

Ikvict Communication Protocol

I created a program that uses my own protocol over udp.

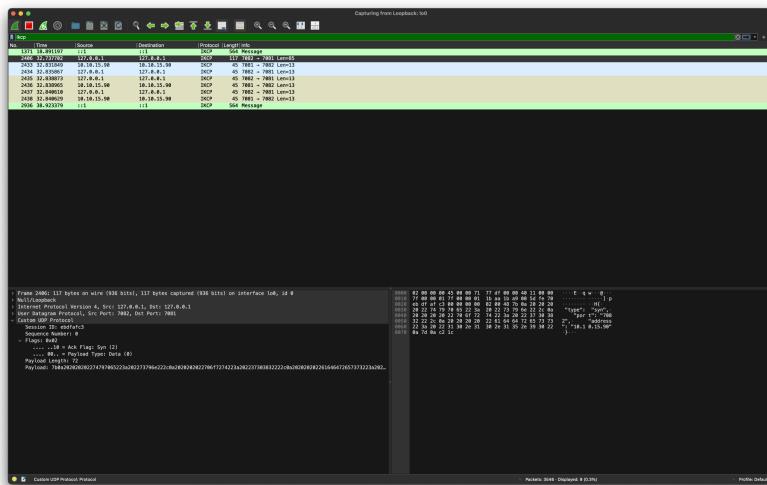
[Github](#)

The program supports two modes: CLI and GUI [see How to run](#)

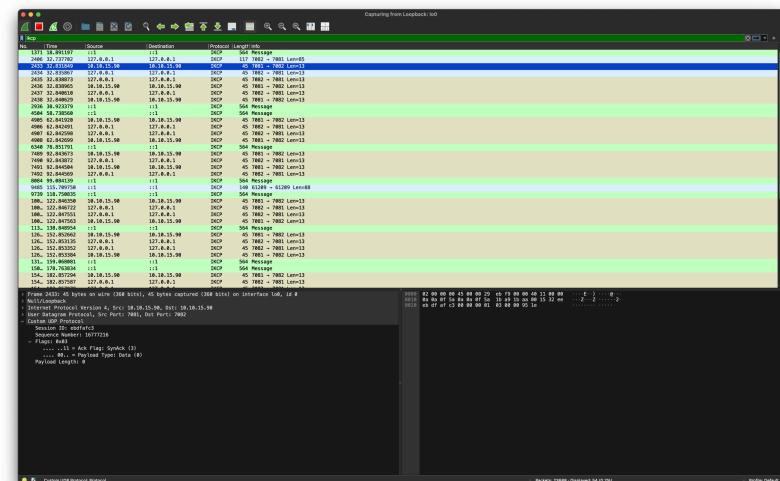
Table of content

- [Connection creating](#)
 - [Message sending](#)
 - [Keep-Alive](#)
 - [Corrupted message](#)
 - [Closing connection](#)
 - [How to run](#)
 - [How to use](#)
 - [Protocol](#)
 - [Speed](#)
-

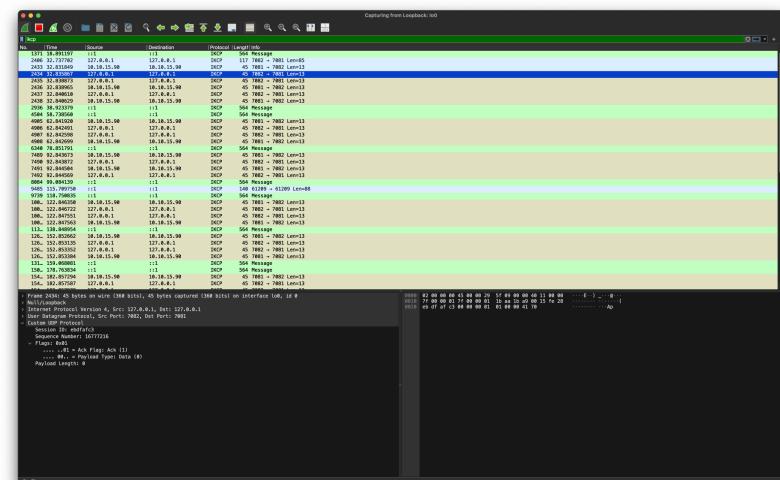
Connection creating



Sending syn



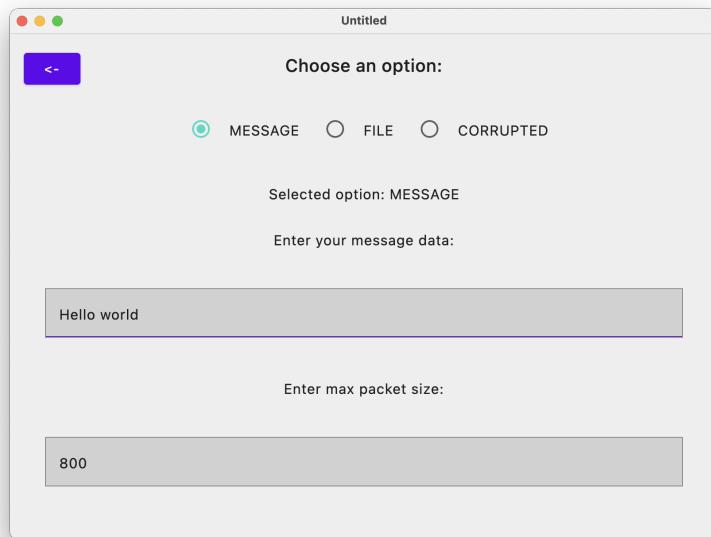
Confirming syn



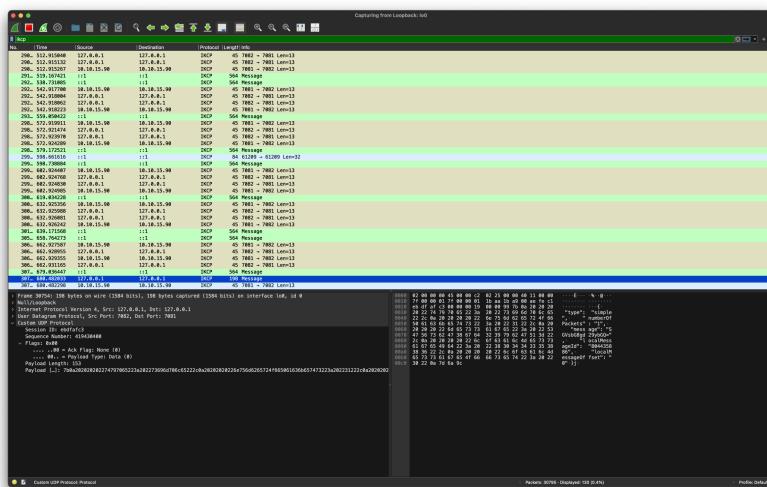
Confirming SynAckSyn

Connection established

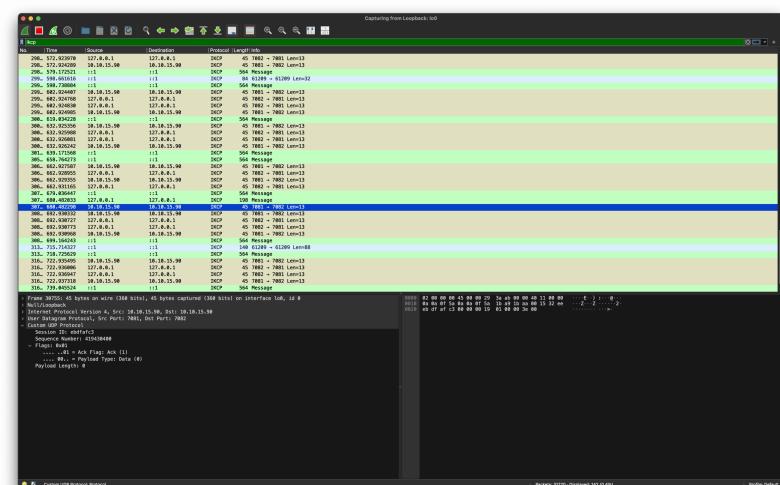
Message sending



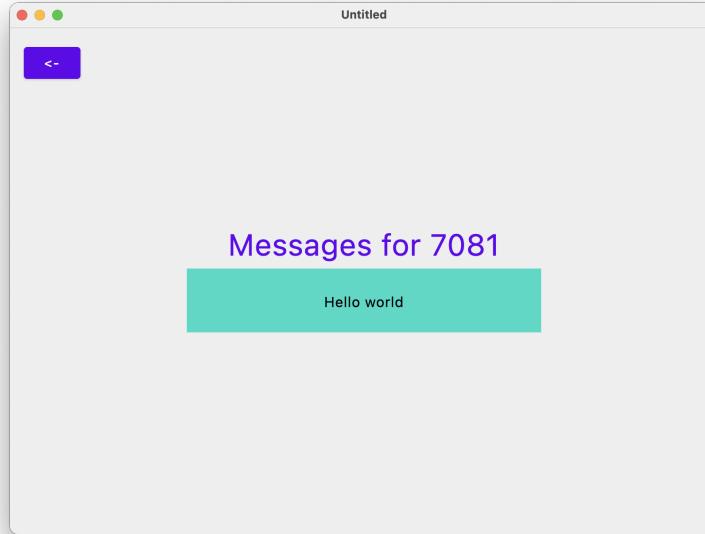
Sending message



Message sent



Message confirmed



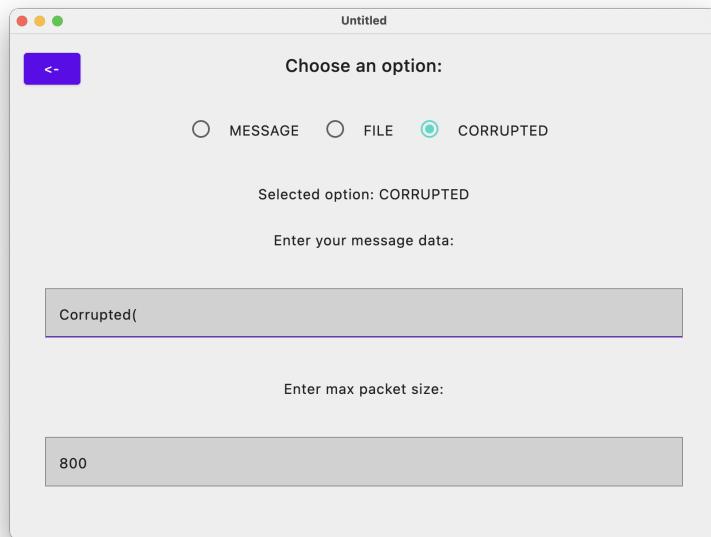
Message received

Keep-alive

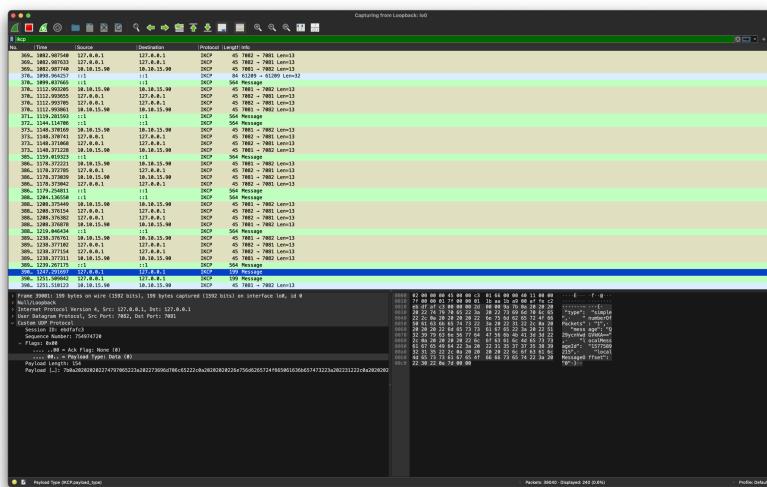
Keep-alive sent

Keep-alive confirmed

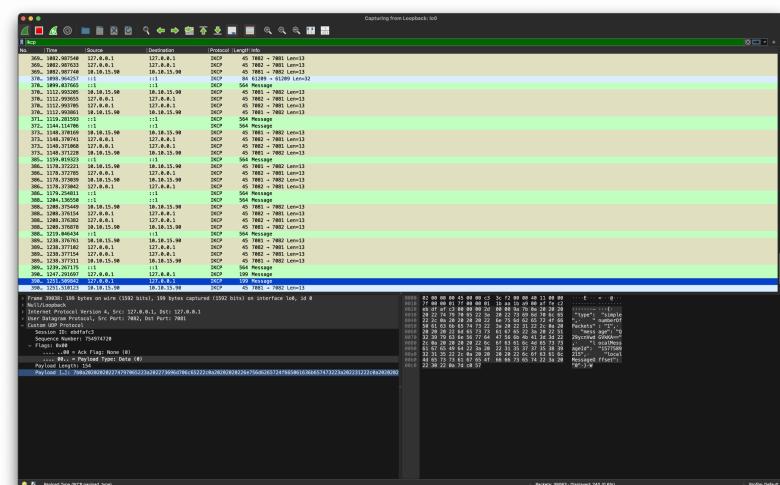
Sending a corrupted message



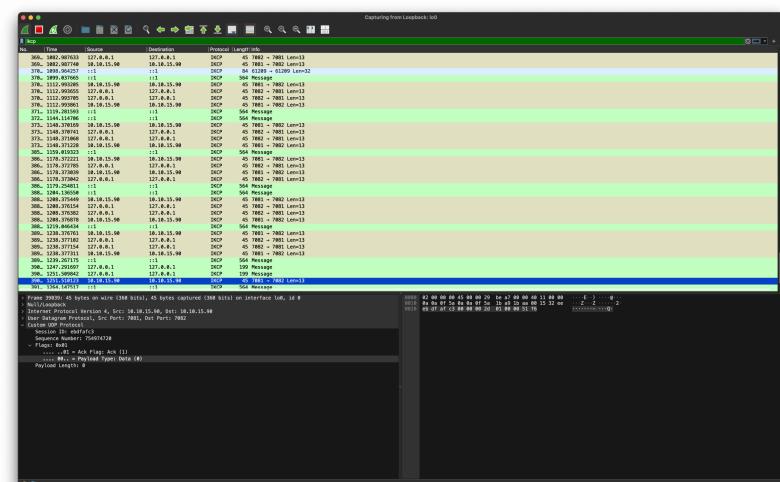
Sending corrupted message



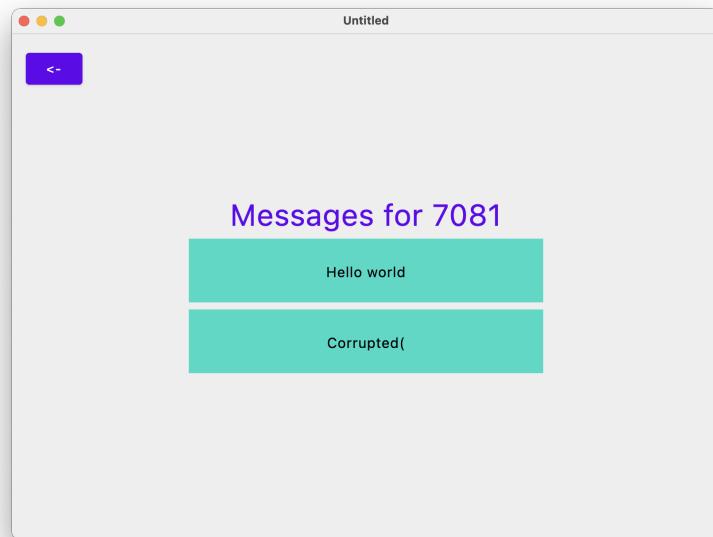
Message sent



Resending message

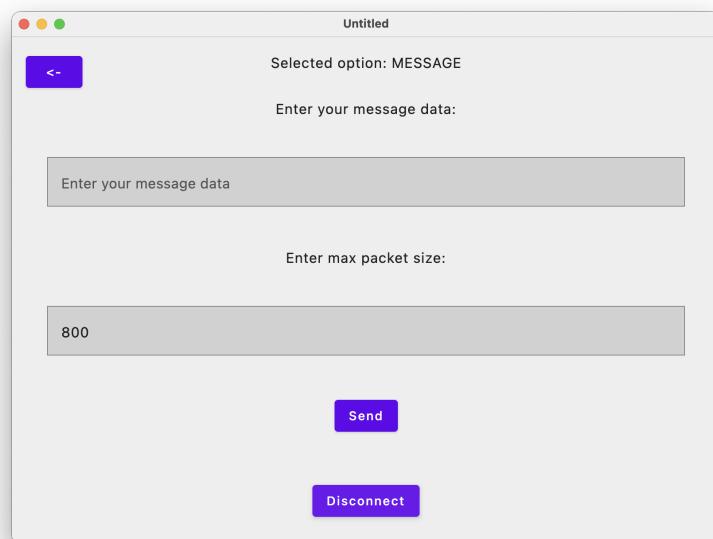


Message confirmed



Message received

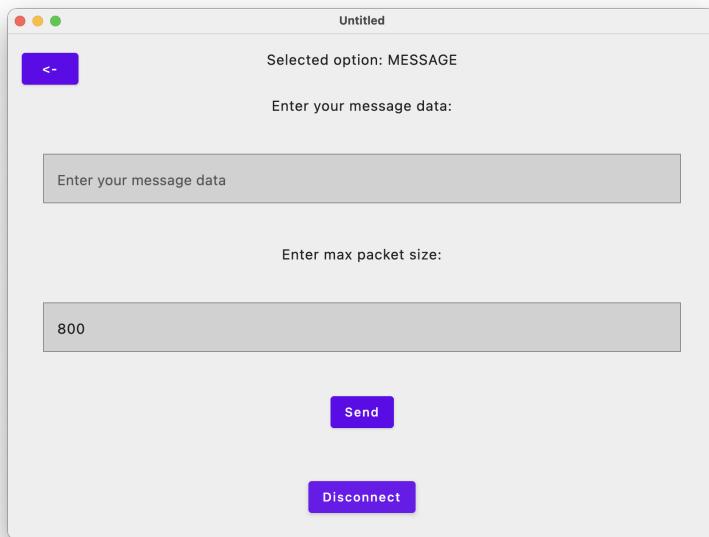
Closing the connection



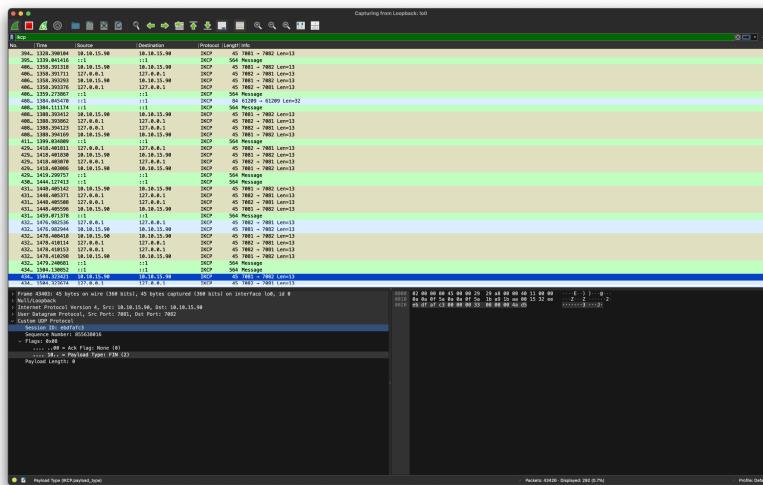
UI closing connection

FIN sent

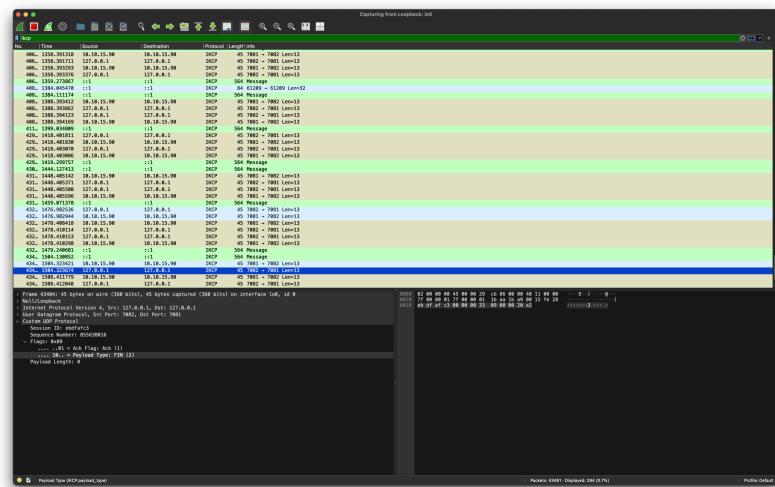
U2 confirms FIN



U2 closing connection



FIN sent



UI confirms FIN

Connection closed

How to run

From console run

```
java -jar pks-1.0.0.jar
```

How to configure

```
java -jar /path/to/jar/pks-1.0.0.jar --key=value --key2=value2
```

Where key can be any of:

```
logging.level.root (off, info, trace) – use trace to see all info about packets
socket.max-payload-size (1-800)
socket.message-resending-confirmation-time-ms
socket.message-resending-frequency-ms
socket.keep-alive-frequency-ms
socket.connection-timeout-ms
socket.attempts-to-reconnect
socket.retry-to-connect-every-ms
general.file-save-location
ui.enabled (true, false)
```

They aren't mandatory as they have default values, However, I recommend changing these:

```
general.file-save-location and ui.enabled
```

How to use

CLI mode

If you set `ui.enabled=false` the program will run in CLI mode

The Console will ask you:

- If you want to open ports for listening
- How many you want to open
- List of ports
- If you want to connect to someone
- How many connections you want
- Remote port to connect to
- Remote Ip to connect to
- Port you will open for listening

Now the program is configured and you can use it

- Choose `ip:port` from the list of available connections
 - or change – to configure packet fragment size
- Select a type of message you want to send (`message`, `file`, `corrupted`)
 - or `close` if you want to close this connection
- Write a message
 - or file path

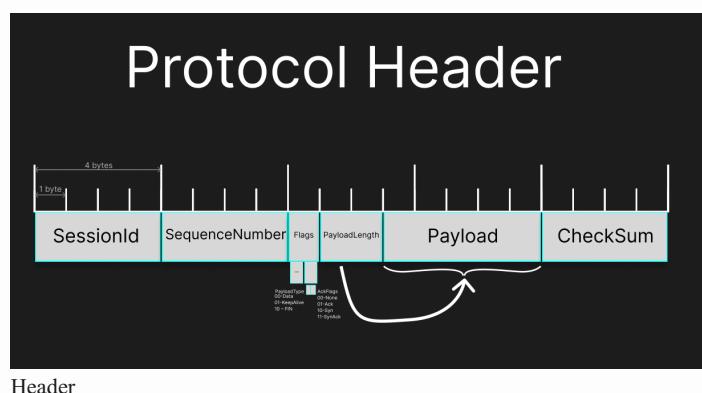
GUI mode

If you set `ui.enabled=true` the program will run in GUI mode

All the configuration now will be done using GUI

Sending and reading messages will be done using GUI as well

Protocol



Keep-Alive

Every N seconds node sends an empty packet with a keep-alive flag

If this node doesn't receive packet Ack keep-alive within x seconds y times

It will automatically disconnect

N, x, y – are configuration params

Receiving confirmation

After Node2 receives a message from Node1, it must send an empty packet with SequenceNumber set to the same value as a received packet and with ack flag

If Node1 doesn't receive this ack packet within a time interval that is configurable, the message is considered as non received and will be resented.

Closing connection

The Connection will be closed when both side receive FIN packet

Speed

It takes ~20s to read, send and store 100 mb file