

Ping Program Report

The Protocol was designed keeping an arbitrary client and server in mind, i.e the server and client programs only serve as proof of concept- the protocol can be implemented regardless of the protocols other components follow.

Protocol Design

Type of message

The messages are encoded as strings. This was because, at a lower level, Python strings are encoded as C strings and so any client/server program written in any language that occupies the same or higher hierarchical position as C will be able to easily read and extract the information in the string.

Message syntax

The message should be a string of at least 55 characters. The upper bound of the number of characters is fixed by the size of packet permitted. The first 54 characters can be interpreted as header, with information packed into it. All characters that

follow constitute the user data. If the user data is empty, it shall be encoded as “_/_”(without quotes)

Message semantics

The message is interpreted as: [0-2]: "UDP", [3-18]: destAddr, [19-22]: port Number, [23-48]: timestamp, [49-54]: platform of the host/server, [55:]: user message ([x-y]: Index x to index y, both included. String is 0-indexed)

Rules

- 1)The message should be a string of atleast 55 characters
- 2)The message string should adhere to the message semantics specified

Ping Client Program

Control flow/Program execution

The program calls the function ping(host), which does the following:

- 1)Calls the doOnePing(host) handler that handles a round trip of the packet, and returns the time taken for the round trip
- 2)Prints a message stating the ID(starting from 1) of the packet and the RTT
- 3)If the RTT exceeds timeout, print the message ‘Timeout’
- 4)Update the maxRTT and/or minRTT if necessary

5) Follow the above sequence for 10 packets, each with a gap of 1 second between them.

`doOnePing(host):`

The function creates a socket, calls `sendOnePing` and then `recieveOnePing` and then returns the RTT

`sendOneping:`

Send a ping message by encoding the appropriate message on the string

`recieveOnePing:`

Receive a packet and parse the message received. Return the RTT

`printStats():`

Print the aggregate statistics once 10 packets are sent.

Ping Server Program

Create a socket and bind the host to the port.

Randomly generate the IDs of the packets to not send a response back to

Create the server message and send it to the address unpacked from the incoming packet