

ECE 456

Lab 09

Pablo Corona Alvarez

No.	Time	Source	Destination	Protocol	Length	Info
348	13.560110	2601:282:8001:f90:79e5:9baf:8949:a392	2607:f8b0:400f:804::2003	HTTP	442	GET /gts1c3/MEBwTTBLMEkwrZAHBgUrDgMCgQUxy55it3h2FYTSzuu1HQr17xsAKB2MEFip0f6n2BFze6VzT2c80JGFPNnNR0nAhAU...
365	13.5835493	2607:f8b0:400f:804::2003	2601:282:8001:f90:79e5:9baf:8949:a392	OCSP	799	Response
1585	14.527489	2601:282:8001:f90:79e5:9baf:8949:a392	2607:f8b0:400f:804::2003	HTTP	442	GET /gts1c3/MEBwTTBLMEkwrZAHBgUrDgMCgQUxy55it3h2FYTSzuu1HQr17xsAKB2MEFip0f6n2BFze6VzT2c80JGFPNnNR0nAhBT...
2211	14.578325	2607:f8b0:400f:804::2003	2601:282:8001:f90:79e5:9baf:8949:a392	OCSP	799	Response
2459	14.600483	2601:282:8001:f90:79e5:9baf:8949:a392	2607:f8b0:400f:804::2003	HTTP	448	GET /gts1c3/MFAwTJBMMEowSDAHBgUrDgMCgQUxy55it3h2FYTSzuu1HQr17xsAKB2MEFip0f6n2BFze6VzT2c80JGFPNnNR0nAhEA...
2753	14.620880	2607:f8b0:400f:804::2003	2601:282:8001:f90:79e5:9baf:8949:a392	OCSP	800	Response
3452	15.147213	2601:282:8001:f90:79e5:9baf:8949:a392	2607:f8b0:400f:804::2003	HTTP	444	GET /gts1c3/MEBwTTBLMEkwrZAHBgUrDgMCgQUxy55it3h2FYTSzuu1HQr17xsAKB2MEFip0f6n2BFze6VzT2c80JGFPNnNR0nAhBE...
3456	15.166683	2607:f8b0:400f:804::2003	2601:282:8001:f90:79e5:9baf:8949:a392	OCSP	799	Response

Figure 1: Http filter of data over network

```

▼ Frame 348: 442 bytes on wire (3536 bits), 442 bytes captured (3536 bits) on interface en0, id 0
  > Interface id: 0 (en0)
    Encapsulation type: Ethernet (1)
    Arrival Time: May 7, 2022 17:50:38.793680000 MDT
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1651967438.793680000 seconds
    [Time delta from previous captured frame: 0.000623000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 13.560110000 seconds]
    Frame Number: 348
    Frame Length: 442 bytes (3536 bits)
    Capture Length: 442 bytes (3536 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ipv6:tcp:http]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
  ▼ Ethernet II, Src: Apple_13:93:d2 (f0:2f:4b:13:93:d2), Dst: ARRISGro_b1:00:28 (5c:8f:e0:b1:00:28)
    ▼ Destination: ARRISGro_b1:00:28 (5c:8f:e0:b1:00:28)
      Address: ARRISGro_b1:00:28 (5c:8f:e0:b1:00:28)
      .... ..0. .... = LG bit: Globally unique address (factory default)
      .... ..0. .... = IG bit: Individual address (unicast)
    ▼ Source: Apple_13:93:d2 (f0:2f:4b:13:93:d2)
      Address: Apple_13:93:d2 (f0:2f:4b:13:93:d2)
      .... ..0. .... = LG bit: Globally unique address (factory default)
      .... ..0. .... = IG bit: Individual address (unicast)
    Type: IPv6 (0x86dd)

```

Figure 2: Expanded packet details

```

Internet Protocol Version 6, Src: 2601:282:8001:f90:79e5:9baf:8949:a392, Dst: 2607:f8b0:400f:804::2003
  0110 .... = Version: 6
  > .... 0000 0000 .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
  .... 1100 1101 0001 1011 0010 = Flow Label: 0xcd1b2
  Payload Length: 388
  Next Header: TCP (6)
  Hop Limit: 64
  Source Address: 2601:282:8001:f90:79e5:9baf:8949:a392
  Destination Address: 2607:f8b0:400f:804::2003
  ▼ Transmission Control Protocol, Src Port: 56248, Dst Port: 80, Seq: 1, Ack: 1, Len: 356
    Source Port: 56248
    Destination Port: 80
    [Stream index: 11]
    [Conversation completeness: Complete, WITH_DATA (63)]
    [TCP Segment Len: 356]
    Sequence Number: 1 (relative sequence number)
    Sequence Number (raw): 866900355
    [Next Sequence Number: 357 (relative sequence number)]
    Acknowledgment Number: 1 (relative ack number)
    Acknowledgment number (raw): 2466074490
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x018 (PSH, ACK)
    Window: 2052
    [Calculated window size: 131328]
    [Window size scaling factor: 64]
    Checksum: 0x1173 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [Timestamps]
  > [SEQ/ACK analysis]
  TCP payload (356 bytes)

```

Figure 3: Expanded packet details pt 2

When running wireshark and monitoring the traffic we are able to see a lot more in depth information about the packets being sent and received. IN these particular screenshots we are looking at a HTTP communication between the source and destination. Something that I noticed from my running wireshark is that I am using a laptop to run this software, because of that I do not have a physical Ethernet connection. When choosing the interface to monitor I decided to focus on the wifi interface. Though I believe that it is because of this that my source and destination numbers are not IP addresses. Below are the source and destination addresses

Source Address: 2601:282:8001: f90:795:9baf:8949: a392

Destination Address: 2607: f8b0:400f: 804::2003

As we can see this is not the standard XXX.XXX.XXX.XXX ip address format but is instead very different. The other thing that I was able to notice from the difference between my interface and the lab manual is that when I look at the info section I get a far more jumbled up representation. The lab manual in the info section says GET/ HTTP/1.1 while my section contains a long string of what seems to be random characters. Though after further inspection this seems to be what is different between the examples that I ran and the Example that is in the lab manual. I was also able to run this experiment again but this time streaming a video from youtube as I monitored the packets this time i was able to see that the Get/ XXXXX/HTTP/1.1 Was actually reaching for a streaming service and then was simply continuing to stream this content that was given.

No.	Time	Source	Di	Protocol	Info
2462	69.220043	10.28.16.129	--	HTTP	GET /filestreamingservice/files/99f49bf6-1a2b-4b47-8d69-8f4e33c5e0f4/pieceshash HTTP/1.1
2467	69.231248	23.58.85.144	--	HTTP	HTTP/1.1 200 OK
2495	69.363910	10.28.16.129	--	HTTP	GET /filestreamingservice/files/99f49bf6-1a2b-4b47-8d69-8f4e33c5e0f4?P1=16526299396P2=4846P3=26P4=XanYuHtoG15fKdVT6zLnZtEo2CnaQv4YjUMvylMMN
2496	69.363943	10.28.16.129	--	HTTP	GET /filestreamingservice/files/99f49bf6-1a2b-4b47-8d69-8f4e33c5e0f4?P1=16526299396P2=4846P3=26P4=XanYuHtoG15fKdVT6zLnZtEo2CnaQv4YjUMvylMMN
4091	69.547287	208.111.186.0	--	HTTP	HTTP/1.1 206 Partial Content
4124	69.549128	10.28.16.129	--	HTTP	GET /filestreamingservice/files/99f49bf6-1a2b-4b47-8d69-8f4e33c5e0f4?P1=16526299396P2=4846P3=26P4=XanYuHtoG15fKdVT6zLnZtEo2CnaQv4YjUMvylMMN
4331	69.563695	208.111.186.128	--	HTTP	HTTP/1.1 206 Partial Content
4333	69.565471	10.28.16.129	--	HTTP	GET /filestreamingservice/files/99f49bf6-1a2b-4b47-8d69-8f4e33c5e0f4?P1=16526299396P2=4846P3=26P4=XanYuHtoG15fKdVT6zLnZtEo2CnaQv4YjUMvylMMN
5000	69.612773	208.111.186.128	--	HTTP	[60 seconds segment not captured] Continuation
5001	69.612777	208.111.186.128	--	HTTP	Continuation
5002	69.612777	208.111.186.128	--	HTTP	Continuation
5003	69.612778	208.111.186.128	--	HTTP	Continuation
5004	69.612778	208.111.186.128	--	HTTP	Continuation
5005	69.612779	208.111.186.128	--	HTTP	Continuation
5006	69.612779	208.111.186.128	--	HTTP	Continuation
5007	69.612780	208.111.186.128	--	HTTP	Continuation
5008	69.612780	208.111.186.128	--	HTTP	Continuation
5009	69.612781	208.111.186.128	--	HTTP	Continuation

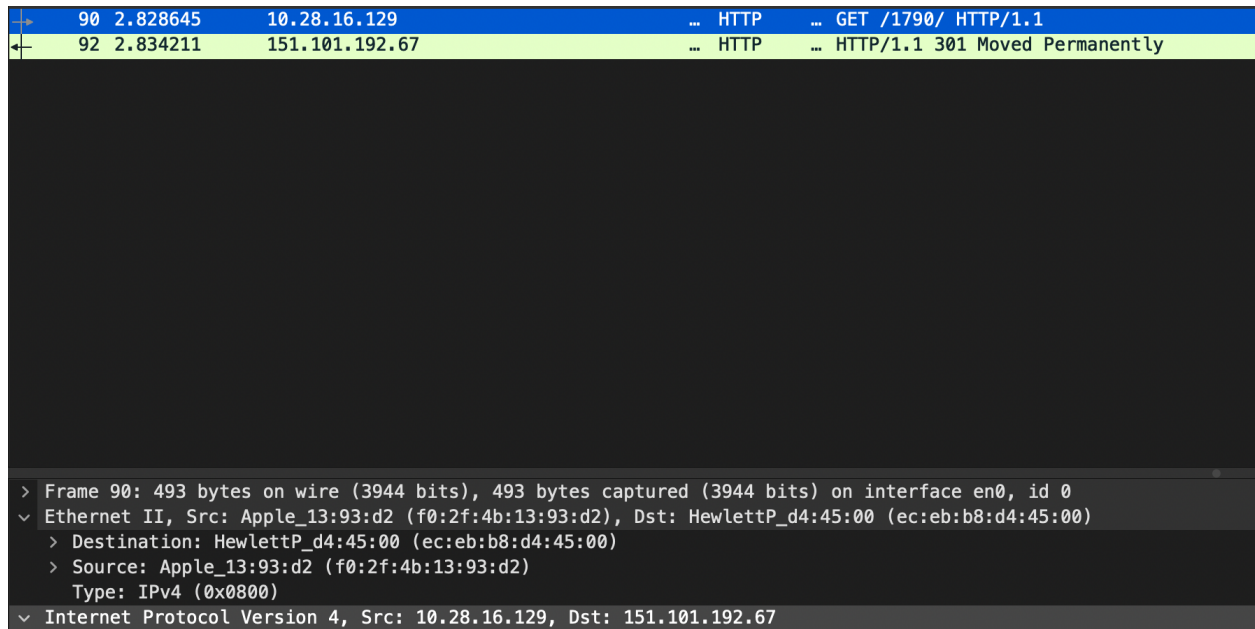


Figure 5: Running Wireshark on the

Here we can see that when we are simply reloading the HTML website after having click the random image and wait for everything to reload that we are operating from 10.28.16.129 with port 55827 while the Destination is 151.101.192.67 with port 80. When the two are contacting eachother whether its requesting information or receiving information we are using an IPv4 type connection to be able to communicate with. Something that I don't see here is the 3 way handshake, there is a request being made and then the request is being fulfilled so I do not believe that there was any form of 3 way handshake being used.

When we run the who is command on the destination it returns the following infomratoin

```
OrgName:    Fastly
OrgId:      SKYCA-3
Address:    PO Box 78266
City:       San Francisco
StateProv:  CA
PostalCode: 94107
Country:    US
RegDate:    2011-09-16
Updated:    2021-09-20
Ref:        https://rdap.arin.net/registry/entity/SKYCA-3
```

When it is ran on the source it returns the following information

```
inetnum:    10.0.0.0 - 10.255.255.255
organisation: IANA - Private Use
status:     RESERVED

remarks:    Reserved for Private-Use Networks [RFC1918].Complete
remarks:    registration details for 10.0.0.0/8 are found
remarks:    iniana-ipv4-special-registry.
```

changed: 1995-06
source: IANA

We can see where the destination is and who it is, in this case it looks like we are accessing a service that is being provided by Fastly a large company, but when we are looking at the information on the source IP we only are able to see the the company is private and doesn't provide much information

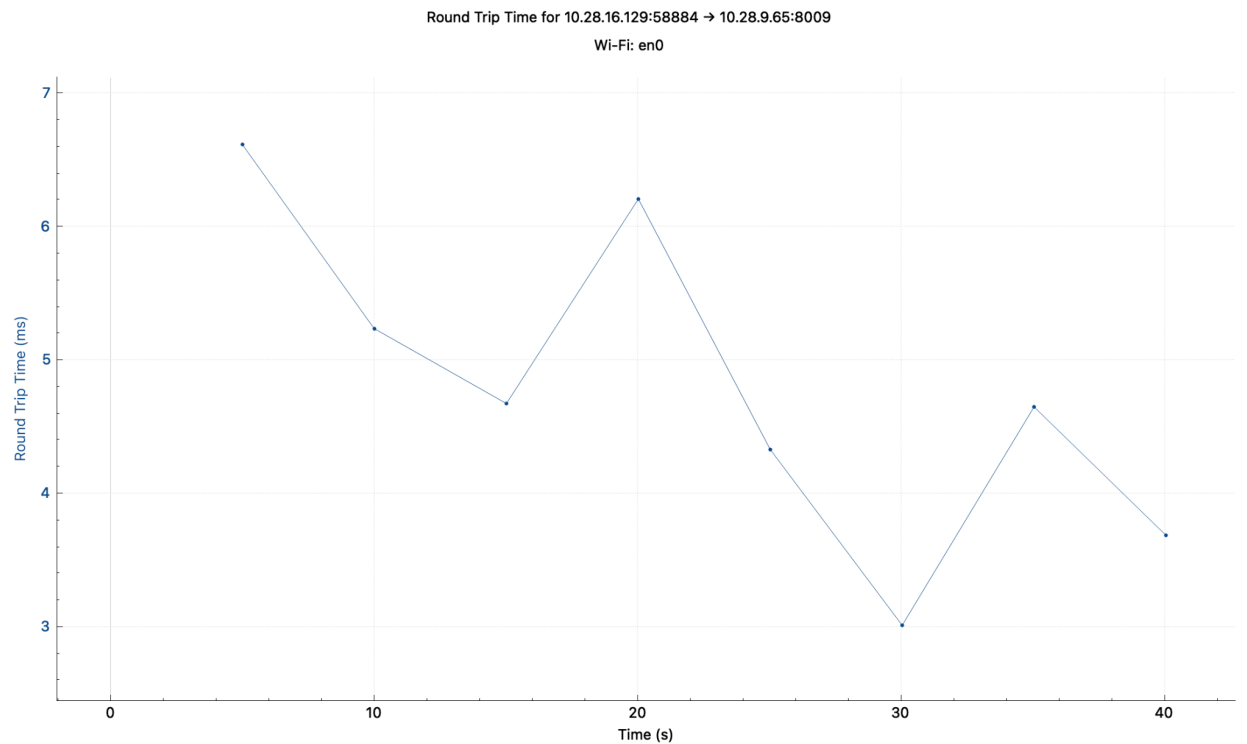


Figure 6: TCP Round Trip Time Graph

As can be seen from Figure 6 and Figure 7 We are able to see that THE two graphs can be thought of as inverses of eachother. When one Figure 7 shows a lot of congestion/has a high y value on the graph. We can also see that when there is a change in the sequence number that we also see a change along the TCP Round Trip Time Graph. Thus we are able to see that as we reload more websites or as the congestion through TCP increases that the round trip time decreases. This is not what we expected to happen as when you see more congestion the general intuition would be that the as the congestion increases over a TCP connection you would find the total round trip time would increase aswell. What i believe this indicates that since this only tracks the single packet from one connection you cannot truly see the decrease in performance as we are only monitoring the single packet and not the overall connction.

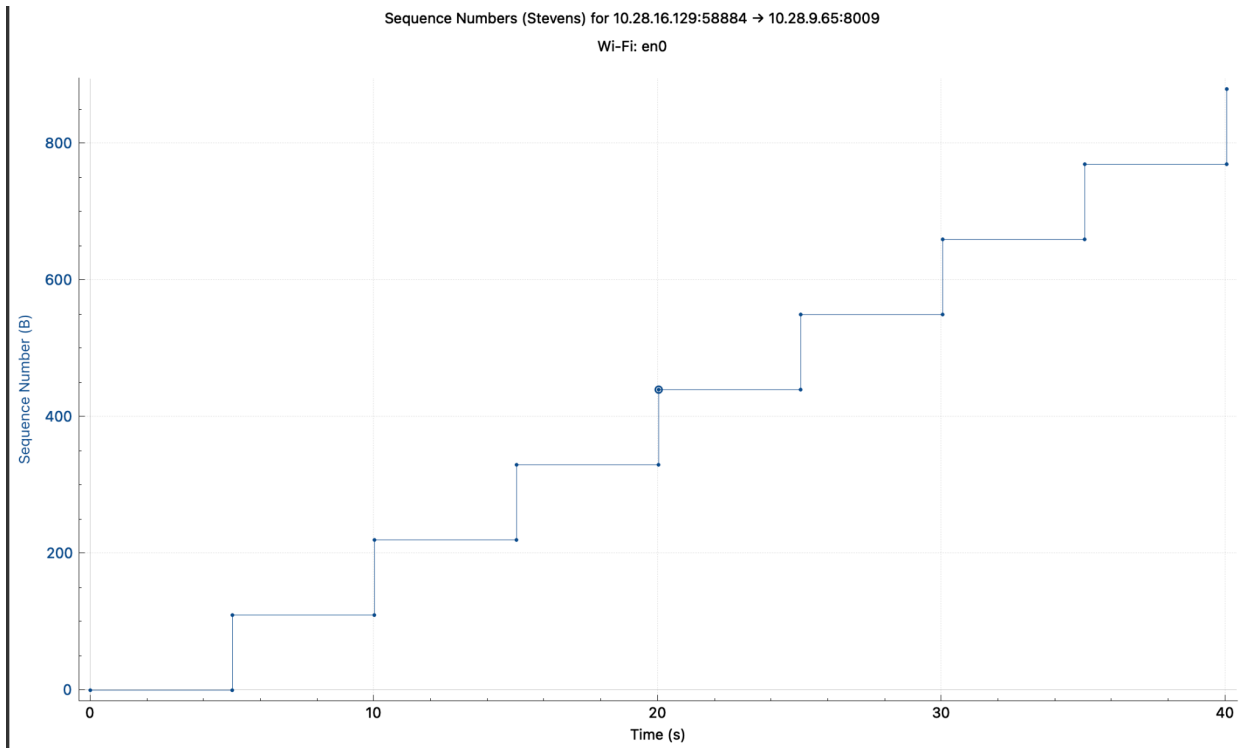


Figure 7: TCP Time Sequence Graph (Stevens)

Part 3:

```
C:\Users\corona>tracert www.google.com

Tracing route to www.google.com [142.250.72.36]
over a maximum of 30 hops:

  1    7 ms    5 ms    5 ms  den16s08-in-f4.1e100.net [142.250.72.36]

Trace complete.

C:\Users\corona>tracert www.google.com

Tracing route to www.google.com [142.250.72.36]
over a maximum of 30 hops:

  1    5 ms    5 ms    5 ms  den16s08-in-f4.1e100.net [142.250.72.36]

Trace complete.

C:\Users\corona>
```

Figure 8: Running TraceRT www.google.com

Tracert is a function that uses ICMP echo request in order to determine the path that the ping took. The way that it does this is through incrementally Decreasing the TTL for each router along the path taken to reach our destination. This effectively acts as a counter for the maximum number of hops the connection can make in order to reach its destination. IF the TTL were to reach 0 prior to reaching its destination, the router is expected to return an ICMP time exceeded message to the source. So here when analyzing the traffic that this command has with wire shark we want to understand the amount of hops that were taken to arrive at the final destination or not, meaning we want to focus on the TTL

No.	Time	Source	Destination	Protocol	Info
64	3.147735	10.28.16.129	142.250.72.36	ICMP	Echo (ping) request id=0x0001, seq=28/7168, ttl=64 (reply in 65)
66	3.153860	10.28.16.129	142.250.72.36	ICMP	Echo (ping) request id=0x0001, seq=29/7424, ttl=64 (reply in 67)
68	3.168015	10.28.16.129	142.250.72.36	ICMP	Echo (ping) request id=0x0001, seq=30/7680, ttl=64 (reply in 69)
65	3.152836	142.250.72.36	10.28.16.129	ICMP	Echo (ping) reply id=0x0001, seq=28/7168, ttl=117 (request in 64)
67	3.158790	142.250.72.36	10.28.16.129	ICMP	Echo (ping) reply id=0x0001, seq=29/7424, ttl=117 (request in 66)
69	3.164759	142.250.72.36	10.28.16.129	ICMP	Echo (ping) reply id=0x0001, seq=30/7680, ttl=117 (request in 68)

> Frame 68: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface en0, id 0
> Ethernet II, Src: Apple_13:93:d2 (f8:2f:4b:13:93:d2), Dst: HewlettP_d4:45:00 (ec:eb:b8:d4:45:00)
> Destination: HewlettP_d4:45:00 (ec:eb:b8:d4:45:00)
> Source: Apple_13:93:d2 (f8:2f:4b:13:93:d2)
Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 10.28.16.129, Dst: 142.250.72.36
0100 ... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 92
Identification: 0xca5f (51807)
> Flags: 0x00
... 0 0000 0000 = Fragment Offset: 0
Time to Live: 64
Protocol: ICMP (1)
Header Checksum: 0xb8b6 [validation disabled]
[Header checksum status: Unverified]
Source Address: 10.28.16.129
Destination Address: 142.250.72.36
> Internet Control Message Protocol

Figure 9: ICMP request being sent from source to destination TTL 64

Nb.	Time	Source	- [D] Protocol [L] Info
64	3.147735	10.28.16.129	- ICMP .. Echo (ping) request id=0x0001, seq=28/7168, ttl=64 (reply in 65)
66	3.153868	10.28.16.129	- ICMP .. Echo (ping) request id=0x0001, seq=29/7424, ttl=64 (reply in 67)
68	3.160015	10.28.16.129	- ICMP .. Echo (ping) request id=0x0001, seq=30/7680, ttl=64 (reply in 69)
65	3.152836	142.250.72.36	- ICMP .. Echo (ping) reply id=0x0001, seq=28/7168, ttl=117 (request in 64)
67	3.158790	142.250.72.36	- ICMP .. Echo (ping) reply id=0x0001, seq=29/7424, ttl=117 (request in 66)
69	3.164739	142.250.72.36	- ICMP .. Echo (ping) reply id=0x0001, seq=30/7680, ttl=117 (request in 68)


```

> Frame 69: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface en0, id 0
> Ethernet II, Src: HewlettP_d4:45:00 (ec:eb:b8:d4:45:00), Dst: Apple_13:93:d2 (f0:2f:4b:13:93:d2)
> Destination: Apple_13:93:d2 (f0:2f:4b:13:93:d2)
> Source: HewlettP_d4:45:00 (ec:eb:b8:d4:45:00)
> Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 142.250.72.36, Dst: 10.28.16.129
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 92
Identification: 0x0000 (0)
> Flags: 0x00
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 117
Protocol: ICMP (1)
Header Checksum: 0x5366 [validation disabled]
[Header checksum status: Unverified]
Source Address: 142.250.72.36
Destination Address: 10.28.16.129
> Internet Control Message Protocol

```

Figure 10: ICMP echo responding from destination TTL 117

Here we are able to see that 3 ICMP echo requests to the destination, and each of them had a TTL of 64 while when the server was communicating back to the local machine we see that there is a TTL of 117. From this we are able to tell that when we requested the information/echo from the server, this was talking far more hops in order to reach its final destination then the server communicating back with my machine. Though there is one other thing that i found interesting from this command was that the more ICMP request that the tracert sent we would find that the reply would take longer for the requests being sent after to reply.

Part 4:

1. The SSIDs of the two access points that issue most of the beacon frames in this trace are 30 munroe St as well as linksys12.

No.	Time	Source	Destination	Protocol	Length	Info
2	0.023373	Cisco-Li_f7:1d:51	Broadcast	802.11	183	Beacon frame, SN=2855, FN=0, Flags=.....C, BI=100, SSID=30 Munroe St

Frame 2: 183 bytes on wire (1464 bits), 183 bytes captured (1464 bits)

Figure 11:

No.	Time	Source	Destination	Protocol	Length	Info
13	0.437096	LinksysG_67:22:94	Broadcast	802.11	90	Beacon frame, SN=3074, FN=0, Flags=.....C, BI=100, SSID=linksys12

Figure 12:

2. From the linksys12 we see that the interval of time between the transmissions was [Time delta from previous captured frame: 0.004165000 seconds]

```
[Time delta from previous captured frame: 0.004165000 seconds]
[Time delta from previous displayed frame: 0.004165000 seconds]
[Time since reference or first frame: 0.437096000 seconds]
Frame Number: 13
```

Figure 13: time delta for linksys12 SSID

From the 30 Munroe st we see that the interval of time between transmissions was

[Time delta from previous displayed frame: 0.023373000 seconds]

Epoch Time: 1183082707.157931000 seconds

[Time delta from previous captured frame: 0.023373000 seconds]

[Time delta from previous displayed frame: 0.023373000 seconds]

[Time since reference or first frame: 0.023373000 seconds]

Figure 14: time delta for 30 munroe st

3. Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
Transmitter address: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
Source address: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
BSS Id: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)

```
IEEE 802.11 Beacon frame, Flags: .....C
  Type/Subtype: Beacon frame (0x0008)
  > Frame Control Field: 0x8000
    .000 0000 0000 0000 = Duration: 0 microseconds
    Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
    Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    Transmitter address: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
    Source address: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
    BSS Id: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
    .... 0000 = Fragment number: 0
    1011 0010 0111 .... = Sequence number: 2855
    Frame check sequence: 0x39700f3d [unverified]
    [FCS Status: Unverified]
```

Figure 15: Mac information for 30 munroe st

4. Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
5. BSS Id: Cisco-Li_f7:1d:51 (00:16:b6:f7:1d:51)
- 6.
- 7.