

Assignment 1 Part 2

OSI model

SER 321

Elliot Davis

10/22/2025

GitHub Repository Link: <https://github.com/snacksnackin/ser321-fall25-B-eldavi20.git>

** I am presently fixing my desktop computer, and my laptop does not run virtual machines well. **The assignment was completed using macOS Sequoia 15.5 and the Terminal.**

Part I.

1. Command line tasks

macOS Sequoia 15.5 Terminal:

```
1: mkdir cli_assignment
2: cd cli_assignment
3: touch stuff.txt
4: cat > stuff.txt
   ⇒ The quick brown fox
   ⇒ jumps over the lazy dog.
   ⇒ Ctrl+D
5: wc stuff.txt
6: echo "This is a new appended line." >> stuff.txt
7: mkdir draft
8: mv stuff.txt draft/
9: cd draft && touch secret.txt
10: cp -R . ../final
11: cd ..
   ⇒ mv draft draft.remove
12: mv draft.remove final/
13: ls -al
   ⇒ ls
14: zless ../NASA_access_log_Aug95.gz
   ⇒ q
15: gunzip ../NASA_access_log_Aug95.gz
16: mv ../NASA_access_log_Aug95 ../logs.txt
17: mv ../logs.txt .
18: head -n 100 logs.txt
19: head -n 100 logs.txt > logs_top_100.txt
20: tail -n 100 logs.txt
21: tail -n 100 logs.txt > logs_bottom_100.txt
```

```
22:cat logs_top_100.txt logs_bottom_100.txt > logs_snapshot.txt
23:echo "eldavi20: This is a great assignment: $(date)" >> logs_snapshot.txt
24:less logs.txt
    ⇒ q
25:tail -n +2 ..marks.csv | cut -d '%' -f 2
26:tail -n +2 ..marks.csv | cut -d '%' -f 4 | sort -n
27:awk -F '%' '{sum+=$3} END {print sum/(NR-1)}' ..marks.csv
28:awk -F '%' '{sum+=$3} END {print sum/(NR-1)}' ..marks.csv > done.txt
29:mv done.txt final/
30:mv final/done.txt final/average.txt
```

2. Some Setup and Examples

2.1. Setup a GitHub repo so you version control your work

GitHub Repository Link: <https://github.com/snacksnackin/ser321-fall25-B-eldavi20.git>

2.2 Running examples

Example 1 – Serialization/ClassMigration

```
Last login: Thu Oct 23 23:09:21 on ttys000
[dog@elliots-MacBook-Air ~ % cd ser32lexamples/Serilization/ClassMigration
[dog@elliots-MacBook-Air ClassMigration % gradle runUser

> Task :runUser
User: private static final long serialVersionUID = -3153542405433484184L;

BUILD SUCCESSFUL in 318ms
2 actionable tasks: 1 executed, 1 up-to-date
Consider enabling configuration cache to speed up this build: https://docs.gradle.org/9.1.0/userguide/configuration_cache_enabling.html
[dog@elliots-MacBook-Air ClassMigration % gradle runFileSerialize --args 'write'

> Task :runFileSerialize
Done externalizing User with id: Joe has password: hisWord to file user.ser

BUILD SUCCESSFUL in 338ms
2 actionable tasks: 1 executed, 1 up-to-date
Consider enabling configuration cache to speed up this build: https://docs.gradle.org/9.1.0/userguide/configuration_cache_enabling.html
[dog@elliots-MacBook-Air ClassMigration % gradle runFileSerialize --args 'read'

> Task :runFileSerialize
User de-serialize from file is: User with id: Joe has password: hisWord

BUILD SUCCESSFUL in 31ms
2 actionable tasks: 1 executed, 1 up-to-date
Consider enabling configuration cache to speed up this build: https://docs.gradle.org/9.1.0/userguide/configuration_cache_enabling.html
[dog@elliots-MacBook-Air ClassMigration % nano User.java
[dog@elliots-MacBook-Air ClassMigration % cd src/main/java
[dog@elliots-MacBook-Air java % nano User.java
[dog@elliots-MacBook-Air java % cd ..
[dog@elliots-MacBook-Air main % cd ..
[dog@elliots-MacBook-Air src % cd ..
[dog@elliots-MacBook-Air ClassMigration % gradle runFileSerialize --args 'read'

> Task :runFileSerialize
User de-serialize from file is: User with id: Joe has password: hisWord

BUILD SUCCESSFUL in 356ms
2 actionable tasks: 2 executed
Consider enabling configuration cache to speed up this build: https://docs.gradle.org/9.1.0/userguide/configuration_cache_enabling.html
[dog@elliots-MacBook-Air ClassMigration % gradle runUser

> Task :runUser
User: private static final long serialVersionUID = -3153542405433484184L;

BUILD SUCCESSFUL in 342ms
2 actionable tasks: 2 executed
Consider enabling configuration cache to speed up this build: https://docs.gradle.org/9.1.0/userguide/configuration_cache_enabling.html
[dog@elliots-MacBook-Air ClassMigration % gradle runFileSerialize --args 'read'

> Task :runFileSerialize
User de-serialize from file is: User with id: Joe has password: hisWord

BUILD SUCCESSFUL in 312ms
2 actionable tasks: 1 executed, 1 up-to-date
[dog@elliots-MacBook-Air ClassMigration % gradle runUser

> Task :runUser
User: private static final long serialVersionUID = -3153542405433484184L;

BUILD SUCCESSFUL in 318ms
2 actionable tasks: 1 executed, 1 up-to-date
Consider enabling configuration cache to speed up this build: https://docs.gradle.org/9.1.0/userguide/configuration_cache_enabling.html
[dog@elliots-MacBook-Air ClassMigration % gradle runFileSerialize --args 'write'

> Task :runFileSerialize
Done externalizing User with id: Joe has password: hisWord to file user.ser

BUILD SUCCESSFUL in 338ms
2 actionable tasks: 1 executed, 1 up-to-date
Consider enabling configuration cache to speed up this build: https://docs.gradle.org/9.1.0/userguide/configuration_cache_enabling.html
[dog@elliots-MacBook-Air ClassMigration % gradle runFileSerialize --args 'read'

> Task :runFileSerialize
User de-serialize from file is: User with id: Joe has password: hisWord

BUILD SUCCESSFUL in 31ms
2 actionable tasks: 1 executed, 1 up-to-date
Consider enabling configuration cache to speed up this build: https://docs.gradle.org/9.1.0/userguide/configuration_cache_enabling.html
[dog@elliots-MacBook-Air ClassMigration % nano User.java
[dog@elliots-MacBook-Air ClassMigration % cd src/main/java
[dog@elliots-MacBook-Air java % nano User.java
[dog@elliots-MacBook-Air java % cd ..
[dog@elliots-MacBook-Air main % cd ..
[dog@elliots-MacBook-Air src % cd ..
[dog@elliots-MacBook-Air ClassMigration % gradle runFileSerialize --args 'read'

> Task :runFileSerialize
User de-serialize from file is: User with id: Joe has password: hisWord

BUILD SUCCESSFUL in 358ms
2 actionable tasks: 2 executed
Consider enabling configuration cache to speed up this build: https://docs.gradle.org/9.1.0/userguide/configuration_cache_enabling.html
[dog@elliots-MacBook-Air ClassMigration % gradle runUser

> Task :runUser
User: private static final long serialVersionUID = -3153542405433484184L;

BUILD SUCCESSFUL in 342ms
2 actionable tasks: 2 executed
Consider enabling configuration cache to speed up this build: https://docs.gradle.org/9.1.0/userguide/configuration_cache_enabling.html
[dog@elliots-MacBook-Air ClassMigration % gradle runFileSerialize --args 'read'

> Task :runFileSerialize
User de-serialize from file is: User with id: Joe has password: hisWord

[BUILD SUCCESSFUL in 312ms
2 actionable tasks: 1 executed, 1 up-to-date
Consider enabling configuration cache to speed up this build: https://docs.gradle.org/9.1.0/userguide/configuration_cache_enabling.html
[dog@elliots-MacBook-Air ClassMigration % ]
```

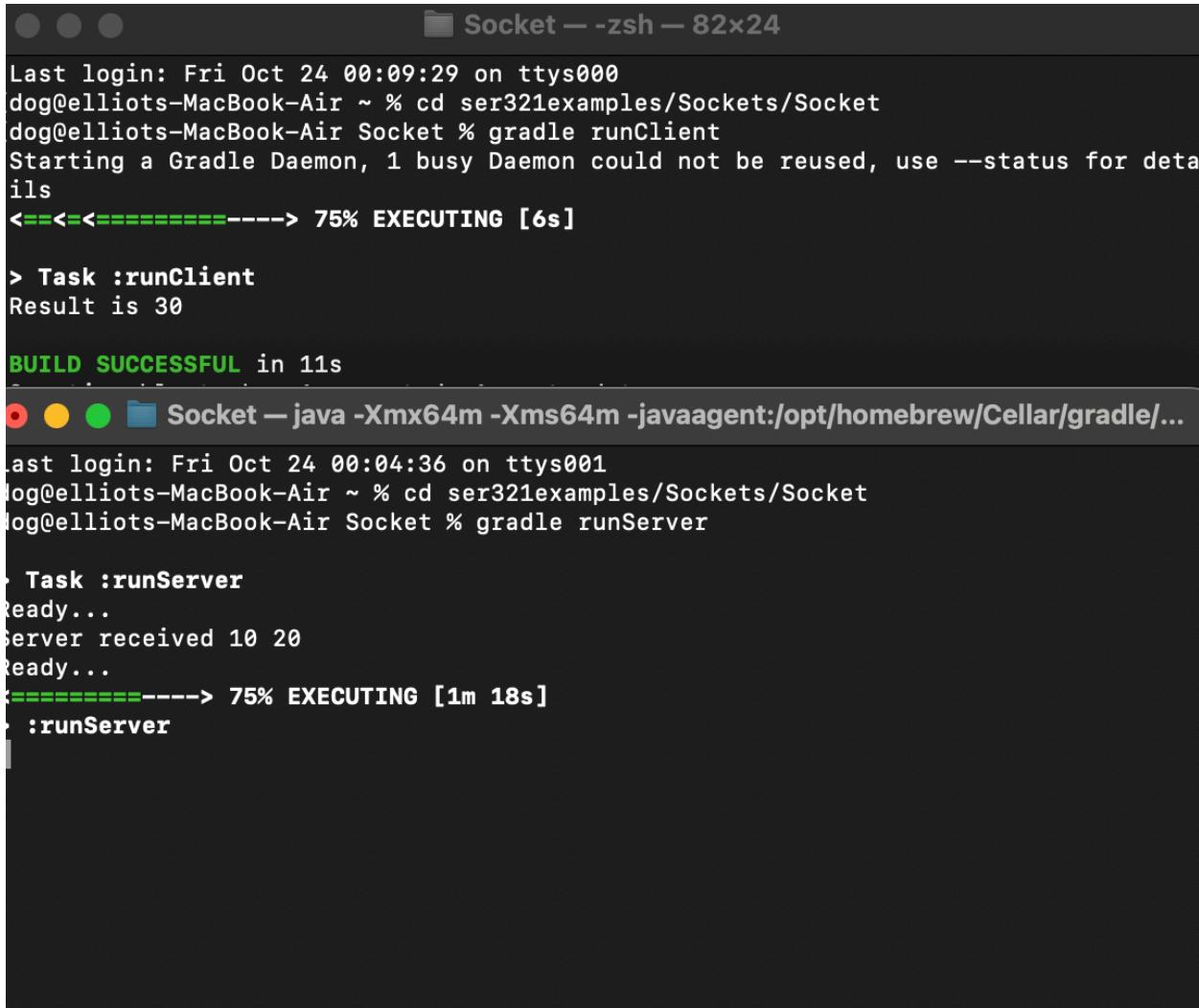
Commands:

- cd ser321examples/Serialization/ClassMigration
- gradle runUser
- gradle runFileSerialize --args 'write'
- gradle runFileSerialize --args 'read'
- cd src/main/java
- nano User.java
- cd ..
- cd ..
- cd ..
- gradle runFileSerialize --args 'read'
- cd src/main/java
- nano User.java
- cd ..
- cd ..
- cd ..
- gradle runUser
- gradle runFileSerialize --args 'read'

What it does:

The example shows how Java handles a class that changes its structure after objects of its older version have already been serialized to a file.

Example 2 – Sockets/Socket



The screenshot shows two terminal windows side-by-side. The left window is titled "Socket — -zsh — 82x24" and the right window is titled "Socket — java -Xmx64m -Xms64m -javaagent:/opt/homebrew/Cellar/gradle/...". Both windows show command-line output for running a Gradle task named "runClient" or "runServer". The output includes logs from the Gradle daemon, task execution progress (e.g., "75% EXECUTING [6s]"), and the final result ("Result is 30" or "BUILD SUCCESSFUL in 11s").

```
Last login: Fri Oct 24 00:09:29 on ttys000
dog@elliots-MacBook-Air ~ % cd ser321examples/Sockets/Socket
dog@elliots-MacBook-Air Socket % gradle runClient
Starting a Gradle Daemon, 1 busy Daemon could not be reused, use --status for details
<==<<=====--> 75% EXECUTING [6s]

> Task :runClient
Result is 30

BUILD SUCCESSFUL in 11s

Last login: Fri Oct 24 00:04:36 on ttys001
dog@elliots-MacBook-Air ~ % cd ser321examples/Sockets/Socket
dog@elliots-MacBook-Air Socket % gradle runServer

> Task :runServer
Ready...
Server received 10 20
Ready...
=====--> 75% EXECUTING [1m 18s]
:runServer
```

Commands:

- In 1st Terminal window:
 - cd ser321examples/Sockets/Socket
 - gradle runServer
 - In 2nd Terminal window:
 - cd ser321examples/Sockets/Socket
 - gradle runClient

What it does:

The example allows the user to locally host a server and client in the two Terminal windows that is able to send data back and forth.

Example 3 – Network/SimpleGrabURL

```
Last login: Thu Oct 23 23:52:56 on ttys000
dog@elliots-MacBook-Air ~ % cd sr321examples/Network/SimpleGrabURL
dog@elliots-MacBook-Air SimpleGrabURL % gradle run

> Task :run
> [DOCTYPE html]<html lang="en"> <head><meta charset="utf-8"><meta name="viewport" content="width=device-width"><meta name="generator" content="Astro v4.5.4"><link rel="shortcut icon" type="image/png" href="/assets/favicon.png"><script type="text/partytown" src="https://www.googletagmanager.com/gtag/js?id=G-N7TC6B227L"></script>
<script type="text/partytown" data-ga-measurement-id="G-N7TC6B227L" id="ga-init">
  const measurementId = document
    .getElementsByClassName('ga-init')
    .getAttribute('data-ga-measurement-id')
  window.dataLayer = window.dataLayer || []
  function gtag() {
    dataLayer.push(arguments) // eslint-disable-line
  }
  gtag('js', new Date())
  gtag('config', measurementId)
</script> <title>Bash scripting cheatsheet</title><meta name="description" content="Variables · Functions · Interpolation · Brace expansions · Loops · Conditional execution · Command substitution · One-page guide to Bash scripting"><meta name="app:pageurl" content="https://devhints.io/bash"><meta property="og:description" content="Variables · Functions · Interpolation · Brace expansions · Loops · Conditional execution · Command substitution · One-page guide to Bash scripting"><meta property="og:image" content="https://assets.devhints.io/previews/bash.jpg"><meta property="og:title" content="Bash scripting cheatsheet"><meta property="og:image:width" content="900"><meta property="og:image:height" content="471"><meta property="og:image" content="https://assets.devhints.io/previews/bash.jpg"><meta property="og:type" content="article"><meta property="og:site_name" content="Devhints.io cheatsheets"><meta property="og:title" content="Bash scripting cheatsheet"><meta property="og:type" content="article:section" content="CLI"><meta property="og:image" content="https://devhints.io/bash"/><meta property="article:tag" content="featured"><meta property="article:section" content="CLI"><meta property="twitter:title" content="Bash scripting cheatsheet"><meta property="twitter:image" content="https://assets.devhints.io/previews/bash.jpg"><meta property="twitter:description" content="Variables · Functions · Interpolation · Brace expansions · Loops · Conditional execution · Command substitution · One-page guide to Bash scripting"><meta property="twitter:image:width" content="900"><meta property="twitter:image:height" content="471"><meta property="twitter:site" content="https://devhints.io/#cli"/><meta property="twitter:card" content="summary"><meta property="twitter:label_1" content="Variables · Functions · Interpolation · Brace expansions · Loops · Conditional execution · Command substitution · One-page guide to Bash scripting"><meta property="twitter:label_2" content="CLI"><meta property="twitter:label_3" content="Bash scripting cheatsheet"><meta property="twitter:label_4" content="Devhints.io"/>
```

Commands:

- cd ser321examples/Network/SimpleGrabURL
 - gradle run

What it does:

The example fetches the content of a specified URL using Java's `URLConnection` mechanism.

2.3 Understanding Gradle

Changes committed to GitHub repository.

2.4 Set up your second system

Link to screencast:

<https://www.loom.com/share/88a2cb7f84ed4d0a8ecf73358a25ba93?sid=fe7fcf05-c937-407e-b356-88f4c9be260f>

Part II.

Networking

3.1 Understanding TCP network sockets

Script:

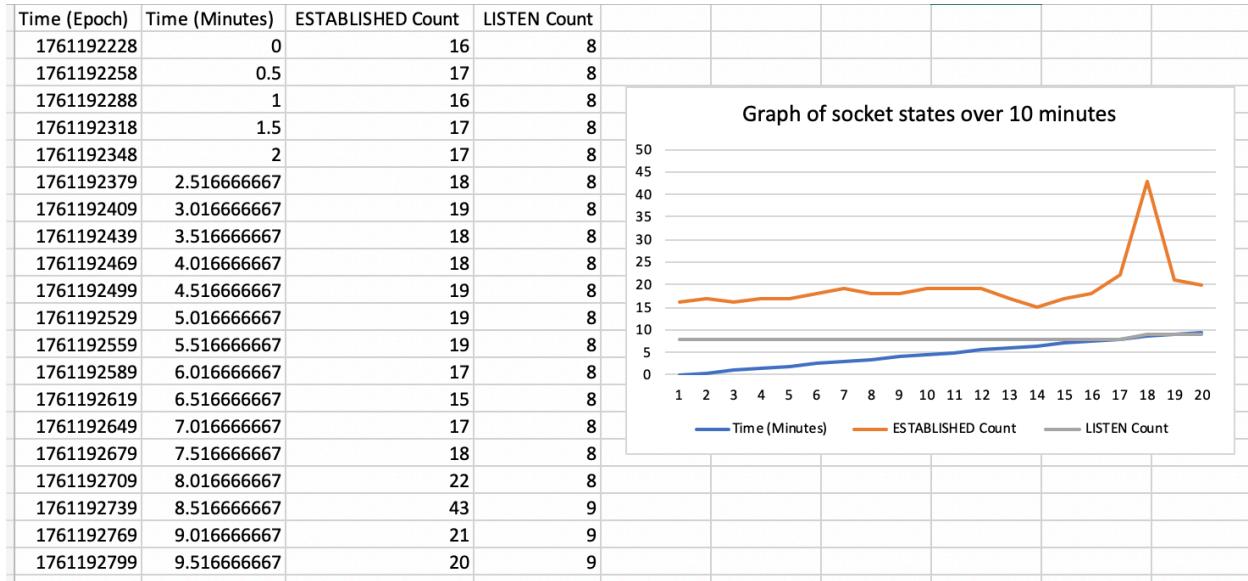
```
#!/bin/bash
DURATION=600 # 10 mins in sec
INTERVAL=30 # 30 sec interval
LOG_FILE="netstat_log_$(date +%Y%m%d_%H%M%S).txt"
END_TIME=$(( $(date +%s) + DURATION ))
echo "Starting netstat monitoring for 10 minutes. Log file: $LOG_FILE"
echo "Time (Epoch) | ESTABLISHED Count | LISTEN Count" > "$LOG_FILE"
while [ $(date +%s) -lt $END_TIME ]; do
    CURRENT_TIME=$(date +%s)

    # run netstat, filter for ESTABLISHED or LISTEN, count.
    COUNTS=$(netstat -n -a -p tcp | grep -E 'ESTABLISHED|LISTEN' | awk '{print $NF}' |
    sort | uniq -c | awk '/ESTABLISHED/{EST=$1} /LISTEN/{LISTEN=$1} END{print EST " "
LISTEN}')
    # extract counts. if not found, then 0.
    ESTABLISHED_COUNT=$(echo $COUNTS | awk '{print $1}')
    LISTEN_COUNT=$(echo $COUNTS | awk '{print $2}')

    # case handling wherein one or both states are zero.
    if [[ -z "$ESTABLISHED_COUNT" ]]; then
        ESTABLISHED_COUNT=0
    fi
    if [[ -z "$LISTEN_COUNT" ]]; then
        LISTEN_COUNT=0
    fi
    echo "$CURRENT_TIME | $ESTABLISHED_COUNT | $LISTEN_COUNT" >>
"$LOG_FILE"
    echo "Snapshot taken at $(date -r $CURRENT_TIME):"
    ESTABLISHED=$ESTABLISHED_COUNT, LISTEN=$LISTEN_COUNT

    sleep $INTERVAL
done
echo "Monitoring complete. Data is in $LOG_FILE"
```

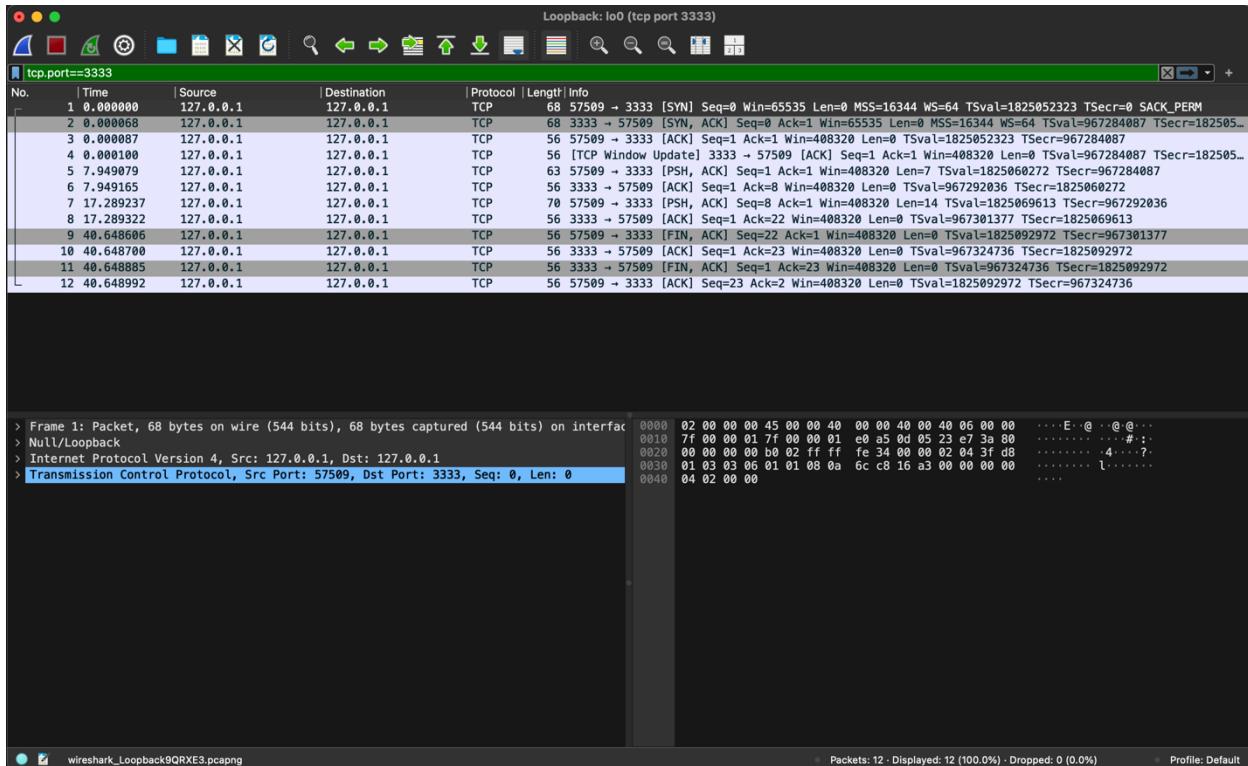
Graph of socket states over 10 minutes:

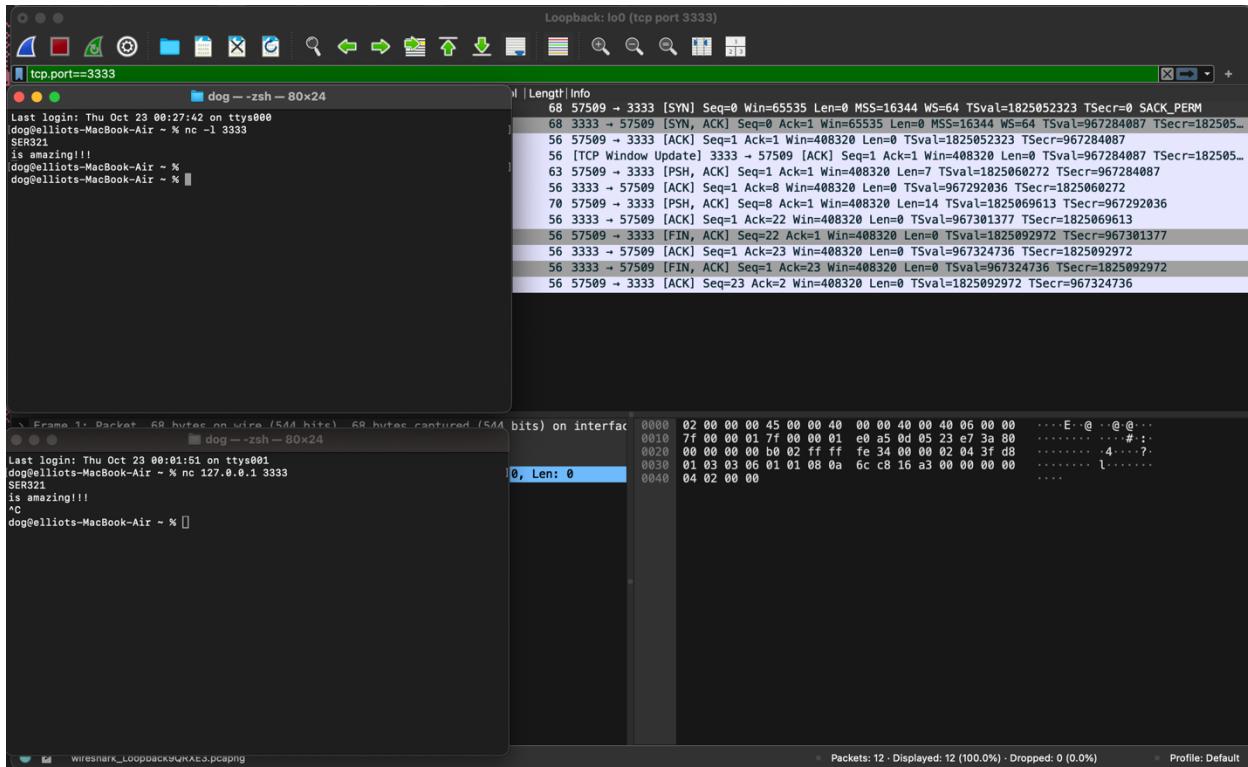


3.2 Sniffing TCP/UDP traffic

Step 1 (TCP)

Screenshot of Wireshark data and Terminal windows:



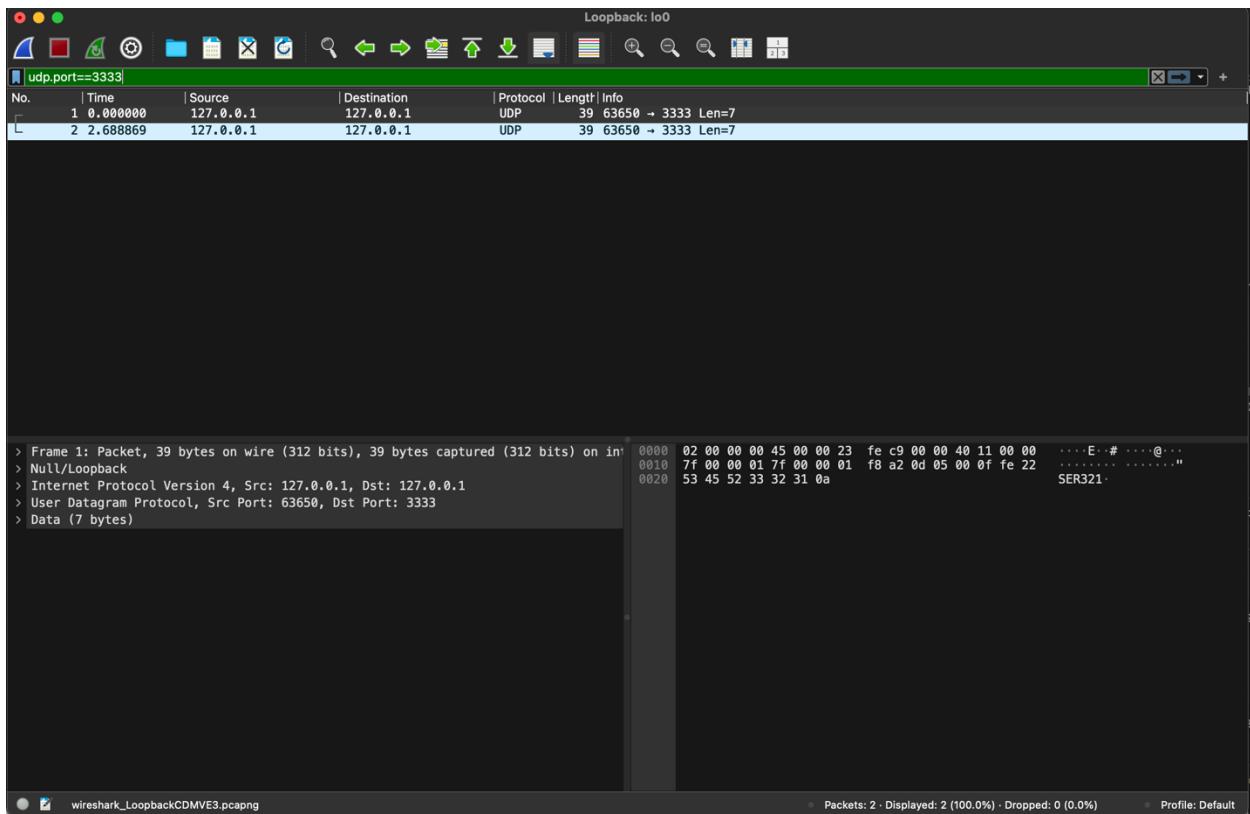


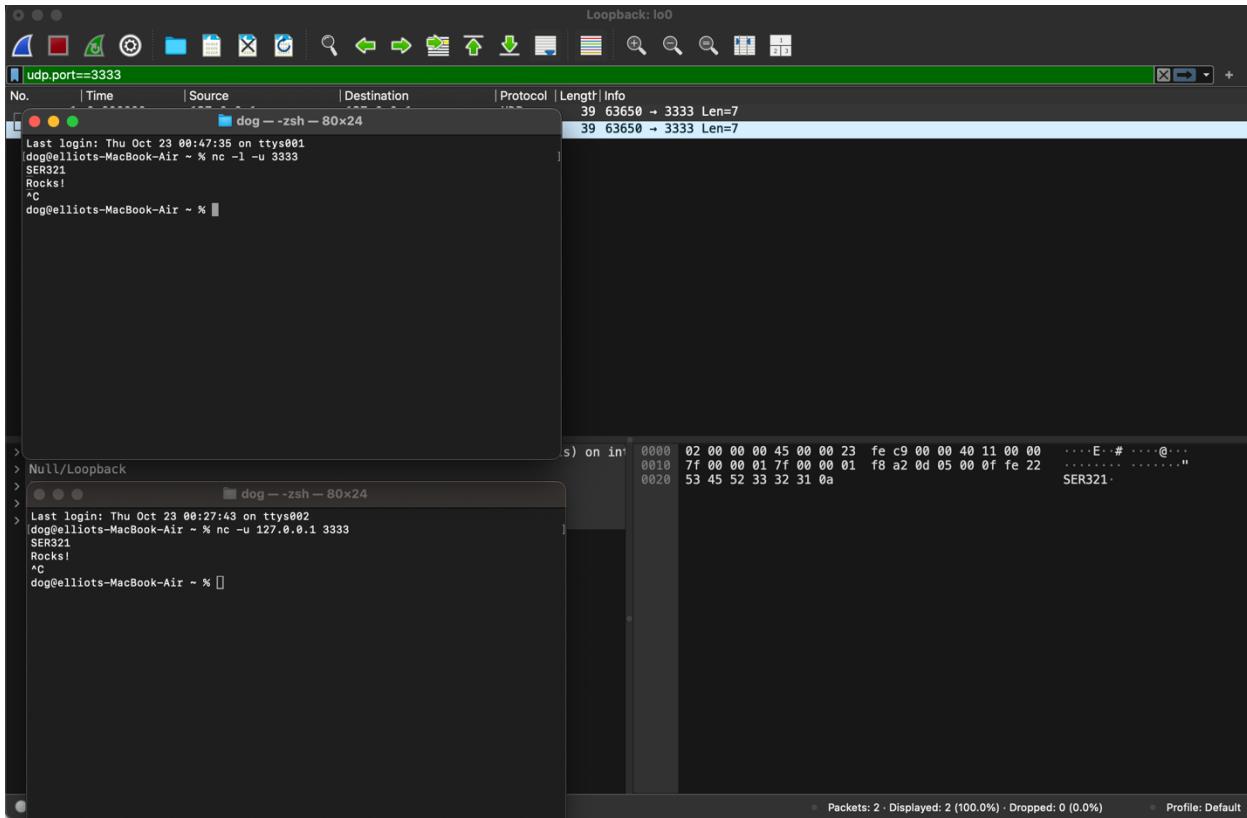
- The two commands used were the Netcat listener (`nc -l 3333`) and the Netcat client (`nc 127.0.0.1 3333`). The Netcat utility (`nc`) is used for reading from and writing to network connections. The listen flag (`-l`) tells Netcat to run in server mode and wait for an incoming connection instead of establishing an outgoing one. 3333 is the port number on which the program will listen to connections. The listener command started a server on my machine bound to TCP port 3333 on the Loopback interface (`lo0`) and waited for a client to connect. Data received was printed to the terminal. 127.0.0.1 is the target IP address (the loopback address), referring to my machine. The client command established an active TCP connection attempt to my machine at 127.0.0.1 on port 3333. Once the connection was established, the client could type data and send it across the network connection to the listener/server.
- The total packets sent back and forth were 4 packets.
 - Client -> Server: packet containing SER321\n.
 - Server -> Client: Acknowledgement packet (ACK) for the first data packet.
 - Client -> Server: packet containing is amazing!!!\n.
 - Server -> Client: Acknowledgement packet (ACK) for the second data packet.
- The total packets was 11 packets (3 Setup, 4 Data, 4 Termination).
 - Connection Setup
 - Client -> Server: SYN
 - Server -> Client: SYN/ACK
 - Client -> Server: ACK
 - Data Transfer
 - 2 data, 2 ACKs (see part b)

- c) Connection Termination
 - i. Client -> Server: FIN/ACK
 - ii. Server -> Client: ACK
 - iii. Server -> Client: FIN/ACK
 - iv. Client -> Server: ACK
- d) 21 bytes.
 - a. Line 1: 6 characters, 1 newline = 7 bytes. Line 2: 13 characters, 1 newline = 14 bytes.
- e) Total bytes over the wire = 381 bytes.
 - a. Header Overhead per packet
 - i. IP header = 20 bytes
 - ii. TCP header = 20 bytes
 - b. Total packets = 11
 - c. Bytes in control packets
 - i. 7 control packets * 40 bytes/header = 280 bytes.
 - d. Bytes in data packets
 - i. 2 data packets * 40 bytes/header = 80 bytes.
 - ii. Data payload = 21 bytes (part d).
 - iii. Total data traffic = $80 + 21 = 101$ bytes.
- f) The overhead is 360 bytes, ~17 times greater than application data itself.
 - a. Overhead = $381 \text{ bytes} - 21 \text{ bytes} = 360 \text{ bytes}$.

Step 2 (UDP)

Screenshots of Wireshark data and Terminal windows:





- a) The command `nc -l -u 3333` utilizes `-l` (listen) and `-u` (UDP) flags to bind to a port and wait for incoming traffic. This command started a server on port 3333 using UDP protocol and waited passively to receive data. The command `nc -u 127.0.0.1 3333` utilizes the `-u` flag (UDP) to send data to a specified host and port. This command set up a client to send UDP data to the local loopback address 127.0.0.1 on port 3333. It does not require a handshake and immediately starts sending data upon input.
- b) 2 packets. Acknowledgements are not sent in UDP.
- c) 2 packets. UDP does not have formal connection setup like TCP 3-way handshake or termination like TCP 4-way teardown.
- d) 13 bytes.
 - a. Line 1: 6 characters, 1 newline.
 - b. Line 2: 5 characters, 1 newline.
- e) 69 bytes.
 - a. Header overhead
 - i. IP header = 20 bytes.
 - ii. UDP header = 8 bytes.
 - iii. Total header per packet: 28 bytes.
 - b. Packet 1 = $28 + 7 = 35$ bytes.
 - c. Packet 2 = $28 + 6 = 34$ bytes.
- f) 56 bytes.
 - a. Overhead = $69 \text{ bytes} - 13 \text{ bytes} = 56 \text{ bytes}$.

g) The difference in overhead is dramatic, demonstrating that there is a tradeoff when it comes to speed and simplicity (UDP) versus reliability (TCP). UDP has much lower relative overhead due to it being connectionless, eliminating the need for most control packets and header fields that TCP requires. This makes it less reliable. More information is exchanged by TCP, such as synchronize (SYN) and FIN (finish) flags, which indicate the establishment and termination of connections. Sequence numbers (SEQ) and Acknowledgements (ACK) are exchanged in TCP, guaranteeing reliable delivery of data. UDP sends data without confirming it has been received.

a. Relative parts of the packet traces:

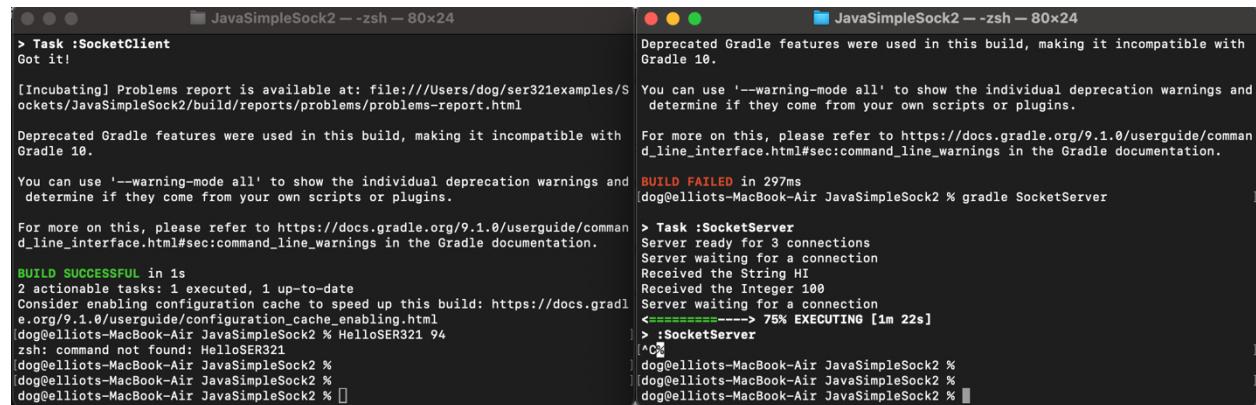
- i. TCP Trace Screenshot: Shows the initial SYN packet (Connection Setup), multiple ACK packets (Reliability), and FIN/ACK packets (Termination). The TCP header is 20 bytes minimum.
- ii. UDP Trace Screenshot: Only shows the two data packets. There are no setup, termination, or acknowledgment packets. The UDP header is only 8 bytes.

3.3.1 Running things locally

Screencast Link:

<https://www.loom.com/share/d254315cdee14e1baff13e496cb3225e?sid=5a5d1a15-30e4-4cb0-8649-2b3ddd9d1052>

Screenshot of Terminal windows:



The screenshot displays two terminal windows side-by-side, both titled "JavaSimpleSock2 — zsh — 80x24".

Left Terminal Window (Client):

```
> Task :SocketClient
Got it!

[Incubating] Problems report is available at: file:///Users/dog/ser321examples/S
ockets/JavaSimpleSock2/build/reports/problems/problems-report.html

Deprecated Gradle features were used in this build, making it incompatible with
Gradle 10.

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

For more on this, please refer to https://docs.gradle.org/9.1.0/userguide/comman
d_line_interface.html#sec:command_line_warnings in the Gradle documentation.

BUILD SUCCESSFUL in 1s
2 actionable tasks: 1 executed, 1 up-to-date
Consider enabling configuration cache to speed up this build: https://docs.gradl
e.org/9.1.0/userguide/configuration_cache_enabling.html
|dog@elliots-MacBook-Air JavaSimpleSock2 % HelloSER321 94
|zsh: command not found: HelloSER321
|dog@elliots-MacBook-Air JavaSimpleSock2 %
|dog@elliots-MacBook-Air JavaSimpleSock2 %
|dog@elliots-MacBook-Air JavaSimpleSock2 %
```

Right Terminal Window (Server):

```
Deprecated Gradle features were used in this build, making it incompatible with
Gradle 10.

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

For more on this, please refer to https://docs.gradle.org/9.1.0/userguide/comman
d_line_interface.html#sec:command_line_warnings in the Gradle documentation.

BUILD FAILED in 297ms
dog@elliots-MacBook-Air JavaSimpleSock2 % gradle SocketServer

> 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 [1m 22s]
> <SocketServer
|dog@elliots-MacBook-Air JavaSimpleSock2 %
|dog@elliots-MacBook-Air JavaSimpleSock2 %
|dog@elliots-MacBook-Air JavaSimpleSock2 %
|dog@elliots-MacBook-Air JavaSimpleSock2 %
```

3.3.2 Server on AWS

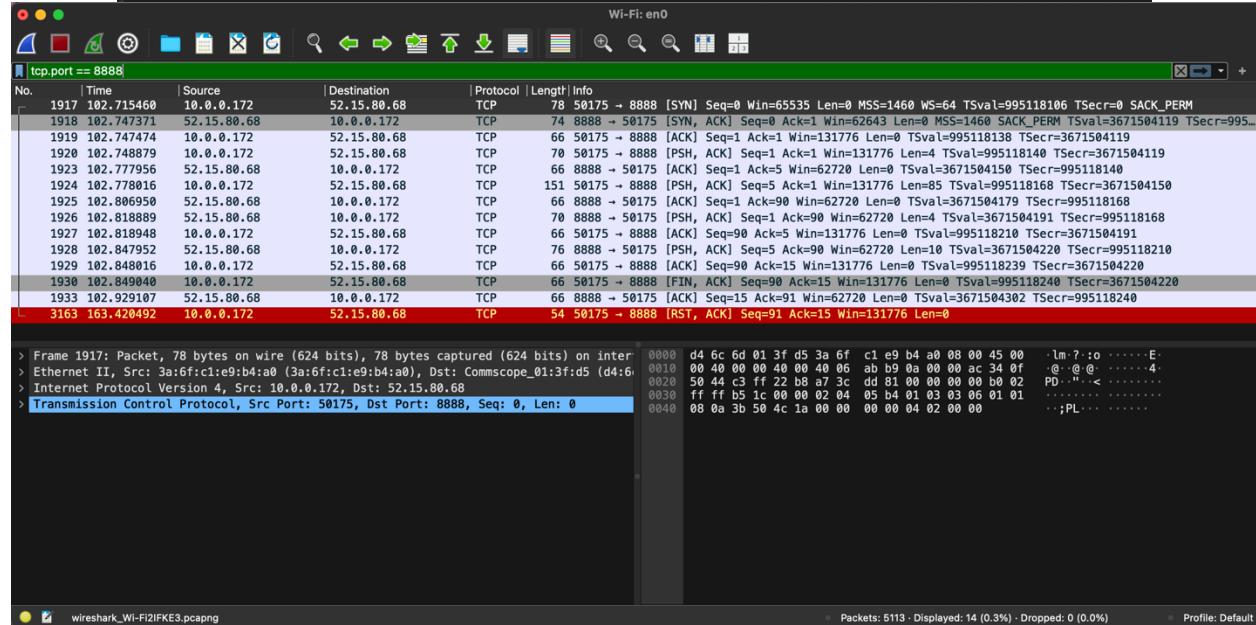
Screenshots of Terminal windows and Wireshark activity:

The screenshot shows two terminal windows. The top window is titled 'Documents' and shows the JavaSimpleSock2 server logs. It includes a welcome message from Amazon Linux 2023, a task switch to 'SocketServer', and a connection from a client. The bottom window is titled 'JavaSimpleSock2' and shows the JavaSimpleSock2 client logs, including a connection attempt from the server and a successful connection message.

```
Last login: Fri Oct 24 04:32:53 2025 from 68.44.121.83
[ec2-user@ip-172-31-26-205 ~]$ cd ser32examples/Sockets/JavaSimpleSock2
[ec2-user@ip-172-31-26-205 JavaSimpleSock2]$ gradle SocketServer
> Task :SocketServer
Server ready for 3 connections
Server waiting for a connection
Received the String HELLO
Received the Integer 100
Server waiting for a connection
<=====--> 75% EXECUTING [2m 58s]
> :SocketServer
[ec2-user@ip-172-31-26-205 ~]$ 

Last login: Fri Oct 24 01:09:54 on ttys001
[dog@elliots-MacBook-Air ~ % cd ser32examples/Sockets/JavaSimpleSock2
[dog@elliots-MacBook-Air JavaSimpleSock2 % gradle SocketClient --args="52.15.80.68 HELLO 100"
> Task :SocketClient
Got it!
[Incubating] Problems report is available at: file:///Users/dog/ser32examples/Sockets/JavaSimpleSock2/build/reports/problems/problems-report.html
Deprecated Gradle features were used in this build, making it incompatible with Gradle 18.
You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.
For more on this, please refer to https://docs.gradle.org/9.1.0/userguide/command_line_interface.html#sec:command_line_warnings in the Gradle documentation.

BUILD SUCCESSFUL in 449ms
2 actionable tasks: 1 executed, 1 up-to-date
Consider enabling configuration cache to speed up this build: https://docs.gradle.org/9.1.0/userguide/configuration_cache_enabling.html
[dog@elliots-MacBook-Air JavaSimpleSock2 %]
```



The client's target changed from the loopback address (127.0.0.1) to the AWS public IP address (52.15.80.68). The command to connect on the Client side reflects this. The Wireshark filter used had to be strict (tcp.port == 8888) to exclude unwanted network

traffic results and isolate the required packets. External network interface was monitored instead of Loopback.

3.3.3. Client on AWS

No, this will not work without issues. It will likely fail to connect immediately without dramatic configuration adjustments. In the previous scenario, only one external firewall had to be considered (AWS). In this case, two separate network components would need to be adjusted, versus a simply Security Group setting. A home public IP address is often dynamic/changes, making it less reliable than a static AWS public IP. Explicit allowance of incoming traffic on port 8888 would need to be configured in system settings.

3.3.4. Client on AWS 2

The difference in connection difficulty between a client on a local network reaching a server on AWS versus an external client (like one on AWS) reaching a server on a local network is due to Network Address Translation (NAT), which is managed by the router. Local devices use private IP addresses that are non-routable on the internet, while the router uses a single unique public IP address for all external communication. When the local client connects to the AWS server, the router replaces the private IP with the public IP, creating a temporary mapping in its NAT table. The router uses this record to successfully route the server's response back to the correct local client.

When an external client attempts to connect to a local server, the request arrives at the router's public IP address as an unsolicited inbound connection. The router fails the connection for security purposes because it cannot identify it. The router blocks these incoming requests. One must explicitly configure a permanent rule to direct incoming traffic on a specific public port to the correct private IP and port of the local server, in order for the connection to be successful.