Matt Stoffel
923127111

Network Programming with UDP and TCP

In Go

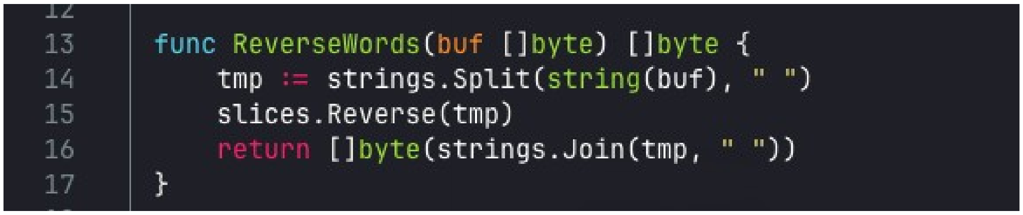
Source Code

Objective:

The goal of this project is to gain a practical understanding of network programming. The focus is on the usage of UDP and TCP protocols for client-server communication. It is a basic implementation that is supposed to focus on how to initialize a connection and gain practical experience with socket programming.

Project Structure:

Because both implementations require the same business logic it is best to write the reversing function once:



```go
func ReverseWords(buf []byte) []byte {
    tmp := strings.Split(string(buf), " ")
    slices.Reverse(tmp)
    return []byte(strings.Join(tmp, " "))
}
```

Other than that and gathering input from the clients I will structure my code in a way that the TCP and UDP servers and clients are in their own directory. I will set up a simple little CLI so you can select which protocol and server/client part of the program you want to run in the current session.

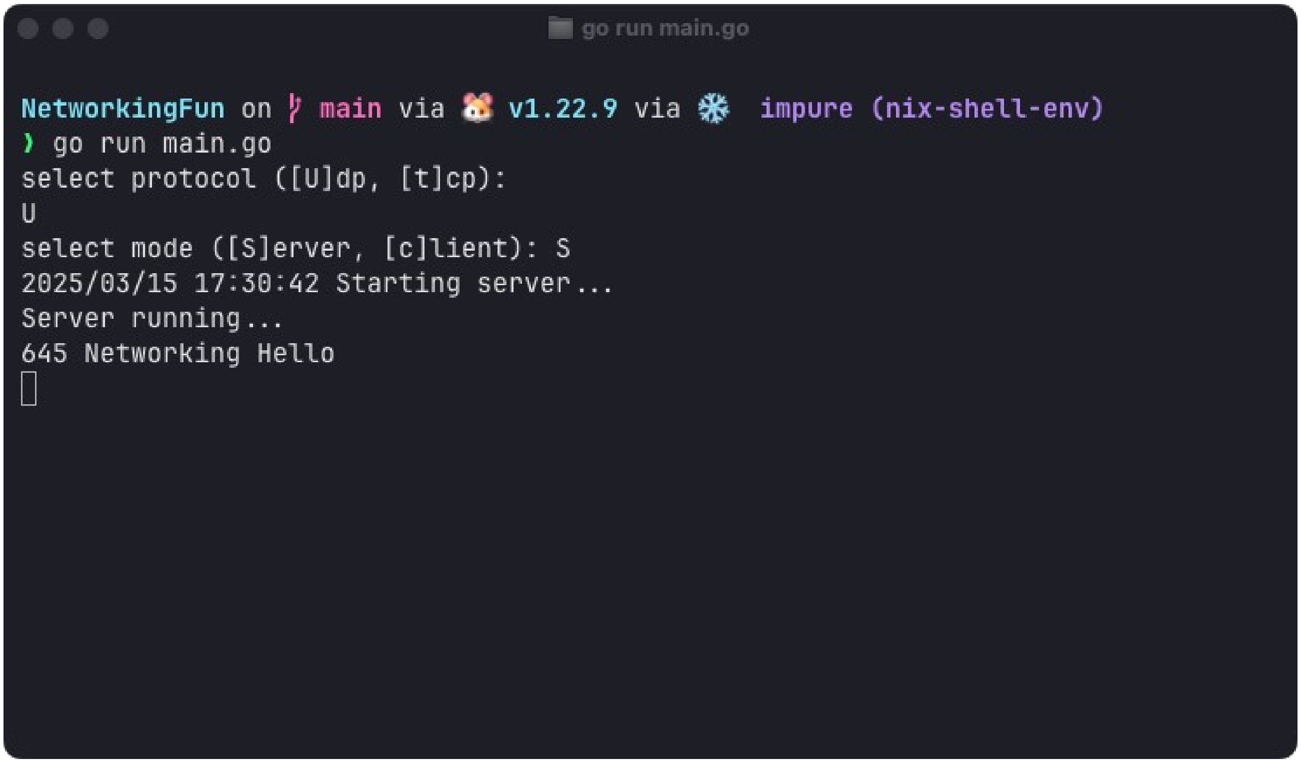
Observations:

I found that there were certain things about UDP that are more simple to implement because there is less overhead. Because UDP doesn’t rely on a connection the same way as TCP you have less to verify upon a request from the client. Also, because of the lesser overhead UDP should be faster. There is nothing in this project that can realistically demonstrate that but simply because of the lack of reliability, i.e. not waiting for ACKs it will definitely be faster. This is a tradeoff because there is no guarantee that the packets you get will be in order when using UDP.

Matt Stoffel
923127111

Testing & Screenshots:

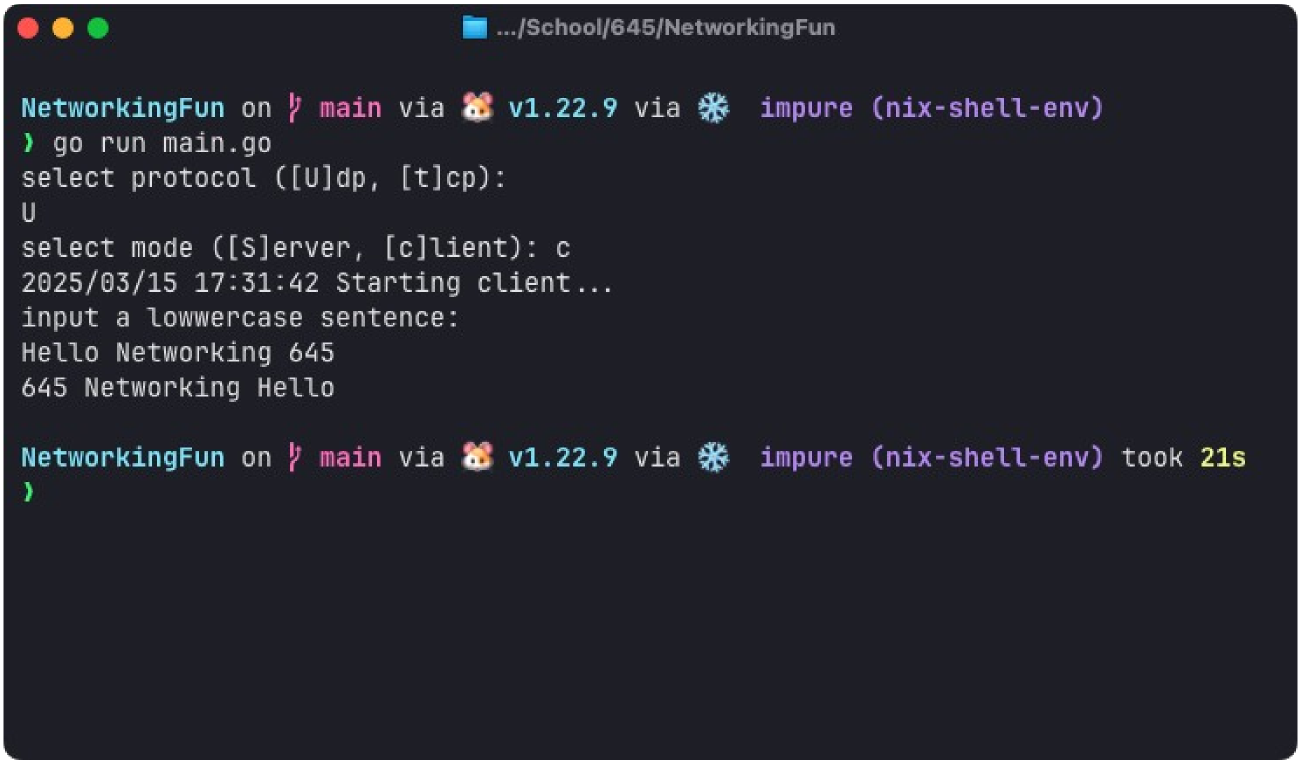
UDP – Server



```
NetworkingFun on  main via 🐹 v1.22.9 via ❄️  impure (nix-shell-env)
❯ go run main.go
select protocol ([U]dp, [t]cp):
U
select mode ([S]erver, [c]lient): S
2025/03/15 17:30:42 Starting server...
Server running...
645 Networking Hello
```

Matt Stoffel
923127111

UDP – Client

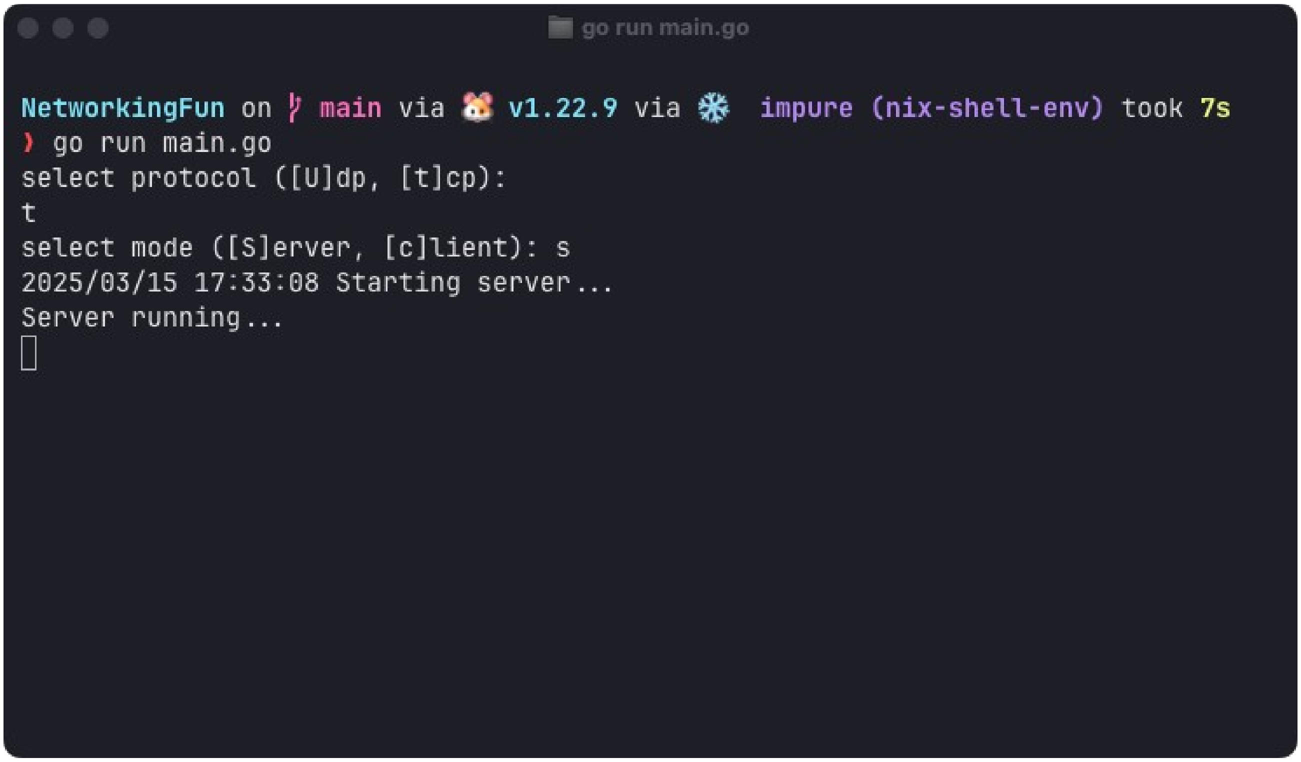


```
NetworkingFun on  main via 🐹 v1.22.9 via ❄️  impure (nix-shell-env)
❯ go run main.go
select protocol ([U]dp, [t]cp):
U
select mode ([S]erver, [c]lient): c
2025/03/15 17:31:42 Starting client...
input a lowwercase sentence:
Hello Networking 645
645 Networking Hello

NetworkingFun on  main via 🐹 v1.22.9 via ❄️  impure (nix-shell-env) took 21s
❯
```

Matt Stoffel
923127111

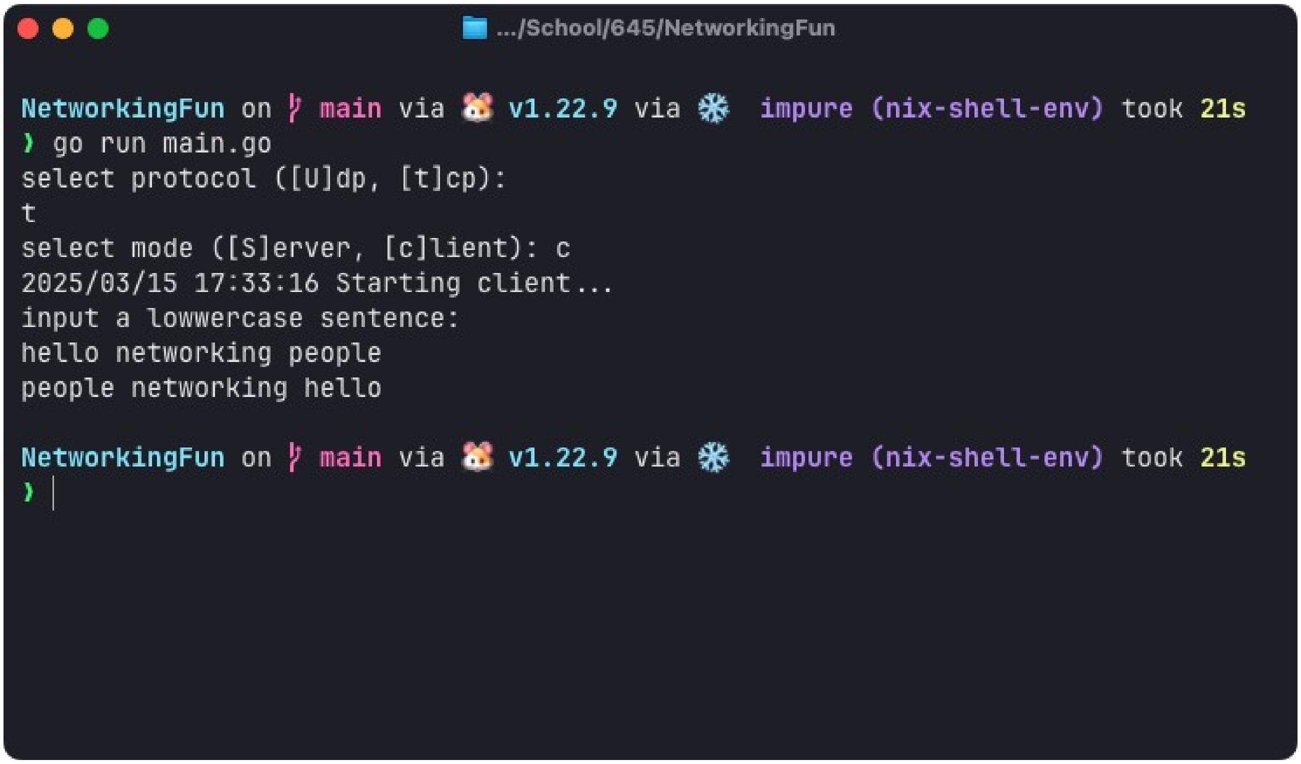
TCP – Server



```
NetworkingFun on  main via 🐹 v1.22.9 via ❄️  impure (nix-shell-env) took 7s
❯ go run main.go
select protocol ([U]dp, [t]cp):
t
select mode ([S]erver, [c]lient): s
2025/03/15 17:33:08 Starting server...
Server running...
```

Matt Stoffel
923127111

TCP – Client



```
NetworkingFun on ᚠ main via 🐹 v1.22.9 via ❄️  impure (nix-shell-env) took 21s
❯ go run main.go
select protocol ([U]dp, [t]cp):
t
select mode ([S]erver, [c]lient): c
2025/03/15 17:33:16 Starting client...
input a lowwercase sentence:
hello networking people
people networking hello

NetworkingFun on ᚠ main via 🐹 v1.22.9 via ❄️  impure (nix-shell-env) took 21s
❯ |
```