Ivan Fernandez

SER 321 – Summer 2021

5/18/2021

Assignment 1

Linux and Setup

Command line tasks: **Linux**

1. Create a directory named "cli_assignment".

```
mkdir cli_assignment
```

2. Change the current working directory to the new directory.

```
cd cli_assignment
```

3. Create a new file named "stuff.txt". Use the touch command to do this. Read about the touch command using the manual (man) pages.

```
touch stuff.txt
```

4. Add some text (multiple lines) to this text file using the cat command.

```
cat > stuff.txt
This part of this assignment
is pretty enjoyable.
I like learning about Linux.
```

5. Count the number of words and the number of lines in the file "stuff.txt".

```
wc stuff.txt
```

6. Append more text to the file "stuff.txt".

```
cat >> stuff.txt
This is part 1 of this assignment.
I am learning many new things.
```

7. In the current working directory, create a new directory "draft".

```
mkdir draft
```

8. Move the "stuff.txt" file to the directory "draft".

```
mv stuff.txt ~/draft
```

9. Change your working directory to "draft" and create a hidden file named "secret.txt".

```
cd draft
touch .secret.txt
```

10. Create a new directory ("final") as a copy of the "draft" directory (final should be on the same level as draft) using the copy command.

```
cp -R draft final
```

11. Rename the "draft" directory to "draft.remove". Use the mv command for this.

```
mv draft draft.remove
```

12. Move the "draft.remove" directory to inside the "final" directory. Use the mv command for this.

```
mv draft.remove final
```

13. From inside the "cli_assignment" directory list all the files and sub-directories and their permissions.

```
cd cli_assignment
ls -l
```

14. List the contents of the given file "NASA_access_log_Aug95.gz" without extracting it.

```
less NASA_access_log_Aug95.gz
```

or

```
zmore NASA_access_log_Aug95.gz
```

15. Extract the given file "NASA_access_log_Aug95.gz".

```
gunzip -v NASA_access_log_Aug95.gz
```

16. Rename the extracted file to "logs.txt".

```
mv NASA_access_log_Aug95 logs.txt
```

17. Move the file "logs.txt" to the "cli_assignment" directory.

```
mv logs.txt cli_assignment/
```

18. Read the top 100 lines of the file "logs.txt".

```
head -n 100 logs.txt
```

19. Create a new file "logs_top_100.txt" containing the top 100 lines using I/O redirection.

```
head -n 100 logs.txt > logs_top_100.txt
```

20. Read the bottom 100 lines of the file "logs.txt".

```
tail -n 100 logs.txt
```

21. Create a new file "logs_bottom_100".txt containing the bottom 100 lines using I/Oredirection.

```
tail -n 100 logs.txt > logs_bottom_100.txt
```

22. Create a new file "logs_snapshot".txt by concatenating files "logs_top_100".txt and"logs_bottom_100".txt.

```
cat logs_top_100.txt logs_bottom_100.txt > logs_snapshot.txt
```

23. Now append to the "logs_snapshot".txt the line "asurite: This is a great assignment" and the current date (asurite is your asurite, e.g. amehlhas for me)

```
cat >> logs_snapshot.txt
```

```
iafernan: This is a great assignment 5/20/2021.
```

24. Read the file "logs.txt" using the less command.

```
less logs.txt
```

25. Using the given file "marks.csv" (delimited by %), print the column "student_names" without the header (you can use the column num as index). Use the cut command for this.

```
cut -f 1 -d % marks.csv
```

26. Using the given file "marks.csv", print the sorted list of marks in "subject_3". Use the sort command piped with the cut command.

```
cut -f 4 -d % marks.csv | sort
```

27. Using the given file "marks.csv", print the average marks for "subject_2".

```
cut -f 3 -d % marks.csv > subject_2.txt
```

```
awk ‘{ total += $0; count ++ } END { print total/count }’ subject_2.txt
```

28. Save the average into a new file "done.txt".

awk '{ total += $0; count ++ } END { print total/count }' subject_2.txt > done.txt

29. Move "done.txt" into your "final" directory.

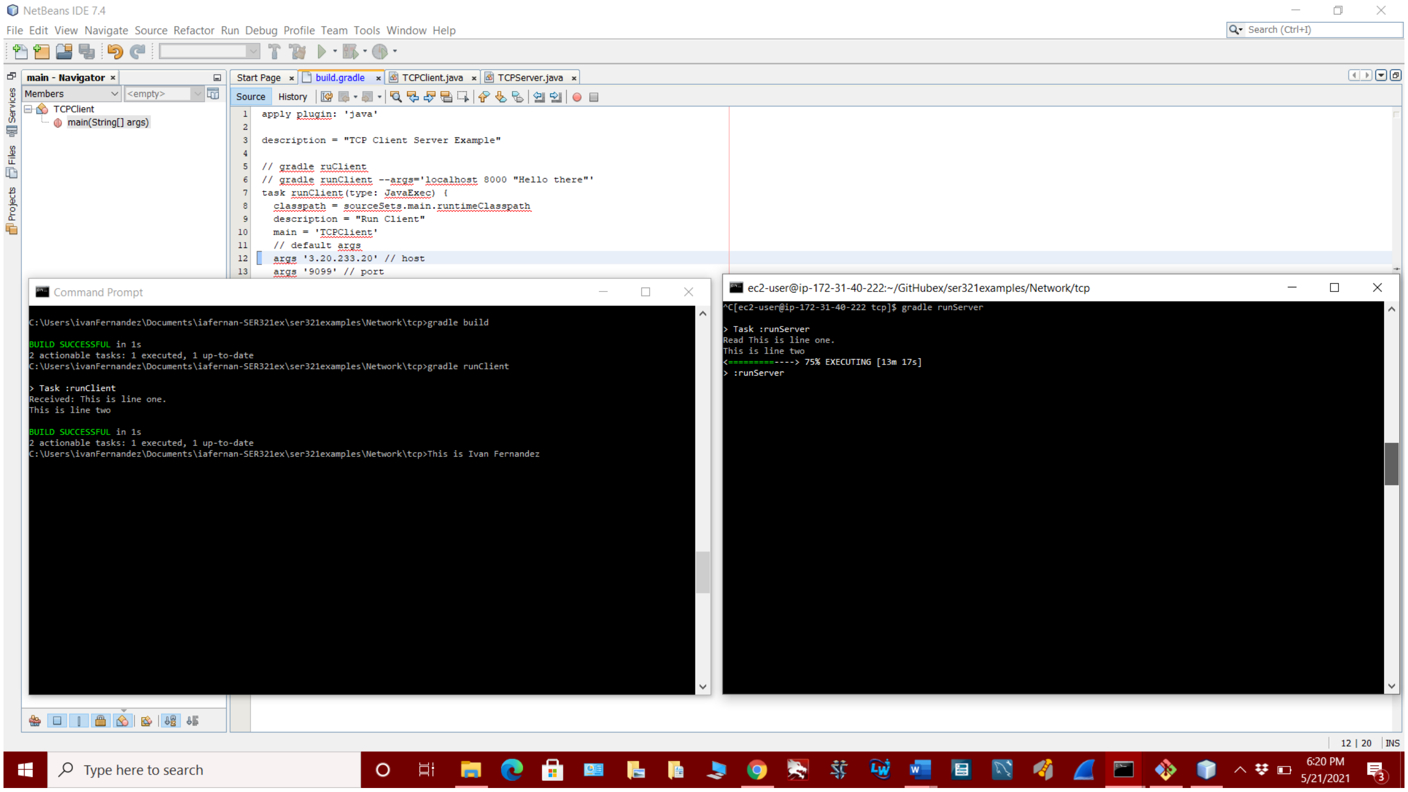
mv done.txt final/

30. Rename the "done.txt" file to "average.txt".

mv done.txt average.txt

2.2. Running examples

Example 1: Network/tcp



```
apply plugin: 'java'

description = "TCP Client Server Example"

// gradle ruClient
// gradle runClient --args='localhost 8000 "Hello there"'
task runClient(type: JavaExec) {
  classpath = sourceSets.main.runtimeClasspath
  description = "Run Client"
  main = 'TCPClient'
  // default args
  args '3.20.233.20' // host
  args '9099' // port
```

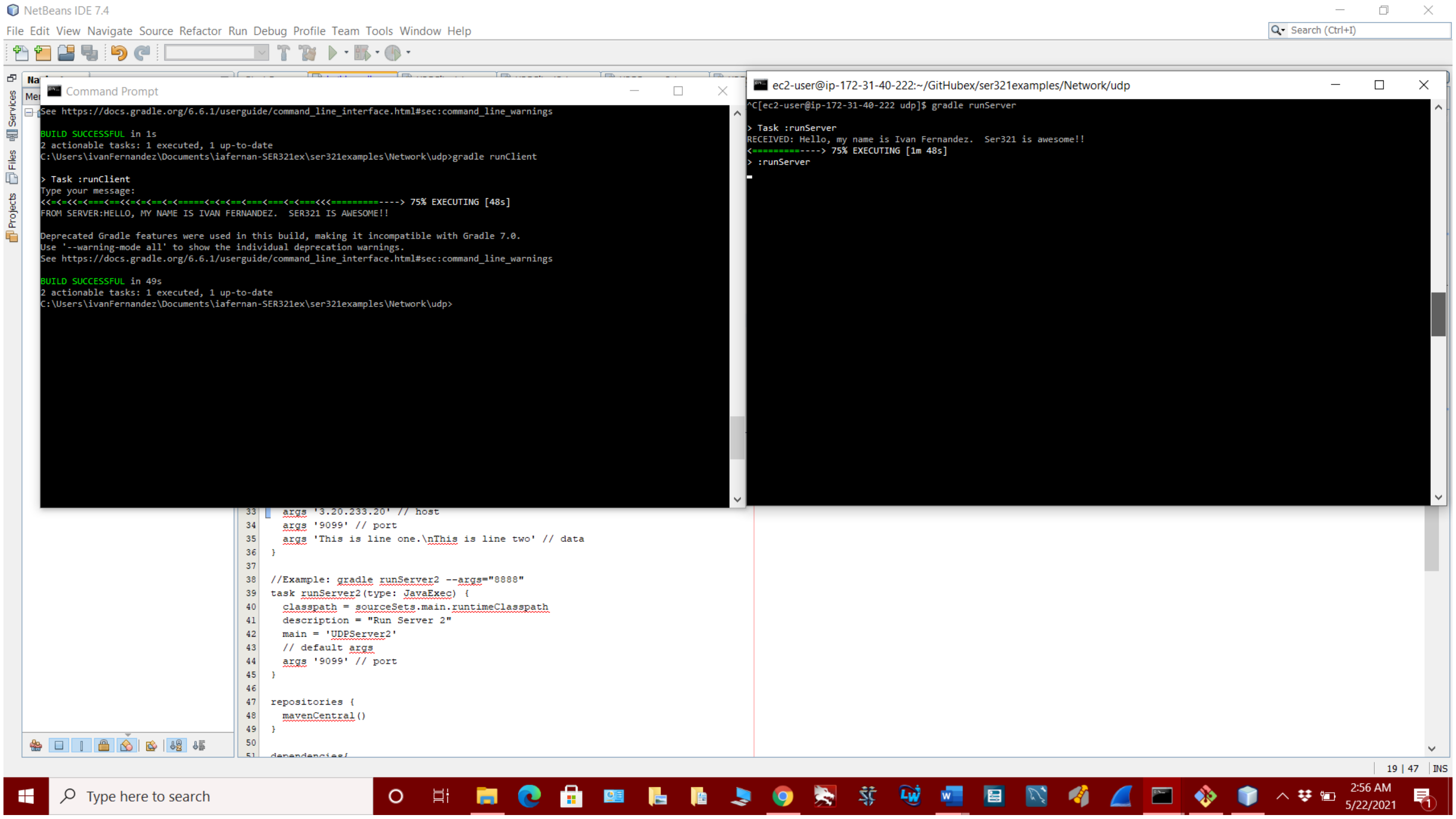
Command Prompt

```
C:\Users\ivanFernandez\Documents\iafernan-SER321ex\ser321examples\Network\tcp>gradle build

BUILD SUCCESSFUL in 1s
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\ivanFernandez\Documents\iafernan-SER321ex\ser321examples\Network\tcp>gradle runClient

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

BUILD SUCCESSFUL in 1s
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\ivanFernandez\Documents\iafernan-SER321ex\ser321examples\Network\tcp>This is Ivan Fernandez
```

ec2-user@ip-172-31-40-222:~/GitHubex/ser321examples/Network/tcp

```
^C[ec2-user@ip-172-31-40-222 tcp]$ gradle runServer

> Task :runServer
Read This is line one.
This is line two
<=========----> 75% EXECUTING [13m 17s]
> :runServer
```

Example 2: Network/UDP Server and UDP Client

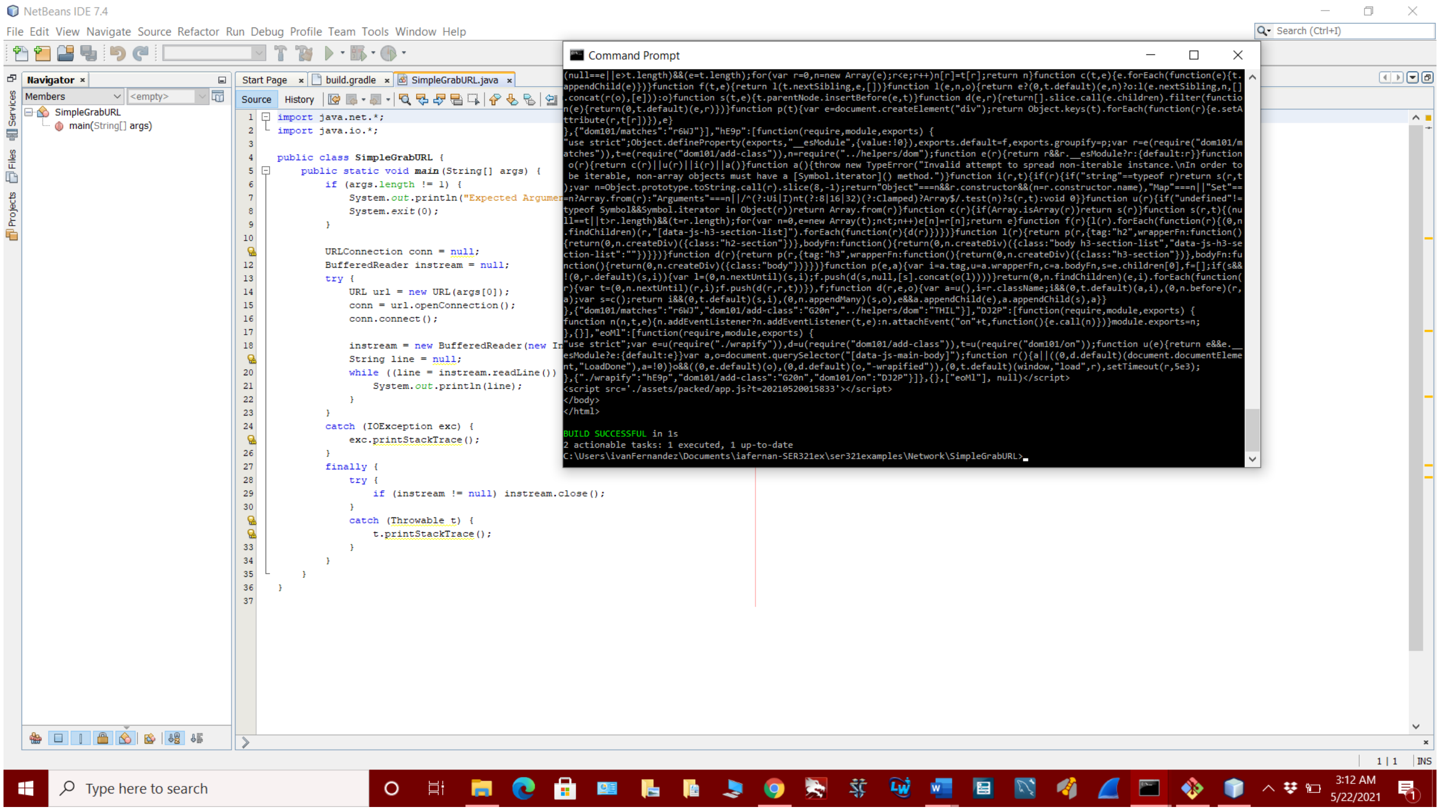

NetBeans IDE 7.4
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Command Prompt
ec2-user@ip-172-31-40-222:~/GitHubex/ser321examples/Network/udp
Services
Files
Projects
19 | 47
INS
Type here to search
2:56 AM
5/22/2021

```
See https://docs.gradle.org/6.6.1/userguide/command_line_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 1s
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\ivanFernandez\Documents\iafernan-SER321ex\ser321examples\Network\udp>gradle runClient

> Task :runClient
Type your message:
<<=<=<<=<===<==<<=<=<==<=<=====<=<=<==<===<===<=<===<<<=========----> 75% EXECUTING [48s]
FROM SERVER:HELLO, MY NAME IS IVAN FERNANDEZ.  SER321 IS AWESOME!!

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 49s
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\ivanFernandez\Documents\iafernan-SER321ex\ser321examples\Network\udp>
```

```
^C[ec2-user@ip-172-31-40-222 udp]$ gradle runServer

> Task :runServer
RECEIVED: Hello, my name is Ivan Fernandez.  Ser321 is awesome!!
<=========----> 75% EXECUTING [1m 48s]
> :runServer
```

```
  args '3.20.233.20' // host
  args '9099' // port
  args 'This is line one.\nThis is line two' // data
}

//Example: gradle runServer2 --args="8888"
task runServer2(type: JavaExec) {
  classpath = sourceSets.main.runtimeClasspath
  description = "Run Server 2"
  main = 'UDPServer2'
  // default args
  args '9099' // port
}

repositories {
  mavenCentral()
}

dependencies{
```

Example 3: Network/SimpleGrabURL



```java
import java.net.*;
import java.io.*;

public class SimpleGrabURL {
    public static void main(String[] args) {
        if (args.length != 1) {
            System.out.println("Expected Argumen
            System.exit(0);
        }

        URLConnection conn = null;
        BufferedReader instream = null;
        try {
            URL url = new URL(args[0]);
            conn = url.openConnection();
            conn.connect();

            instream = new BufferedReader(new In
            String line = null;
            while ((line = instream.readLine())
                System.out.println(line);
            }
        }
        catch (IOException exc) {
            exc.printStackTrace();
        }
        finally {
            try {
                if (instream != null) instream.close();
            }
            catch (Throwable t) {
                t.printStackTrace();
            }
        }
    }
}
```

```
(null==e||e>t.length)&&(e=t.length);for(var r=0,n=new Array(e);r<e;r++)n[r]=t[r];return n}function c(t,e){e.forEach(function(e){t.
appendChild(e)})}function f(t,e){return l(t.nextSibling,e,[])}function l(e,n,o){return e?(0,t.default)(e,n)?o:l(e.nextSibling,n,[]
.concat(r(o),[e])):o}function s(t,e){t.parentNode.insertBefore(e,t)}function d(e,r){return[].slice.call(e.children).filter(functio
n(e){return(0,t.default)(e,r)})}function p(t){var e=document.createElement("div");return Object.keys(t).forEach(function(r){e.setA
ttribute(r,t[r])}),e}
},{"dom101/matches":"r6WJ"}],"hE9p":[function(require,module,exports) {
"use strict";Object.defineProperty(exports,"__esModule",{value:!0}),exports.default=f,exports.groupify=p;var r=e(require("dom101/m
atches")),t=e(require("dom101/add-class")),n=require("../helpers/dom");function e(r){return r&&r.__esModule?r:{default:r}}function
 o(r){return c(r)||u(r)||i(r)||a()}function a(){throw new TypeError("Invalid attempt to spread non-iterable instance.\nIn order to
 be iterable, non-array objects must have a [Symbol.iterator]() method.")}function i(r,t){if(r){if("string"==typeof r)return s(r,t
);var n=Object.prototype.toString.call(r).slice(8,-1);return"Object"===n&&r.constructor&&(n=r.constructor.name),"Map"===n||"Set"==
=n?Array.from(r):"Arguments"===n||/^(?:Ui|I)nt(?:8|16|32)(?:Clamped)?Array$/.test(n)?s(r,t):void 0}}function u(r){if("undefined"!=
typeof Symbol&&Symbol.iterator in Object(r))return Array.from(r)}function c(r){if(Array.isArray(r))return s(r)}function s(r,t){(nu
ll==t||t>r.length)&&(t=r.length);for(var n=0,e=new Array(t);n<t;n++)e[n]=r[n];return e}function f(r){l(r).forEach(function(r){(0,n
.findChildren)(r,"[data-js-h3-section-list]").forEach(function(r){d(r)})})}function l(r){return p(r,{tag:"h2",wrapperFn:function()
{return(0,n.createDiv)({class:"h2-section"})},bodyFn:function(){return(0,n.createDiv)({class:"body h3-section-list","data-js-h3-se
ction-list":""})}})}function d(r){return p(r,{tag:"h3",wrapperFn:function(){return(0,n.createDiv)({class:"h3-section"})},bodyFn:fu
nction(){return(0,n.createDiv)({class:"body"})}})}function p(e,a){var i=a.tag,u=a.wrapperFn,c=a.bodyFn,s=e.children[0],f=[];if(s&&
!(0,r.default)(s,i)){var l=(0,n.nextUntil)(s,i);f.push(d(s,null,[s].concat(o(l))))}return(0,n.findChildren)(e,i).forEach(function(
r){var t=(0,n.nextUntil)(r,i);f.push(d(r,r,t))}),f;function d(r,e,o){var a=u(),i=r.className;i&&(0,t.default)(a,i),(0,n.before)(r,
a);var s=c();return i&&(0,t.default)(s,i),(0,n.appendMany)(s,o),e&&a.appendChild(e),a.appendChild(s),a}}
},{"dom101/matches":"r6WJ","dom101/add-class":"G20n","../helpers/dom":"THIL"}],"DJ2P":[function(require,module,exports) {
function n(n,t,e){n.addEventListener?n.addEventListener(t,e):n.attachEvent("on"+t,function(){e.call(n)})}module.exports=n;
},{}],"eoMl":[function(require,module,exports) {
"use strict";var e=u(require("./wrapify")),d=u(require("dom101/add-class")),t=u(require("dom101/on"));function u(e){return e&&e.__
esModule?e:{default:e}}var a,o=document.querySelector("[data-js-main-body]");function r(){a||((0,d.default)(document.documentEleme
nt,"LoadDone"),a=!0)}o&&((0,e.default)(o),(0,d.default)(o,"-wrapified")),(0,t.default)(window,"load",r),setTimeout(r,5e3);
},{"./wrapify":"hE9p","dom101/add-class":"G20n","dom101/on":"DJ2P"}]},{},["eoMl"], null)</script>
<script src='./assets/packed/app.js?t=20210520015833'></script>
</body>
</html>

BUILD SUCCESSFUL in 1s
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\ivanFernandez\Documents\iafernan-SER321ex\ser321examples\Network\SimpleGrabURL>
```

2.4. Set up your second system

For my second system I have set up AWS. Here is the link to my screencast:

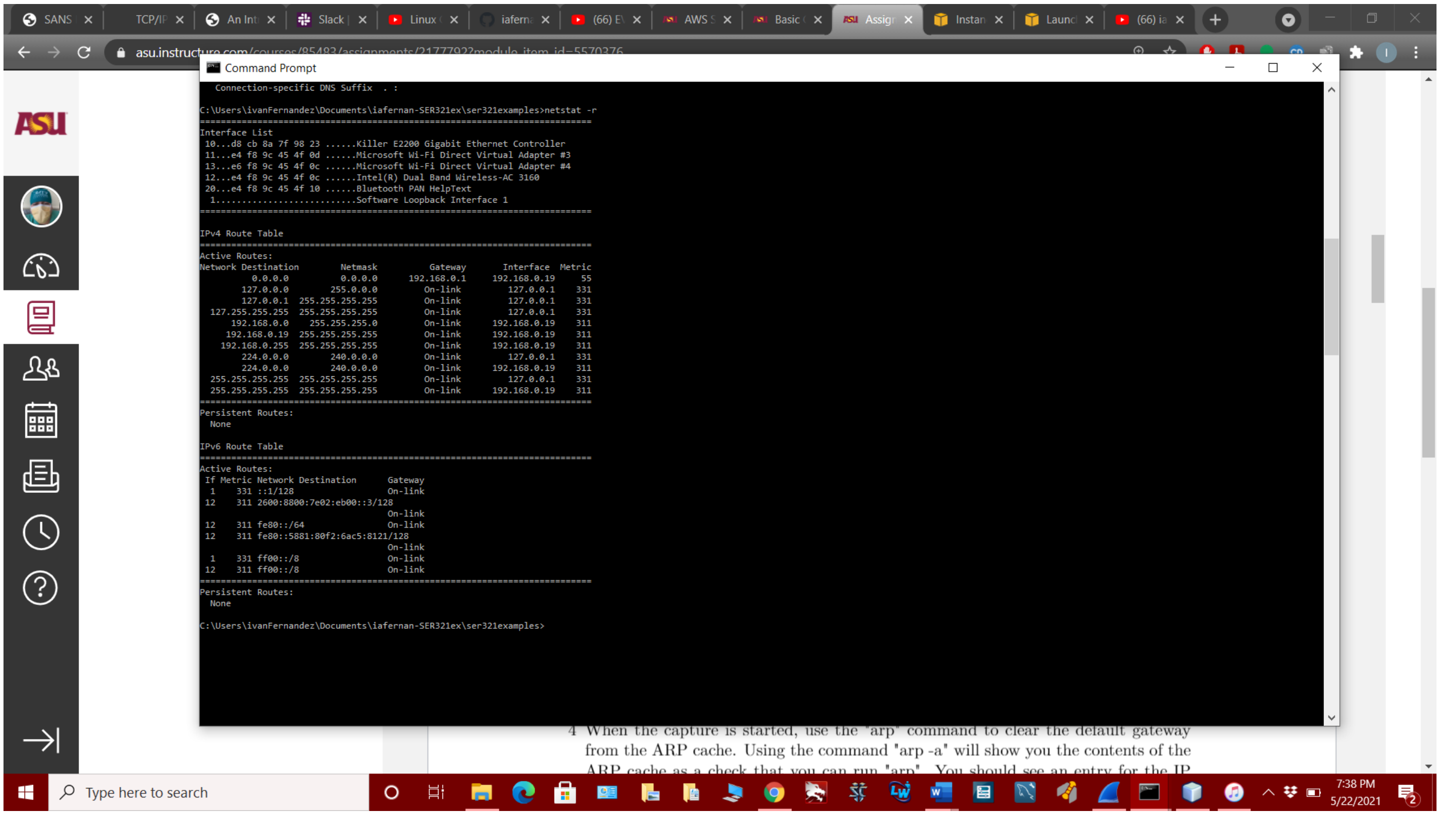
https://youtu.be/uE9d7Jo5QXE

Part II.   Networking

3.1. Explore the Data Link Layer with ARP

Step 1: Capture a Trace

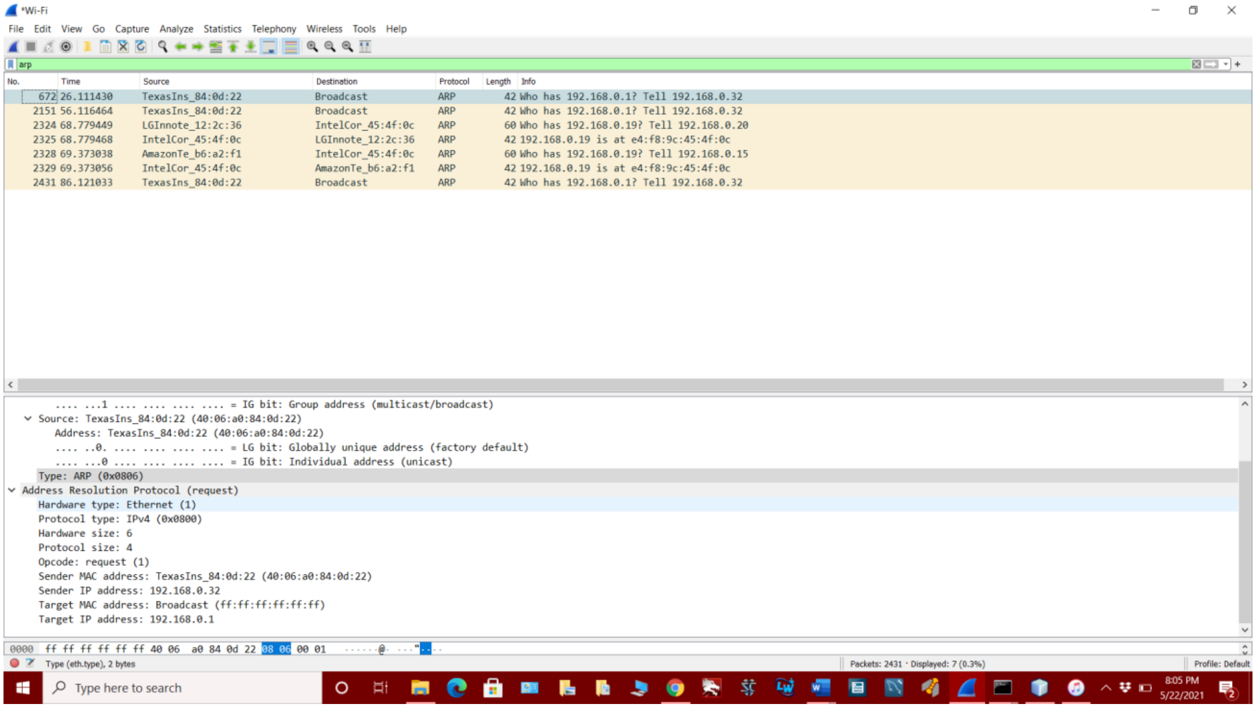
Deliverable: Provide a screen capture of your calls to identify your network interface and gateway.



```
   Connection-specific DNS Suffix  . :

C:\Users\ivanFernandez\Documents\iafernan-SER321ex\ser321examples>netstat -r
===========================================================================
Interface List
 10...d8 cb 8a 7f 98 23 ......Killer E2200 Gigabit Ethernet Controller
 11...e4 f8 9c 45 4f 0d ......Microsoft Wi-Fi Direct Virtual Adapter #3
 13...e6 f8 9c 45 4f 0c ......Microsoft Wi-Fi Direct Virtual Adapter #4
 12...e4 f8 9c 45 4f 0c ......Intel(R) Dual Band Wireless-AC 3160
 20...e4 f8 9c 45 4f 10 ......Bluetooth PAN HelpText
  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     192.168.0.19     55
        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      192.168.0.19    311
     192.168.0.19  255.255.255.255         On-link      192.168.0.19    311
    192.168.0.255  255.255.255.255         On-link      192.168.0.19    311
        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      192.168.0.19    311
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
  255.255.255.255  255.255.255.255         On-link      192.168.0.19    311
===========================================================================
Persistent Routes:
  None

IPv6 Route Table
===========================================================================
Active Routes:
 If Metric Network Destination      Gateway
  1    331 ::1/128                  On-link
 12    311 2600:8800:7e02:eb00::3/128
                                    On-link
 12    311 fe80::/64                On-link
 12    311 fe80::5881:80f2:6ac5:8121/128
                                    On-link
  1    331 ff00::/8                 On-link
 12    311 ff00::/8                 On-link
===========================================================================
Persistent Routes:
  None

C:\Users\ivanFernandez\Documents\iafernan-SER321ex\ser321examples>
```

4 When the capture is started, use the "arp" command to clear the default gateway from the ARP cache. Using the command "arp -a" will show you the contents of the ARP cache as a check that you can run "arp". You should see an entry for the IP

Deliverable: Provide a screen capture of your Wireshark instance with the appropriate filters.



*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

arp

```
No.     Time       Source               Destination          Protocol Length Info
    672 26.111430  TexasIns_84:0d:22    Broadcast            ARP        42 Who has 192.168.0.1? Tell 192.168.0.32
   2151 56.116464  TexasIns_84:0d:22    Broadcast            ARP        42 Who has 192.168.0.1? Tell 192.168.0.32
   2324 68.779449  LGInnote_12:2c:36    IntelCor_45:4f:0c    ARP        60 Who has 192.168.0.19? Tell 192.168.0.20
   2325 68.779468  IntelCor_45:4f:0c    LGInnote_12:2c:36    ARP        42 192.168.0.19 is at e4:f8:9c:45:4f:0c
   2328 69.373038  AmazonTe_b6:a2:f1    IntelCor_45:4f:0c    ARP        60 Who has 192.168.0.19? Tell 192.168.0.15
   2329 69.373056  IntelCor_45:4f:0c    AmazonTe_b6:a2:f1    ARP        42 192.168.0.19 is at e4:f8:9c:45:4f:0c
   2431 86.121033  TexasIns_84:0d:22    Broadcast            ARP        42 Who has 192.168.0.1? Tell 192.168.0.32
```

```
        .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
    Source: TexasIns_84:0d:22 (40:06:a0:84:0d:22)
        Address: TexasIns_84:0d:22 (40:06:a0:84:0d:22)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: ARP (0x0806)
Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: TexasIns_84:0d:22 (40:06:a0:84:0d:22)
    Sender IP address: 192.168.0.32
    Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
    Target IP address: 192.168.0.1
```

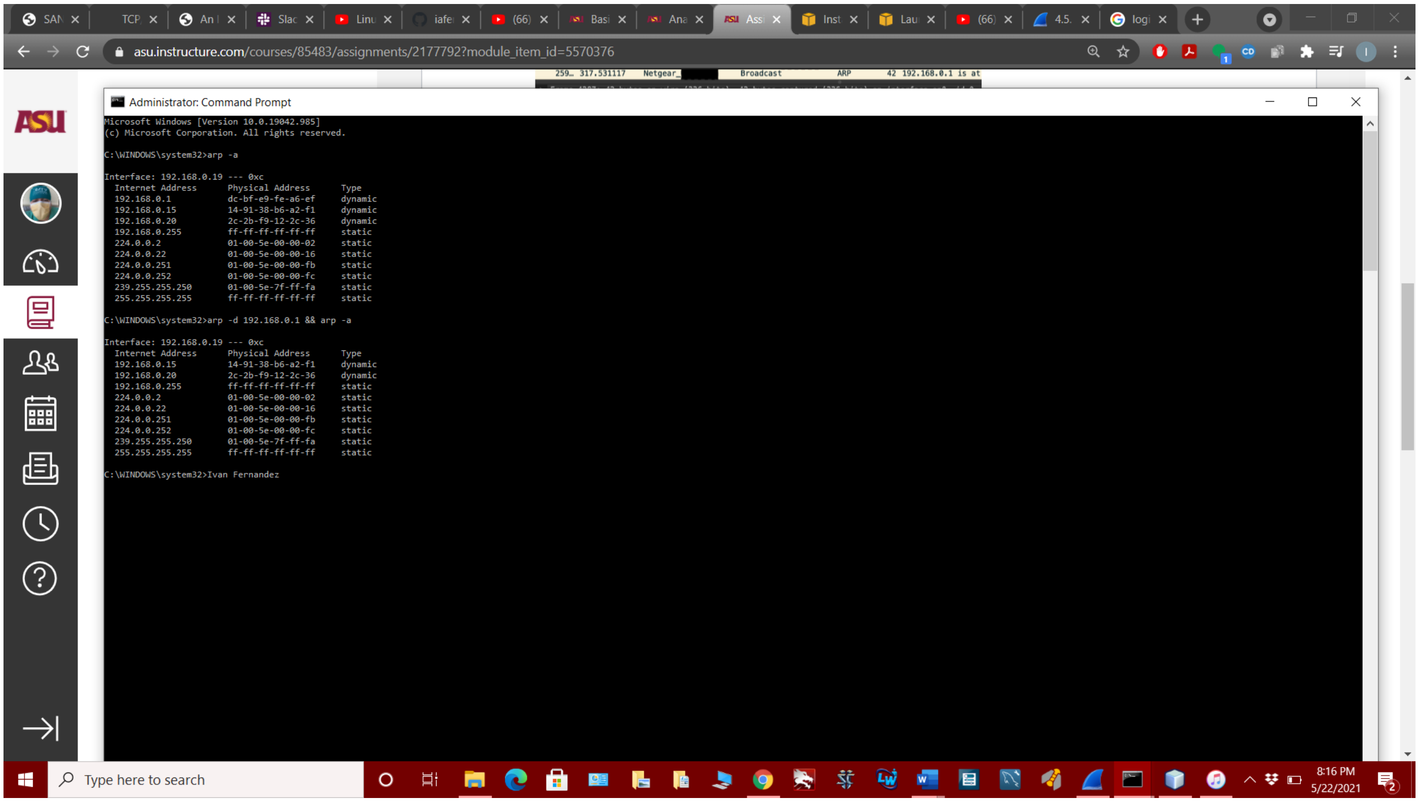
```
0000  ff ff ff ff ff ff 40 06  a0 84 0d 22 08 06 00 01   ······@· ···"····
```

Type (eth.type), 2 bytes

Packets: 2431 · Displayed: 7 (0.3%)

Profile: Default

Deliverable: Provide screen captures of your arp -a and arp -d commands. After runningthe arp -d be sure to run arp -a again to demonstrate the node successfully deleted.

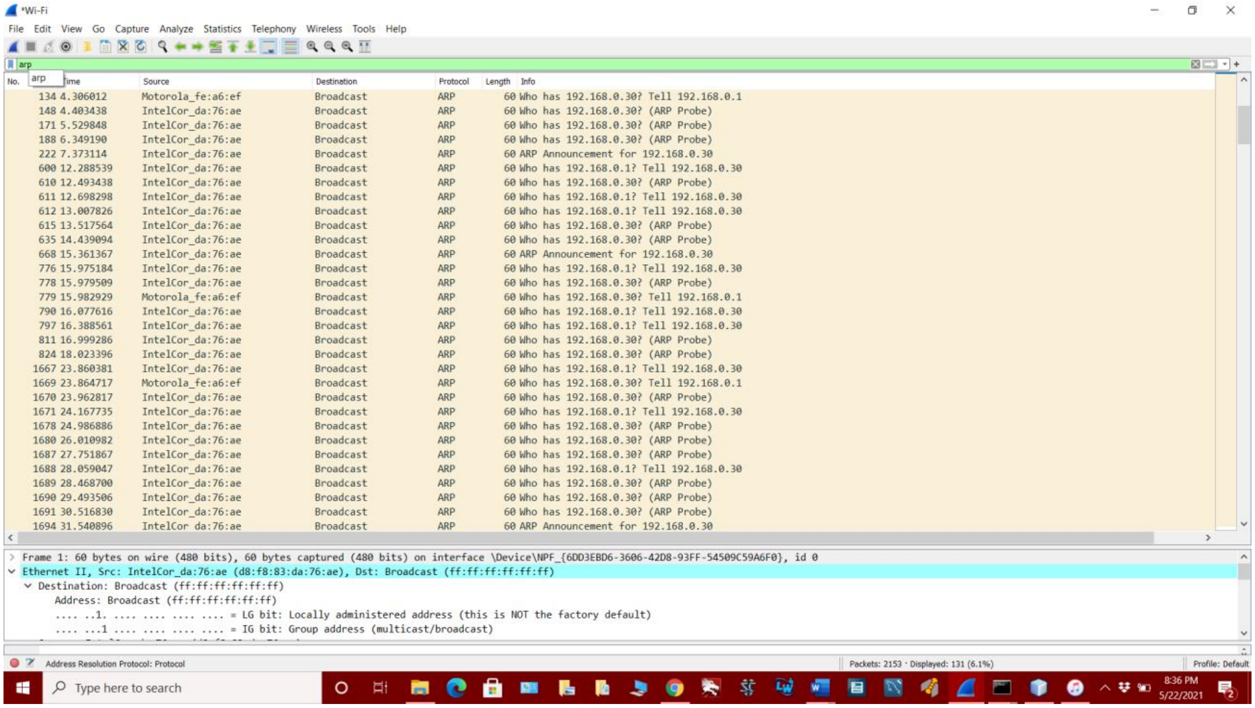


asu.instructure.com/courses/85483/assignments/2177792?module_item_id=5570376

Administrator: Command Prompt

```
Microsoft Windows [Version 10.0.19042.985]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>arp -a

Interface: 192.168.0.19 --- 0xc
  Internet Address      Physical Address      Type
  192.168.0.1           dc-bf-e9-fe-a6-ef     dynamic
  192.168.0.15          14-91-38-b6-a2-f1     dynamic
  192.168.0.20          2c-2b-f9-12-2c-36     dynamic
  192.168.0.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\WINDOWS\system32>arp -d 192.168.0.1 && arp -a

Interface: 192.168.0.19 --- 0xc
  Internet Address      Physical Address      Type
  192.168.0.15          14-91-38-b6-a2-f1     dynamic
  192.168.0.20          2c-2b-f9-12-2c-36     dynamic
  192.168.0.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\WINDOWS\system32>Ivan Fernandez
```

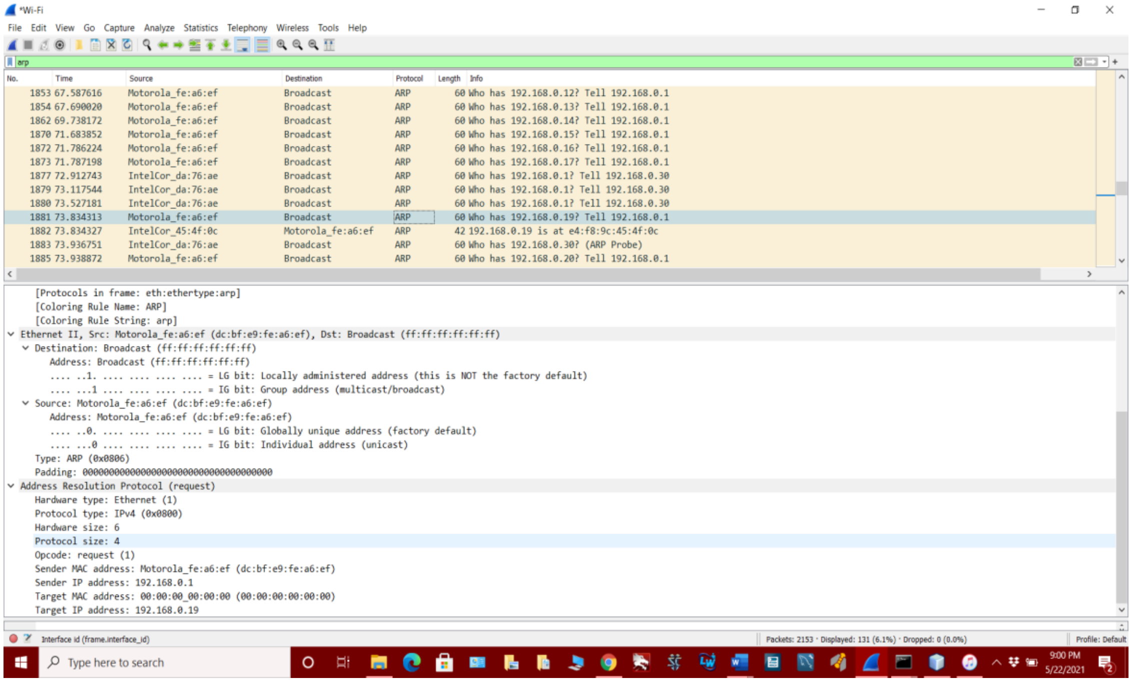
Deliverable: Capture the updated trace in Wireshark and add to your document.


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
arp
Type here to search
8:36 PM
5/22/2021

```
No.   Time        Source              Destination   Protocol  Length  Info
  134 4.306012    Motorola_fe:a6:ef   Broadcast     ARP       60 Who has 192.168.0.30? Tell 192.168.0.1
  148 4.403438    IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.30? (ARP Probe)
  171 5.529848    IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.30? (ARP Probe)
  188 6.349190    IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.30? (ARP Probe)
  222 7.373114    IntelCor_da:76:ae   Broadcast     ARP       60 ARP Announcement for 192.168.0.30
  600 12.288539   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.1? Tell 192.168.0.30
  610 12.493438   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.30? (ARP Probe)
  611 12.698298   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.1? Tell 192.168.0.30
  612 13.007826   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.1? Tell 192.168.0.30
  615 13.517564   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.30? (ARP Probe)
  635 14.439094   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.30? (ARP Probe)
  668 15.361367   IntelCor_da:76:ae   Broadcast     ARP       60 ARP Announcement for 192.168.0.30
  776 15.975184   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.1? Tell 192.168.0.30
  778 15.979509   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.30? (ARP Probe)
  779 15.982929   Motorola_fe:a6:ef   Broadcast     ARP       60 Who has 192.168.0.30? Tell 192.168.0.1
  790 16.077616   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.1? Tell 192.168.0.30
  797 16.388561   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.1? Tell 192.168.0.30
  811 16.999286   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.30? (ARP Probe)
  824 18.023396   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.30? (ARP Probe)
 1667 23.860381   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.1? Tell 192.168.0.30
 1669 23.864717   Motorola_fe:a6:ef   Broadcast     ARP       60 Who has 192.168.0.30? Tell 192.168.0.1
 1670 23.962817   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.30? (ARP Probe)
 1671 24.167735   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.1? Tell 192.168.0.30
 1678 24.986886   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.30? (ARP Probe)
 1680 26.010982   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.30? (ARP Probe)
 1687 27.751867   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.30? (ARP Probe)
 1688 28.059047   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.1? Tell 192.168.0.30
 1689 28.468700   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.30? (ARP Probe)
 1690 29.493506   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.30? (ARP Probe)
 1691 30.516830   IntelCor_da:76:ae   Broadcast     ARP       60 Who has 192.168.0.30? (ARP Probe)
 1694 31.540896   IntelCor_da:76:ae   Broadcast     ARP       60 ARP Announcement for 192.168.0.30

> Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{6DD3EBD6-3606-42D8-93FF-54509C59A6F0}, id 0
v Ethernet II, Src: IntelCor_da:76:ae (d8:f8:83:da:76:ae), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  v Destination: Broadcast (ff:ff:ff:ff:ff:ff)
      Address: Broadcast (ff:ff:ff:ff:ff:ff)
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory default)
      .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)

Address Resolution Protocol: Protocol          Packets: 2153 · Displayed: 131 (6.1%)          Profile: Default
```

Step 2: Inspect the Trace

Deliverable: Capture the ARP request and reply from this step and add to your document.


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
arp
Type here to search
9:00 PM
5/22/2021

```
No.   Time        Source              Destination         Protocol  Length  Info
 1853 67.587616   Motorola_fe:a6:ef   Broadcast           ARP       60 Who has 192.168.0.12? Tell 192.168.0.1
 1854 67.690020   Motorola_fe:a6:ef   Broadcast           ARP       60 Who has 192.168.0.13? Tell 192.168.0.1
 1862 69.738172   Motorola_fe:a6:ef   Broadcast           ARP       60 Who has 192.168.0.14? Tell 192.168.0.1
 1870 71.683852   Motorola_fe:a6:ef   Broadcast           ARP       60 Who has 192.168.0.15? Tell 192.168.0.1
 1872 71.786224   Motorola_fe:a6:ef   Broadcast           ARP       60 Who has 192.168.0.16? Tell 192.168.0.1
 1873 71.787198   Motorola_fe:a6:ef   Broadcast           ARP       60 Who has 192.168.0.17? Tell 192.168.0.1
 1877 72.912743   IntelCor_da:76:ae   Broadcast           ARP       60 Who has 192.168.0.1? Tell 192.168.0.30
 1879 73.117544   IntelCor_da:76:ae   Broadcast           ARP       60 Who has 192.168.0.1? Tell 192.168.0.30
 1880 73.527181   IntelCor_da:76:ae   Broadcast           ARP       60 Who has 192.168.0.1? Tell 192.168.0.30
 1881 73.834313   Motorola_fe:a6:ef   Broadcast           ARP       60 Who has 192.168.0.19? Tell 192.168.0.1
 1882 73.834327   IntelCor_45:4f:0c   Motorola_fe:a6:ef   ARP       42 192.168.0.19 is at e4:f8:9c:45:4f:0c
 1883 73.936751   IntelCor_da:76:ae   Broadcast           ARP       60 Who has 192.168.0.30? (ARP Probe)
 1885 73.938872   Motorola_fe:a6:ef   Broadcast           ARP       60 Who has 192.168.0.20? Tell 192.168.0.1

    [Protocols in frame: eth:ethertype:arp]
    [Coloring Rule Name: ARP]
    [Coloring Rule String: arp]
v Ethernet II, Src: Motorola_fe:a6:ef (dc:bf:e9:fe:a6:ef), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  v Destination: Broadcast (ff:ff:ff:ff:ff:ff)
      Address: Broadcast (ff:ff:ff:ff:ff:ff)
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory default)
      .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
  v Source: Motorola_fe:a6:ef (dc:bf:e9:fe:a6:ef)
      Address: Motorola_fe:a6:ef (dc:bf:e9:fe:a6:ef)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
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: Motorola_fe:a6:ef (dc:bf:e9:fe:a6:ef)
    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: 192.168.0.19

Interface id (frame.interface_id)          Packets: 2153 · Displayed: 131 (6.1%) · Dropped: 0 (0.0%)          Profile: Default
```

Step 3: Details of ARP over Ethernet

1. What opcode is used to indicate a request? What about a reply?

The opcode that indicates a request is 1. Conversely, the opcode that indicates a reply is 2.

2. How large is the ARP header for a request? What about for a reply?

The ARP header size for a request and a reply is 28 bytes.

3. What value is carried on a request for the unknown target MAC address?

00:00:00:00:00:00

4. What Ethernet Type value indicates that ARP is the higher layer protocol?

The Ethernet Type value for ARP is 0x806.

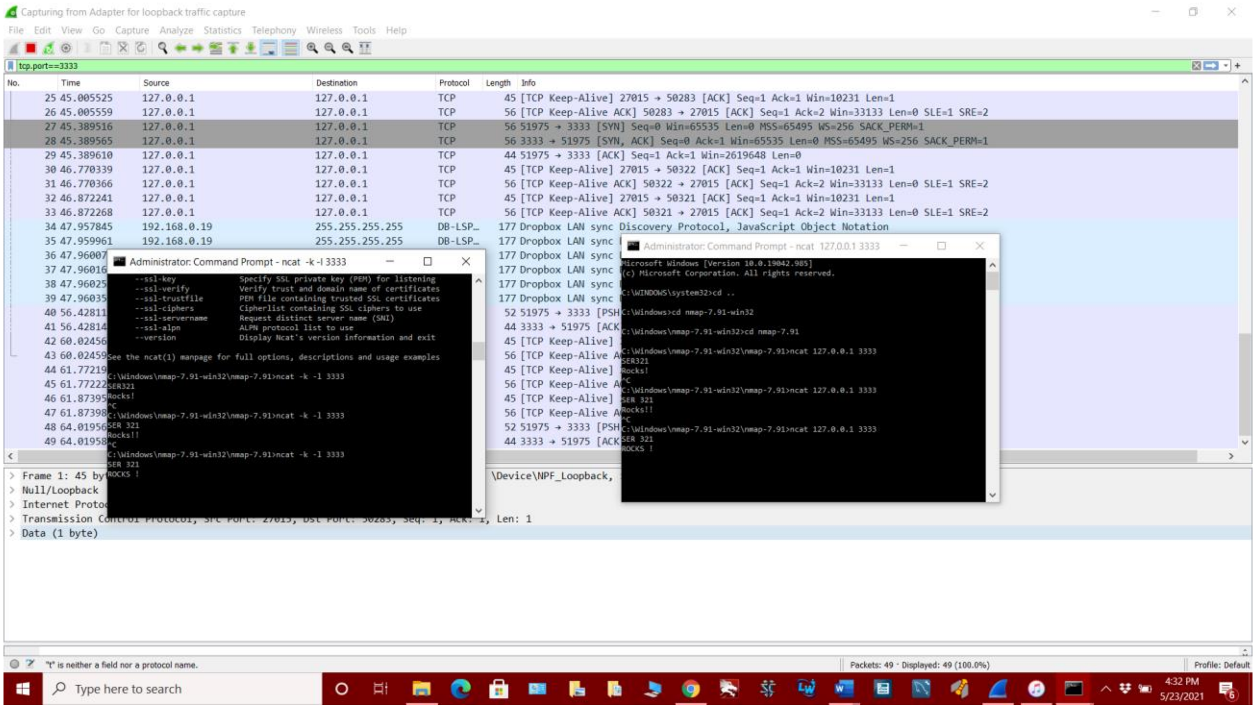
3.2. Understanding TCP network sockets

In windows:

netstat -a 30 | findstr “ESTABLISHED LISTENING” >> output.txt

3.3. Sniffing TCP/UDP traffic

Step 1 (TCP)



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
   25 45.005525   127.0.0.1       127.0.0.1        TCP        45 [TCP Keep-Alive] 27015 → 50283 [ACK] Seq=1 Ack=1 Win=10231 Len=1
   26 45.005559   127.0.0.1       127.0.0.1        TCP        56 [TCP Keep-Alive ACK] 50283 → 27015 [ACK] Seq=1 Ack=2 Win=33133 Len=0 SLE=1 SRE=2
   27 45.389516   127.0.0.1       127.0.0.1        TCP        56 51975 → 3333 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
   28 45.389565   127.0.0.1       127.0.0.1        TCP        56 3333 → 51975 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
   29 45.389610   127.0.0.1       127.0.0.1        TCP        44 51975 → 3333 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
   30 46.770339   127.0.0.1       127.0.0.1        TCP        45 [TCP Keep-Alive] 27015 → 50322 [ACK] Seq=1 Ack=1 Win=10231 Len=1
   31 46.770366   127.0.0.1       127.0.0.1        TCP        56 [TCP Keep-Alive ACK] 50322 → 27015 [ACK] Seq=1 Ack=2 Win=33133 Len=0 SLE=1 SRE=2
   32 46.872241   127.0.0.1       127.0.0.1        TCP        45 [TCP Keep-Alive] 27015 → 50321 [ACK] Seq=1 Ack=1 Win=10231 Len=1
   33 46.872268   127.0.0.1       127.0.0.1        TCP        56 [TCP Keep-Alive ACK] 50321 → 27015 [ACK] Seq=1 Ack=2 Win=33133 Len=0 SLE=1 SRE=2
   34 47.957845   192.168.0.19    255.255.255.255  DB-LSP…   177 Dropbox LAN sync Discovery Protocol, JavaScript Object Notation
   35 47.959961   192.168.0.19    255.255.255.255  DB-LSP…   177 Dropbox LAN sync
   36 47.96007                                             177 Dropbox LAN sync
   37 47.96016                                             177 Dropbox LAN sync
   38 47.96025                                             177 Dropbox LAN sync
   39 47.96035                                             177 Dropbox LAN sync
   40 56.42811                                              52 51975 → 3333 [PSH
   41 56.42814                                              44 3333 → 51975 [ACK
   42 60.02456                                              45 [TCP Keep-Alive]
   43 60.02459                                              56 [TCP Keep-Alive A
   44 61.77219                                              45 [TCP Keep-Alive]
   45 61.77222                                              56 [TCP Keep-Alive A
   46 61.87395                                              45 [TCP Keep-Alive]
   47 61.87398                                              56 [TCP Keep-Alive A
   48 64.01956                                              52 51975 → 3333 [PSH
   49 64.01958                                              44 3333 → 51975 [ACK
```

```
> Frame 1: 45 by                                  \Device\NPF_Loopback,
> Null/Loopback
> Internet Proto
> Transmission Control Protocol, Src Port: 27015, Dst Port: 50283, Seq: 1, Ack: 1, Len: 1
> Data (1 byte)
```

Administrator: Command Prompt - ncat -k -l 3333

```
      --ssl-key              Specify SSL private key (PEM) for listening
      --ssl-verify           Verify trust and domain name of certificates
      --ssl-trustfile        PEM file containing trusted SSL certificates
      --ssl-ciphers          Cipherlist containing SSL ciphers to use
      --ssl-servername       Request distinct server name (SNI)
      --ssl-alpn             ALPN protocol list to use
      --version              Display Ncat's version information and exit

See the ncat(1) manpage for full options, descriptions and usage examples

C:\Windows\nmap-7.91-win32\nmap-7.91>ncat -k -l 3333
SER321
Rocks!
^C
C:\Windows\nmap-7.91-win32\nmap-7.91>ncat -k -l 3333
SER 321
Rocks!!
^C
C:\Windows\nmap-7.91-win32\nmap-7.91>ncat -k -l 3333
SER 321
ROCKS !
```

Administrator: Command Prompt - ncat 127.0.0.1 3333

```
Microsoft Windows [Version 10.0.19042.985]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd ..

C:\Windows>cd nmap-7.91-win32

C:\Windows\nmap-7.91-win32>cd nmap-7.91

C:\Windows\nmap-7.91-win32\nmap-7.91>ncat 127.0.0.1 3333
SER321
Rocks!
^C
C:\Windows\nmap-7.91-win32\nmap-7.91>ncat 127.0.0.1 3333
SER 321
Rocks!!
^C
C:\Windows\nmap-7.91-win32\nmap-7.91>ncat 127.0.0.1 3333
SER 321
ROCKS !
```

"t" is neither a field nor a protocol name.

Packets: 49 · Displayed: 49 (100.0%)

Profile: Default

Using the capture file (open it with Wireshark), answer the following questions.

a) How many frames were needed to capture those 2 lines?

Two frames:

193 36.905413 127.0.0.1 127.0.0.1 TCP 52 50064 → 3333 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=8

195 42.420621 127.0.0.1 127.0.0.1 TCP 51 50064 → 3333 [PSH, ACK] Seq=9 Ack=1 Win=2619648 Len=7

b) How many packets were needed to capture those 2 lines?

Four packets were required:

193 36.905413 127.0.0.1 127.0.0.1 TCP 52 50064 → 3333 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=8

194 36.905447 127.0.0.1 127.0.0.1 TCP 44 3333 → 50064 [ACK] Seq=1 Ack=9 Win=2619648 Len=0

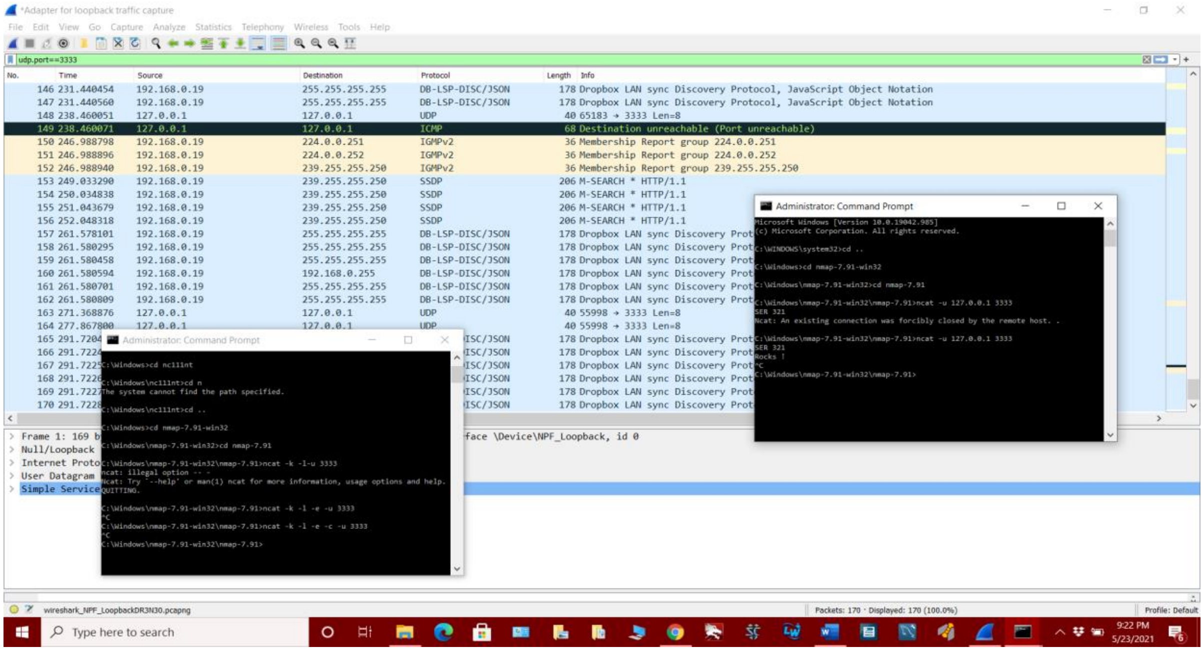
195 42.420621 127.0.0.1 127.0.0.1 TCP 51 50064 → 3333 [PSH, ACK] Seq=9 Ack=1 Win=2619648 Len=7

196 42.420653 127.0.0.1 127.0.0.1 TCP 44 3333 → 50064 [ACK] Seq=1 Ack=16 Win=2619648 Len=0

c) How many total bytes went over the wire? How much overhead was there(basically the percentage of traffic that was not needed to send SER321 Rocks!)?

15 bytes total. 103 total bytes – 15 bytes = 88 bytes. 88 bytes / 103 bytes * 100 = 85%

Step 2 (UDP)



*Adapter for loopback traffic capture
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp.port==3333
Administrator: Command Prompt
Administrator: Command Prompt
wireshark_NPF_LoopbackDR3N30.pcapng
Packets: 170 · Displayed: 170 (100.0%)
Profile: Default
Type here to search
9:22 PM
5/23/2021

```
No.  Time        Source        Destination      Protocol          Length Info
146 231.440454   192.168.0.19  255.255.255.255  DB-LSP-DISC/JSON  178 Dropbox LAN sync Discovery Protocol, JavaScript Object Notation
147 231.440560   192.168.0.19  255.255.255.255  DB-LSP-DISC/JSON  178 Dropbox LAN sync Discovery Protocol, JavaScript Object Notation
148 238.460051   127.0.0.1     127.0.0.1        UDP                40 65183 → 3333 Len=8
149 238.460071   127.0.0.1     127.0.0.1        ICMP               68 Destination unreachable (Port unreachable)
150 246.988798   192.168.0.19  224.0.0.251      IGMPv2             36 Membership Report group 224.0.0.251
151 246.988896   192.168.0.19  224.0.0.252      IGMPv2             36 Membership Report group 224.0.0.252
152 246.988940   192.168.0.19  239.255.255.250  IGMPv2             36 Membership Report group 239.255.255.250
153 249.033290   192.168.0.19  239.255.255.250  SSDP              206 M-SEARCH * HTTP/1.1
154 250.034838   192.168.0.19  239.255.255.250  SSDP              206 M-SEARCH * HTTP/1.1
155 251.043679   192.168.0.19  239.255.255.250  SSDP              206 M-SEARCH * HTTP/1.1
156 252.048318   192.168.0.19  239.255.255.250  SSDP              206 M-SEARCH * HTTP/1.1
157 261.578101   192.168.0.19  255.255.255.255  DB-LSP-DISC/JSON  178 Dropbox LAN sync Discovery Prot
158 261.580295   192.168.0.19  255.255.255.255  DB-LSP-DISC/JSON  178 Dropbox LAN sync Discovery Prot
159 261.580458   192.168.0.19  255.255.255.255  DB-LSP-DISC/JSON  178 Dropbox LAN sync Discovery Prot
160 261.580594   192.168.0.19  192.168.0.255    DB-LSP-DISC/JSON  178 Dropbox LAN sync Discovery Prot
161 261.580701   192.168.0.19  255.255.255.255  DB-LSP-DISC/JSON  178 Dropbox LAN sync Discovery Prot
162 261.580809   192.168.0.19  255.255.255.255  DB-LSP-DISC/JSON  178 Dropbox LAN sync Discovery Prot
163 271.368876   127.0.0.1     127.0.0.1        UDP                40 55998 → 3333 Len=8
164 277.867800   127.0.0.1     127.0.0.1        UDP                40 55998 → 3333 Len=8
165 291.7204                                    ISC/JSON          178 Dropbox LAN sync Discovery Prot
166 291.7224                                    ISC/JSON          178 Dropbox LAN sync Discovery Prot
167 291.7225                                    ISC/JSON          178 Dropbox LAN sync Discovery Prot
168 291.7226                                    ISC/JSON          178 Dropbox LAN sync Discovery Prot
169 291.7227                                    ISC/JSON          178 Dropbox LAN sync Discovery Prot
170 291.7228                                    ISC/JSON          178 Dropbox LAN sync Discovery Prot

> Frame 1: 169 b                                face \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Proto
> User Datagram
> Simple Service
```

```
C:\Windows>cd nc111nt

C:\Windows\nc111nt>cd n
The system cannot find the path specified.

C:\Windows\nc111nt>cd ..

C:\Windows>cd nmap-7.91-win32

C:\Windows\nmap-7.91-win32>cd nmap-7.91

C:\Windows\nmap-7.91-win32\nmap-7.91>ncat -k -l-u 3333
ncat: illegal option -- -
Ncat: Try `--help' or man(1) ncat for more information, usage options and help. QUITTING.

C:\Windows\nmap-7.91-win32\nmap-7.91>ncat -k -l -e -u 3333
^C
C:\Windows\nmap-7.91-win32\nmap-7.91>ncat -k -l -e -c -u 3333
^C
C:\Windows\nmap-7.91-win32\nmap-7.91>
```

```
Microsoft Windows [Version 10.0.19042.985]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd ..

C:\Windows>cd nmap-7.91-win32

C:\Windows\nmap-7.91-win32>cd nmap-7.91

C:\Windows\nmap-7.91-win32\nmap-7.91>ncat -u 127.0.0.1 3333
SER 321
Ncat: An existing connection was forcibly closed by the remote host. .

C:\Windows\nmap-7.91-win32\nmap-7.91>ncat -u 127.0.0.1 3333
SER 321
Rocks !
^C
C:\Windows\nmap-7.91-win32\nmap-7.91>
```

Using the capture file (open it with Wireshark), answer the following questions

a) How many frames were needed to capture those 2 lines?

Two.

```
163    271.368876   127.0.0.1      127.0.0.1      UDP   40      55998 → 3333 Len=8
164    277.867800   127.0.0.1      127.0.0.1      UDP   40      55998 → 3333 Len=8
```

b) How many packets were needed to capture those 2 lines?

Two.

```
163    271.368876   127.0.0.1      127.0.0.1      UDP   40      55998 → 3333 Len=8
164    277.867800   127.0.0.1      127.0.0.1      UDP   40      55998 → 3333 Len=8
```

c) How many total bytes went over the wire? How much overhead was there (percent of bytes not in the above 2 lines)?

16 bytes total. 40 bytes total – 16 bytes = 24 bytes. 24 bytes / 40 bytes total * 100 = 60%

d) What is the difference in relative overhead between UDP and TCP and why? Specifically, what kind of information was exchanged in TCP that was not exchanged in UDP? Show the relative parts of the packet traces.

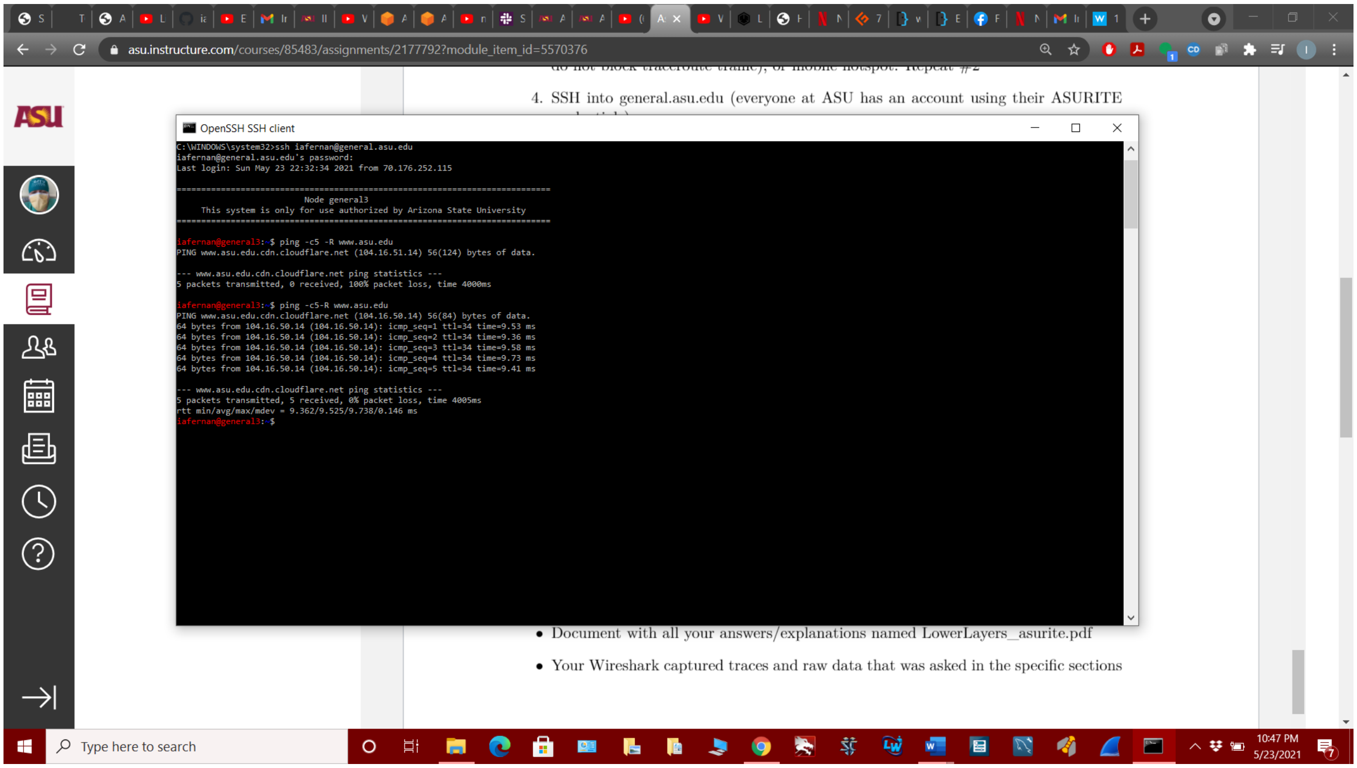
UDP carries less overheads, taking up less space than TCP, and thus is a lot faster than TCP. What we saw in the TCP exchanges in wireshark were a lot more packages and frames when compared to UDP. This is because TCP checks for the readiness of the receiver since it is a connection-oriented protocol. In UDP we see two packets for the transmission of data, whereas in TCP we see four. After the data is transmitted, the received packets are acknowledged by sending back a packet with an ACK bit set.

3.4. Internet Protocol (IP) Routing

Now compare the 3 routes and answer the following questions:

a) Which is the fastest?

The fastest was Route 3, taking only 4005 ms. The slowest was Route 2, taking over 216 seconds with using a hot spot from my mobile device.



asu.instructure.com/courses/85483/assignments/2177792?module_item_id=5570376

do not block traceroute traffic), or mobile hotspot. Repeat #2

4. SSH into general.asu.edu (everyone at ASU has an account using their ASURITE

OpenSSH SSH client

```
C:\WINDOWS\system32>ssh iafernan@general.asu.edu
iafernan@general.asu.edu's password:
Last login: Sun May 23 22:32:34 2021 from 70.176.252.115

============================================================================
                          Node general3
     This system is only for use authorized by Arizona State University
============================================================================

iafernan@general3:~$ ping -c5 -R www.asu.edu
PING www.asu.edu.cdn.cloudflare.net (104.16.51.14) 56(124) bytes of data.

--- www.asu.edu.cdn.cloudflare.net ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4000ms

iafernan@general3:~$ ping -c5-R www.asu.edu
PING www.asu.edu.cdn.cloudflare.net (104.16.50.14) 56(84) bytes of data.
64 bytes from 104.16.50.14 (104.16.50.14): icmp_seq=1 ttl=34 time=9.53 ms
64 bytes from 104.16.50.14 (104.16.50.14): icmp_seq=2 ttl=34 time=9.36 ms
64 bytes from 104.16.50.14 (104.16.50.14): icmp_seq=3 ttl=34 time=9.58 ms
64 bytes from 104.16.50.14 (104.16.50.14): icmp_seq=4 ttl=34 time=9.73 ms
64 bytes from 104.16.50.14 (104.16.50.14): icmp_seq=5 ttl=34 time=9.41 ms

--- www.asu.edu.cdn.cloudflare.net ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 9.362/9.525/9.738/0.146 ms
iafernan@general3:~$
```

- Document with all your answers/explanations named LowerLayers_asurite.pdf
- Your Wireshark captured traces and raw data that was asked in the specific sections

b) Which has the fewest hops?

Route 3 had the fewest hops too.