –

- a. Version (we are using IPv4 for all packets)
- b. Header Length (because these are ICMP packets)
- c. source IP (the source stays the same)
- d. destination IP (the destination stays the same)
- e. Differentiated Services (all packets have the same Type of Service class)
- f. Upper Layer Protocol (because these are ICMP packets)

All of the fields above must stay constant for the stated reasons above.

The fields that must change are –

- a. Time-to-live (the changes based on traceroute and what packet we are one)
- b. Identification (must change depending on what packet we are on)
- c. Header checksum (changes with the header)

7. Describe the pattern you see in the values in the Identification field of the UP datagram.

The pattern in the values in the Identification field of the UP datagram is that the fields increment with each ICMP Echo request.

8. What is the value in the Identification field and the TTL field?

2477	71.350850	72.31.67.44	192.168.0.16	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
2329	68.850694	72.31.67.44	192.168.0.16	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
2198	66.350705	72.31.67.44	192.168.0.16	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
2062	63.850654	72.31.67.44	192.168.0.16	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
1930	61.348360	72.31.67.44	192.168.0.16	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
1788	58.848419	72.31.67.44	192.168.0.16	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
1655	56.348480	72.31.67.44	192.168.0.16	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
1572	53.848277	72.31.67.44	192.168.0.16	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
1505	51.348255	72.31.67.44	192.168.0.16	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)

```
> Frame 2477: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface 0
> Ethernet II, Src: ArrisGro_b7:21:d3 (bc:64:4b:b7:21:d3), Dst: AsustekC_a7:0f:fa (70:8b:cd:a7:0f:fa)
v Internet Protocol Version 4, Src: 72.31.67.44, Dst: 192.168.0.16
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  v Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    0000 00.. = Differentiated Services Codepoint: Default (0)
    .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
  Total Length: 96
  Identification: 0x4592 (17810)
  v Flags: 0x00
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
  Fragment offset: 0
  Time to live: 253
```

The value in the Identification field is 17810 and is it 253 in the TTL field.

9. Do these values remain unchanged for all of the ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router? Why?

No – because each datagram has a unique Identification field, any new datagram will have a changed Identification field. If the field hasn't changed then we know that the datagram is a part of the same IP datagram, broken into parts.

10. Find the first ICMP Echo Request message that was sent by your computer after you changed the Packet Size in pingplotter to be 2000. Has that message been fragmented across more than one IP datagram?

Yes, after changing the Packet Size in pingplotter to be 2000, the message has been fragmented across more than one IP datagram.

11. Screenshot the first fragment of the fragmented IP datagram (with sufficient details to answer these questions). What information in the IP header indicates that the datagram has been fragmented. What information in the IP header indicates whether this is the first fragment versus a latter fragment? How long is this IP datagram?

598	26.110978	192.168.0.16	192.168.0.255	UDP	305 54915 → 54915 Len=263
599	26.126694	192.168.0.16	151.101.1.140	ICMP	70 Echo (ping) request id=0x0001, seq=992/57347, ttl=8 (no response found!)
600	26.176936	192.168.0.16	151.101.1.140	ICMP	70 Echo (ping) request id=0x0001, seq=993/57603, ttl=9 (reply in 601)
601	26.198067	151.101.1.140	192.168.0.16	ICMP	70 Echo (ping) reply id=0x0001, seq=993/57603, ttl=56 (request in 600)
602	27.090923	192.168.0.16	192.168.0.255	UDP	305 54915 → 54915 Len=263
603	27.134361	192.168.0.16	151.101.4.133	SSL	55 Continuation Data
604	27.155419	151.101.4.133	192.168.0.16	TCP	66 443 → 63123 [ACK] Seq=1 Ack=2 Win=70 Len=0 SLE=1 SRE=2
605	27.628390	fe80::be64:4bff:feb...	ff02::1	ICMPv6	110 Router Advertisement from bc:64:4b:b7:21:d3
606	27.715382	192.168.0.16	151.101.65.140	IPv4	1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=7640) [Reassembled in #607]
607	27.715399	192.168.0.16	151.101.65.140	ICMP	534 Echo (ping) request id=0x0001, seq=994/57859, ttl=255 (reply in 609)
608	27.738374	151.101.65.140	192.168.0.16	IPv4	1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=8c03) [Reassembled in #609]
609	27.738376	151.101.65.140	192.168.0.16	ICMP	534 Echo (ping) reply id=0x0001, seq=994/57859, ttl=56 (request in 607)
610	27.766216	192.168.0.16	151.101.65.140	IPv4	1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=7641) [Reassembled in #611]
611	27.766230	192.168.0.16	151.101.65.140	ICMP	534 Echo (ping) request id=0x0001, seq=995/58115, ttl=1 (no response found!)
612	27.770707	192.168.0.1	192.168.0.16	TCP	590 Time-to-live exceeded (Time to live exceeded in transit)

  

0100	....	= Version: 4
....	0101	= Header Length: 20 bytes (5)
▼	Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)	
	0000	00.. = Differentiated Services Codepoint: Default (0)
	....	..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
	Total Length: 1500	
	Identification: 0x7640 (30272)	
▼	Flags: 0x01 (More Fragments)	
	0...	.... = Reserved bit: Not set
	.0...	.... = Don't fragment: Not set
	..1.	.... = More fragments: Set
	Fragment offset: 0	

We know that the datagram has been fragmented because the “More fragment” flag has been set to 1. We can also tell that this is the first fragment because the Fragment offset is 0. The datagram has a Total Length of 1500 (including the header).

12. Screenshot the second fragment of the fragmented IP datagram (with sufficient details to answer these questions). What information in the IP header indicates that this is not the first datagram fragment? Are there more fragments? How can you tell?

→	607	27.715399	192.168.0.16	151.101.65.140	ICMP	534 Echo (ping) request id=0x0001, seq=994/57859, ttl=255 (reply in 609)
	608	27.738374	151.101.65.140	192.168.0.16	IPv4	1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=8c03) [Reassembled in #609]
→	609	27.738376	151.101.65.140	192.168.0.16	ICMP	534 Echo (ping) reply id=0x0001, seq=994/57859, ttl=56 (request in 607)
	610	27.766216	192.168.0.16	151.101.65.140	IPv4	1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=7641) [Reassembled in #611]
	611	27.766230	192.168.0.16	151.101.65.140	ICMP	534 Echo (ping) request id=0x0001, seq=995/58115, ttl=1 (no response found!)
	612	27.770707	192.168.0.16	192.168.0.16	ICMP	590 Time-to-live exceeded (Time to live exceeded in transit)
0100 .... = Version: 4						
... 0101 = Header Length: 20 bytes (5)						
▼ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)						
0000 00.. = Differentiated Services Codepoint: Default (0)						
... 00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)						
Total Length: 520						
Identification: 0x7640 (30272)						
▼ Flags: 0x00						
0... .... = Reserved bit: Not set						
.0. .... = Don't fragment: Not set						
..0. .... = More fragments: Not set						
Fragment offset: 1480						
Time to live: 255						

You can tell that this is not the first datagram fragment because the Fragment offset is not 0 (it is 1480). You can also tell that there are not any more fragments because the “More fragments” flag is set to 0.

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

The fields that changed in the IP header between the first and second fragments are the total length flags, fragment offset, checksum, and the flags. Everything else remained the same.

14. How many fragments were created from the original datagram?

	1599	55.245825	151.101.65.140	192.168.0.16	ICMP	554 Echo (ping) reply id=0x0001, seq=1294/3589, ttl=56 (request in 1596)
	1600	55.273140	192.168.0.16	151.101.65.140	IPv4	1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=776d) [Reassembled in #1602]
	1601	55.273157	192.168.0.16	151.101.65.140	IPv4	1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=776d) [Reassembled in #1602]
	1602	55.273164	192.168.0.16	151.101.65.140	ICMP	554 Echo (ping) request id=0x0001, seq=1295/3845, ttl=1 (no response found!)
	1603	55.276034	192.168.0.1	192.168.0.16	ICMP	590 Time-to-live exceeded (Time to live exceeded in transit)
	1604	55.323172	192.168.0.16	151.101.65.140	IPv4	1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=776e) [Reassembled in #1606]
	1605	55.323188	192.168.0.16	151.101.65.140	IPv4	1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=776e) [Reassembled in #1606]
	1606	55.323195	192.168.0.16	151.101.65.140	ICMP	554 Echo (ping) request id=0x0001, seq=1296/4101, ttl=2 (no response found!)
	1607	55.335613	10.110.48.1	192.168.0.16	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
	1608	55.372911	192.168.0.16	151.101.65.140	IPv4	1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=776f) [Reassembled in #1610]
▼ Frame 1600: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0						
▼ Ethernet II, Src: AsustekC_a7:0f:fa (70:8b:cd:a7:0f:fa), Dst: ArrisGro_b7:21:d3 (bc:64:4b:b7:21:d3)						
▼ Internet Protocol Version 4, Src: 192.168.0.16, Dst: 151.101.65.140						
0100 .... = Version: 4						
... 0101 = Header Length: 20 bytes (5)						
▼ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)						
0000 00.. = Differentiated Services Codepoint: Default (0)						
... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)						
Total Length: 1500						
Identification: 0x776d (30573)						
▼ Flags: 0x01 (More Fragments)						
0... .... = Reserved bit: Not set						
.0. .... = Don't fragment: Not set						
..1. .... = More fragments: Set						
Fragment offset: 0						

First fragment after changing Packet Size in pingplotter to be 3500.

```

Internet Protocol Version 4, Src: 192.168.0.16, Dst: 151.101.65
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
  ▾ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT
    0000 00.. = Differentiated Services Codepoint: Default (0
      .... ..00 = Explicit Congestion Notification: Not ECN-Cap
    Total Length: 1500
    Identification: 0x776d (30573)
  ▾ Flags: 0x01 (More Fragments)
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..1. .... = More fragments: Set
    Fragment offset: 1480

```

Second Fragment.

```

  ▾ Internet Protocol Version 4, Src: 192.168.0.16, Dst:
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▾ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT
    0000 00.. = Differentiated Services Codepoint: Default (0
      .... ..00 = Explicit Congestion Notification: Not ECN-Cap
    Total Length: 540
    Identification: 0x776d (30573)
  ▾ Flags: 0x00
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
    Fragment offset: 2960

```

Third Fragment.

There were three fragments created from the original datagram. You can see this because the first datagram has a total length of 1500 with “More fragments” set to 1 and a Fragment offset of 0.

The second packet has a total length of 1500 with “More fragments” set to 1 and a Fragment offset of 1480.

And the third packet has a total length of 540 with “More fragments” set to 0 and a Fragment offset of 2960.

15. What fields change in the IP header among the fragments?

The Total Length stays the same for packets one and two, but change for packet three. The Fragment Offset changes for all three packets. And the “More fragments” flag stays the same for packets one and two, but change for packet three.