Phase 1

Samstag, 14. November 2020 19:30

On new con: Phase 1

Client simply connects via a Net-Socket

Client sends: REQ > SERVERS

RES >{1:<IP_1>; 2:<IP_2}

Phase 2

Samstag, 14. November 2020 19:36

Ping-Phase:

Client sends Pings to all 3 Servers (20 Ping / 2 per sec.

↳ (up to)

Client: PING

PONG

Phase 3

Samstag, 14. November 2020 20:37

Transmitting Sample Data:

Clients submit all their Ping-Data to the Main-Server

Clients:

SEND> {0:[<MAIN_SERER DPs>], 1:[<1>], 2:[<2>]}

ACK

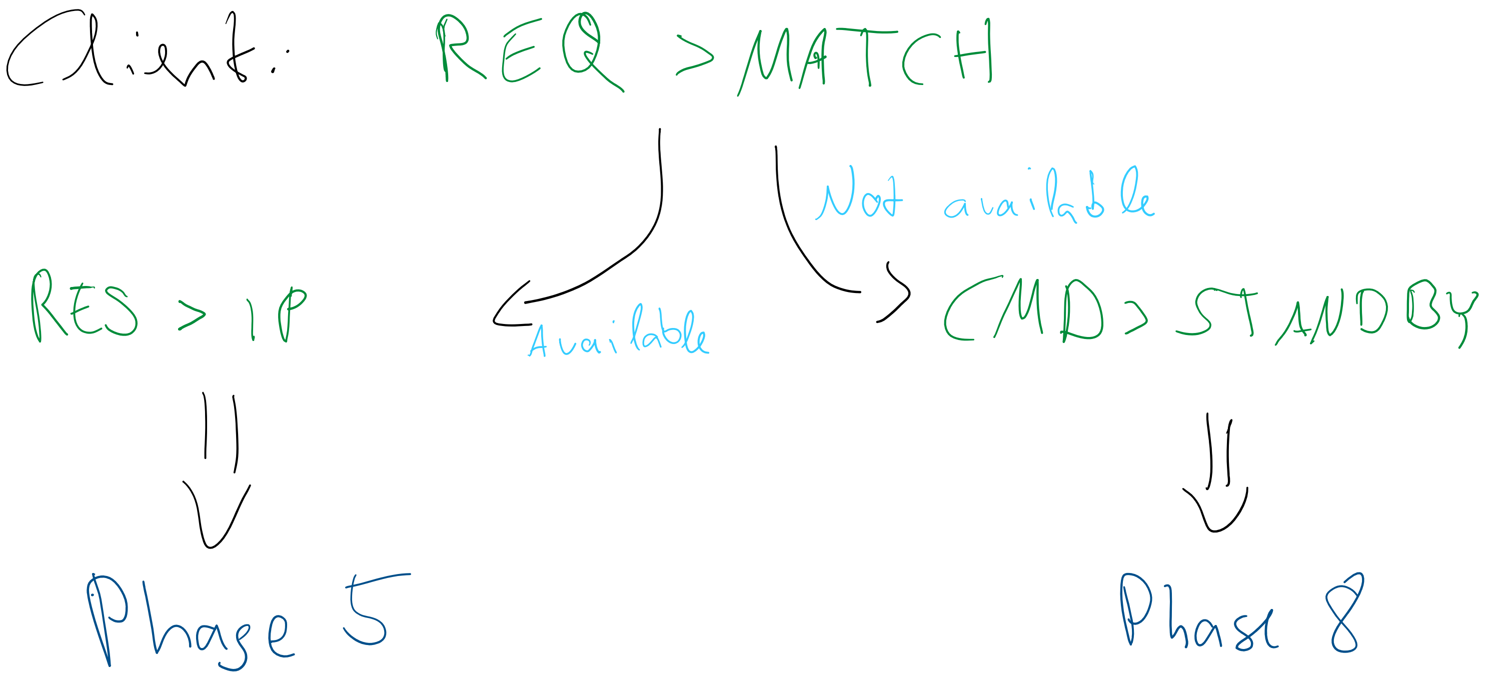
Phase 4

Samstag, 14. November 2020 21:21

Get possible Matches:

Request Matches from the Server

↳ based on the already transferred Results finding new Clients



Phase 5

Samstag, 14. November 2020 21:35

Register measurement:

Main-Server registers Client 1 on Client 2 for measurements

Main-Server: REG><IP_Cl_1>

ACK

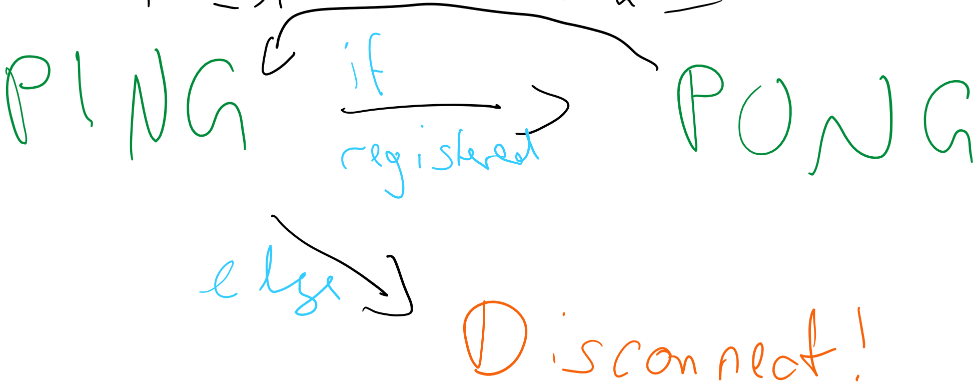
Phase 6

Samstag, 14. November 2020 21:30

Ping Match:

Client pings the match 20 times, 2 times per sec. (up to)

Client_1 → Client 2



Phase 7

Samstag, 14. November 2020 21:39

Transmitting Data 2:

Client 1 transmitts new data-points to Main-Server

Client_1 : SEND>[<DP,>]

ACK

Phase 8

Samstag, 14. November 2020 21:43

Hold & Wait:

Client stands by and requests Match every 5 sec.

Client_1: Wait for 5 sec.

Phase_4

every second:

Client_1: LIFE → ACK

Message Definition

Sonntag, 15. November 2020 16:51

All msgs are composed the same way.

After extracting UDP's payload:

-> mandatory!

CMD > ID [> PAYLOAD]

One of:

{ ACK,
ERR,
REQ,
RES,
PING,
PONG,
SEND,
REG,
LIFE }

For each Client unique: 1 Byte

⇓

Package-loss can be identified

⇓

- initialized with 0
- incremented with every sent package

optional!

↳ not all CMDs need payload

Depending on the command this can differ a lot!

↳ ACK: Acknowledge -> Payload contains the
code of the successful message

ERR: Error -> Payload consists of
a.) The message causing the error (id)
b.) Either one or a list of pre-defined error-codes

REQ: Request -> Payload contains
cmd which the receiver is requested to answer

RES: Response which is not ACK or ERR
↳ first byte of the payload: msg-id

PING: Ping-Packet -> no payload

PONG: Responce to PING -> payload ≙ PING's msg-id

SEND: Sending unrequested data
↳ payload can be custom

REG: Register Client for PING-evaluation
↳ payload is the IP of the other Client!

LIFE: A packet without payload indicating that a waiting Client ist still alive