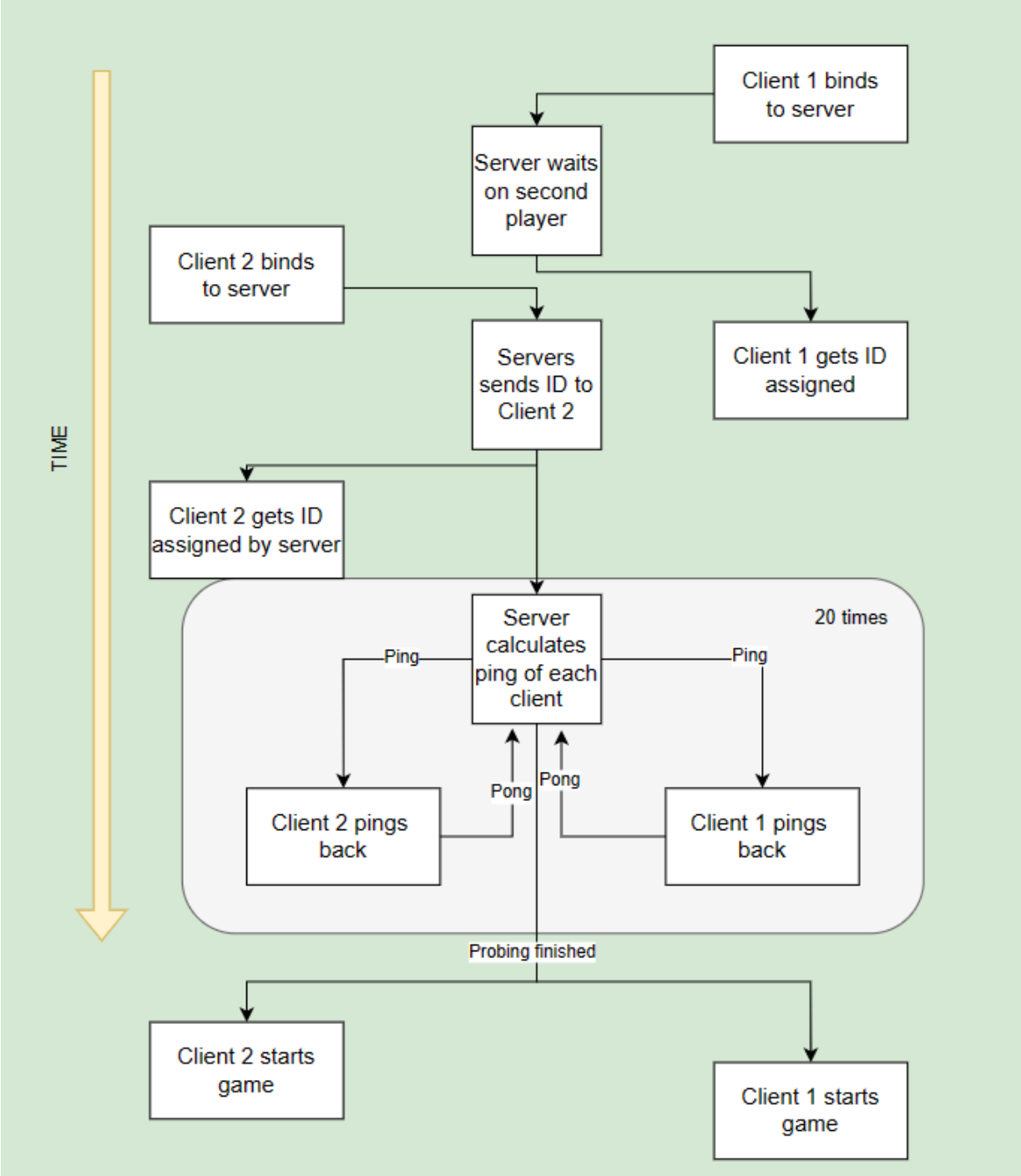
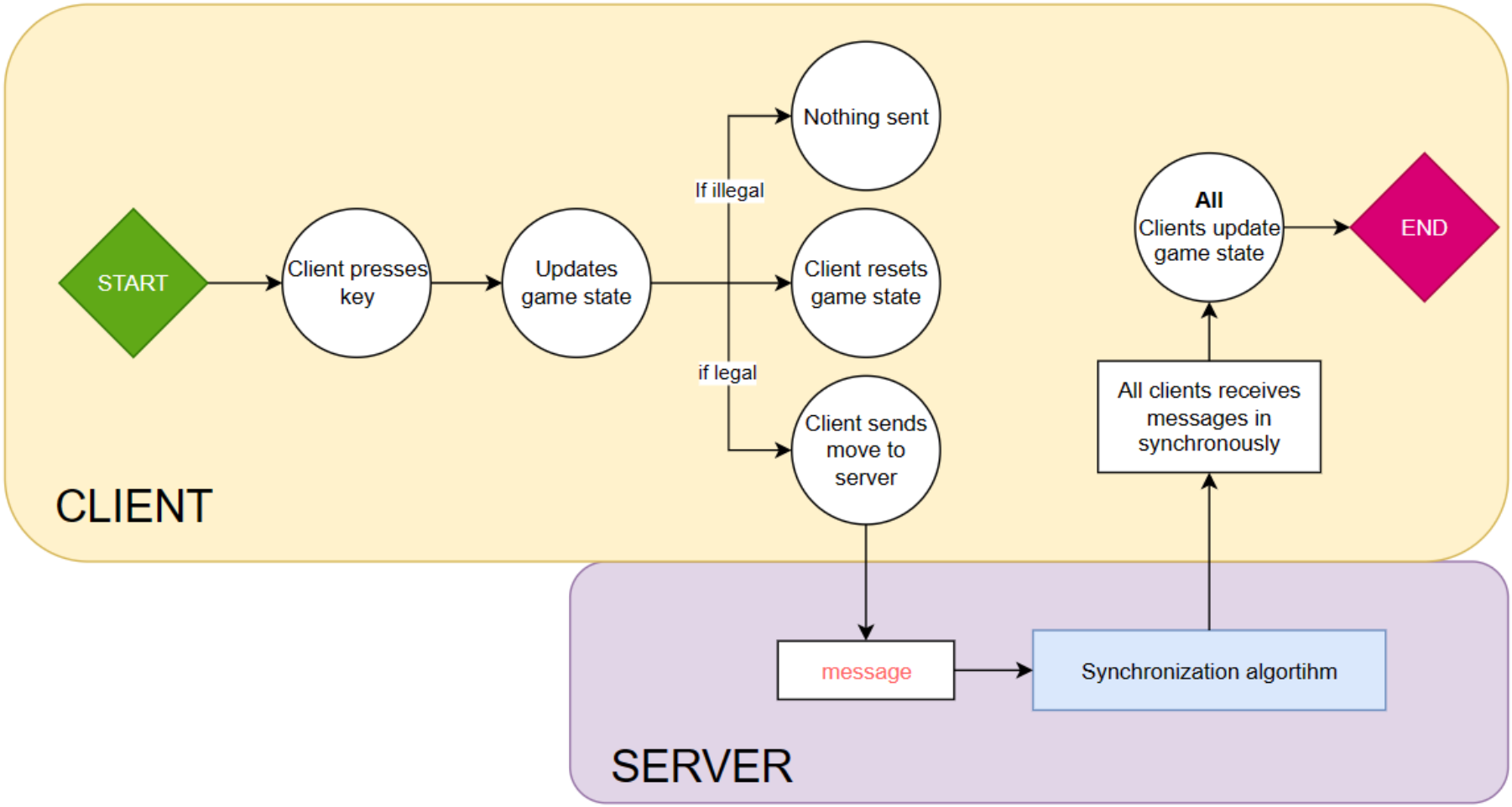
Synchronized Gaming Protocol (SGP)

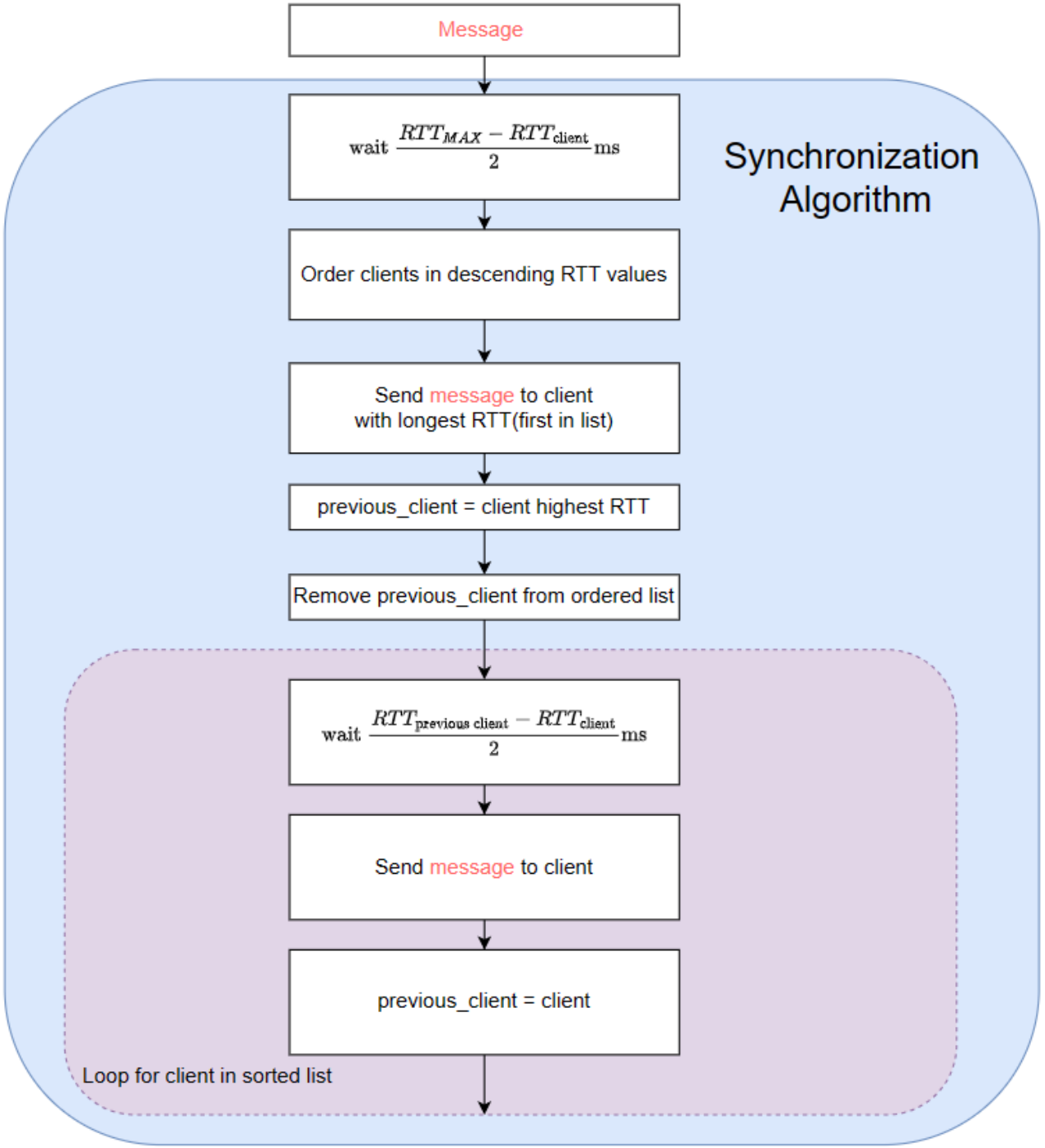
Kucuktopal Burak

Faculty of Sciences
University of Hasselt

December 2023

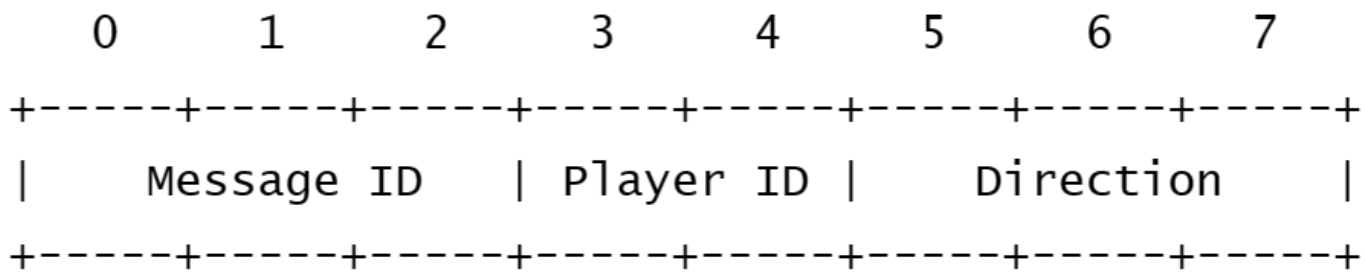

Client 1 binds to server
Server waits on second player
Client 2 binds to server
Servers sends ID to Client 2
Client 1 gets ID assigned
TIME
Client 2 gets ID assigned by server
20 times
Server calculates ping of each client
Ping
Ping
Pong
Pong
Client 2 pings back
Client 1 pings back
Probing finished
Client 2 starts game
Client 1 starts game


START
Client presses key
Updates game state
If illegal
Nothing sent
Client resets game state
if legal
Client sends move to server
CLIENT
message
Synchronization algortihm
All clients receives messages in synchronously
All
Clients update game state
END
SERVER


Message
wait $\frac{RTT_{MAX} - RTT_{\text{client}}}{2}$ ms
Synchronization Algorithm
Order clients in descending RTT values
Send message to client with longest RTT(first in list)
previous_client = client highest RTT
Remove previous_client from ordered list
wait $\frac{RTT_{\text{previous client}} - RTT_{\text{client}}}{2}$ ms
Send message to client
previous_client = client
Loop for client in sorted list

SGP Protocol

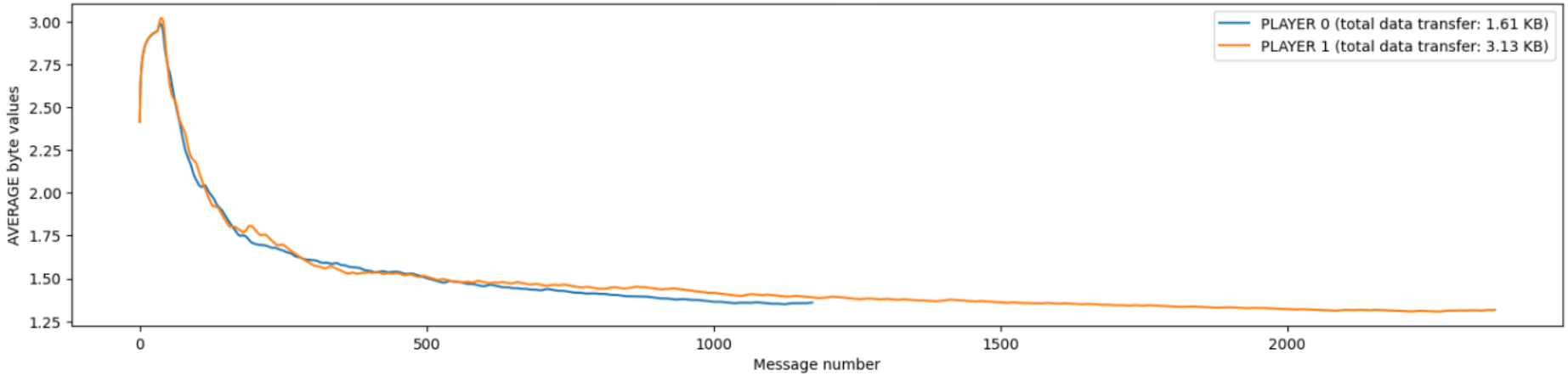
SGP works on TCP and has a header of 1 byte:

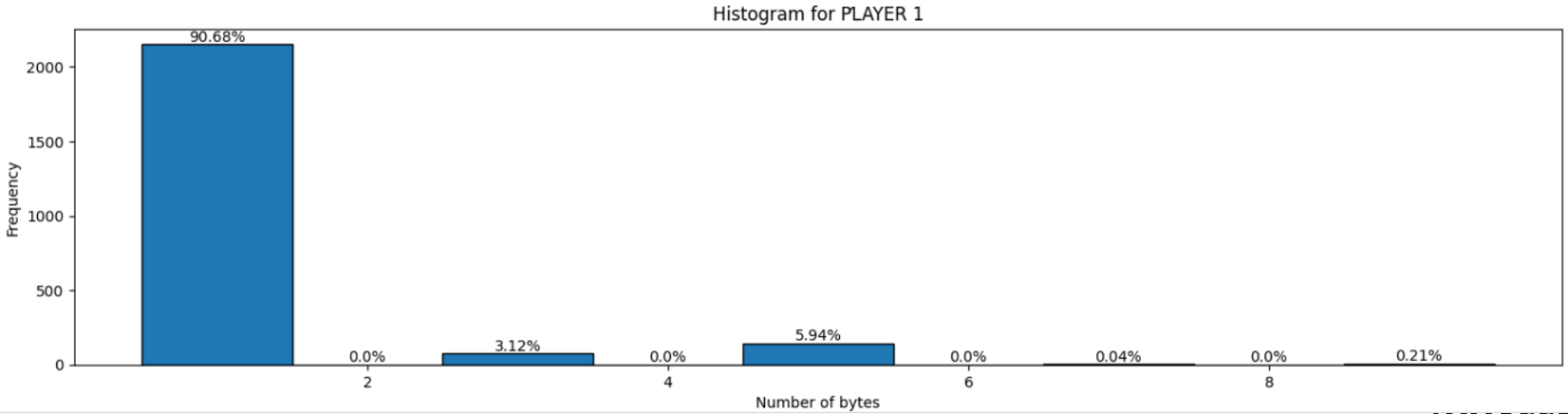


6 type of messages, with all the same header but different identifiers:

- ID 1: Assigning ID to player(1 byte)
- ID 2: Time and first fruit initialization(7 bytes)
- ID 3: Position update(1 byte or 5 bytes)
- ID 4: Leaving message(1 byte)
- ID 5: Ping message(3 bytes)
- ID 6: Fruit and position update(5 bytes or 9 bytes)

Performance





Refinements

Points to think about for further implementations:

- Only total game time synchronized. For the rest; trusted on inner clocks of host devices.
- Protocol punishes players with a better connection. Protocol still usable until 150-200 ms ping.
- Future optimization: using UDP for the 1 byte message. Keeping TCP for initialization and new fruit position.