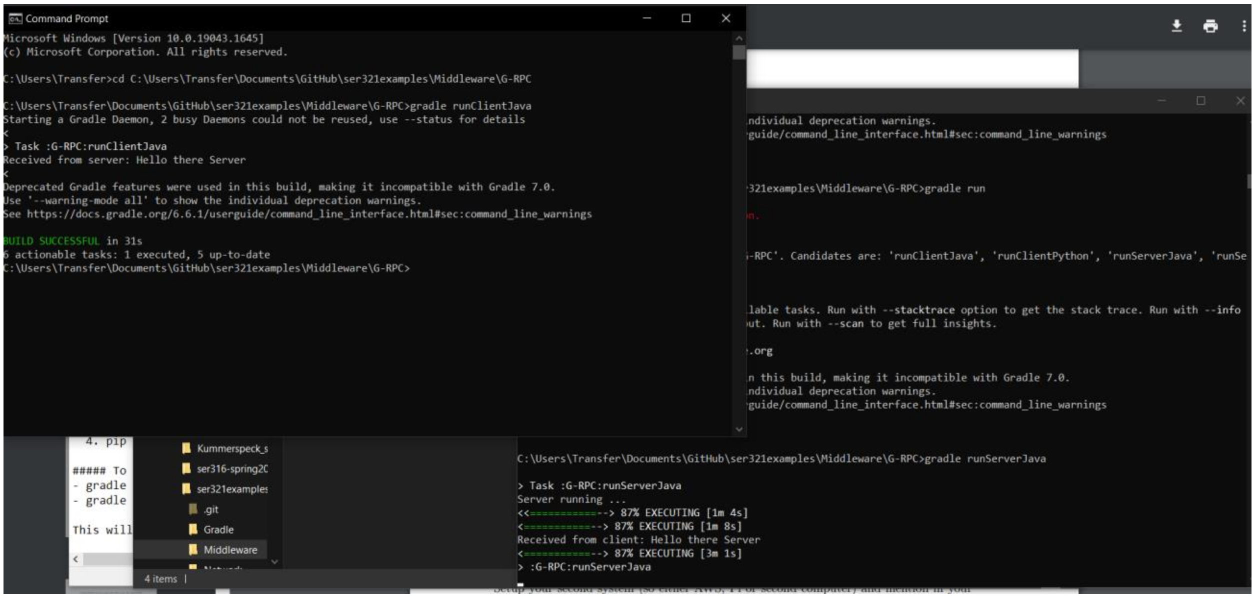
https://github.com/pwsmith4/ser321-summer2022-C-pwsmith4.git

Example 1:

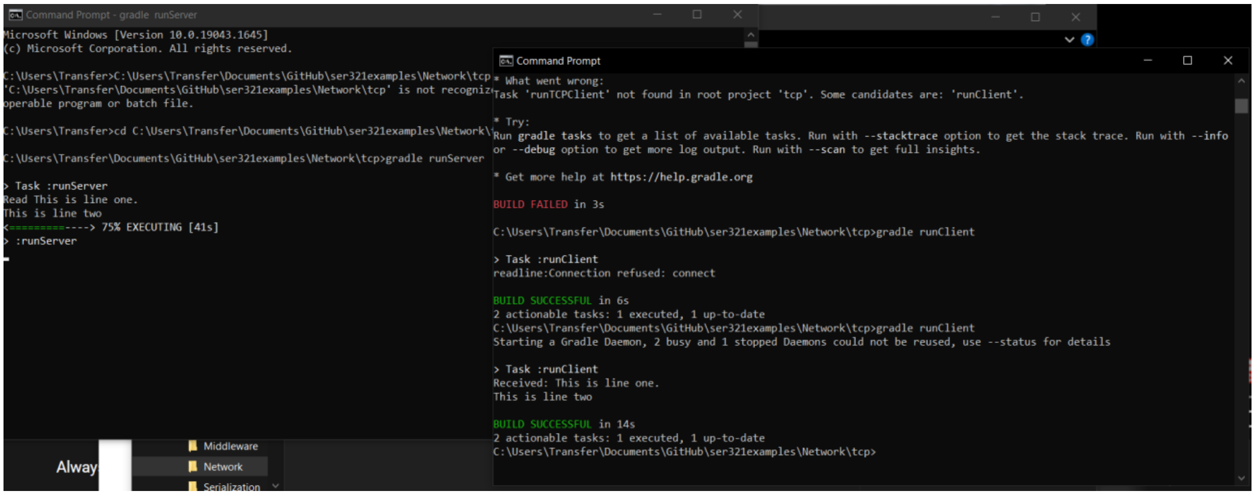


```
Microsoft Windows [Version 10.0.19043.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Transfer>cd C:\Users\Transfer\Documents\GitHub\ser321examples\Middleware\G-RPC

C:\Users\Transfer\Documents\GitHub\ser321examples\Middleware\G-RPC>gradle runClientJava
Starting a Gradle Daemon, 2 busy Daemons could not be reused, use --status for details
<
> Task :G-RPC:runClientJava
Received from server: Hello there Server
<
Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/6.6.1/userguide/command_line_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 31s
6 actionable tasks: 1 executed, 5 up-to-date
C:\Users\Transfer\Documents\GitHub\ser321examples\Middleware\G-RPC>
```

```
ndividual deprecation warnings.
guide/command_line_interface.html#sec:command_line_warnings


321examples\Middleware\G-RPC>gradle run

n.


-RPC'. Candidates are: 'runClientJava', 'runClientPython', 'runServerJava', 'runSe


lable tasks. Run with --stacktrace option to get the stack trace. Run with --info
ut. Run with --scan to get full insights.

.org

n this build, making it incompatible with Gradle 7.0.
ndividual deprecation warnings.
guide/command_line_interface.html#sec:command_line_warnings


C:\Users\Transfer\Documents\GitHub\ser321examples\Middleware\G-RPC>gradle runServerJava

> Task :G-RPC:runServerJava
Server running ...
<<===========--> 87% EXECUTING [1m 4s]
<===========--> 87% EXECUTING [1m 8s]
Received from client: Hello there Server
<===========--> 87% EXECUTING [3m 1s]
> :G-RPC:runServerJava
```

This example is from MiddleWare, G-RPC. I had to run two instances of the program. One with runServerJava and the other with runClientJava. When I typed in the runClientJava, the client stopped running and the server printed Received from client: Hello there Server.

Example 2:

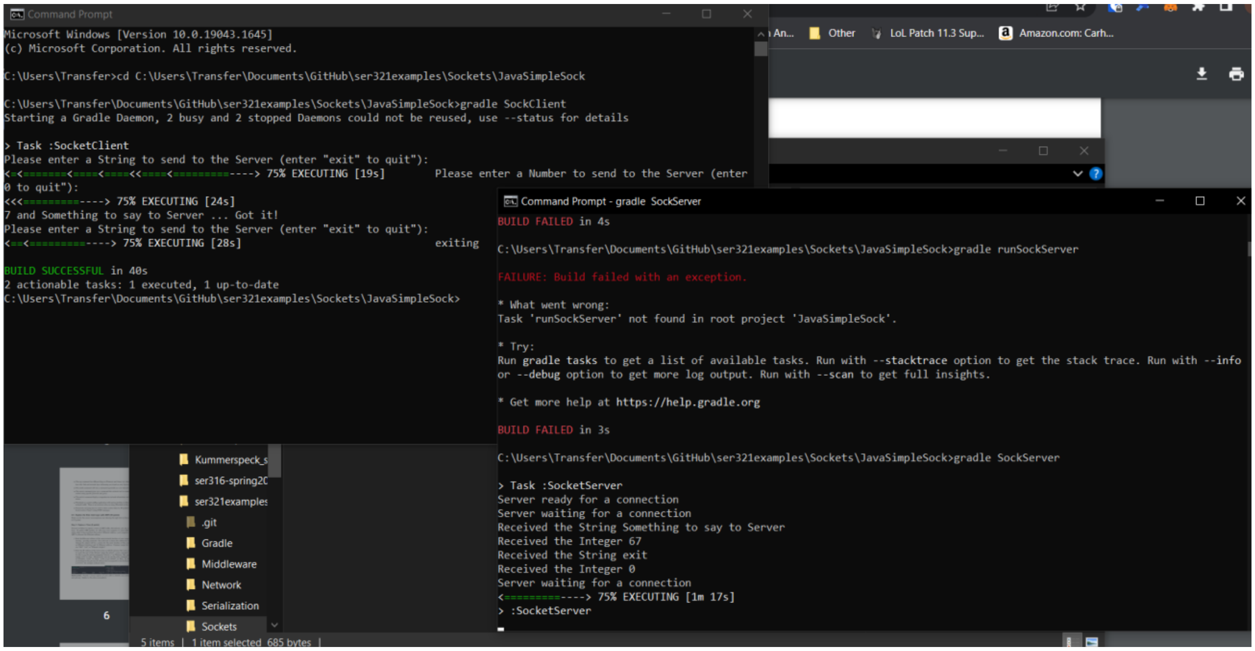


```
Microsoft Windows [Version 10.0.19043.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Transfer>C:\Users\Transfer\Documents\GitHub\ser321examples\Network\tcp
'C:\Users\Transfer\Documents\GitHub\ser321examples\Network\tcp' is not recogniz
operable program or batch file.

C:\Users\Transfer>cd C:\Users\Transfer\Documents\GitHub\ser321examples\Network\

C:\Users\Transfer\Documents\GitHub\ser321examples\Network\tcp>gradle runServer

> Task :runServer
Read This is line one.
This is line two
<=========----> 75% EXECUTING [41s]
> :runServer
```

```
* What went wrong:
Task 'runTCPClient' not found in root project 'tcp'. Some candidates are: 'runClient'.

* Try:
Run gradle tasks to get a list of available tasks. Run with --stacktrace option to get the stack trace. Run with --info
or --debug option to get more log output. Run with --scan to get full insights.

* Get more help at https://help.gradle.org

BUILD FAILED in 3s

C:\Users\Transfer\Documents\GitHub\ser321examples\Network\tcp>gradle runClient

> Task :runClient
readline:Connection refused: connect

BUILD SUCCESSFUL in 6s
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\Transfer\Documents\GitHub\ser321examples\Network\tcp>gradle runClient
Starting a Gradle Daemon, 2 busy and 1 stopped Daemons could not be reused, use --status for details

> Task :runClient
Received: This is line one.
This is line two

BUILD SUCCESSFUL in 14s
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\Transfer\Documents\GitHub\ser321examples\Network\tcp>
```

This example is from Network, tcp. After the server is running, the client can connect. This will send two messages to the server. This first line says: “Read This is line one.” And the second line says: “This is line two”. After printing these, the client stops running.

Example 3:



```
Microsoft Windows [Version 10.0.19043.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Transfer>cd C:\Users\Transfer\Documents\GitHub\ser321examples\Sockets\JavaSimpleSock

C:\Users\Transfer\Documents\GitHub\ser321examples\Sockets\JavaSimpleSock>gradle SockClient
Starting a Gradle Daemon, 2 busy and 2 stopped Daemons could not be reused, use --status for details

> Task :SocketClient
Please enter a String to send to the Server (enter "exit" to quit"):
<=<=======<====<====<<====<=========----> 75% EXECUTING [19s]      Please enter a Number to send to the Server (enter
0 to quit"):
<<<=========----> 75% EXECUTING [24s]
7 and Something to say to Server ... Got it!
Please enter a String to send to the Server (enter "exit" to quit"):
<==<=========----> 75% EXECUTING [28s]                              exiting

BUILD SUCCESSFUL in 40s
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\Transfer\Documents\GitHub\ser321examples\Sockets\JavaSimpleSock>
```

```
BUILD FAILED in 4s

C:\Users\Transfer\Documents\GitHub\ser321examples\Sockets\JavaSimpleSock>gradle runSockServer

FAILURE: Build failed with an exception.

* What went wrong:
Task 'runSockServer' not found in root project 'JavaSimpleSock'.

* Try:
Run gradle tasks to get a list of available tasks. Run with --stacktrace option to get the stack trace. Run with --info
or --debug option to get more log output. Run with --scan to get full insights.

* Get more help at https://help.gradle.org

BUILD FAILED in 3s

C:\Users\Transfer\Documents\GitHub\ser321examples\Sockets\JavaSimpleSock>gradle SockServer

> Task :SocketServer
Server ready for a connection
Server waiting for a connection
Received the String Something to say to Server
Received the Integer 67
Received the String exit
Received the Integer 0
Server waiting for a connection
<=========----> 75% EXECUTING [1m 17s]
> :SocketServer
```

This example was from Sockets, JavaSimpleSocket. The Server waits for the client and the client prompts the user to enter a string to send to the server. After the user enters the string and hits enter, the same string prints out on the server. Then it does the same thing but asks for a number instead. At any point, the user can type 'exit' and the Client stops running.

Understanding Gradle:

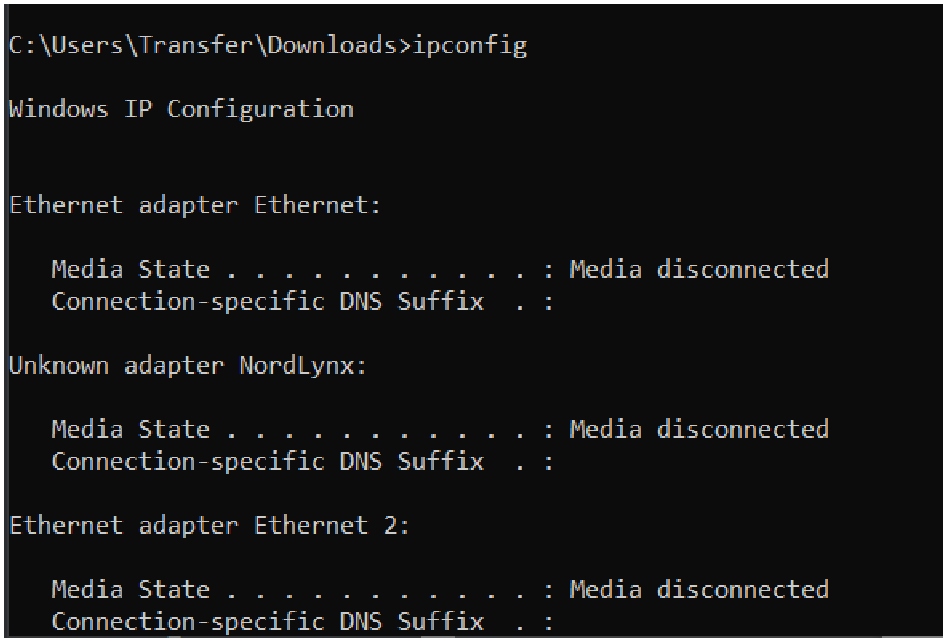
If no arguments are provided, it prints out that exactly 2 arguments need to be provided. If 1 argument is provided, it multiplies that number by 1, which would be the default number.

I used AWS as my second system.

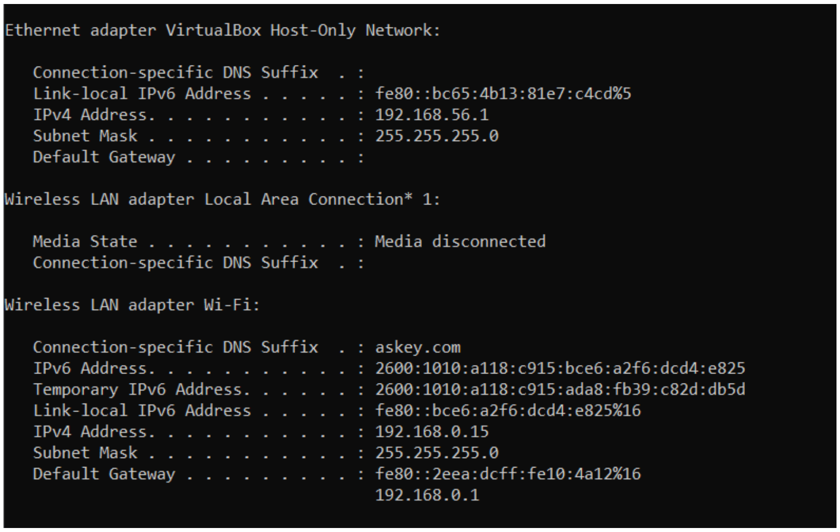
Server Message:

https://youtu.be/nstv_yhXwY0

Network Interface:

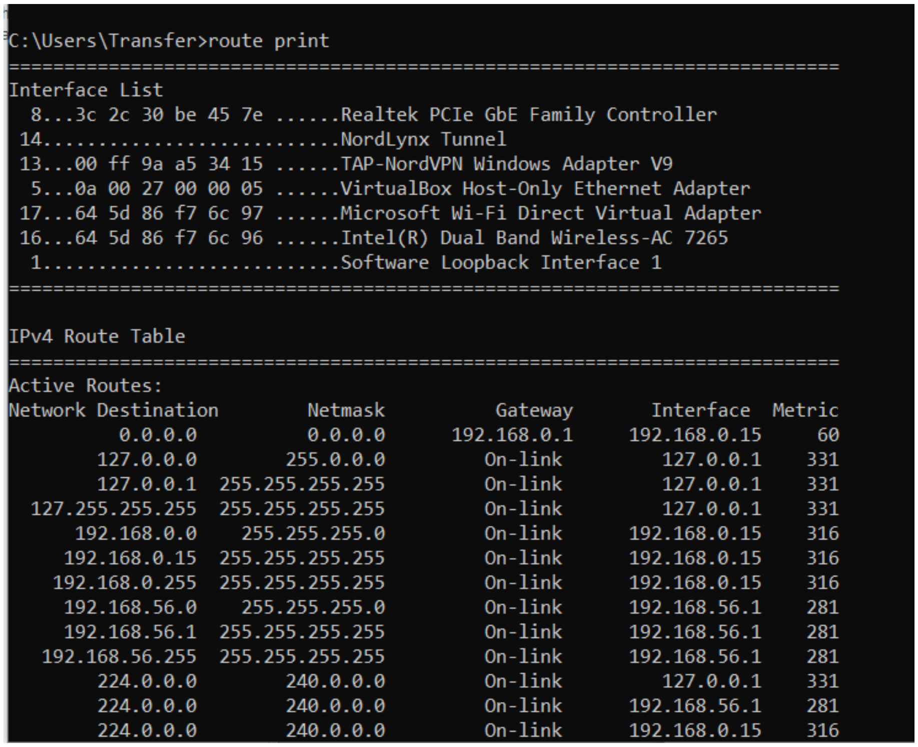


```
C:\Users\Transfer\Downloads>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Unknown adapter NordLynx:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```



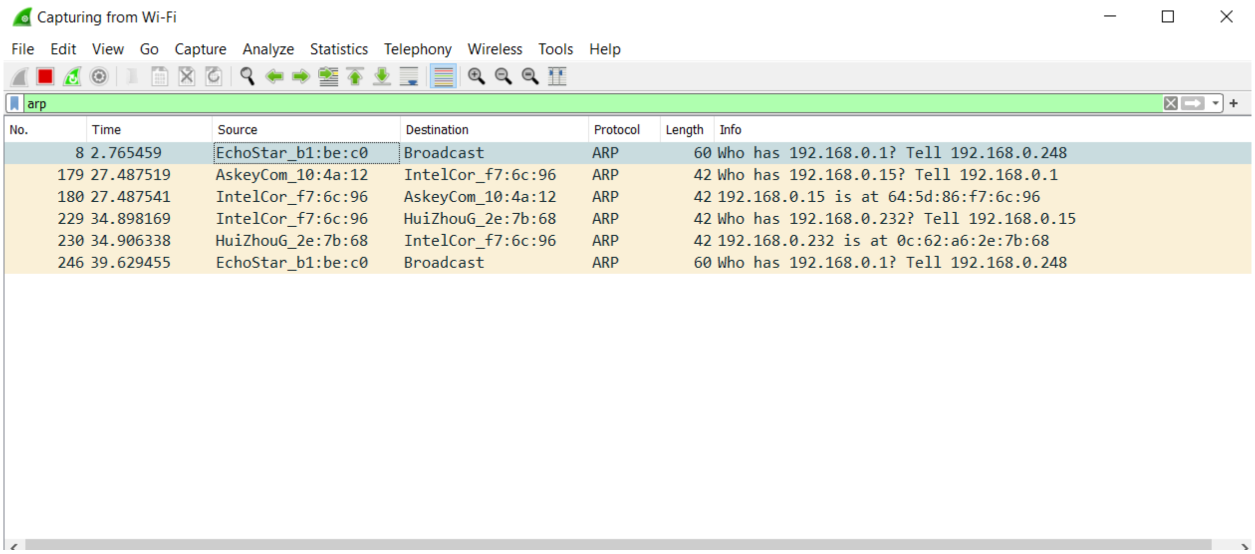
```
Ethernet adapter VirtualBox Host-Only Network:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::bc65:4b13:81e7:c4cd%5
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : askey.com
   IPv6 Address. . . . . . . . . . . : 2600:1010:a118:c915:bce6:a2f6:dcd4:e825
   Temporary IPv6 Address. . . . . . : 2600:1010:a118:c915:ada8:fb39:c82d:db5d
   Link-local IPv6 Address . . . . . : fe80::bce6:a2f6:dcd4:e825%16
   IPv4 Address. . . . . . . . . . . : 192.168.0.15
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::2eea:dcff:fe10:4a12%16
                                       192.168.0.1
```

Gateway:



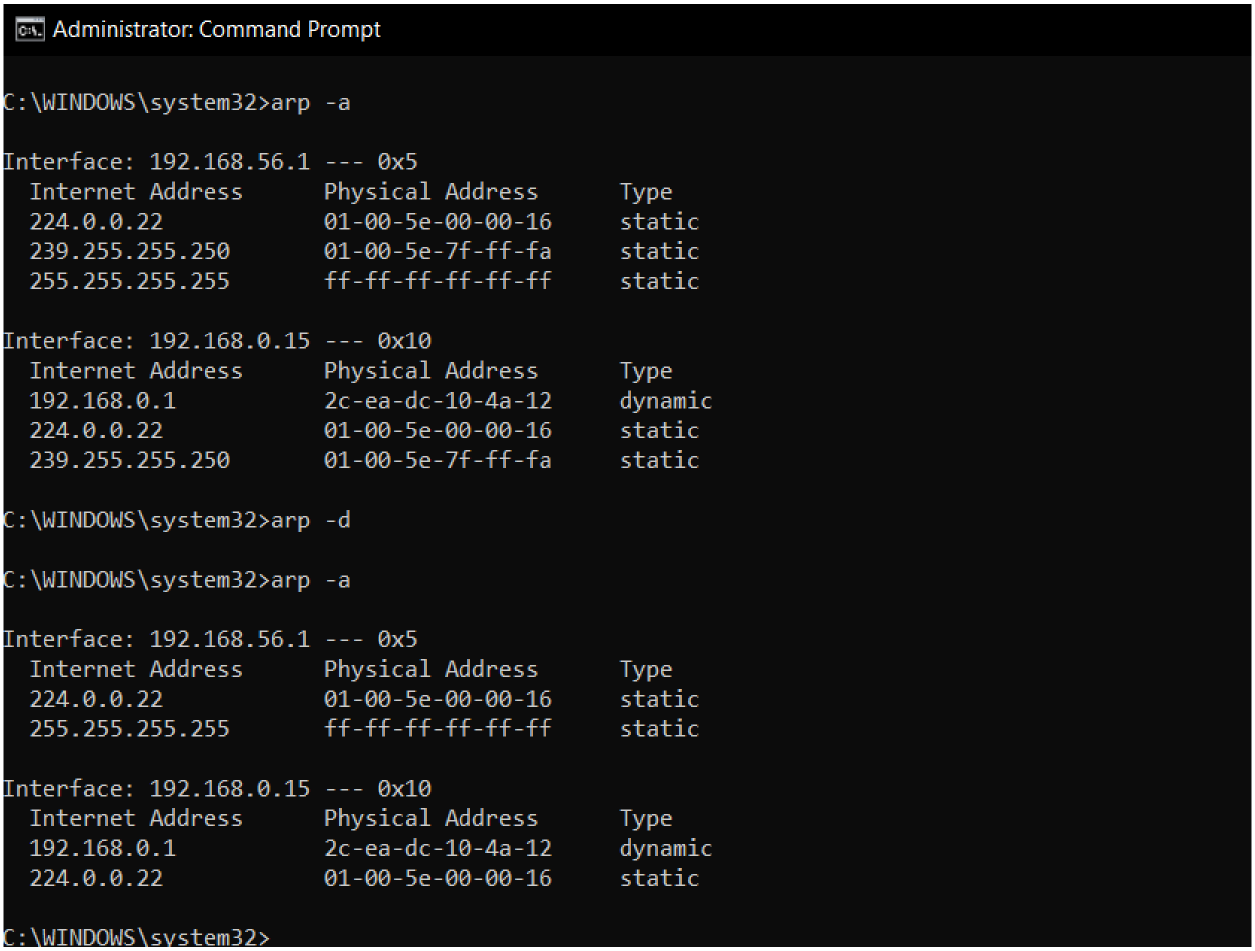
```
C:\Users\Transfer>route print
===========================================================================
Interface List
  8...3c 2c 30 be 45 7e ......Realtek PCIe GbE Family Controller
 14...........................NordLynx Tunnel
 13...00 ff 9a a5 34 15 ......TAP-NordVPN Windows Adapter V9
  5...0a 00 27 00 00 05 ......VirtualBox Host-Only Ethernet Adapter
 17...64 5d 86 f7 6c 97 ......Microsoft Wi-Fi Direct Virtual Adapter
 16...64 5d 86 f7 6c 96 ......Intel(R) Dual Band Wireless-AC 7265
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      192.168.0.1    192.168.0.15     60
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
      192.168.0.0    255.255.255.0         On-link      192.168.0.15    316
     192.168.0.15  255.255.255.255         On-link      192.168.0.15    316
    192.168.0.255  255.255.255.255         On-link      192.168.0.15    316
     192.168.56.0    255.255.255.0         On-link      192.168.56.1    281
     192.168.56.1  255.255.255.255         On-link      192.168.56.1    281
   192.168.56.255  255.255.255.255         On-link      192.168.56.1    281
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link      192.168.56.1    281
        224.0.0.0        240.0.0.0         On-link      192.168.0.15    316
```

WireShark:



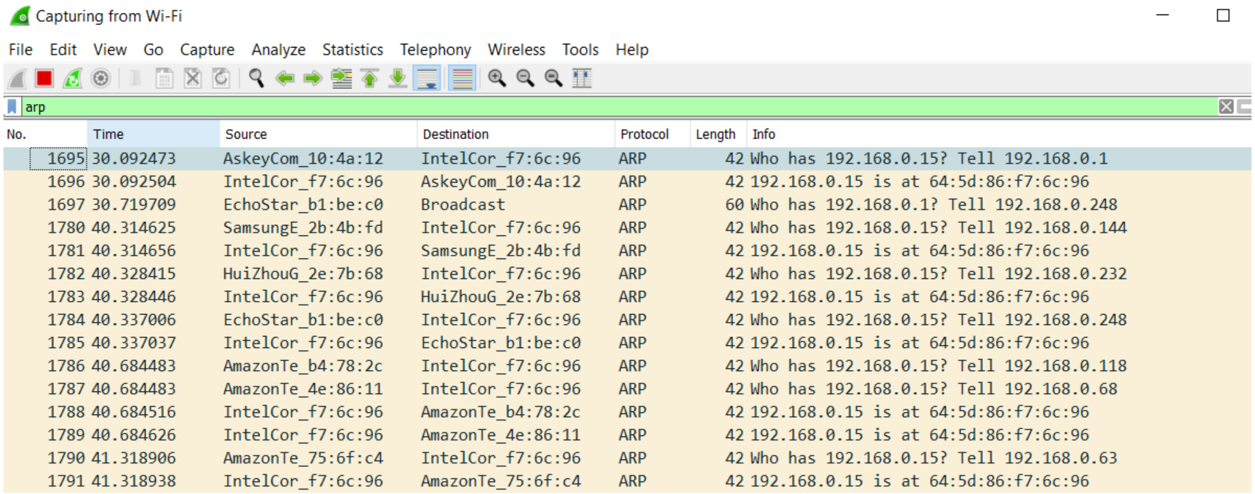
arp

No.	Time	Source	Destination	Protocol	Length	Info
8	2.765459	EchoStar_b1:be:c0	Broadcast	ARP	60	Who has 192.168.0.1? Tell 192.168.0.248
179	27.487519	AskeyCom_10:4a:12	IntelCor_f7:6c:96	ARP	42	Who has 192.168.0.15? Tell 192.168.0.1
180	27.487541	IntelCor_f7:6c:96	AskeyCom_10:4a:12	ARP	42	192.168.0.15 is at 64:5d:86:f7:6c:96
229	34.898169	IntelCor_f7:6c:96	HuiZhouG_2e:7b:68	ARP	42	Who has 192.168.0.232? Tell 192.168.0.15
230	34.906338	HuiZhouG_2e:7b:68	IntelCor_f7:6c:96	ARP	42	192.168.0.232 is at 0c:62:a6:2e:7b:68
246	39.629455	EchoStar_b1:be:c0	Broadcast	ARP	60	Who has 192.168.0.1? Tell 192.168.0.248



```
C:\WINDOWS\system32>arp -a

Interface: 192.168.56.1 --- 0x5
  Internet Address      Physical Address      Type
  224.0.0.22            01-00-5e-00-00-16     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 192.168.0.15 --- 0x10
  Internet Address      Physical Address      Type
  192.168.0.1           2c-ea-dc-10-4a-12     dynamic
  224.0.0.22            01-00-5e-00-00-16     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

C:\WINDOWS\system32>arp -d

C:\WINDOWS\system32>arp -a

Interface: 192.168.56.1 --- 0x5
  Internet Address      Physical Address      Type
  224.0.0.22            01-00-5e-00-00-16     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 192.168.0.15 --- 0x10
  Internet Address      Physical Address      Type
  192.168.0.1           2c-ea-dc-10-4a-12     dynamic
  224.0.0.22            01-00-5e-00-00-16     static

C:\WINDOWS\system32>
```

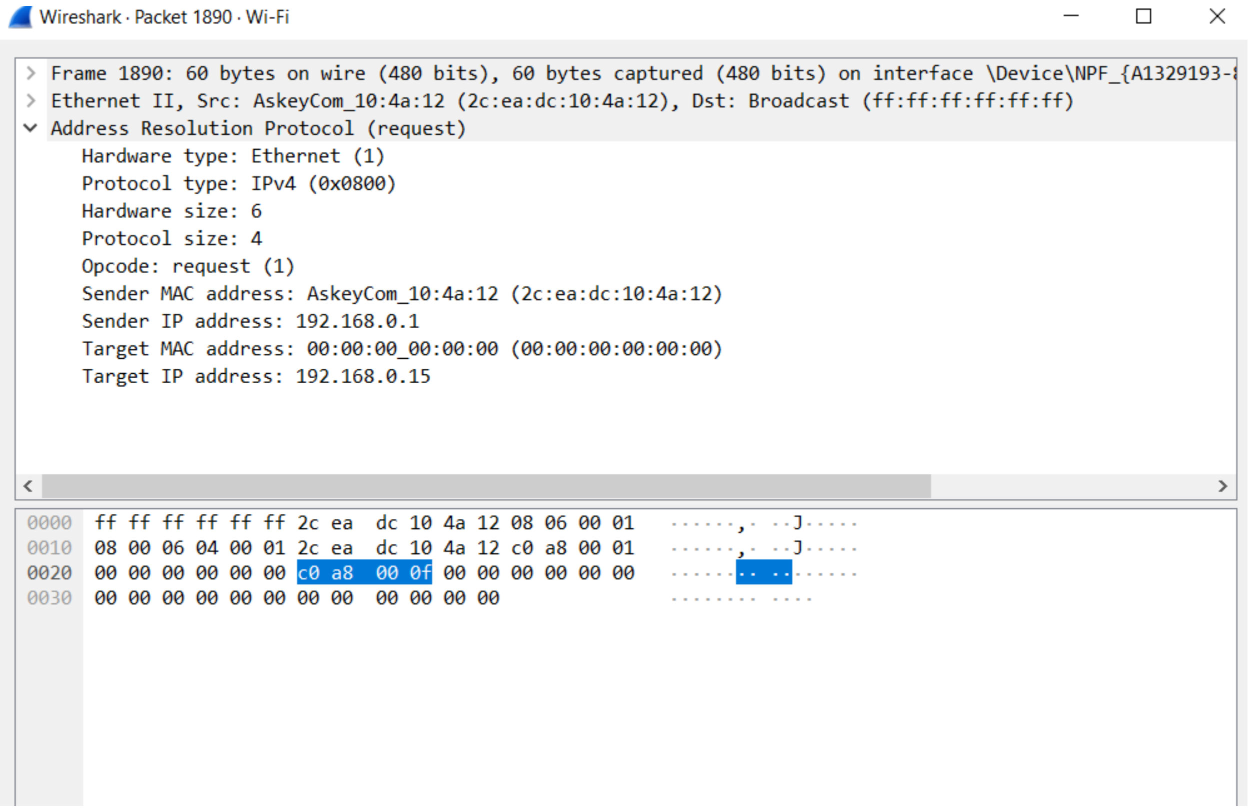
ARP Trace:



arp

No.	Time	Source	Destination	Protocol	Length	Info
1695	30.092473	AskeyCom_10:4a:12	IntelCor_f7:6c:96	ARP	42	Who has 192.168.0.15? Tell 192.168.0.1
1696	30.092504	IntelCor_f7:6c:96	AskeyCom_10:4a:12	ARP	42	192.168.0.15 is at 64:5d:86:f7:6c:96
1697	30.719709	EchoStar_b1:be:c0	Broadcast	ARP	60	Who has 192.168.0.1? Tell 192.168.0.248
1780	40.314625	SamsungE_2b:4b:fd	IntelCor_f7:6c:96	ARP	42	Who has 192.168.0.15? Tell 192.168.0.144
1781	40.314656	IntelCor_f7:6c:96	SamsungE_2b:4b:fd	ARP	42	192.168.0.15 is at 64:5d:86:f7:6c:96
1782	40.328415	HuiZhouG_2e:7b:68	IntelCor_f7:6c:96	ARP	42	Who has 192.168.0.15? Tell 192.168.0.232
1783	40.328446	IntelCor_f7:6c:96	HuiZhouG_2e:7b:68	ARP	42	192.168.0.15 is at 64:5d:86:f7:6c:96
1784	40.337006	EchoStar_b1:be:c0	IntelCor_f7:6c:96	ARP	42	Who has 192.168.0.15? Tell 192.168.0.248
1785	40.337037	IntelCor_f7:6c:96	EchoStar_b1:be:c0	ARP	42	192.168.0.15 is at 64:5d:86:f7:6c:96
1786	40.684483	AmazonTe_b4:78:2c	IntelCor_f7:6c:96	ARP	42	Who has 192.168.0.15? Tell 192.168.0.118
1787	40.684483	AmazonTe_4e:86:11	IntelCor_f7:6c:96	ARP	42	Who has 192.168.0.15? Tell 192.168.0.68
1788	40.684516	IntelCor_f7:6c:96	AmazonTe_b4:78:2c	ARP	42	192.168.0.15 is at 64:5d:86:f7:6c:96
1789	40.684626	IntelCor_f7:6c:96	AmazonTe_4e:86:11	ARP	42	192.168.0.15 is at 64:5d:86:f7:6c:96
1790	41.318906	AmazonTe_75:6f:c4	IntelCor_f7:6c:96	ARP	42	Who has 192.168.0.15? Tell 192.168.0.63
1791	41.318938	IntelCor_f7:6c:96	AmazonTe_75:6f:c4	ARP	42	192.168.0.15 is at 64:5d:86:f7:6c:96

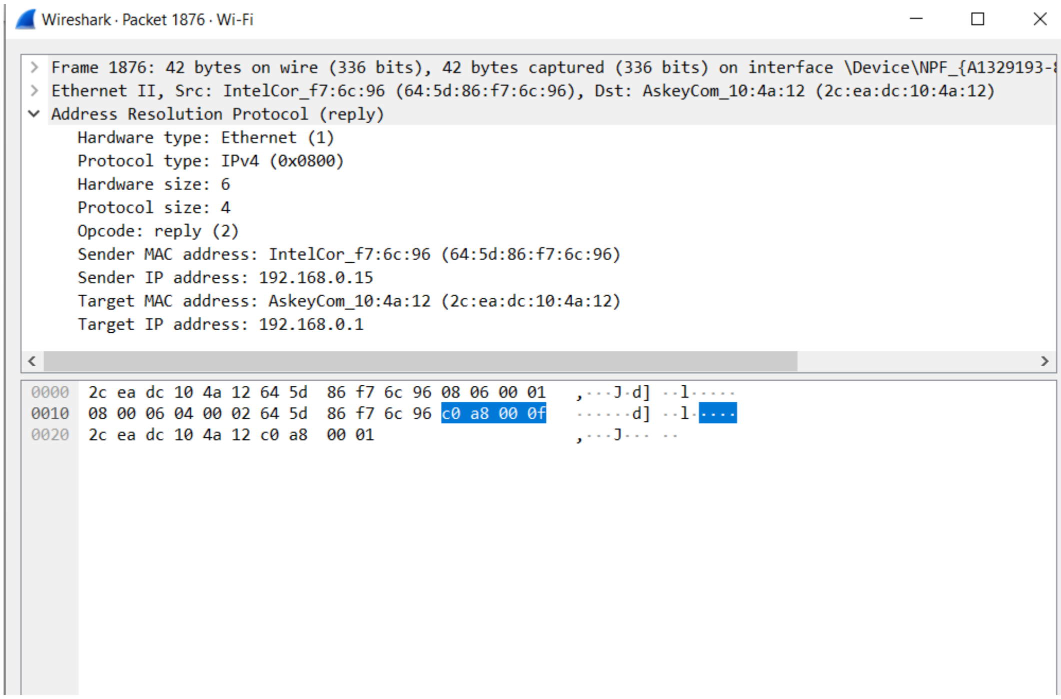
ARP Request:



```
> Frame 1890: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{A1329193-
> Ethernet II, Src: AskeyCom_10:4a:12 (2c:ea:dc:10:4a:12), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: AskeyCom_10:4a:12 (2c:ea:dc:10:4a:12)
    Sender IP address: 192.168.0.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.0.15
```

```
0000  ff ff ff ff ff ff 2c ea  dc 10 4a 12 08 06 00 01   ······,· ··J·····
0010  08 00 06 04 00 01 2c ea  dc 10 4a 12 c0 a8 00 01   ······,· ··J·····
0020  00 00 00 00 00 00 c0 a8  00 0f 00 00 00 00 00 00   ········ ········
0030  00 00 00 00 00 00 00 00  00 00 00 00               ········ ····
```

ARP Reply:



```
> Frame 1876: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{A1329193-
> Ethernet II, Src: IntelCor_f7:6c:96 (64:5d:86:f7:6c:96), Dst: AskeyCom_10:4a:12 (2c:ea:dc:10:4a:12)
v Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: IntelCor_f7:6c:96 (64:5d:86:f7:6c:96)
    Sender IP address: 192.168.0.15
    Target MAC address: AskeyCom_10:4a:12 (2c:ea:dc:10:4a:12)
    Target IP address: 192.168.0.1
```

```
0000  2c ea dc 10 4a 12 64 5d  86 f7 6c 96 08 06 00 01   ,···J·d] ··l·····
0010  08 00 06 04 00 02 64 5d  86 f7 6c 96 c0 a8 00 0f   ······d] ··l·····
0020  2c ea dc 10 4a 12 c0 a8  00 01                     ,···J··· ··
```

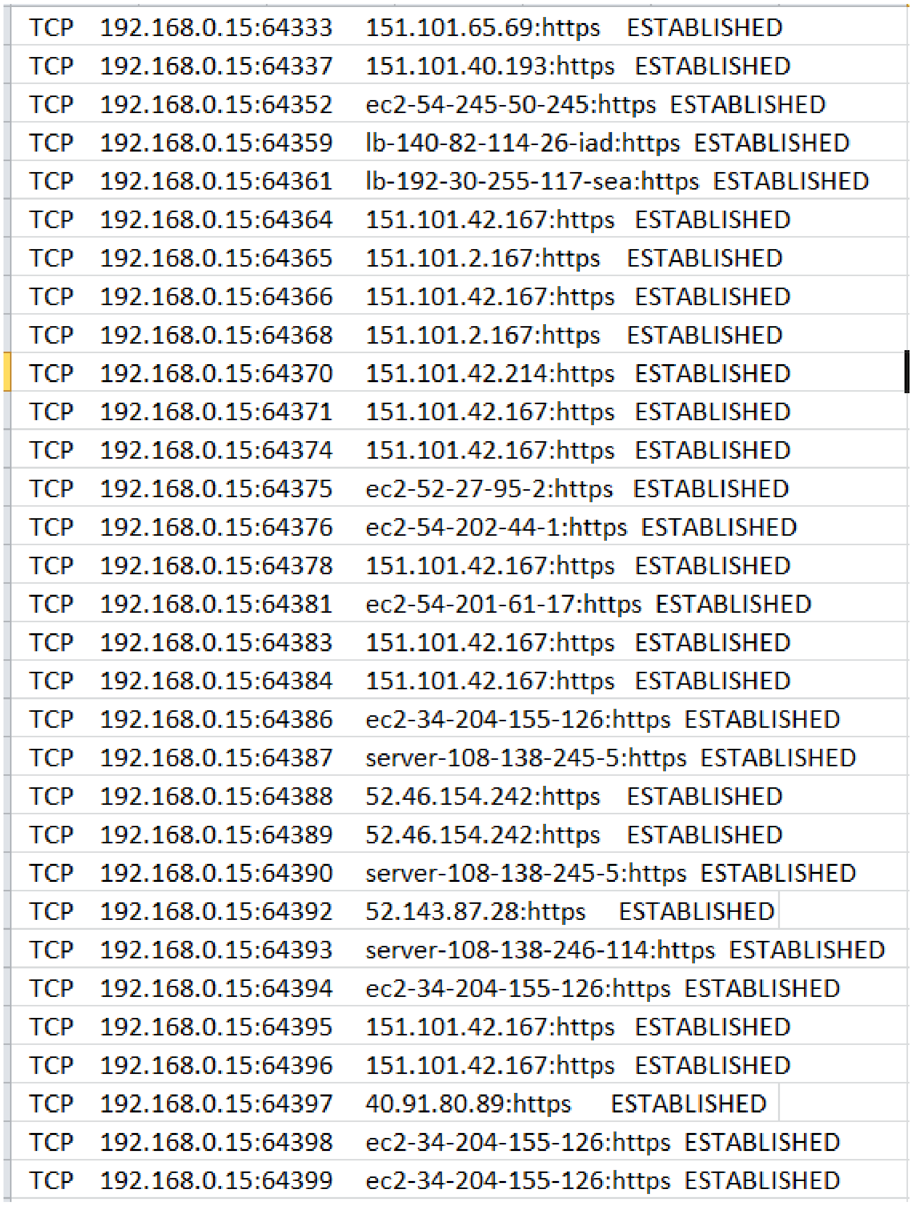
Details of ARP:

Opcode of 1 is used for Request and Opcode of 2 is used for Reply.

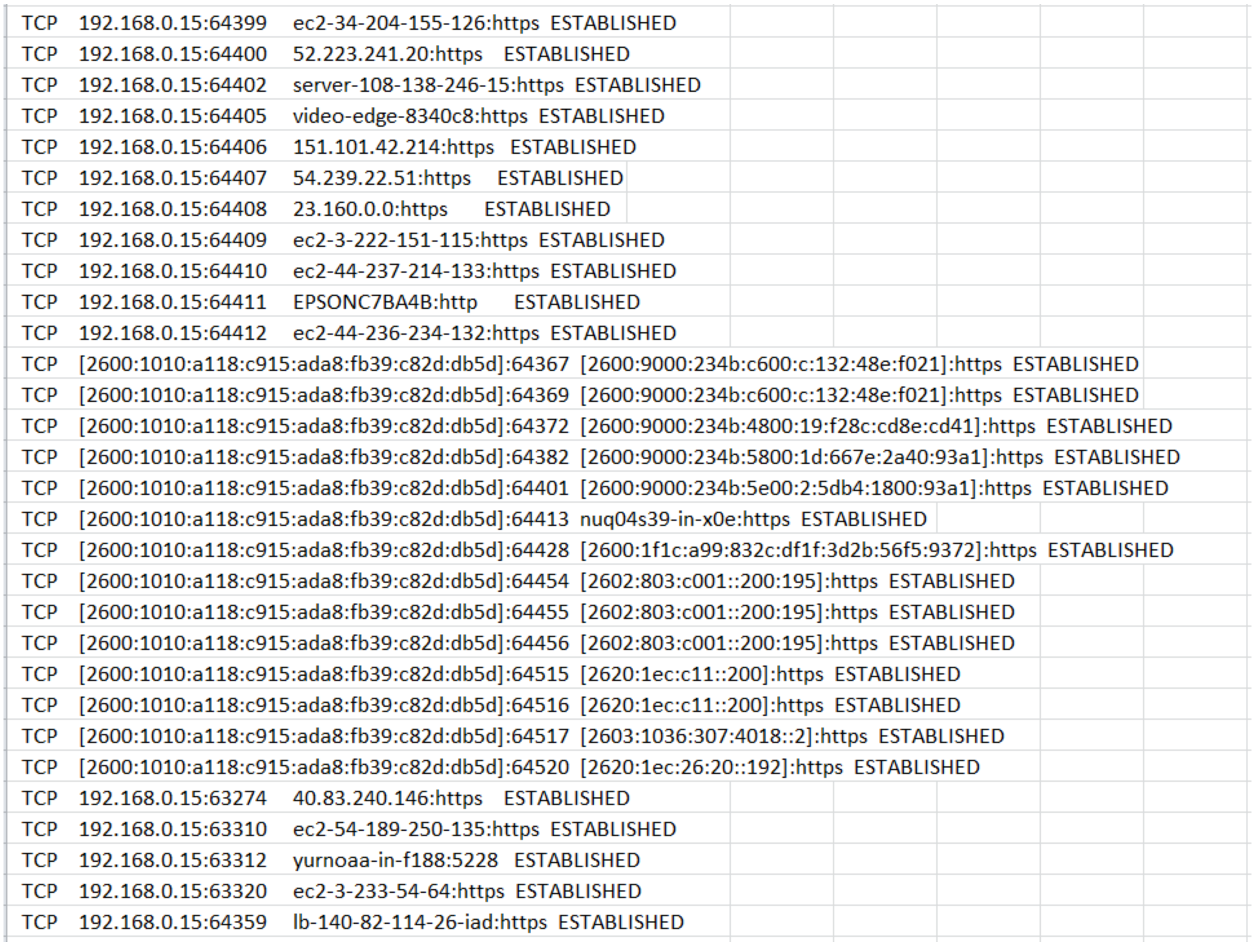
The ARP request and reply header are both 28 bytes according to kevincurran.org

The value would be ff ff ff ff ff ff

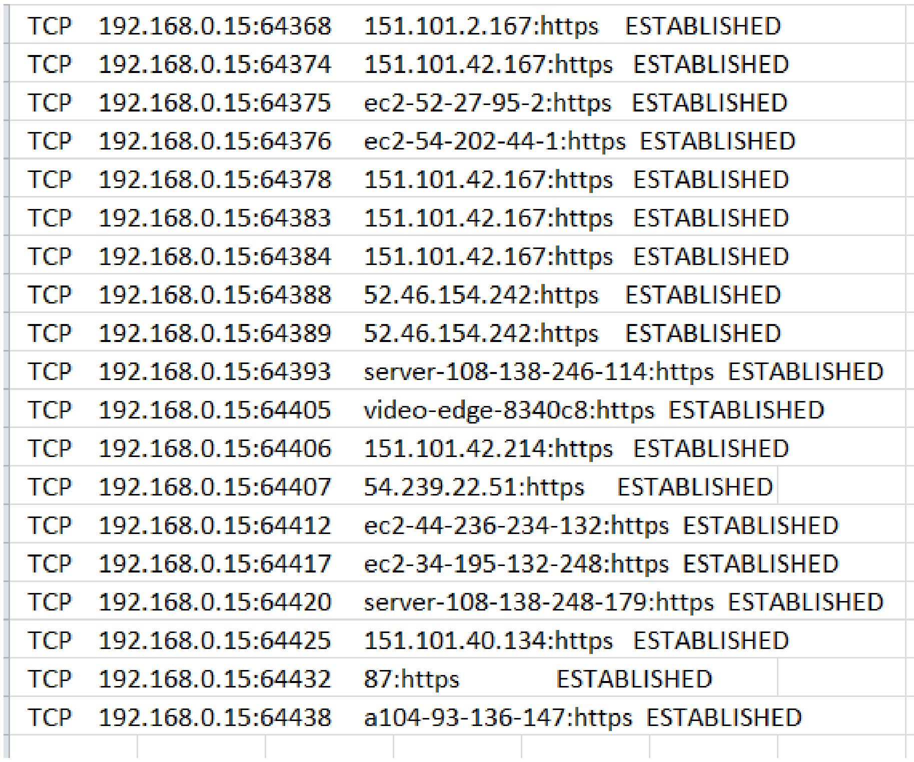
It is a type 1 for ARP.



```
TCP   192.168.0.15:64333   151.101.65.69:https   ESTABLISHED
TCP   192.168.0.15:64337   151.101.40.193:https  ESTABLISHED
TCP   192.168.0.15:64352   ec2-54-245-50-245:https  ESTABLISHED
TCP   192.168.0.15:64359   lb-140-82-114-26-iad:https  ESTABLISHED
TCP   192.168.0.15:64361   lb-192-30-255-117-sea:https  ESTABLISHED
TCP   192.168.0.15:64364   151.101.42.167:https  ESTABLISHED
TCP   192.168.0.15:64365   151.101.2.167:https   ESTABLISHED
TCP   192.168.0.15:64366   151.101.42.167:https  ESTABLISHED
TCP   192.168.0.15:64368   151.101.2.167:https   ESTABLISHED
TCP   192.168.0.15:64370   151.101.42.214:https  ESTABLISHED
TCP   192.168.0.15:64371   151.101.42.167:https  ESTABLISHED
TCP   192.168.0.15:64374   151.101.42.167:https  ESTABLISHED
TCP   192.168.0.15:64375   ec2-52-27-95-2:https  ESTABLISHED
TCP   192.168.0.15:64376   ec2-54-202-44-1:https  ESTABLISHED
TCP   192.168.0.15:64378   151.101.42.167:https  ESTABLISHED
TCP   192.168.0.15:64381   ec2-54-201-61-17:https  ESTABLISHED
TCP   192.168.0.15:64383   151.101.42.167:https  ESTABLISHED
TCP   192.168.0.15:64384   151.101.42.167:https  ESTABLISHED
TCP   192.168.0.15:64386   ec2-34-204-155-126:https  ESTABLISHED
TCP   192.168.0.15:64387   server-108-138-245-5:https  ESTABLISHED
TCP   192.168.0.15:64388   52.46.154.242:https   ESTABLISHED
TCP   192.168.0.15:64389   52.46.154.242:https   ESTABLISHED
TCP   192.168.0.15:64390   server-108-138-245-5:https  ESTABLISHED
TCP   192.168.0.15:64392   52.143.87.28:https    ESTABLISHED
TCP   192.168.0.15:64393   server-108-138-246-114:https  ESTABLISHED
TCP   192.168.0.15:64394   ec2-34-204-155-126:https  ESTABLISHED
TCP   192.168.0.15:64395   151.101.42.167:https  ESTABLISHED
TCP   192.168.0.15:64396   151.101.42.167:https  ESTABLISHED
TCP   192.168.0.15:64397   40.91.80.89:https     ESTABLISHED
TCP   192.168.0.15:64398   ec2-34-204-155-126:https  ESTABLISHED
TCP   192.168.0.15:64399   ec2-34-204-155-126:https  ESTABLISHED
```

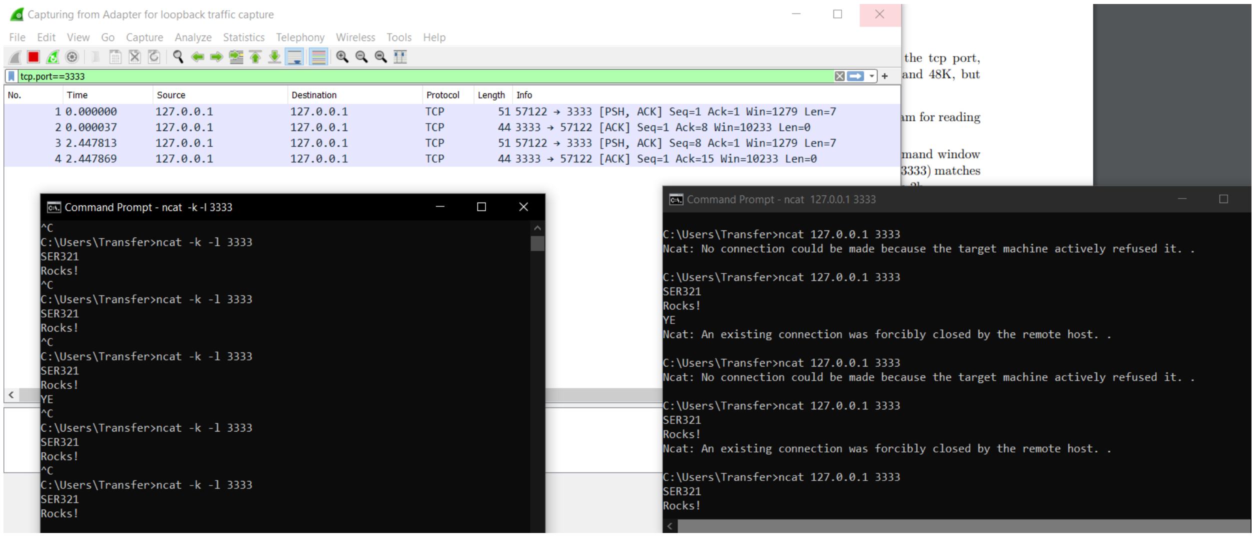


```
TCP   192.168.0.15:64399    ec2-34-204-155-126:https  ESTABLISHED
TCP   192.168.0.15:64400    52.223.241.20:https    ESTABLISHED
TCP   192.168.0.15:64402    server-108-138-246-15:https  ESTABLISHED
TCP   192.168.0.15:64405    video-edge-8340c8:https  ESTABLISHED
TCP   192.168.0.15:64406    151.101.42.214:https   ESTABLISHED
TCP   192.168.0.15:64407    54.239.22.51:https    ESTABLISHED
TCP   192.168.0.15:64408    23.160.0.0:https      ESTABLISHED
TCP   192.168.0.15:64409    ec2-3-222-151-115:https  ESTABLISHED
TCP   192.168.0.15:64410    ec2-44-237-214-133:https  ESTABLISHED
TCP   192.168.0.15:64411    EPSONC7BA4B:http      ESTABLISHED
TCP   192.168.0.15:64412    ec2-44-236-234-132:https  ESTABLISHED
TCP   [2600:1010:a118:c915:ada8:fb39:c82d:db5d]:64367  [2600:9000:234b:c600:c:132:48e:f021]:https  ESTABLISHED
TCP   [2600:1010:a118:c915:ada8:fb39:c82d:db5d]:64369  [2600:9000:234b:c600:c:132:48e:f021]:https  ESTABLISHED
TCP   [2600:1010:a118:c915:ada8:fb39:c82d:db5d]:64372  [2600:9000:234b:4800:19:f28c:cd8e:cd41]:https  ESTABLISHED
TCP   [2600:1010:a118:c915:ada8:fb39:c82d:db5d]:64382  [2600:9000:234b:5800:1d:667e:2a40:93a1]:https  ESTABLISHED
TCP   [2600:1010:a118:c915:ada8:fb39:c82d:db5d]:64401  [2600:9000:234b:5e00:2:5db4:1800:93a1]:https  ESTABLISHED
TCP   [2600:1010:a118:c915:ada8:fb39:c82d:db5d]:64413  nuq04s39-in-x0e:https  ESTABLISHED
TCP   [2600:1010:a118:c915:ada8:fb39:c82d:db5d]:64428  [2600:1f1c:a99:832c:df1f:3d2b:56f5:9372]:https  ESTABLISHED
TCP   [2600:1010:a118:c915:ada8:fb39:c82d:db5d]:64454  [2602:803:c001::200:195]:https  ESTABLISHED
TCP   [2600:1010:a118:c915:ada8:fb39:c82d:db5d]:64455  [2602:803:c001::200:195]:https  ESTABLISHED
TCP   [2600:1010:a118:c915:ada8:fb39:c82d:db5d]:64456  [2602:803:c001::200:195]:https  ESTABLISHED
TCP   [2600:1010:a118:c915:ada8:fb39:c82d:db5d]:64515  [2620:1ec:c11::200]:https  ESTABLISHED
TCP   [2600:1010:a118:c915:ada8:fb39:c82d:db5d]:64516  [2620:1ec:c11::200]:https  ESTABLISHED
TCP   [2600:1010:a118:c915:ada8:fb39:c82d:db5d]:64517  [2603:1036:307:4018::2]:https  ESTABLISHED
TCP   [2600:1010:a118:c915:ada8:fb39:c82d:db5d]:64520  [2620:1ec:26:20::192]:https  ESTABLISHED
TCP   192.168.0.15:63274    40.83.240.146:https    ESTABLISHED
TCP   192.168.0.15:63310    ec2-54-189-250-135:https  ESTABLISHED
TCP   192.168.0.15:63312    yurnoaa-in-f188:5228   ESTABLISHED
TCP   192.168.0.15:63320    ec2-3-233-54-64:https  ESTABLISHED
TCP   192.168.0.15:64359    lb-140-82-114-26-iad:https  ESTABLISHED
```



```
TCP   192.168.0.15:64368    151.101.2.167:https    ESTABLISHED
TCP   192.168.0.15:64374    151.101.42.167:https   ESTABLISHED
TCP   192.168.0.15:64375    ec2-52-27-95-2:https   ESTABLISHED
TCP   192.168.0.15:64376    ec2-54-202-44-1:https  ESTABLISHED
TCP   192.168.0.15:64378    151.101.42.167:https   ESTABLISHED
TCP   192.168.0.15:64383    151.101.42.167:https   ESTABLISHED
TCP   192.168.0.15:64384    151.101.42.167:https   ESTABLISHED
TCP   192.168.0.15:64388    52.46.154.242:https    ESTABLISHED
TCP   192.168.0.15:64389    52.46.154.242:https    ESTABLISHED
TCP   192.168.0.15:64393    server-108-138-246-114:https  ESTABLISHED
TCP   192.168.0.15:64405    video-edge-8340c8:https  ESTABLISHED
TCP   192.168.0.15:64406    151.101.42.214:https   ESTABLISHED
TCP   192.168.0.15:64407    54.239.22.51:https     ESTABLISHED
TCP   192.168.0.15:64412    ec2-44-236-234-132:https  ESTABLISHED
TCP   192.168.0.15:64417    ec2-34-195-132-248:https  ESTABLISHED
TCP   192.168.0.15:64420    server-108-138-248-179:https  ESTABLISHED
TCP   192.168.0.15:64425    151.101.40.134:https   ESTABLISHED
TCP   192.168.0.15:64432    87:https              ESTABLISHED
TCP   192.168.0.15:64438    a104-93-136-147:https  ESTABLISHED
```

Command Line Used: netstat 30 | findstr "ESTABLISHED LISTENING"


Capturing from Adapter for loopback traffic capture
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
tcp.port==3333
```

```
No.   Time       Source      Destination  Protocol  Length  Info
    1 0.000000   127.0.0.1   127.0.0.1    TCP           51 57122 → 3333 [PSH, ACK] Seq=1 Ack=1 Win=1279 Len=7
    2 0.000037   127.0.0.1   127.0.0.1    TCP           44 3333 → 57122 [ACK] Seq=1 Ack=8 Win=10233 Len=0
    3 2.447813   127.0.0.1   127.0.0.1    TCP           51 57122 → 3333 [PSH, ACK] Seq=8 Ack=1 Win=1279 Len=7
    4 2.447869   127.0.0.1   127.0.0.1    TCP           44 3333 → 57122 [ACK] Seq=1 Ack=15 Win=10233 Len=0
```

the tcp port,
and 48K, but

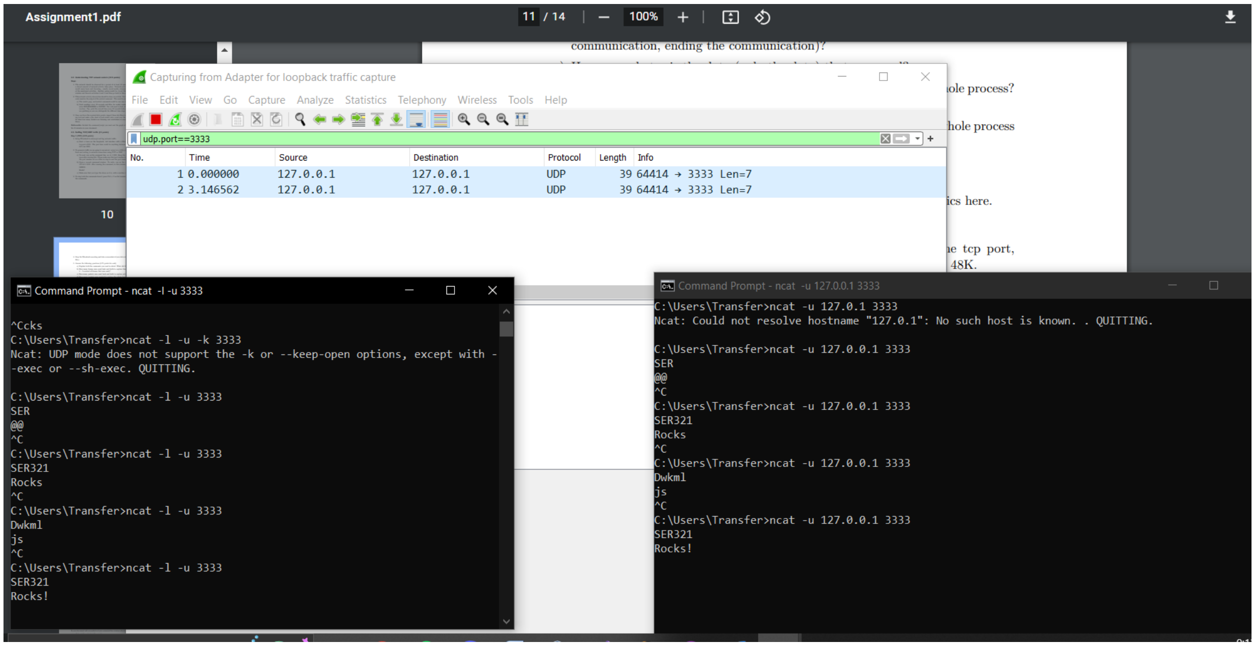
am for reading

mand window
3333) matches

```
^C
C:\Users\Transfer>ncat -k -l 3333
SER321
Rocks!
^C
C:\Users\Transfer>ncat -k -l 3333
SER321
Rocks!
^C
C:\Users\Transfer>ncat -k -l 3333
SER321
Rocks!
YE
^C
C:\Users\Transfer>ncat -k -l 3333
SER321
Rocks!
^C
C:\Users\Transfer>ncat -k -l 3333
SER321
Rocks!
```

```
C:\Users\Transfer>ncat 127.0.0.1 3333
Ncat: No connection could be made because the target machine actively refused it. .

C:\Users\Transfer>ncat 127.0.0.1 3333
SER321
Rocks!
YE
Ncat: An existing connection was forcibly closed by the remote host. .

C:\Users\Transfer>ncat 127.0.0.1 3333
Ncat: No connection could be made because the target machine actively refused it. .

C:\Users\Transfer>ncat 127.0.0.1 3333
SER321
Rocks!
Ncat: An existing connection was forcibly closed by the remote host. .

C:\Users\Transfer>ncat 127.0.0.1 3333
SER321
Rocks!
```

a. The command on the left command prompt was the Client that waited for a prompt by the Server to continuously display what the Server was saying. The command prompt on the right was the Server that accepted data t then send to the Client using port 3333 for both of them.
b. There were 4 frames sent to capture the lines.
c. 2 packets were used to capture just the two lines.
d. 82 packets were used to capture the whole thing
e. 14 bytes were used to send the data.
f. 190 bytes were used for the whole process.
g. 176 extra bytes were used for the overhead.


Assignment1.pdf
11 / 14
100%
communication, ending the communication)?
ole process?
hole process
ics here.
e tcp port,
48K.
10
Capturing from Adapter for loopback traffic capture
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Command Prompt - ncat -l -u 3333
Command Prompt - ncat -u 127.0.0.1 3333

udp.port==3333

```
No.   Time        Source       Destination   Protocol  Length  Info
    1 0.000000    127.0.0.1    127.0.0.1     UDP           39  64414 → 3333 Len=7
    2 3.146562    127.0.0.1    127.0.0.1     UDP           39  64414 → 3333 Len=7
```

```
^Ccks
C:\Users\Transfer>ncat -l -u -k 3333
Ncat: UDP mode does not support the -k or --keep-open options, except with -
-exec or --sh-exec. QUITTING.

C:\Users\Transfer>ncat -l -u 3333
SER
@@
^C
C:\Users\Transfer>ncat -l -u 3333
SER321
Rocks
^C
C:\Users\Transfer>ncat -l -u 3333
Dwkml
js
^C
C:\Users\Transfer>ncat -l -u 3333
SER321
Rocks!
```

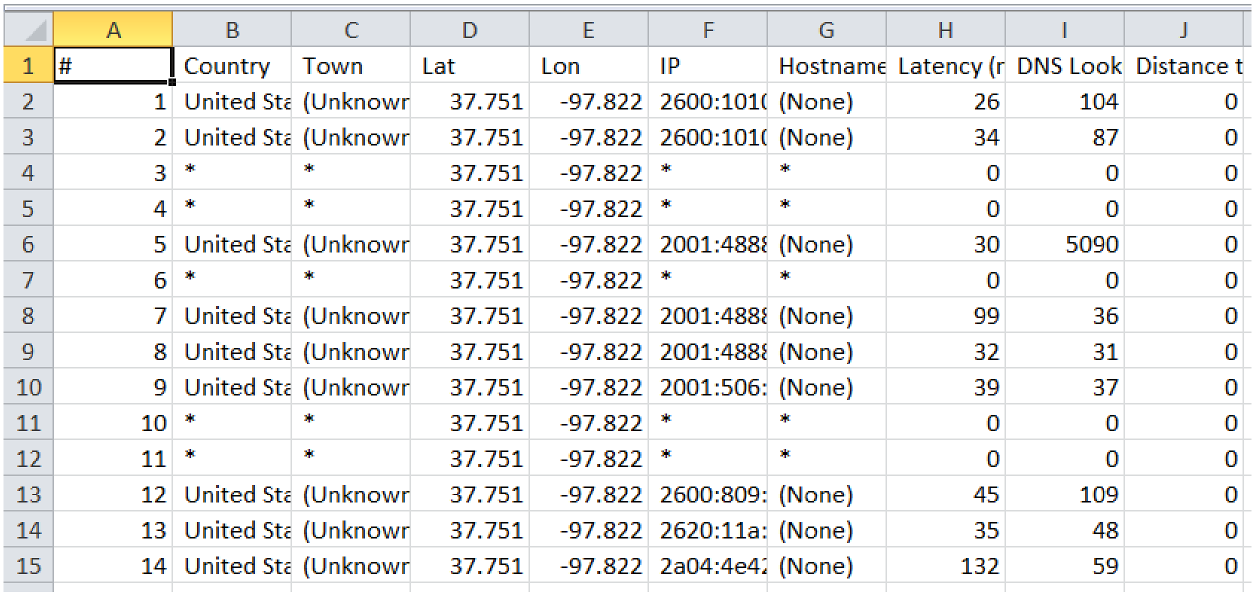
```
C:\Users\Transfer>ncat -u 127.0.1 3333
Ncat: Could not resolve hostname "127.0.1": No such host is known. . QUITTING.

C:\Users\Transfer>ncat -u 127.0.0.1 3333
SER
@@
^C
C:\Users\Transfer>ncat -u 127.0.0.1 3333
SER321
Rocks
^C
C:\Users\Transfer>ncat -u 127.0.0.1 3333
Dwkml
js
^C
C:\Users\Transfer>ncat -u 127.0.0.1 3333
SER321
Rocks!
```

a. The command on the left command prompt was the Client that waited for a prompt by the Server to continuously display what the Server was saying. The command prompt on the right was the Server that accepted data t then send to the Client using port 3333 for both of them.
b. 2 frames were needed to capture these lines.
c. 2 packets were used to capture the lines.
d. 32 packets were used to capture the entire process.
e. 78 bytes were used to capture the entire process.
f. 14 bytes were used just for the data
g. 64 bytes were used for the process (without the data)
h. Both of the processes had the same amount of space for the data itself, but the overhead space is much more on TCP compared to UDP. This is because UDP is connectionless and TCP uses connections. TCP would be more secure, but is using more data to make sure the data is more secure. This increases the security at the expense of much more space used for it.

IP Routing:

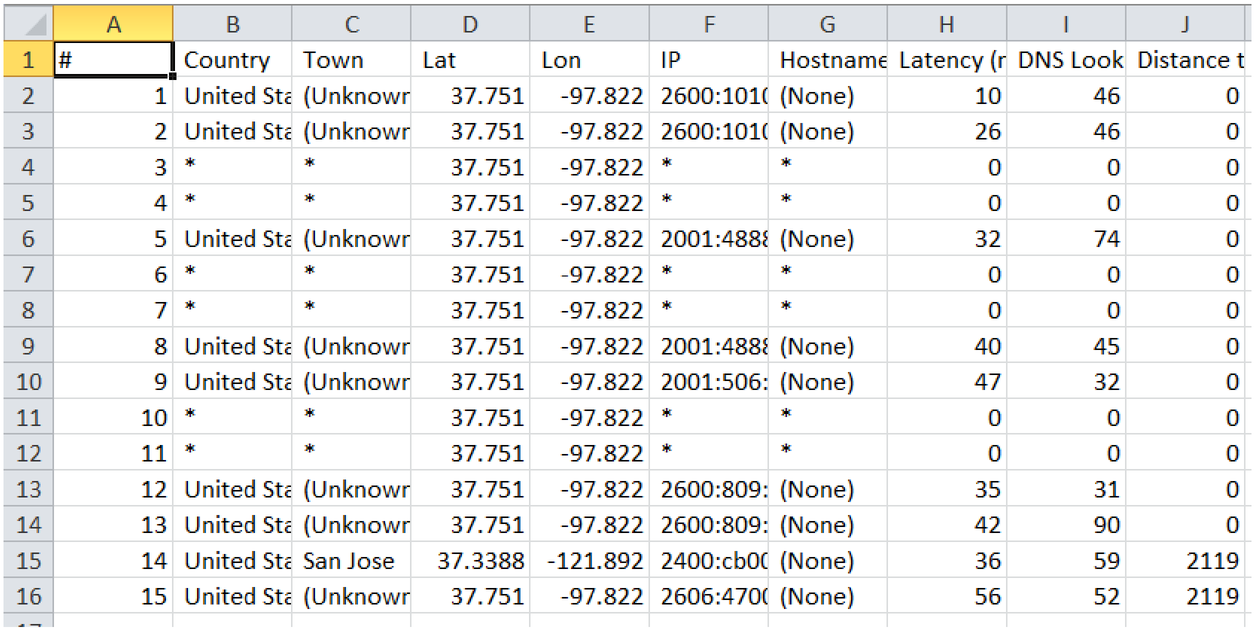
They were about the same speed and the local shop had one more hop.

ASU:

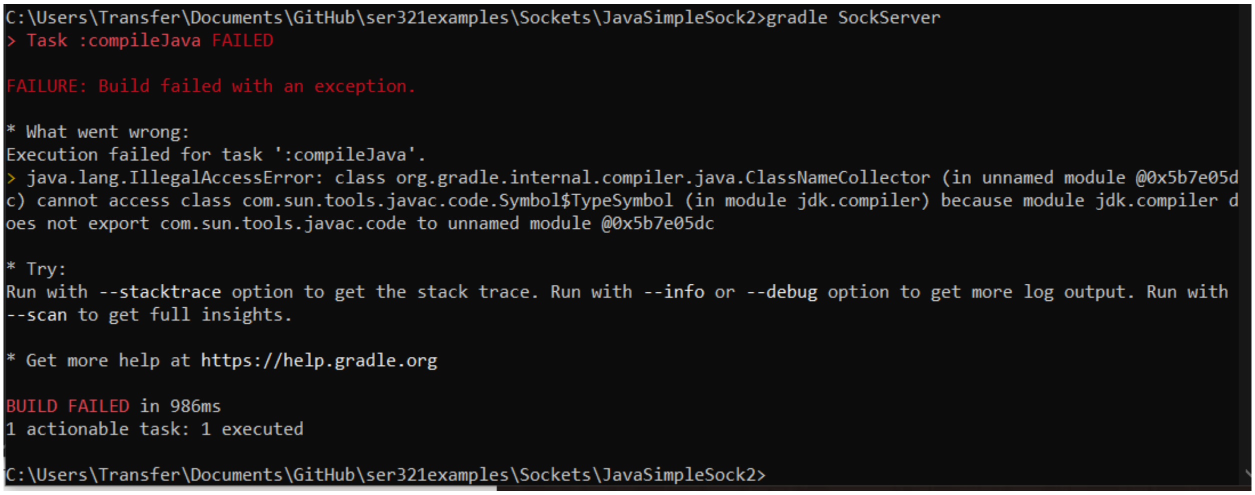


	A	B	C	D	E	F	G	H	I	J
1	#	Country	Town	Lat	Lon	IP	Hostname	Latency (r	DNS Look	Distance t
2	1	United Sta	(Unknowr	37.751	-97.822	2600:101(	(None)	26	104	0
3	2	United Sta	(Unknowr	37.751	-97.822	2600:101(	(None)	34	87	0
4	3	*	*	37.751	-97.822	*	*	0	0	0
5	4	*	*	37.751	-97.822	*	*	0	0	0
6	5	United Sta	(Unknowr	37.751	-97.822	2001:4888	(None)	30	5090	0
7	6	*	*	37.751	-97.822	*	*	0	0	0
8	7	United Sta	(Unknowr	37.751	-97.822	2001:4888	(None)	99	36	0
9	8	United Sta	(Unknowr	37.751	-97.822	2001:4888	(None)	32	31	0
10	9	United Sta	(Unknowr	37.751	-97.822	2001:506:	(None)	39	37	0
11	10	*	*	37.751	-97.822	*	*	0	0	0
12	11	*	*	37.751	-97.822	*	*	0	0	0
13	12	United Sta	(Unknowr	37.751	-97.822	2600:809:	(None)	45	109	0
14	13	United Sta	(Unknowr	37.751	-97.822	2620:11a:	(None)	35	48	0
15	14	United Sta	(Unknowr	37.751	-97.822	2a04:4e42	(None)	132	59	0

Local Coffee Shop:



	A	B	C	D	E	F	G	H	I	J
1	#	Country	Town	Lat	Lon	IP	Hostname	Latency (r	DNS Look	Distance t
2	1	United Sta	(Unknowr	37.751	-97.822	2600:101(	(None)	10	46	0
3	2	United Sta	(Unknowr	37.751	-97.822	2600:101(	(None)	26	46	0
4	3	*	*	37.751	-97.822	*	*	0	0	0
5	4	*	*	37.751	-97.822	*	*	0	0	0
6	5	United Sta	(Unknowr	37.751	-97.822	2001:4888	(None)	32	74	0
7	6	*	*	37.751	-97.822	*	*	0	0	0
8	7	*	*	37.751	-97.822	*	*	0	0	0
9	8	United Sta	(Unknowr	37.751	-97.822	2001:4888	(None)	40	45	0
10	9	United Sta	(Unknowr	37.751	-97.822	2001:506:	(None)	47	32	0
11	10	*	*	37.751	-97.822	*	*	0	0	0
12	11	*	*	37.751	-97.822	*	*	0	0	0
13	12	United Sta	(Unknowr	37.751	-97.822	2600:809:	(None)	35	31	0
14	13	United Sta	(Unknowr	37.751	-97.822	2600:809:	(None)	42	90	0
15	14	United Sta	San Jose	37.3388	-121.892	2400:cb0(	(None)	36	59	2119
16	15	United Sta	(Unknowr	37.751	-97.822	2606:470(	(None)	56	52	2119



```
C:\Users\Transfer\Documents\GitHub\ser321examples\Sockets\JavaSimpleSock2>gradle SockServer
> Task :compileJava FAILED

FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':compileJava'.
> java.lang.IllegalAccessError: class org.gradle.internal.compiler.java.ClassNameCollector (in unnamed module @0x5b7e05d
c) cannot access class com.sun.tools.javac.code.Symbol$TypeSymbol (in module jdk.compiler) because module jdk.compiler d
oes not export com.sun.tools.javac.code to unnamed module @0x5b7e05dc

* Try:
Run with --stacktrace option to get the stack trace. Run with --info or --debug option to get more log output. Run with
--scan to get full insights.

* Get more help at https://help.gradle.org

BUILD FAILED in 986ms
1 actionable task: 1 executed

C:\Users\Transfer\Documents\GitHub\ser321examples\Sockets\JavaSimpleSock2>
```

I tried the same things I did earlier and it keeps saying IllegalAccessError. I’m not sure what else to do.

4.5.3 and 4.5.4

By having the AWS as the client and your local computer and the server, you would effectively have two servers talking to each other. Your local ip address would be reaching out to both of them to communicate, which would make things harder, but not impossible. When your computer is the Client, you can simply reach out to your router and use that to fetch the information coming from the server that already has a different ip address. However, when this is backwards, your computer is reaching out to a server, that is then asking for information from a different server coming from your computer. The data going everywhere makes this much more complicated since you are never coming back to your own ip address. Its just servers talking to servers.