**Networking Protocols**

­­­

Packet Layout:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 Byte | 1 Byte | 1 Byte | 1 Byte | 1 Byte | … |
| 0-1 | 0-255 | 0-255 | 0-1 | 0-255 | … |
| Data Type | Command | Data Size | Fragmentation | Fragment Part | Data |
| **Packet Header** | | | | |  |

|  |  |
| --- | --- |
| Data Type  *(Boolean)* | 0 (False) – Packet is to be treated as a command for the server  1 (True) – Packet is to be treated as raw data |
| Command  *(Integer)* | Up to 255 commands possible  Currently only 19 commands planned  Commands list below |
| Data Size  *(Integer)* | Specifies the size of the data  Allows up to a length of 255 characters for Data  Primarily used to check for data corruption |
| Fragmentation  *(Boolean)* | Boolean value  0 (False) – Not fragmented  1 (True) – Fragmented  Not yet implemented |
| Fragment Part  *(Integer)* | Specifies the Fragment Part number  Up to 255 fragment packets |
| Data  *(String)* | Data to be sent |

Commands Table:

|  |  |  |
| --- | --- | --- |
| **ID** | **CMD TYPE** | **DESCRIPTION** |
| 0 | PING | Ping  Used to test for latency. |
| 1 | REQ\_STAT | Require status command  Ask client/server for status. |
| 2 | REQ\_CON | Request Connection (Only client sends)  Ask to connect to server. Username and Password is sent. |
| 3 | DEC\_CON | Decline Connection (Only server sends)  Request for connection is declined. |
| 4 | ACC\_CON | Accept Connection (Only server sends)  Request for connection is accepted. Authentication Token is sent to client. |
| 5 | ERR\_CON | Error Connecting  Sent when error connecting. |
| 6 | ACK | Acknowledge  “Yes” or “Received” response for a command. |
| 7 | ECHO | Echo Command |
| 8 | KEEP\_ALIVE | Keep Alive  Tells receiver to keep connection alive |
| 9 | TERM\_CON | Terminate Connection  Tells receiver to terminate the current connection. |
| 10 | UPD\_PKT | Update Packet  A request/response update packet containing updated data. |
| 11 | SGN\_UP | Sign Up packet (Only client sends)  Packet containing sign up data. |
| 12 | RSND\_DATA | Resend Data (Only server sends)  Server ask client to resend last transmission. |
| 13 | FRAG\_PART | Fragment Part  Specifies packet as a part of a fragmented data group. |
| 14 | DATA | Data packet  Used to specify that packet is not a command. |
| 15 | RET\_STAT | Return Status  Response to a REQ\_STAT command |
| 16 | *Reserved* |  |
| 17 | *Reserved* |  |
| 18 | RM\_CMD | Room Command (Only client sends)  Data part of packet contains the command to be performed for the specified chat room. |
| 19 | RET\_RSP | Return Response (Only server sends)  Response to RM\_CMD command. |

**Networking operations**

Phase 1 – Connection Handling:

* Client connects to server
* Server accepts connection and creates a new *ClientHandler* thread
  + New *ClientHandler* thread generates an *AuthToken* on initialization
  + New thread is added to thread pool

Phase 2 – Authentication Procedure:

* Client sends a *REQ\_CON* packet to the server
  + *REQ\_CON* packet contains:
    - Username
    - Password Hash (SHA-256)
* Server checks *REQ\_CON* packet
  + Extract Username and Password from Data section of packet
  + Check against database
  + If Username exist and Password is correct:
    - Send *ACC\_CON* with Authentication Token (AuthToken) to client.
  + Else:
    - Send *DEC\_CON* to client and terminate connection.
* Client receives ACC\_CON packet
  + Extracts AuthToken
  + Send ACK packet to server
* Server receives ACK packet and officially recognizes client as authenticated

Phase 3 – Message Handling:

* *ClientHandler* thread listen for incoming packets
* *MessageHandler* function handles all received packets
  + Check if packet is not malformed or corrupted
  + AuthToken is checked for every incoming packets
    - If a mismatch occurred then connection is immediately terminated for security
  + Performs operations specified in *Command.java* based on type and command
* Continue handling packets until made to terminate

Phase 4 – Termination:

* Client sends *TERM\_CON* to server
* Server receives command and terminates connection
  + Server stops handling packets
  + Socket between server and client is closed
  + *ClientHandler* thread terminates
  + Dead thread is removed from thread pool

**Networking Concepts**

|  |  |
| --- | --- |
| Authentication Token  *(AuthToken)* | A string used to provide proof of authentication and is required in all communications with the server. A mismatched AuthToken means that the client is not authenticated/connection is being intercepted.  The token is generated using a security purpose random number generator to produce a *long int* number. The number is then converted to a base 64 number using the *toBase64* function. |
| Message Packet  *(Packet)* | Message Packet or Packet for short is different from ipv4/ipv6 packets. Message Packet is a protocol on top of the ipv4/ipv6 packet. |