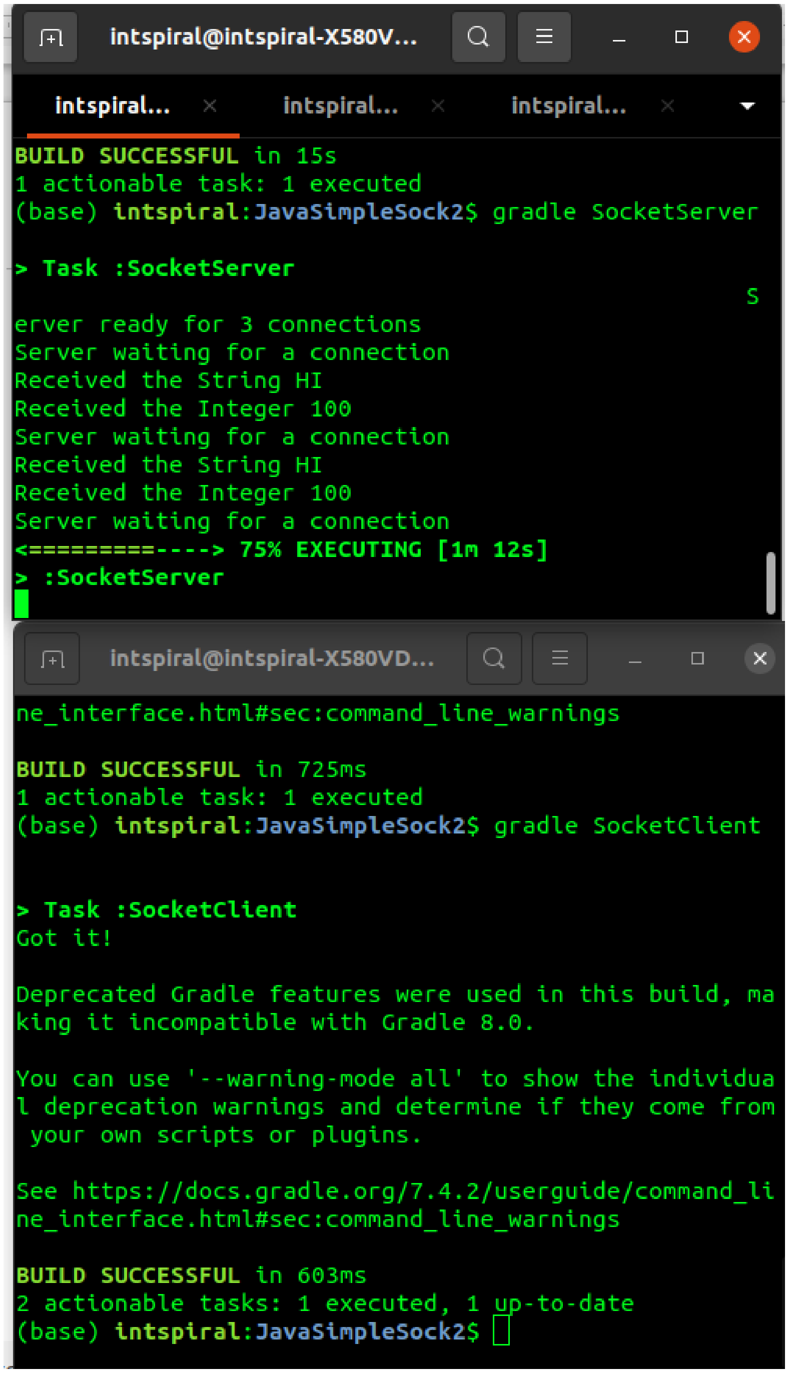
Assignment 1

Github repo link: https://github.com/Kallory/ser321-fall2022-B-krpalme3

Command line tasks:

(Linux System, Ubuntu 20.04)

```
1: mkdir cli_assignment
2: cd cli_assignment
3: touch stuff.txt
4: cat > stuff.txt  (two lines, ctrl+d to stop adding text)
5: wc stuff.txt
6: cat >> stuff.txt (more text, ctrl+d)
7: mkdir draft
8: mk stuff.txt draft
9: cd draft, touch .secret.txt
10: cp -r $PWD ../final/ (from /draft)
10.5: cd .. (can’t do step 11 from inside, mv, from inside it’s own directory)
11: mv draft draft.remove
12: mv draft.remove/ final
13: ls -AlR
14: zcat NASA_access_log_Aug95.gz
15: gunzip NASA_access_log_Aug95.gz
16: mv NASA_access_log_Aug95 logs.txt
17: mv logs.txt cli_assignment/
18: head -100 logs.txt
19: head -100 logs.txt >> logs_top_100.txt
20: tail -100 logs.txt
21: tail -100 logs.txt >> logs_bottom_100.txt
22: cat logs_top_100.txt logs_bottom_100.txt >> logs_snapshot.txt
23: echo 'krpalme3: This is a great assignment - 10/18/2022' >> logs_snapshot.txt
24: less logs.txt
25: cat marks.csv | tail -n +2 | cut -d "%" -f 1
26: cat marks.csv | cut -d "%" -f 4 | sort -n
27: cat marks.csv | tail -n +2 | awk -F "%" ' {sum+=$2; n++;} END {print sum/n}'
28: cat marks.csv | tail -n +2 | awk -F "%" ' {sum+=$2; n++;} END {print sum/n}' >>
        cli_assignment/done.txt
29:  mv cli_assignment/done.txt cli_assignment/final/
30: mv cli_assignment/final/done.txt cli_assignment/final/average.txt
```

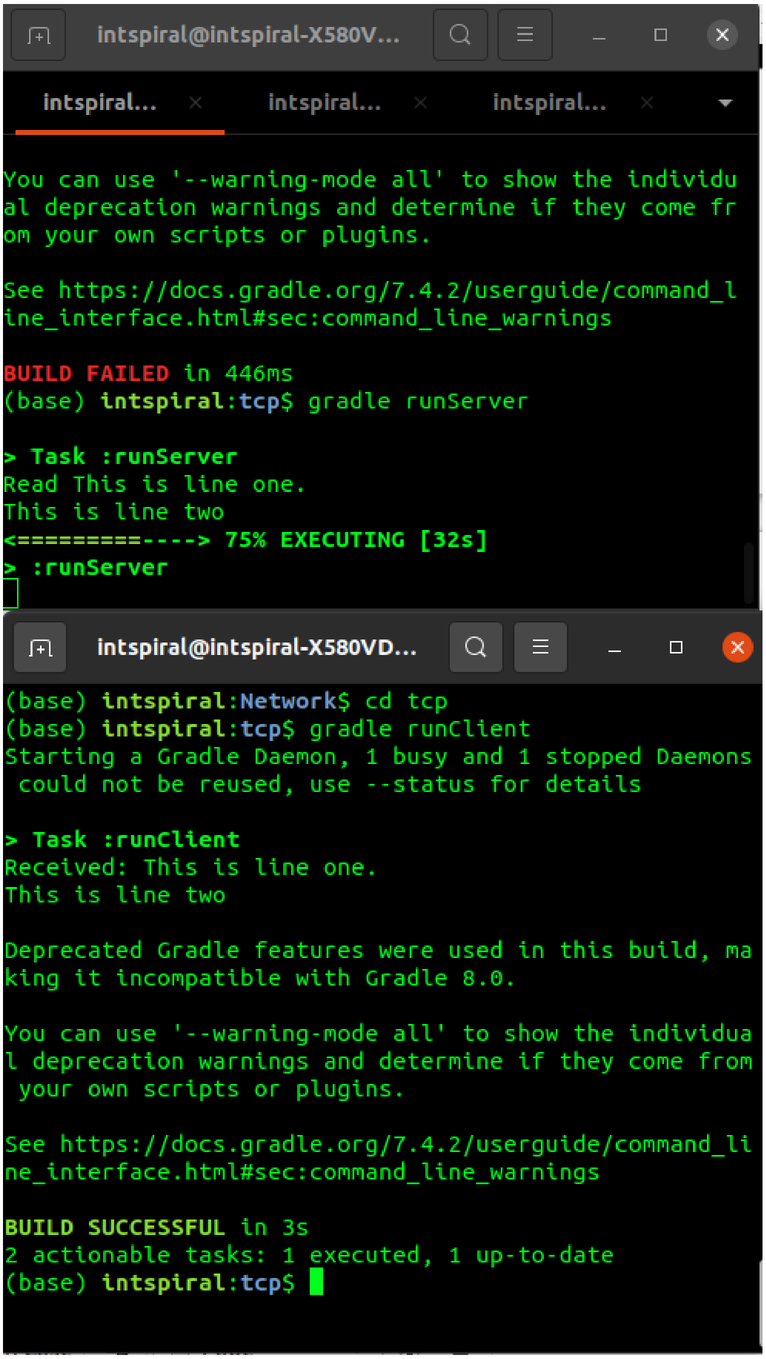


```
BUILD SUCCESSFUL in 15s
1 actionable task: 1 executed
(base) intspiral:JavaSimpleSock2$ gradle SocketServer

> Task :SocketServer
                                                          S
erver ready for 3 connections
Server waiting for a connection
Received the String HI
Received the Integer 100
Server waiting for a connection
Received the String HI
Received the Integer 100
Server waiting for a connection
<==========----> 75% EXECUTING [1m 12s]
> :SocketServer
```

```
ne_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 725ms
1 actionable task: 1 executed
(base) intspiral:JavaSimpleSock2$ gradle SocketClient


> Task :SocketClient
Got it!

Deprecated Gradle features were used in this build, ma
king it incompatible with Gradle 8.0.

You can use '--warning-mode all' to show the individua
l deprecation warnings and determine if they come from
 your own scripts or plugins.

See https://docs.gradle.org/7.4.2/userguide/command_li
ne_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 603ms
2 actionable tasks: 1 executed, 1 up-to-date
(base) intspiral:JavaSimpleSock2$
```

Ex1: Running JavaSimpleSock2 server and client through gradle

Shows server socket waiting for input from client
Returns a message of affirmation to the client



```
You can use '--warning-mode all' to show the individu
al deprecation warnings and determine if they come fr
om your own scripts or plugins.

See https://docs.gradle.org/7.4.2/userguide/command_l
ine_interface.html#sec:command_line_warnings

BUILD FAILED in 446ms
(base) intspiral:tcp$ gradle runServer

> Task :runServer
Read This is line one.
This is line two
<==========----> 75% EXECUTING [32s]
> :runServer
```

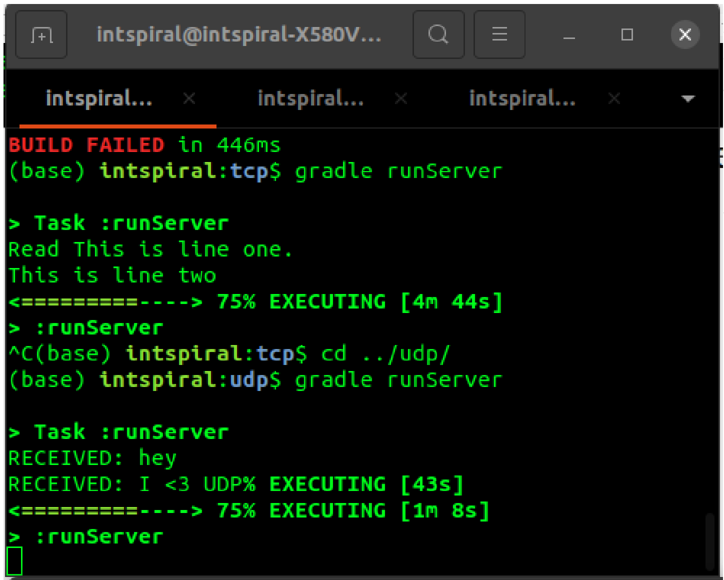
```
(base) intspiral:Network$ cd tcp
(base) intspiral:tcp$ gradle runClient
Starting a Gradle Daemon, 1 busy and 1 stopped Daemons
 could not be reused, use --status for details

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

Deprecated Gradle features were used in this build, ma
king it incompatible with Gradle 8.0.

You can use '--warning-mode all' to show the individua
l deprecation warnings and determine if they come from
 your own scripts or plugins.

See https://docs.gradle.org/7.4.2/userguide/command_li
ne_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 3s
2 actionable tasks: 1 executed, 1 up-to-date
(base) intspiral:tcp$
```

Ex2: Running tcp server and client through gradle

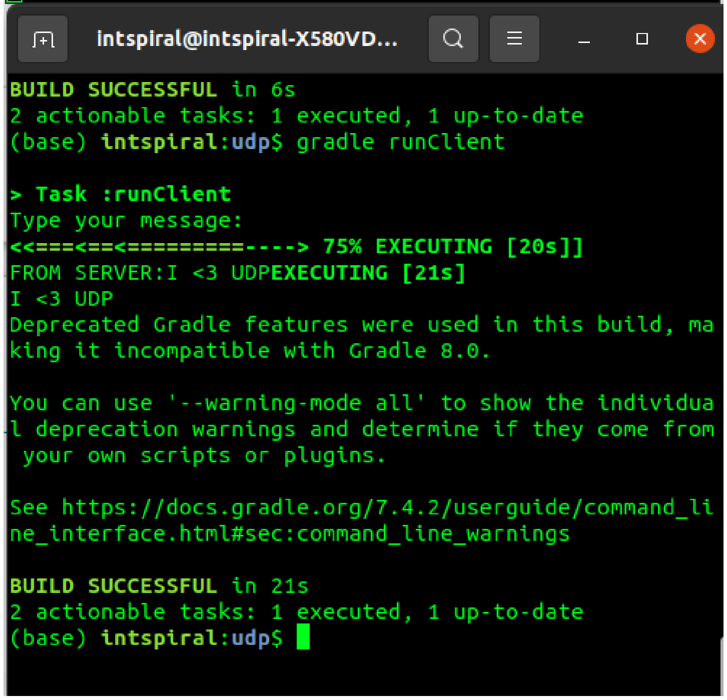
Sends text utilizing tcp protocol from the server to the client.

Ex3: Running UDP example through gradle, server and client

Server waits for input from the client via UDP protocol



```
BUILD FAILED in 446ms
(base) intspiral:tcp$ gradle runServer

> Task :runServer
Read This is line one.
This is line two
<==========----> 75% EXECUTING [4m 44s]
> :runServer
^C(base) intspiral:tcp$ cd ../udp/
(base) intspiral:udp$ gradle runServer

> Task :runServer
RECEIVED: hey
RECEIVED: I <3 UDP% EXECUTING [43s]
<==========----> 75% EXECUTING [1m 8s]
> :runServer
```

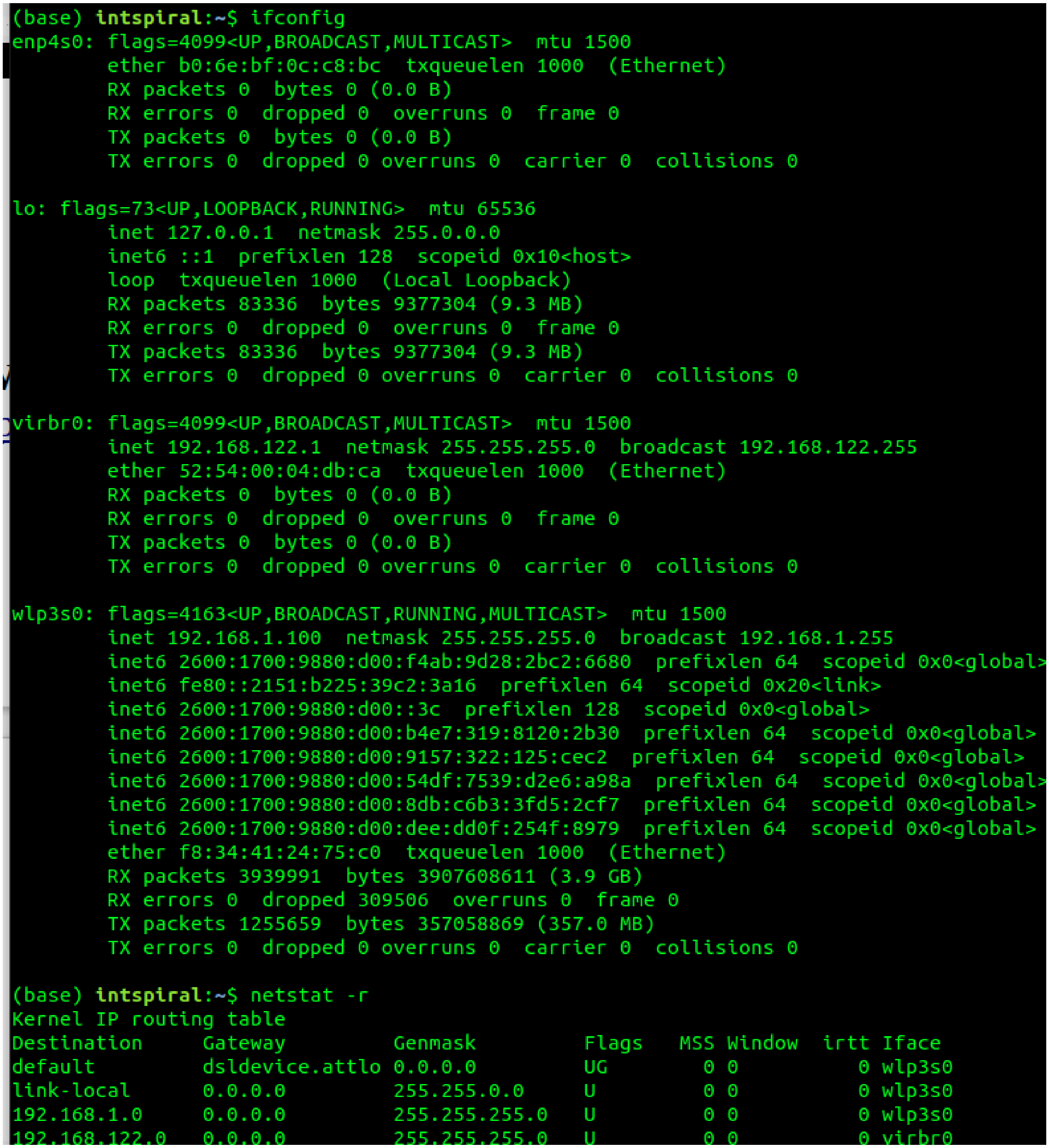


```
BUILD SUCCESSFUL in 6s
2 actionable tasks: 1 executed, 1 up-to-date
(base) intspiral:udp$ gradle runClient

> Task :runClient
Type your message:
<<===<==<=========----> 75% EXECUTING [20s]]
FROM SERVER:I <3 UDPEXECUTING [21s]
I <3 UDP
Deprecated Gradle features were used in this build, ma
king it incompatible with Gradle 8.0.

You can use '--warning-mode all' to show the individua
l deprecation warnings and determine if they come from
 your own scripts or plugins.

See https://docs.gradle.org/7.4.2/userguide/command_li
ne_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 21s
2 actionable tasks: 1 executed, 1 up-to-date
(base) intspiral:udp$
```

AWS JavaSimpleSock2 Screencast:

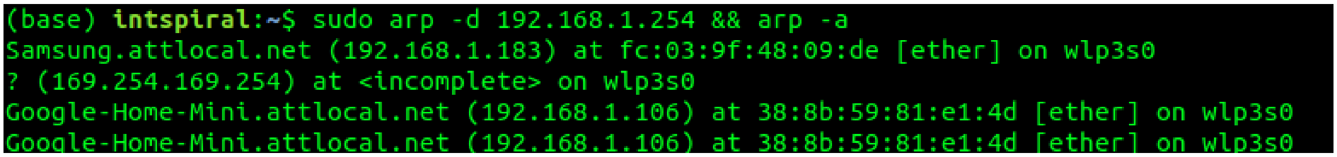
https://youtu.be/ecpvzeyjuA4

Part II
4.1
ifconfig, netstat -r



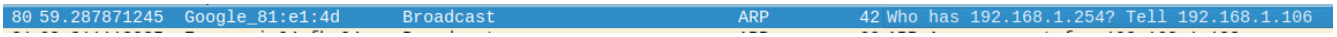
```
(base) intspiral:~$ ifconfig
enp4s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether b0:6e:bf:0c:c8:bc  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 83336  bytes 9377304 (9.3 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 83336  bytes 9377304 (9.3 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.122.1  netmask 255.255.255.0  broadcast 192.168.122.255
        ether 52:54:00:04:db:ca  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.100  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 2600:1700:9880:d00:f4ab:9d28:2bc2:6680  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::2151:b225:39c2:3a16  prefixlen 64  scopeid 0x20<link>
        inet6 2600:1700:9880:d00::3c  prefixlen 128  scopeid 0x0<global>
        inet6 2600:1700:9880:d00:b4e7:319:8120:2b30  prefixlen 64  scopeid 0x0<global>
        inet6 2600:1700:9880:d00:9157:322:125:cec2  prefixlen 64  scopeid 0x0<global>
        inet6 2600:1700:9880:d00:54df:7539:d2e6:a98a  prefixlen 64  scopeid 0x0<global>
        inet6 2600:1700:9880:d00:8db:c6b3:3fd5:2cf7  prefixlen 64  scopeid 0x0<global>
        inet6 2600:1700:9880:d00:dee:dd0f:254f:8979  prefixlen 64  scopeid 0x0<global>
        ether f8:34:41:24:75:c0  txqueuelen 1000  (Ethernet)
        RX packets 3939991  bytes 3907608611 (3.9 GB)
        RX errors 0  dropped 309506  overruns 0  frame 0
        TX packets 1255659  bytes 357058869 (357.0 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

(base) intspiral:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         dsldevice.attlo 0.0.0.0         UG        0 0          0 wlp3s0
link-local      0.0.0.0         255.255.0.0     U         0 0          0 wlp3s0
192.168.1.0     0.0.0.0         255.255.255.0   U         0 0          0 wlp3s0
192.168.122.0   0.0.0.0         255.255.255.0   U         0 0          0 virbr0
```

arp -d, arp- a



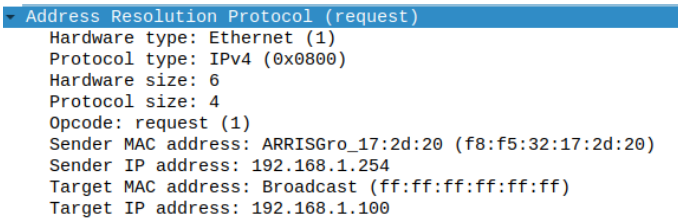
```
(base) intspiral:~$ sudo arp -d 192.168.1.254 && arp -a
Samsung.attlocal.net (192.168.1.183) at fc:03:9f:48:09:de [ether] on wlp3s0
? (169.254.169.254) at <incomplete> on wlp3s0
Google-Home-Mini.attlocal.net (192.168.1.106) at 38:8b:59:81:e1:4d [ether] on wlp3s0
Google-Home-Mini.attlocal.net (192.168.1.106) at 38:8b:59:81:e1:4d [ether] on wlp3s0
```

Trace:

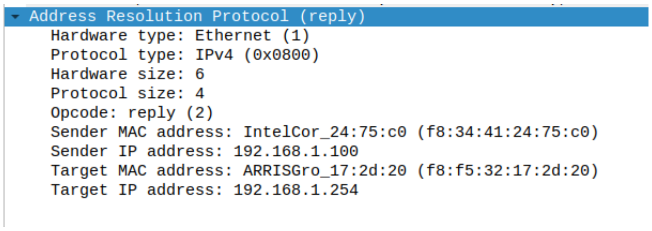


```
80 59.287871245  Google_81:e1:4d    Broadcast              ARP        42 Who has 192.168.1.254? Tell 192.168.1.106
```

Request & Reply



```
▾ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: ARRISGro_17:2d:20 (f8:f5:32:17:2d:20)
    Sender IP address: 192.168.1.254
    Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
    Target IP address: 192.168.1.100
```



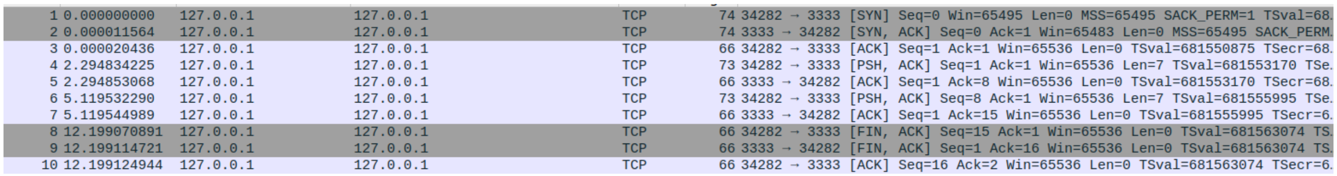
```
▾ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: IntelCor_24:75:c0 (f8:34:41:24:75:c0)
    Sender IP address: 192.168.1.100
    Target MAC address: ARRISGro_17:2d:20 (f8:f5:32:17:2d:20)
    Target IP address: 192.168.1.254
```

Step 3:

1) an opcode of 1 is used for a request, 2 for a reply
2) ARP header request size – 28 bytes. Same for reply, this includes the Host/target MAC addresses and IP addresses
3) Broadcast address ff:ff:ff:ff:ff:ff
4) A value of 0x0806 under the Ethernet II tab shows Type: ARP (0x0806)

4.3 Sniffing TCP/UDP traffic

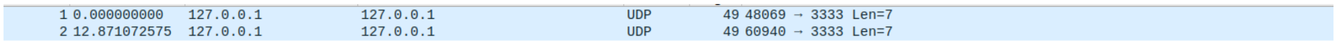
TCP



```
     1 0.000000000   127.0.0.1              127.0.0.1                TCP        74 34282 → 3333 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=68.
     2 0.000011564   127.0.0.1              127.0.0.1                TCP        74 3333 → 34282 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM.
     3 0.000020436   127.0.0.1              127.0.0.1                TCP        66 34282 → 3333 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=681550875 TSecr=68.
     4 2.294834225   127.0.0.1              127.0.0.1                TCP        73 34282 → 3333 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=7 TSval=681553170 TSe.
     5 2.294853068   127.0.0.1              127.0.0.1                TCP        66 3333 → 34282 [ACK] Seq=1 Ack=8 Win=65536 Len=0 TSval=681553170 TSecr=68.
     6 5.119532290   127.0.0.1              127.0.0.1                TCP        73 34282 → 3333 [PSH, ACK] Seq=8 Ack=1 Win=65536 Len=7 TSval=681555995 TSe.
     7 5.119544989   127.0.0.1              127.0.0.1                TCP        66 3333 → 34282 [ACK] Seq=1 Ack=15 Win=65536 Len=0 TSval=681555995 TSecr=6.
     8 12.199070891  127.0.0.1              127.0.0.1                TCP        66 34282 → 3333 [FIN, ACK] Seq=15 Ack=1 Win=65536 Len=0 TSval=681563074 TS.
     9 12.199114721  127.0.0.1              127.0.0.1                TCP        66 3333 → 34282 [FIN, ACK] Seq=1 Ack=16 Win=65536 Len=0 TSval=681563074 TS.
    10 12.199124944  127.0.0.1              127.0.0.1                TCP        66 34282 → 3333 [ACK] Seq=16 Ack=2 Win=65536 Len=0 TSval=681563074 TSecr=6.
```

a)nc with the -l command tells netcat to listen and not initiate a connection, the -k just tells it to repeat once it's made contact with something, all of this on the declared port (3333). nc with 127.0.0.1 3333 is telling nc to initiate communication on port 3333 of the computer I'm using.
b) a total of 10 frames were exchanged between the two terminals while sending these messages
c) 10 packets were exchanged
d) 9 packets were needed to start and end the whole process
e) the sent data is only 14 bytes (7 for the data declaring SER321, 7 for the data declaring Rocks!)
f) 690 bytes was sent over the wire for the whole process
g) 690 – 14 = 686 bytes of overhead

UDP

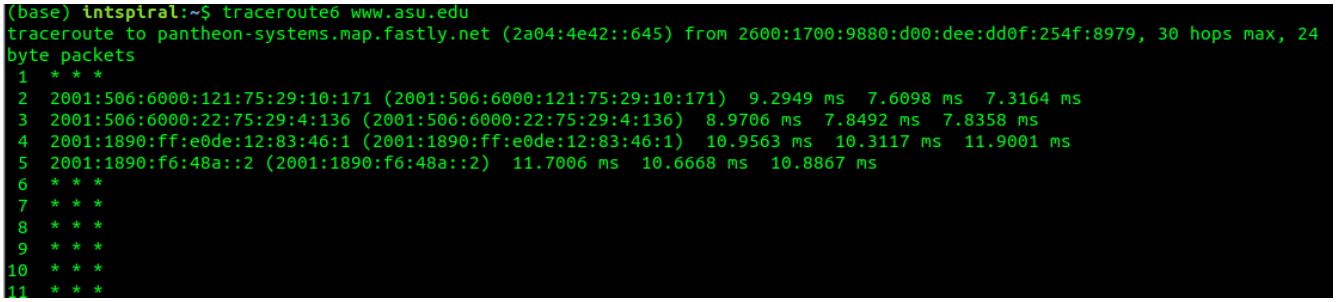


```
     1 0.000000000   127.0.0.1              127.0.0.1                UDP        49 48069 → 3333 Len=7
     2 12.871072575  127.0.0.1              127.0.0.1                UDP        49 60940 → 3333 Len=7
```

a) nc with the -l command tells netcat to listen and not initiate a connection, the -k just tells it to repeat once it's made contact with something, the -u command tells it to utilize UDP instead of TCP. All of this on the declared port (3333). nc with 127.0.0.1 3333 is telling nc to initiate communication on port 3333 of the computer I'm using.
b) 2 frames were needed
c) 2 packets were needed
d) 2 packets were needed to capture the entire process
e) 98 bytes went over the wire
f) the data was only 14 bytes
g) 98 – 14 = 85 bytes of overhead
h) The difference in relative overhead is that TCP has a lot more due to the extra work it does to ensure that packets arrive in the right order.

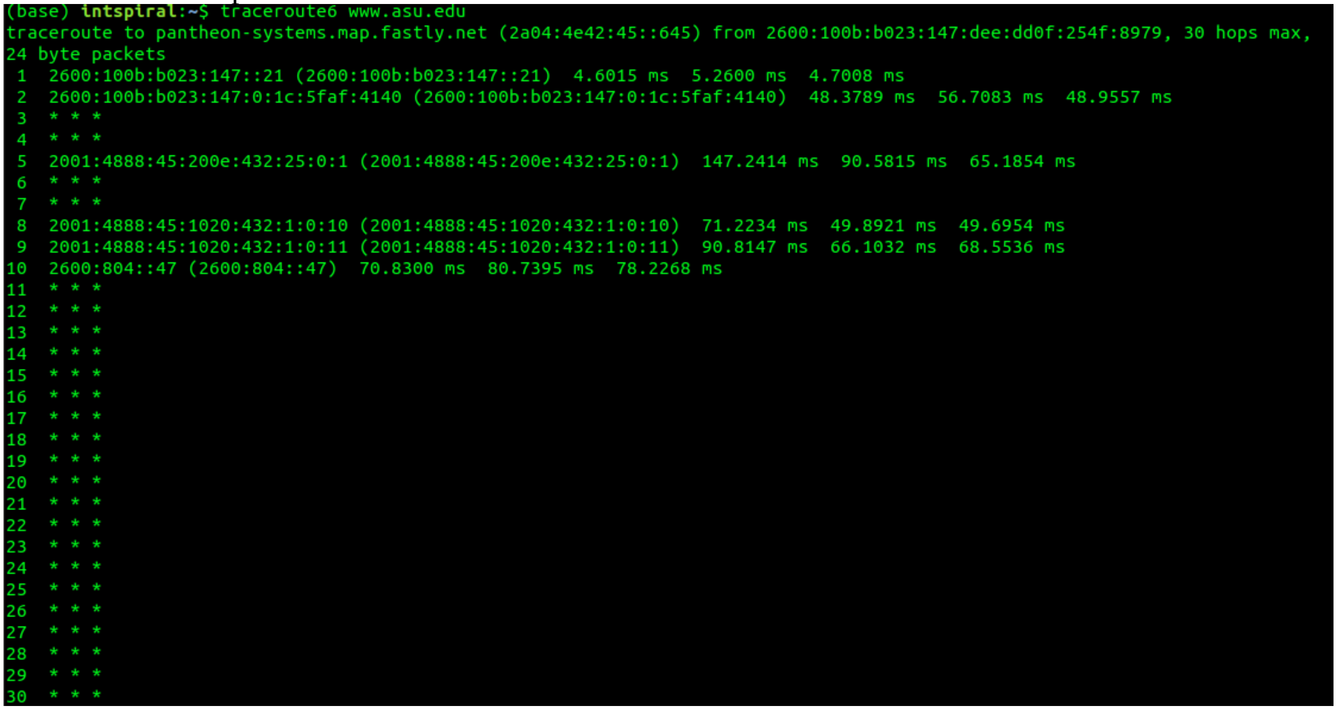
4.4

From home to ASU



```
(base) intspiral:~$ traceroute6 www.asu.edu
traceroute to pantheon-systems.map.fastly.net (2a04:4e42::645) from 2600:1700:9880:d00:dee:dd0f:254f:8979, 30 hops max, 24
byte packets
 1  * * *
 2  2001:506:6000:121:75:29:10:171 (2001:506:6000:121:75:29:10:171)  9.2949 ms  7.6098 ms  7.3164 ms
 3  2001:506:6000:22:75:29:4:136 (2001:506:6000:22:75:29:4:136)  8.9706 ms  7.8492 ms  7.8358 ms
 4  2001:1890:ff:e0de:12:83:46:1 (2001:1890:ff:e0de:12:83:46:1)  10.9563 ms  10.3117 ms  11.9001 ms
 5  2001:1890:f6:48a::2 (2001:1890:f6:48a::2)  11.7006 ms  10.6668 ms  10.8867 ms
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
```

From mobile hotspot to ASU



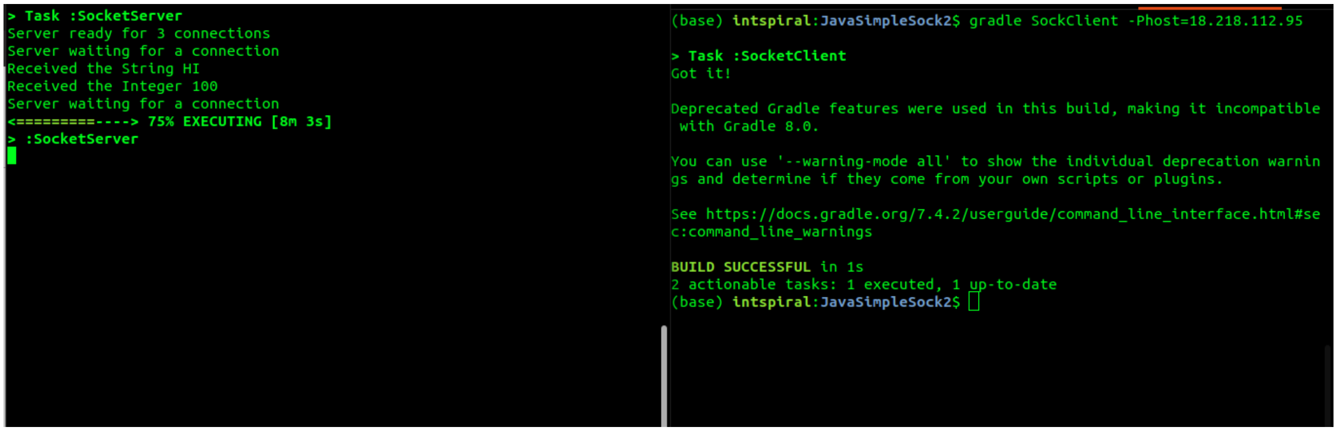
```
(base) intspiral:~$ traceroute6 www.asu.edu
traceroute to pantheon-systems.map.fastly.net (2a04:4e42:45::645) from 2600:100b:b023:147:dee:dd0f:254f:8979, 30 hops max,
24 byte packets
 1  2600:100b:b023:147::21 (2600:100b:b023:147::21)  4.6015 ms  5.2600 ms  4.7008 ms
 2  2600:100b:b023:147:0:1c:5faf:4140 (2600:100b:b023:147:0:1c:5faf:4140)  48.3789 ms  56.7083 ms  48.9557 ms
 3  * * *
 4  * * *
 5  2001:4888:45:200e:432:25:0:1 (2001:4888:45:200e:432:25:0:1)  147.2414 ms  90.5815 ms  65.1854 ms
 6  * * *
 7  * * *
 8  2001:4888:45:1020:432:1:0:10 (2001:4888:45:1020:432:1:0:10)  71.2234 ms  49.8921 ms  49.6954 ms
 9  2001:4888:45:1020:432:1:0:11 (2001:4888:45:1020:432:1:0:11)  90.8147 ms  66.1032 ms  68.5536 ms
10  2600:804::47 (2600:804::47)  70.8300 ms  80.7395 ms  78.2268 ms
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

a) My home network is the fastest
b) My home network has the fewest hops

4.5

Running things locally
https://youtu.be/EqC9Ur5Y8sg

Running Server on AWS
https://youtu.be/JqiYauPdWk4



```
> Task :SocketServer
Server ready for 3 connections
Server waiting for a connection
Received the String HI
Received the Integer 100
Server waiting for a connection
<=========----> 75% EXECUTING [8m 3s]
> :SocketServer
```

```
(base) intspiral:JavaSimpleSock2$ gradle SockClient -Phost=18.218.112.95

> Task :SocketClient
Got it!

Deprecated Gradle features were used in this build, making it incompatible
 with Gradle 8.0.

You can use '--warning-mode all' to show the individual deprecation warnin
gs and determine if they come from your own scripts or plugins.

See https://docs.gradle.org/7.4.2/userguide/command_line_interface.html#se
c:command_line_warnings

BUILD SUCCESSFUL in 1s
2 actionable tasks: 1 executed, 1 up-to-date
(base) intspiral:JavaSimpleSock2$
```

The changes were that I needed to set wireshark to listen to my ethernet as opposed to the loop back. My gradle call, specifically my client, required the IP address from my ec2 instance on AWS.

Client on AWS

This would not work without issues as my ec2 instance is currently only configured to allow traffic INTO port 8888, not sent traffic out. My personal computer probably isn’t setup to receive traffic on that port from an outside source. Either way some connection based variables would need to be adjusted on either side.

Client on AWS 2

I can easily reach my server on AWS because it’s configured to receive input on the port I’ve pre-selected, 8888. As stated above, the same cannot be said for anything leaving my AWS server or anything coming in on my personal machine through that port. My router would reject incoming traffic unless I configured it otherwise.