Network Architecture Assignment #3 Spring 2018

Pragathi Thammaneni

16230695

PragathiThammaneni_____HomeWork_____16230695_

Question-1:

Consider sending a large file from a host to another over a TCP connection that has no loss.

a. Suppose TCP uses AIMD for its congestion control without slow start. Assuming cwnd increases by 1 MSS every time a batch of ACKs is received and assuming approximately constant RTTs, how long does it take for cwnd increase from 1 MSS to 8 MSS?

Answer:

Since we are considering the case of AIMD without slow start, TCP directly enters the congestion avoidance phase, during which the value of cwnd increases by just 1 MSS every RTT.

It takes:

1 RTT is required to increase to 2 MSS.
2 RTT is required to increase to 3 MSS.
3 RTT is required to increase to 4 MSS.
4 RTT is required to increase to 5 MSS.
5 RTT is required to increase to 6 MSS.
6 RTT is required to increase to 7 MSS.
Finally, it takes 7 RTT to increase to 8 MSS.

b. What is the average throughput in terms of MSS and RTT for this connection up through time = 6 RTT (consider 1st 6 RTT after the start of the scenario)?

Answer:

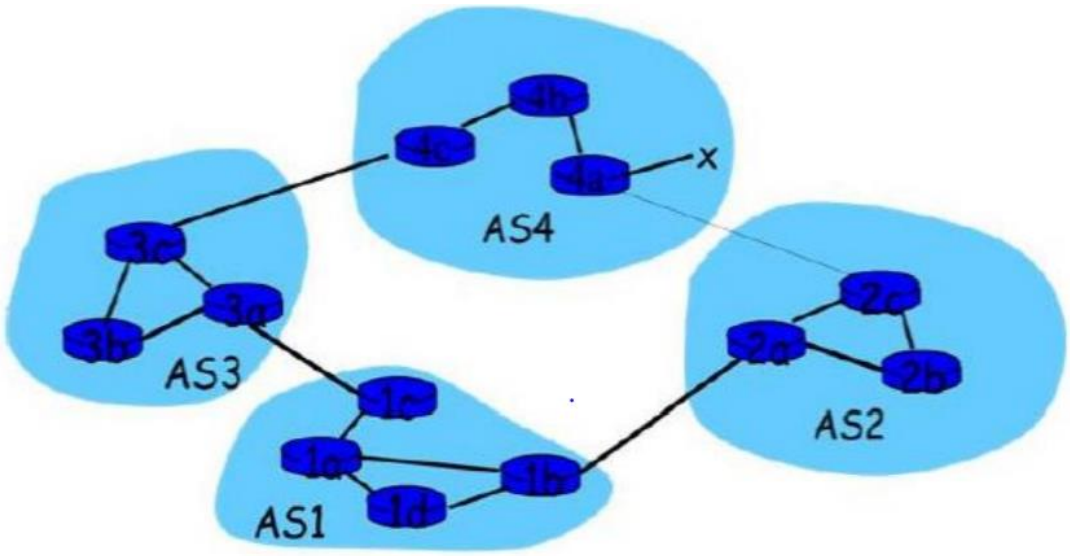
The average throughput (in terms of MSS and RTT) for this connection up through time = 6 RTT is,

In the 1st RTT, 1 MSS was sent.
In the 2nd RTT, 2 MSS was sent.
In the 3rd RTT, 3 MSS was sent.
In the 4th RTT, 4 MSS was sent.
In the 5th RTT, 5 MSS was sent.
In the 6th RTT, 6 MSS was sent.

Thus, up to time 6 RTT, 1 + 2 + 3 + 4 + 5 + 6 = 21 MSS were sent and acknowledged. Therefore, we can say that the average throughput up to time 6 RTT = (21 MSS) / (6 RTT) = 3.5 MSS/RTT

Question-2:

Consider the network shown below. Suppose AS3 and AS2 are running OSPF for their intra-AS routing protocol. Suppose AS1 and AS4 are running RIP for their intraAS routing protocol. Suppose eBGP and iBGP are used for the inter-AS routing protocol. Initially suppose there is no physical link between AS2 and AS4.



(a) Router 3b learns about prefix x from which routing protocol: OSPF, RIP, eBGP or iBGP?

Answer:

Router 3b in AS3 learns about destination x through 3c using iBGP inter-AS routing protocol. Router 3c is in AS3 gateway and it learns about the destination x connected to 4a of AS4 by using inter-AS routing protocol. So Router 3b learns about prefix x from iBGP routing protocol.

(b) Router 1c learns about prefix x from which routing protocol?

Answer:

Router 1c is in AS1 gateway and it learns about destination x that is connected to Router 4a of AS4 by using inter-AS routing protocol that is eBGP. So Router 1c learns about prefix x from eBGP routing protocol.

(c) Router 1b learns about prefix x from which routing protocol?

Answer:

Router 1b is in AS1 gateway and it learns about destination x that is connected to Router 4a of AS4 by using inter-AS routing protocol that is eBGP. So Router 1b learns about prefix x from eBGP routing protocol.

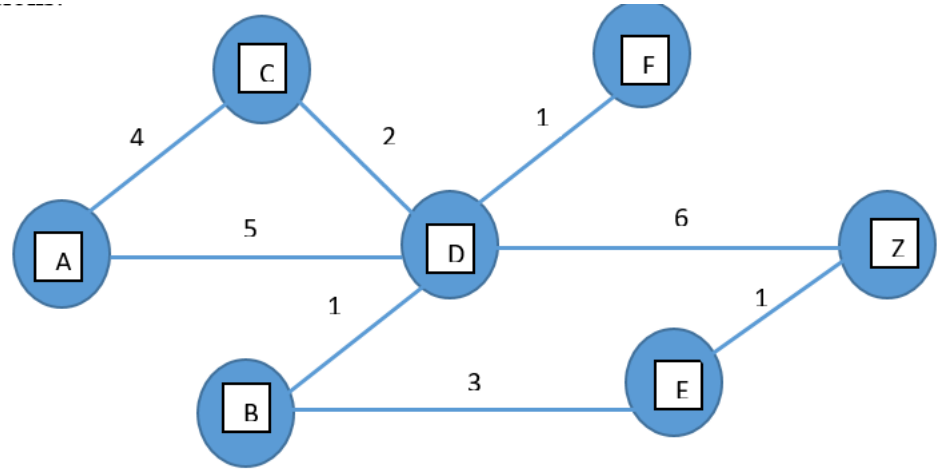
(d) Router 2a learns about prefix x from which routing protocol?

Answer:

Router 2a in AS2 learns about destination x through 2a using iBGP inter-AS routing protocol. Router 2a is in AS2 gateway and it learns about the destination x connected to 4a of AS4 by using inter-AS routing protocol. So Router 2a learns about prefix x from iBGP routing protocol.

Question-3:

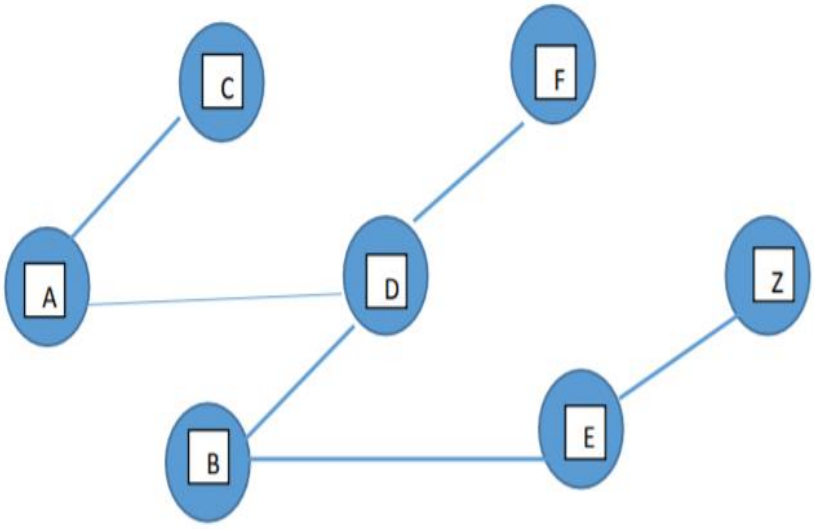
Consider the network shown below (the labels are the cost on the links). Show the operation of Dijkstra's (Link State) algorithm for computing the shortest path from A to all destinations.



Answer: The operation of Dijkstra's algorithm for computing the shortest path from A to all destinations is shown below

Step	N'	D(B) P(B)	D(C) P(C)	D(D) P(D)	D(E) P(E)	D(F) P(F)	D(Z) P(Z)
0	A	∞	4A	5A	∞	∞	∞
1	AC	∞		5A	∞	∞	∞
2	ACD	6D			∞	6D	11Z
3	ACDB				∞	6D	11Z
4	ACDBF				9B		11Z
5	ACDBFE						10E
6	ACDBFEZ						

Shortest Path:

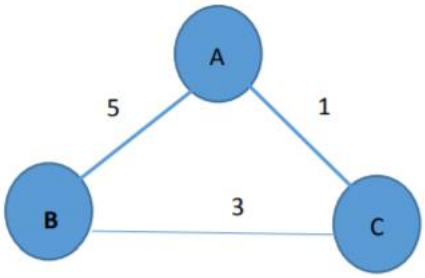


Forwarding Table:

DESTINATION	LINK
B	(A,D)
C	(A,C)
D	(A,D)
E	(A,D)
F	(A,D)
Z	(A,D)

Question-4:

Show the distance table for all nodes (A, B and C) that would be computed by the distance vector algorithm.



Answer: The distance table for all nodes (A,B and C) is shown below:

Step-1

	A	B	C
A	0	5	1
B	∞	∞	∞
C	∞	∞	∞

	A	B	C
A	∞	∞	∞
B	5	0	3
C	∞	∞	∞

	A	B	C
A	∞	∞	∞
B	∞	∞	∞
C	1	3	0

Step-2

	A	B	C
A	0	4	1
B	5	0	3
C	1	3	0

	A	B	C
A	0	5	1
B	4	0	3
C	1	3	0

	A	B	C
A	0	5	1
B	5	0	3
C	1	3	0

Step-3

	A	B	C
A	0	4	1
B	4	0	3
C	1	3	0

	A	B	C
A	0	4	1
B	4	0	3
C	1	3	0

	A	B	C
A	0	4	1
B	4	0	3
C	1	3	0

Computation:

D(A,B) = min{C(A,B) + D(B,B), C(A,C) + D(C,B)}

= min{5, 4} = 4

D(A,C) = min{C(A,C) + D(C,C), C(A,B) + D(B,C)}

= min{1, 8} = 1

D(B,A) = min{C(B,A) + D(A,A), C(B,C) + D(C,A)}

= min{5, 4} = 4

D(B,C) = min{C(B,C) + D(C,C), C(B,A) + D(A,C)}

= min{3, 6} = 3

D(C,A) = min{C(C,A) + D(A,A), C(C,B) + D(B,A)}

= min{1, 8} = 1

D(C,B) = min{C(C,B) + D(B,B), C(C,A) + D(A,B)}

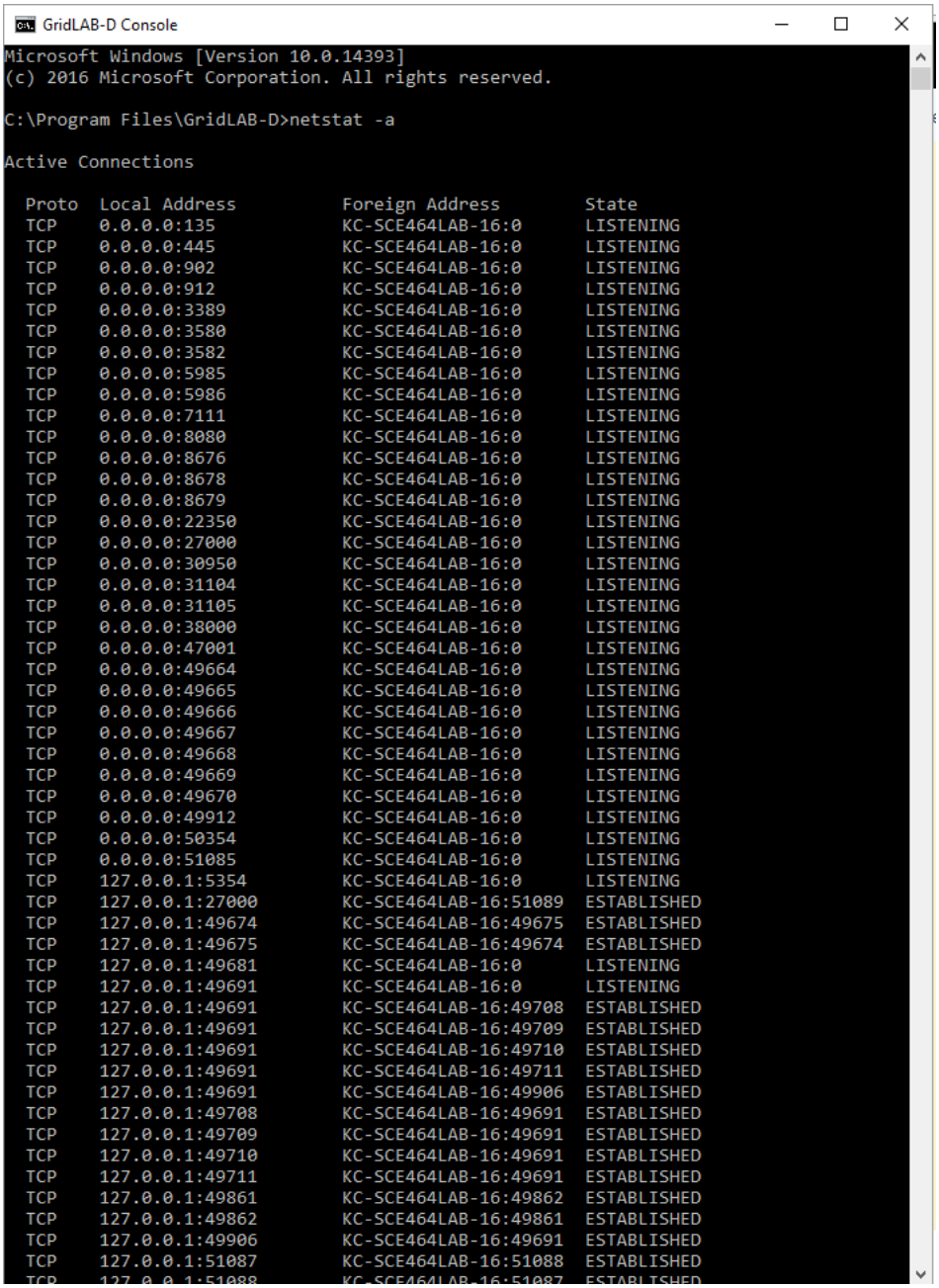
= min{3, 6} = 3

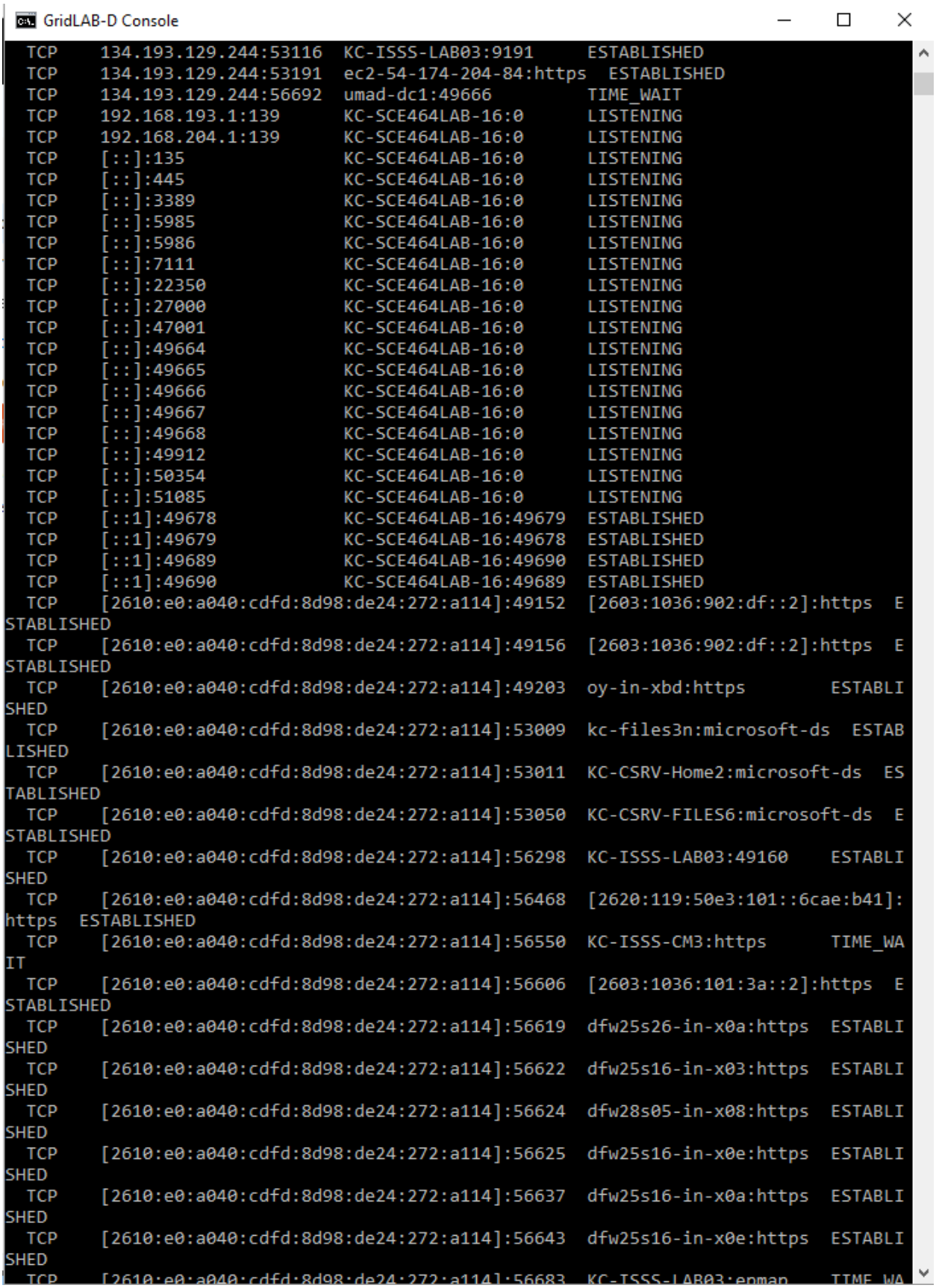
Laboratory Homework Part 1-1: using netstat (20 Points)

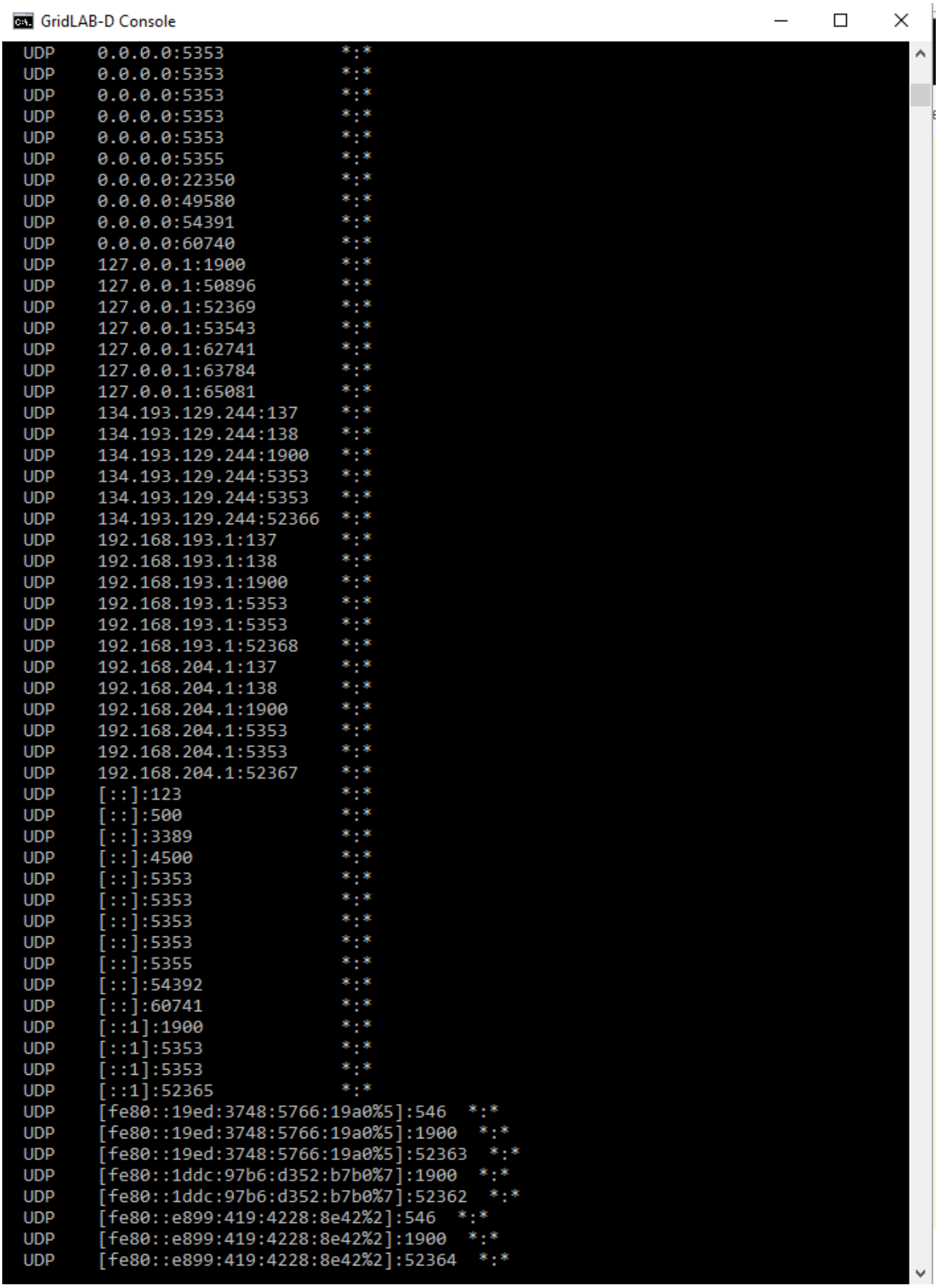
1.Use netstat on your local host to find current UDP sessions and TCP connections. How many of them do you find and what port numbers are used?

Answer:

I have found 62 TCP Connections and 86 UDP Connections. The port numbers used are shown in the below screenshot.

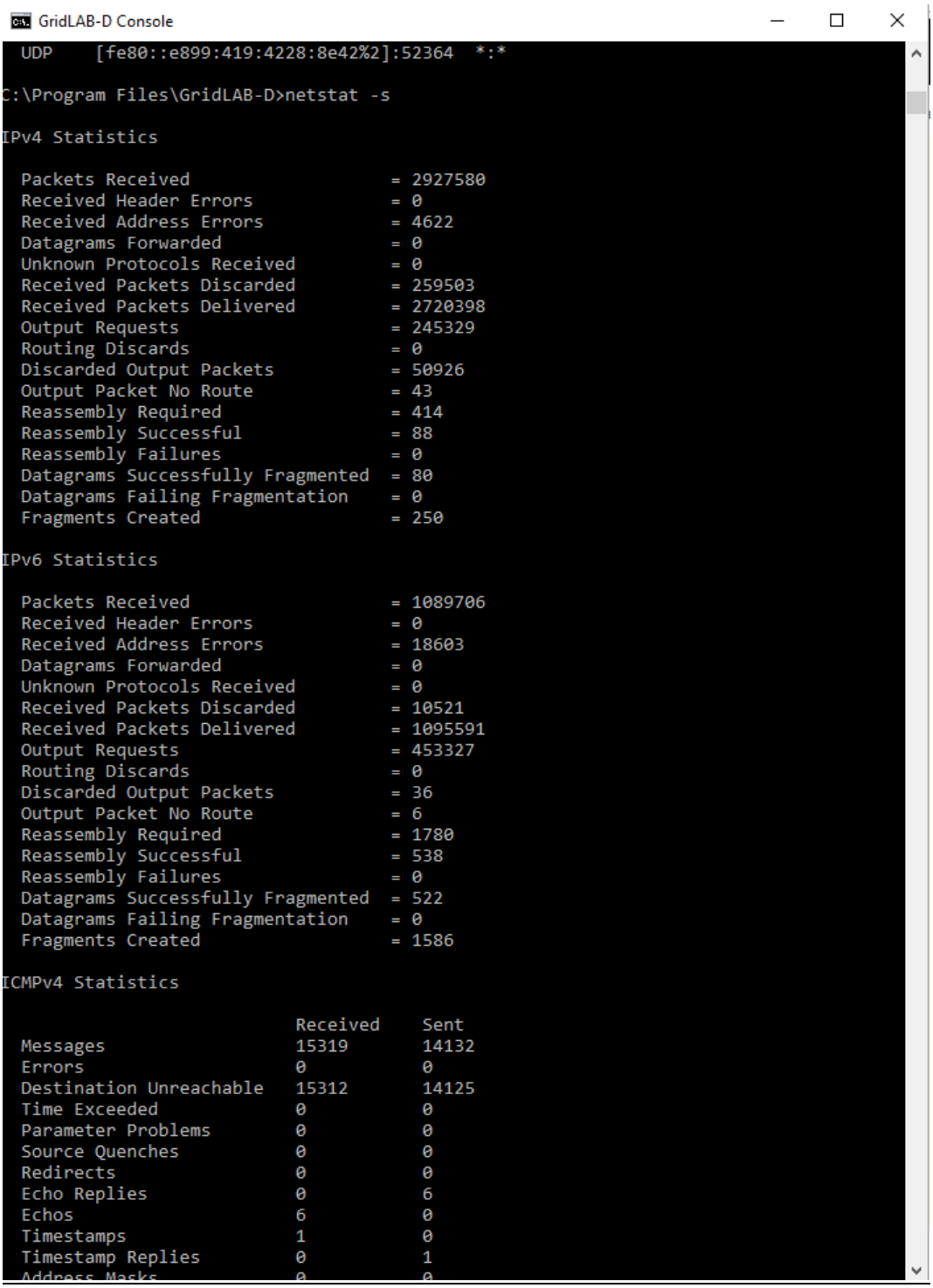

GridLAB-D Console
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.
C:\Program Files\GridLAB-D>netstat -a
Active Connections
Proto Local Address Foreign Address State
TCP 0.0.0.0:135 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:445 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:902 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:912 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:3389 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:3580 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:3582 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:5985 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:5986 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:7111 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:8080 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:8676 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:8678 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:8679 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:22350 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:27000 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:30950 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:31104 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:31105 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:38000 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:47001 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:49664 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:49665 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:49666 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:49667 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:49668 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:49669 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:49670 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:49912 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:50354 KC-SCE464LAB-16:0 LISTENING
TCP 0.0.0.0:51085 KC-SCE464LAB-16:0 LISTENING
TCP 127.0.0.1:5354 KC-SCE464LAB-16:0 LISTENING
TCP 127.0.0.1:27000 KC-SCE464LAB-16:51089 ESTABLISHED
TCP 127.0.0.1:49674 KC-SCE464LAB-16:49675 ESTABLISHED
TCP 127.0.0.1:49675 KC-SCE464LAB-16:49674 ESTABLISHED
TCP 127.0.0.1:49681 KC-SCE464LAB-16:0 LISTENING
TCP 127.0.0.1:49691 KC-SCE464LAB-16:0 LISTENING
TCP 127.0.0.1:49691 KC-SCE464LAB-16:49708 ESTABLISHED
TCP 127.0.0.1:49691 KC-SCE464LAB-16:49709 ESTABLISHED
TCP 127.0.0.1:49691 KC-SCE464LAB-16:49710 ESTABLISHED
TCP 127.0.0.1:49691 KC-SCE464LAB-16:49711 ESTABLISHED
TCP 127.0.0.1:49691 KC-SCE464LAB-16:49906 ESTABLISHED
TCP 127.0.0.1:49708 KC-SCE464LAB-16:49691 ESTABLISHED
TCP 127.0.0.1:49709 KC-SCE464LAB-16:49691 ESTABLISHED
TCP 127.0.0.1:49710 KC-SCE464LAB-16:49691 ESTABLISHED
TCP 127.0.0.1:49711 KC-SCE464LAB-16:49691 ESTABLISHED
TCP 127.0.0.1:49861 KC-SCE464LAB-16:49862 ESTABLISHED
TCP 127.0.0.1:49862 KC-SCE464LAB-16:49861 ESTABLISHED
TCP 127.0.0.1:49906 KC-SCE464LAB-16:49691 ESTABLISHED
TCP 127.0.0.1:51087 KC-SCE464LAB-16:51088 ESTABLISHED

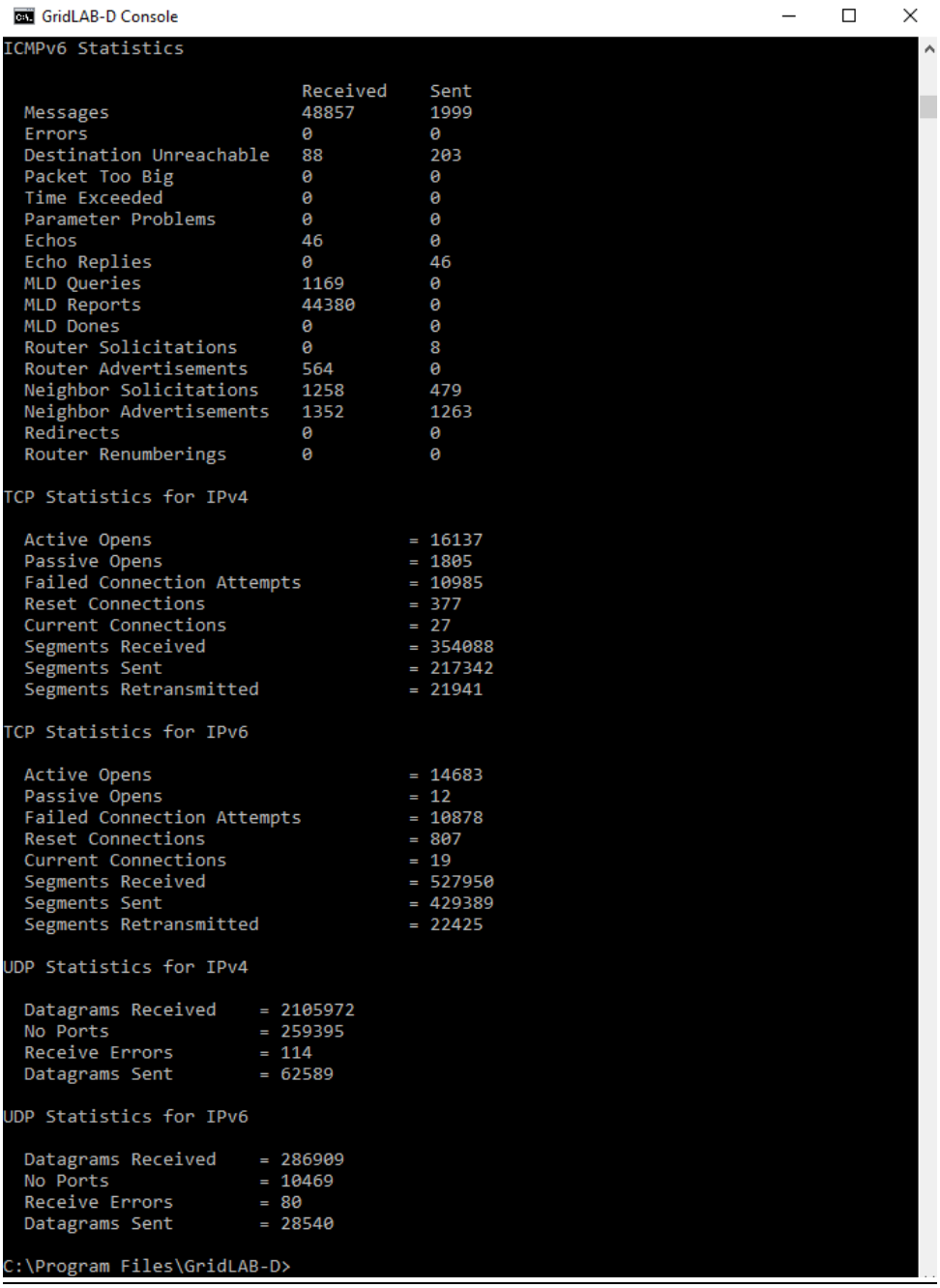

GridLAB-D Console
TCP 134.193.129.244:53116 KC-ISSS-LAB03:9191 ESTABLISHED
TCP 134.193.129.244:53191 ec2-54-174-204-84:https ESTABLISHED
TCP 134.193.129.244:56692 umad-dc1:49666 TIME_WAIT
TCP 192.168.193.1:139 KC-SCE464LAB-16:0 LISTENING
TCP 192.168.204.1:139 KC-SCE464LAB-16:0 LISTENING
TCP [::]:135 KC-SCE464LAB-16:0 LISTENING
TCP [::]:445 KC-SCE464LAB-16:0 LISTENING
TCP [::]:3389 KC-SCE464LAB-16:0 LISTENING
TCP [::]:5985 KC-SCE464LAB-16:0 LISTENING
TCP [::]:5986 KC-SCE464LAB-16:0 LISTENING
TCP [::]:7111 KC-SCE464LAB-16:0 LISTENING
TCP [::]:22350 KC-SCE464LAB-16:0 LISTENING
TCP [::]:27000 KC-SCE464LAB-16:0 LISTENING
TCP [::]:47001 KC-SCE464LAB-16:0 LISTENING
TCP [::]:49664 KC-SCE464LAB-16:0 LISTENING
TCP [::]:49665 KC-SCE464LAB-16:0 LISTENING
TCP [::]:49666 KC-SCE464LAB-16:0 LISTENING
TCP [::]:49667 KC-SCE464LAB-16:0 LISTENING
TCP [::]:49668 KC-SCE464LAB-16:0 LISTENING
TCP [::]:49912 KC-SCE464LAB-16:0 LISTENING
TCP [::]:50354 KC-SCE464LAB-16:0 LISTENING
TCP [::]:51085 KC-SCE464LAB-16:0 LISTENING
TCP [::1]:49678 KC-SCE464LAB-16:49679 ESTABLISHED
TCP [::1]:49679 KC-SCE464LAB-16:49678 ESTABLISHED
TCP [::1]:49689 KC-SCE464LAB-16:49690 ESTABLISHED
TCP [::1]:49690 KC-SCE464LAB-16:49689 ESTABLISHED
TCP [2610:e0:a040:cdfd:8d98:de24:272:a114]:49152 [2603:1036:902:df::2]:https E
STABLISHED
TCP [2610:e0:a040:cdfd:8d98:de24:272:a114]:49156 [2603:1036:902:df::2]:https E
STABLISHED
TCP [2610:e0:a040:cdfd:8d98:de24:272:a114]:49203 oy-in-xbd:https ESTABLI
SHED
TCP [2610:e0:a040:cdfd:8d98:de24:272:a114]:53009 kc-files3n:microsoft-ds ESTAB
LISHED
TCP [2610:e0:a040:cdfd:8d98:de24:272:a114]:53011 KC-CSRV-Home2:microsoft-ds ES
TABLISHED
TCP [2610:e0:a040:cdfd:8d98:de24:272:a114]:53050 KC-CSRV-FILES6:microsoft-ds E
STABLISHED
TCP [2610:e0:a040:cdfd:8d98:de24:272:a114]:56298 KC-ISSS-LAB03:49160 ESTABLI
SHED
TCP [2610:e0:a040:cdfd:8d98:de24:272:a114]:56468 [2620:119:50e3:101::6cae:b41]:
https ESTABLISHED
TCP [2610:e0:a040:cdfd:8d98:de24:272:a114]:56550 KC-ISSS-CM3:https TIME_WA
IT
TCP [2610:e0:a040:cdfd:8d98:de24:272:a114]:56606 [2603:1036:101:3a::2]:https E
STABLISHED
TCP [2610:e0:a040:cdfd:8d98:de24:272:a114]:56619 dfw25s26-in-x0a:https ESTABLI
SHED
TCP [2610:e0:a040:cdfd:8d98:de24:272:a114]:56622 dfw25s16-in-x03:https ESTABLI
SHED
TCP [2610:e0:a040:cdfd:8d98:de24:272:a114]:56624 dfw28s05-in-x08:https ESTABLI
SHED
TCP [2610:e0:a040:cdfd:8d98:de24:272:a114]:56625 dfw25s16-in-x0e:https ESTABLI
SHED
TCP [2610:e0:a040:cdfd:8d98:de24:272:a114]:56637 dfw25s16-in-x0a:https ESTABLI
SHED
TCP [2610:e0:a040:cdfd:8d98:de24:272:a114]:56643 dfw25s16-in-x0e:https ESTABLI
SHED


GridLAB-D Console
UDP 0.0.0.0:5353 *:*
UDP 0.0.0.0:5353 *:*
UDP 0.0.0.0:5353 *:*
UDP 0.0.0.0:5353 *:*
UDP 0.0.0.0:5353 *:*
UDP 0.0.0.0:5355 *:*
UDP 0.0.0.0:22350 *:*
UDP 0.0.0.0:49580 *:*
UDP 0.0.0.0:54391 *:*
UDP 0.0.0.0:60740 *:*
UDP 127.0.0.1:1900 *:*
UDP 127.0.0.1:50896 *:*
UDP 127.0.0.1:52369 *:*
UDP 127.0.0.1:53543 *:*
UDP 127.0.0.1:62741 *:*
UDP 127.0.0.1:63784 *:*
UDP 127.0.0.1:65081 *:*
UDP 134.193.129.244:137 *:*
UDP 134.193.129.244:138 *:*
UDP 134.193.129.244:1900 *:*
UDP 134.193.129.244:5353 *:*
UDP 134.193.129.244:5353 *:*
UDP 134.193.129.244:52366 *:*
UDP 192.168.193.1:137 *:*
UDP 192.168.193.1:138 *:*
UDP 192.168.193.1:1900 *:*
UDP 192.168.193.1:5353 *:*
UDP 192.168.193.1:5353 *:*
UDP 192.168.193.1:52368 *:*
UDP 192.168.204.1:137 *:*
UDP 192.168.204.1:138 *:*
UDP 192.168.204.1:1900 *:*
UDP 192.168.204.1:5353 *:*
UDP 192.168.204.1:5353 *:*
UDP 192.168.204.1:52367 *:*
UDP [::]:123 *:*
UDP [::]:500 *:*
UDP [::]:3389 *:*
UDP [::]:4500 *:*
UDP [::]:5353 *:*
UDP [::]:5353 *:*
UDP [::]:5353 *:*
UDP [::]:5353 *:*
UDP [::]:5355 *:*
UDP [::]:54392 *:*
UDP [::]:60741 *:*
UDP [::1]:1900 *:*
UDP [::1]:5353 *:*
UDP [::1]:5353 *:*
UDP [::1]:52365 *:*
UDP [fe80::19ed:3748:5766:19a0%5]:546 *:*
UDP [fe80::19ed:3748:5766:19a0%5]:1900 *:*
UDP [fe80::19ed:3748:5766:19a0%5]:52363 *:*
UDP [fe80::1ddc:97b6:d352:b7b0%7]:1900 *:*
UDP [fe80::1ddc:97b6:d352:b7b0%7]:52362 *:*
UDP [fe80::e899:419:4228:8e42%2]:546 *:*
UDP [fe80::e899:419:4228:8e42%2]:1900 *:*
UDP [fe80::e899:419:4228:8e42%2]:52364 *:*

2. Find out per-protocol (IP, ICMP, TCP and UDP) statistics (using –s option). Then try ping or tracesroute to a well-known server (eg. www.google.com). Now check per-protocol (IP, ICMP, TCP and UDP) statistics again. Summarize your findings.

Answer: The per-protocol statistcs (IP, ICMP, TCP and UDP) are found after running the netstat –s command. The output of the command results with the below output.




GridLAB-D Console
ICMPv6 Statistics
Received Sent
Messages 48857 1999
Errors 0 0
Destination Unreachable 88 203
Packet Too Big 0 0
Time Exceeded 0 0
Parameter Problems 0 0
Echos 46 0
Echo Replies 0 46
MLD Queries 1169 0
MLD Reports 44380 0
MLD Dones 0 0
Router Solicitations 0 8
Router Advertisements 564 0
Neighbor Solicitations 1258 479
Neighbor Advertisements 1352 1263
Redirects 0 0
Router Renumberings 0 0
TCP Statistics for IPv4
Active Opens = 16137
Passive Opens = 1805
Failed Connection Attempts = 10985
Reset Connections = 377
Current Connections = 27
Segments Received = 354088
Segments Sent = 217342
Segments Retransmitted = 21941
TCP Statistics for IPv6
Active Opens = 14683
Passive Opens = 12
Failed Connection Attempts = 10878
Reset Connections = 807
Current Connections = 19
Segments Received = 527950
Segments Sent = 429389
Segments Retransmitted = 22425
UDP Statistics for IPv4
Datagrams Received = 2105972
No Ports = 259395
Receive Errors = 114
Datagrams Sent = 62589
UDP Statistics for IPv6
Datagrams Received = 286909
No Ports = 10469
Receive Errors = 80
Datagrams Sent = 28540
C:\Program Files\GridLAB-D>

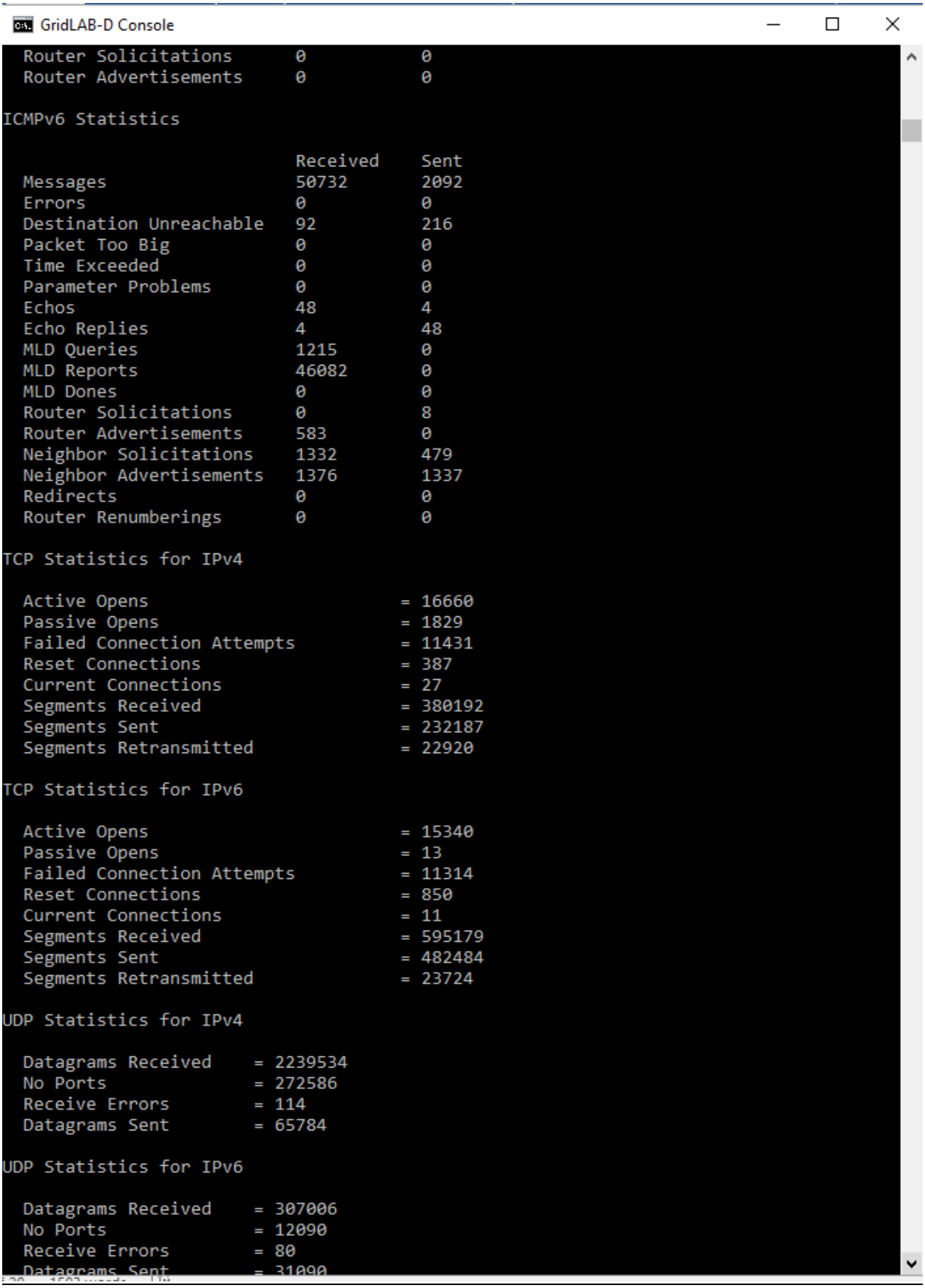
pinging www.google.com**:**

```
C:\Program Files\GridLAB-D>ping google.com

Pinging google.com [2607:f8b0:4000:80f::200e] with 32 bytes of data:
Reply from 2607:f8b0:4000:80f::200e: time=22ms
Reply from 2607:f8b0:4000:80f::200e: time=22ms
Reply from 2607:f8b0:4000:80f::200e: time=22ms
Reply from 2607:f8b0:4000:80f::200e: time=22ms

Ping statistics for 2607:f8b0:4000:80f::200e:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 22ms, Maximum = 22ms, Average = 22ms
```

Statistics after pinging www.google.com**:**

GridLAB-D Console

```
    Minimum = 22ms, Maximum = 22ms, Average = 22ms

C:\Program Files\GridLAB-D>netstat -s

IPv4 Statistics

  Packets Received                    = 3086212
  Received Header Errors              = 0
  Received Address Errors             = 4822
  Datagrams Forwarded                 = 0
  Unknown Protocols Received          = 0
  Received Packets Discarded          = 272694
  Received Packets Delivered          = 2868263
  Output Requests                     = 263566
  Routing Discards                    = 0
  Discarded Output Packets            = 52963
  Output Packet No Route              = 43
  Reassembly Required                 = 431
  Reassembly Successful               = 93
  Reassembly Failures                 = 0
  Datagrams Successfully Fragmented   = 85
  Datagrams Failing Fragmentation     = 0
  Fragments Created                   = 266

IPv6 Statistics

  Packets Received                    = 1184758
  Received Header Errors              = 0
  Received Address Errors             = 19308
  Datagrams Forwarded                 = 0
  Unknown Protocols Received          = 0
  Received Packets Discarded          = 12142
  Received Packets Delivered          = 1189692
  Output Requests                     = 508971
  Routing Discards                    = 0
  Discarded Output Packets            = 37
  Output Packet No Route              = 6
  Reassembly Required                 = 1892
  Reassembly Successful               = 565
  Reassembly Failures                 = 0
  Datagrams Successfully Fragmented   = 550
  Datagrams Failing Fragmentation     = 0
  Fragments Created                   = 1670

ICMPv4 Statistics

                            Received    Sent
  Messages                  15950       14706
  Errors                    0           0
  Destination Unreachable   15943       14699
  Time Exceeded             0           0
  Parameter Problems        0           0
  Source Quenches           0           0
  Redirects                 0           0
  Echo Replies              0           6
  Echos                     6           0
  Timestamps                1           0
  Timestamp Replies         0           1
  Address Masks             0           0
```


GridLAB-D Console
Router Solicitations 0 0
Router Advertisements 0 0
ICMPv6 Statistics
Received Sent
Messages 50732 2092
Errors 0 0
Destination Unreachable 92 216
Packet Too Big 0 0
Time Exceeded 0 0
Parameter Problems 0 0
Echos 48 4
Echo Replies 4 48
MLD Queries 1215 0
MLD Reports 46082 0
MLD Dones 0 0
Router Solicitations 0 8
Router Advertisements 583 0
Neighbor Solicitations 1332 479
Neighbor Advertisements 1376 1337
Redirects 0 0
Router Renumberings 0 0
TCP Statistics for IPv4
Active Opens = 16660
Passive Opens = 1829
Failed Connection Attempts = 11431
Reset Connections = 387
Current Connections = 27
Segments Received = 380192
Segments Sent = 232187
Segments Retransmitted = 22920
TCP Statistics for IPv6
Active Opens = 15340
Passive Opens = 13
Failed Connection Attempts = 11314
Reset Connections = 850
Current Connections = 11
Segments Received = 595179
Segments Sent = 482484
Segments Retransmitted = 23724
UDP Statistics for IPv4
Datagrams Received = 2239534
No Ports = 272586
Receive Errors = 114
Datagrams Sent = 65784
UDP Statistics for IPv6
Datagrams Received = 307006
No Ports = 12090
Receive Errors = 80
Datagrams Sent = 31090

Summarization of the Findings:

After pinging we have observed that, the statistics (IP, ICMP, TCP and UDP) got increased. (I.E we can summarize that packets sent and received increased from the above data)

Laboratory Homework Part 1-2: using ipconfig(Windows) or ifconfig (Unix/Linux)

3. What are the Physical and IP addresses of the host?

Answer: The Physical Address of the Host is 28-D2-44-31-8A-94 and the IP Address is 192.168.56.1.

4. How many bits are for the subnet mask? What is the subnet (not subnet mask) of the host?

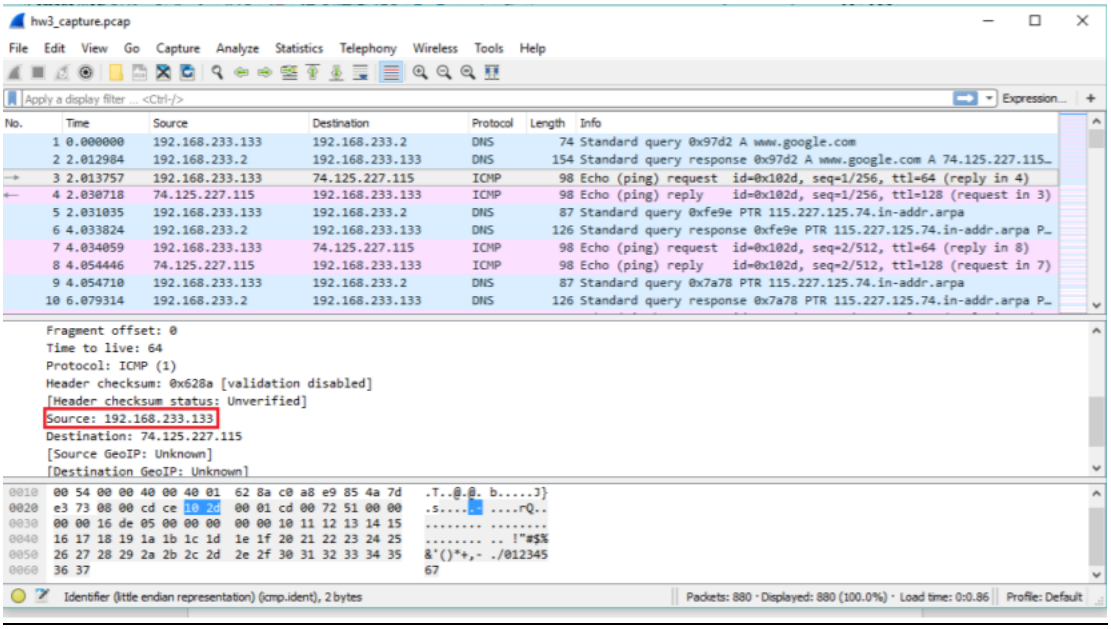
Answer:

The subnet mask is: 255.255.255.0 and the bits for the subnet mask are: 32bits (11111111 11111111 11111111 00000000). The subnet is 192.168.56.1/24. (Refer above figure)

PART2: Wireshark (35pts):

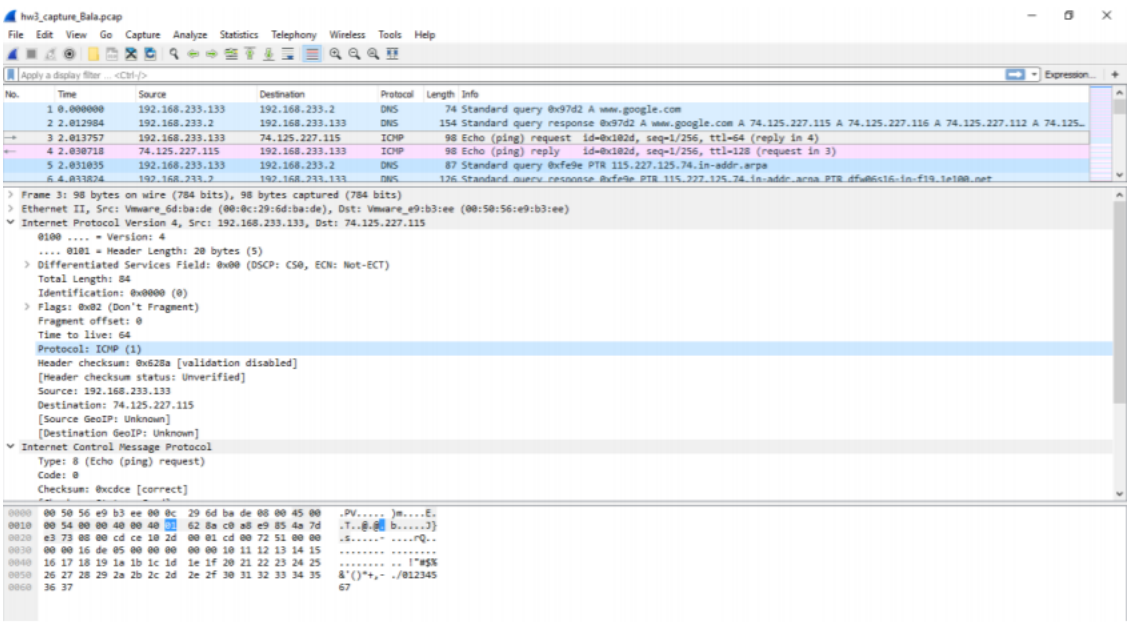
1. Select the first ICMP Echo Request message sent by the computer, and expand the Internet Protocol part of the packet in the packet details window. What is the IP address of the user's computer?

Answer: The IP address of the user's computer is: **192.168.233.133.**



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

Answer: By observing the following screenshot From the IP header, 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.

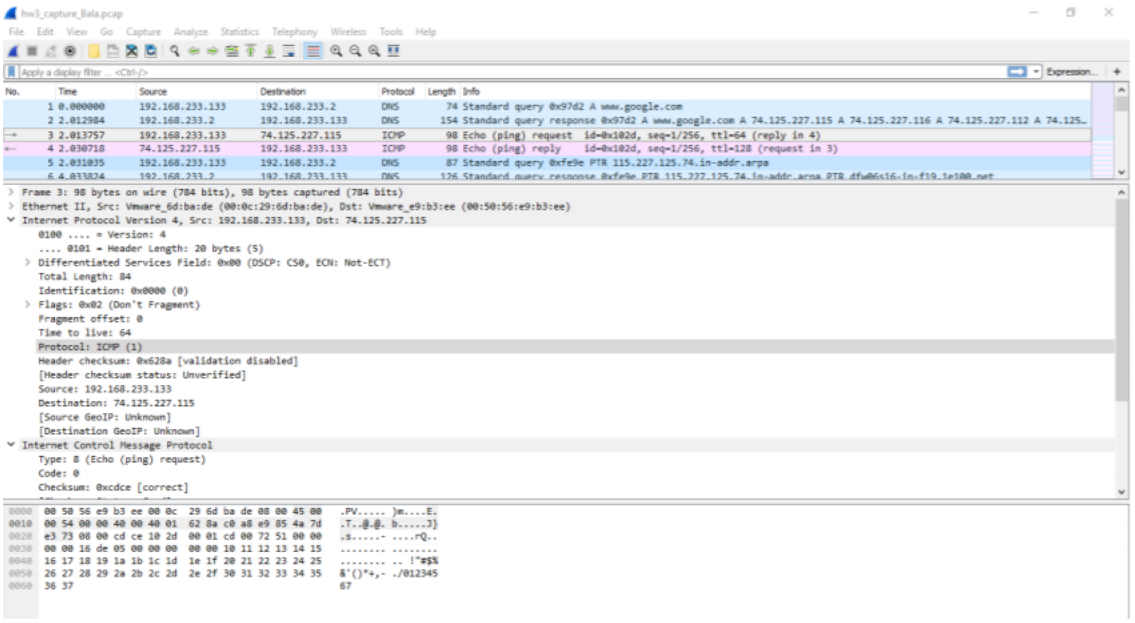
Answer: By observing the following screenshot

IP header length: 20 bytes

Total length: 84

Payload length = total length-header length =84-20

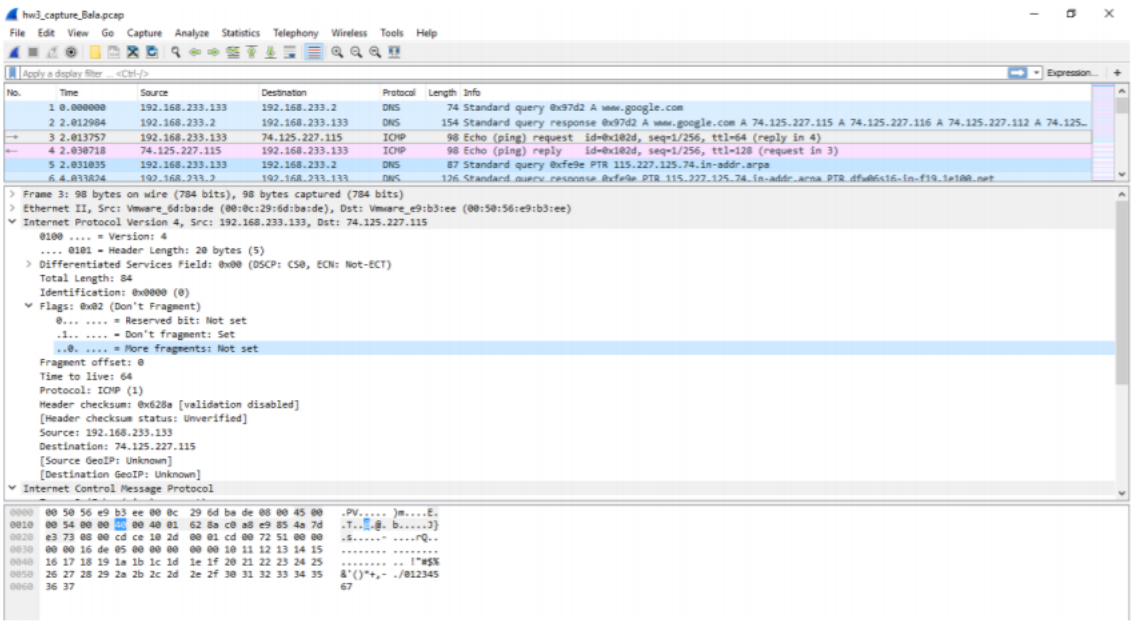
Payload length = 64



4. Has this IP datagram been fragmented? Explain how you determined whether the datagram has been fragmented.

Answer: By observing the following screenshot

The more fragments bit flag in IP datagram header has been set to **zero(0)** , so the **data is not fragmented.**

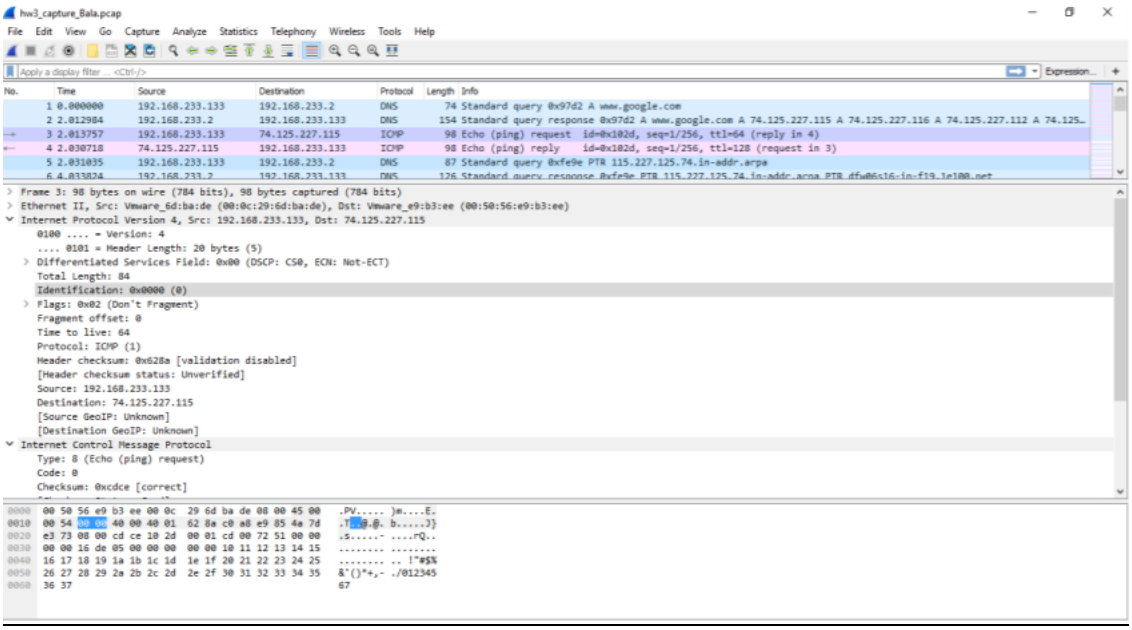


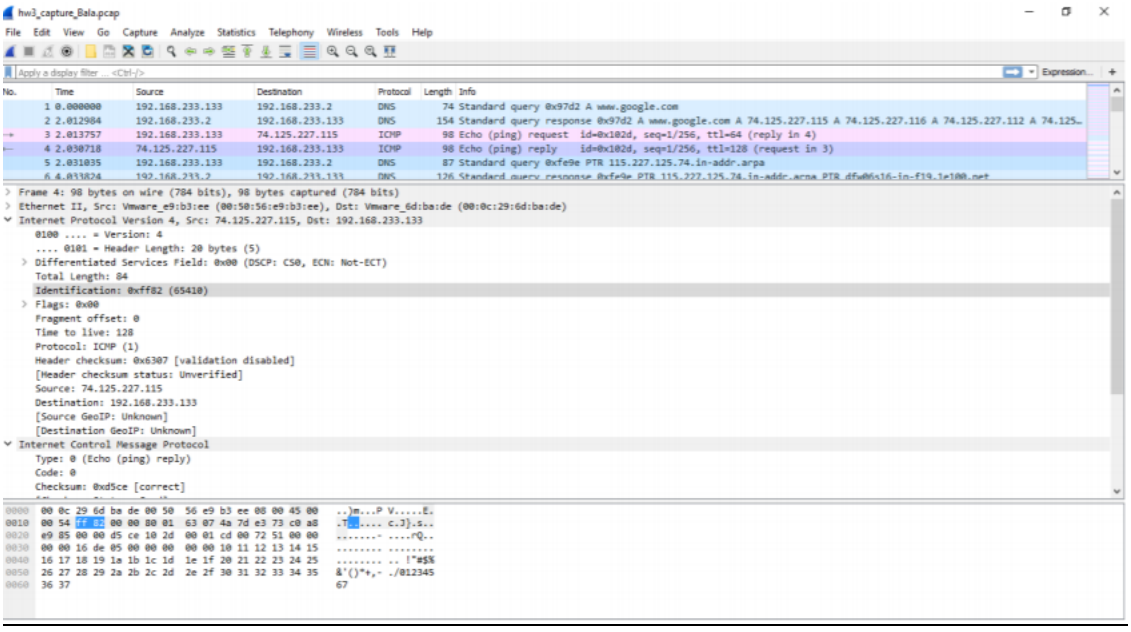
---

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

Answer: By observing the following screenshot

The following fields changes from message to message always: **Identification, Time to Live, checksum values.**





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

Answer:

The fields that stay constant across the IP datagrams are:

- Version (since we are using IPv4 for all packets)
- header length (since these are ICMP packets)
- source IP (since we are sending from the same source)

- destination IP (since we are sending to the same dest)
- Differentiated Services (since all packets are ICMP they use the same
- Type of Service class)
- Upper Layer Protocol (since these are ICMP packets)

<u>The fields that must stay constant are:</u>

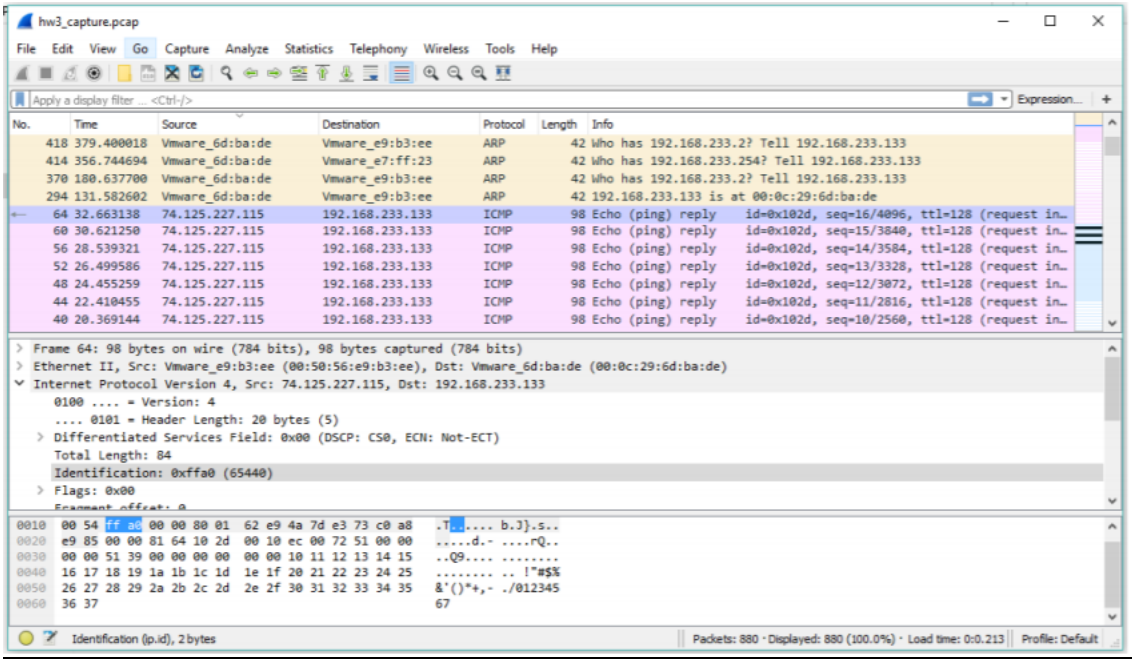
- Version (since we are using IPv4 for all packets)
- header length (since these are ICMP packets)
- source IP (since we are sending from the same source)
- destination IP (since we are sending to the same dest)
- Differentiated Services (since all packets are ICMP they use the same
- Type of Service class)
- Upper Layer Protocol (since these are ICMP packets)

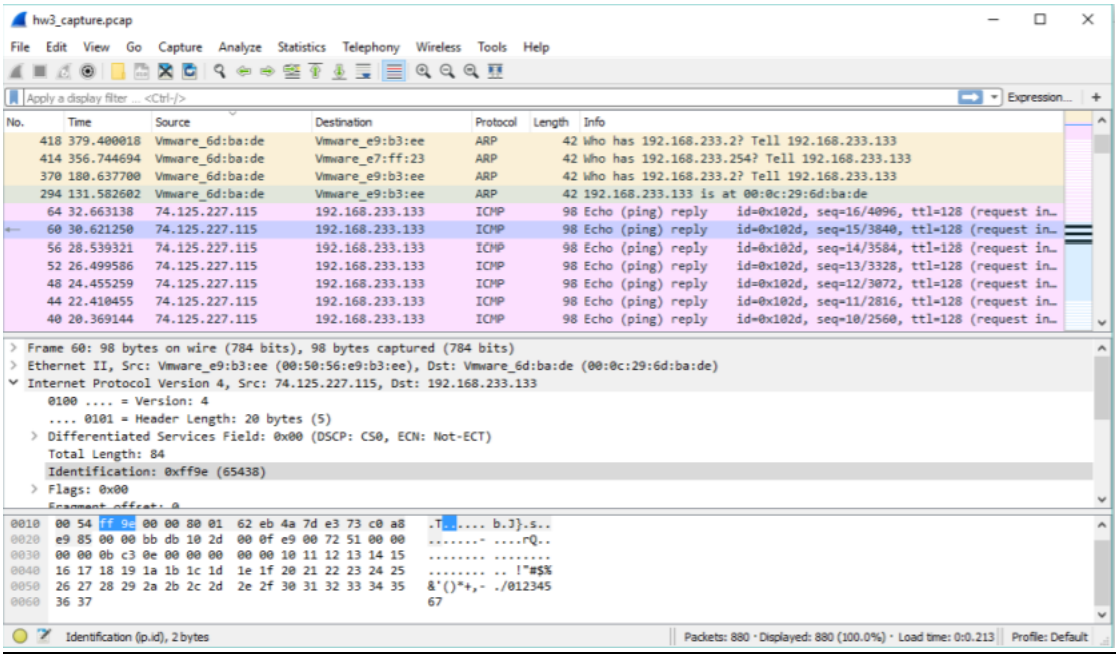
<u>The fields that must change are:</u>

- Identification (IP packets must have different ids)
- Time to live (traceroute increments each subsequent packet)
- Header checksum (since header changes, so must checksum)

7. Describe the pattern you see in the values in the Identification field and TTL field of the IP datagram in the series of ICMP TTL exceeded replies sent to the computer by the nearest (first hop) router.

<u>Answer:</u> The pattern in the IP header identification fields increment with each ICMP Echo (ping) request. Identification increases from 65440 to 65438 and Time to Live is constant in both as 128.




hw3_capture.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
No. Time Source Destination Protocol Length Info
418 379.400018 Vmware_6d:ba:de Vmware_e9:b3:ee ARP 42 Who has 192.168.233.2? Tell 192.168.233.133
414 356.744694 Vmware_6d:ba:de Vmware_e7:ff:23 ARP 42 Who has 192.168.233.254? Tell 192.168.233.133
370 180.637700 Vmware_6d:ba:de Vmware_e9:b3:ee ARP 42 Who has 192.168.233.2? Tell 192.168.233.133
294 131.582602 Vmware_6d:ba:de Vmware_e9:b3:ee ARP 42 192.168.233.133 is at 00:0c:29:6d:ba:de
64 32.663138 74.125.227.115 192.168.233.133 ICMP 98 Echo (ping) reply id=0x102d, seq=16/4096, ttl=128 (request in…
60 30.621250 74.125.227.115 192.168.233.133 ICMP 98 Echo (ping) reply id=0x102d, seq=15/3840, ttl=128 (request in…
56 28.539321 74.125.227.115 192.168.233.133 ICMP 98 Echo (ping) reply id=0x102d, seq=14/3584, ttl=128 (request in…
52 26.499586 74.125.227.115 192.168.233.133 ICMP 98 Echo (ping) reply id=0x102d, seq=13/3328, ttl=128 (request in…
48 24.455259 74.125.227.115 192.168.233.133 ICMP 98 Echo (ping) reply id=0x102d, seq=12/3072, ttl=128 (request in…
44 22.410455 74.125.227.115 192.168.233.133 ICMP 98 Echo (ping) reply id=0x102d, seq=11/2816, ttl=128 (request in…
40 20.369144 74.125.227.115 192.168.233.133 ICMP 98 Echo (ping) reply id=0x102d, seq=10/2560, ttl=128 (request in…
Frame 60: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
Ethernet II, Src: Vmware_e9:b3:ee (00:50:56:e9:b3:ee), Dst: Vmware_6d:ba:de (00:0c:29:6d:ba:de)
Internet Protocol Version 4, Src: 74.125.227.115, Dst: 192.168.233.133
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 84
Identification: 0xff9e (65438)
Flags: 0x00
0010 00 54 ff 9e 00 00 80 01 62 eb 4a 7d e3 73 c0 a8 .T..... b.J}.s..
0020 e9 85 00 00 bb db 10 2d 00 0f e9 00 72 51 00 00 ....- ....rQ..
0030 00 00 0b c3 0e 00 00 00 00 00 10 11 12 13 14 15 ....
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 .... !"#$%
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &'()*+,- ./012345
0060 36 37 67
Identification (ip.id), 2 bytes
Packets: 880 · Displayed: 880 (100.0%) · Load time: 0:0.213
Profile: Default