**CPE464 – Program 2 Chat Design Work**

**Packet Structures: (Each cell is a byte)**

Flag = 1 – Client Initial Packet to Server:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Packet Length 1 | Packet Length 0 | Flag = 0000 0001 | Handle Length | Handle |

Flag = 2 – Server back to Client, confirming a good handle

|  |  |  |
| --- | --- | --- |
| Packet Length 1 | Packet Length 0 | Flag = 0000 0010 |

Flag = 3 – Server back to Client, error on initial packet (handle already exists)

|  |  |  |
| --- | --- | --- |
| Packet Length 1 | Packet Length 0 | Flag = 0000 0011 |

Flag = 5 – Client to another Client via server; Message packet (to 1 other client shown below)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| PL 1 | PL 0 | Flag = 5 | Sender Length | Sender Handle | Number of Recipients | Destination Size | Destination Handle |

Flag = 7 – Server to client, error packet if a destination handle does not exist

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Packet Length 1 | Packet Length 0 | Flag = 0000 0111 | Handle Length | Handle |

Flag = 8 – Client to Server when client is exiting:

|  |  |  |
| --- | --- | --- |
| Packet Length 1 | Packet Length 0 | Flag = 0000 1000 |

Flag = 9 – Server to client, ACKing the client’s exit:

|  |  |  |
| --- | --- | --- |
| Packet Length 1 | Packet Length 0 | Flag = 0000 1001 |

Flag = 10 – Client to Server, client requesting the list of handles:

|  |  |  |
| --- | --- | --- |
| Packet Length 1 | Packet Length 0 | Flag = 0000 1010 |

Flag = 11 – Server to Client, responding to flag = 10, giving number of handles stored

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PL 1 | PL 0 | Flag = 11 | Number of Handles 3 | Number of Handles 2 | Number of Handles 1 | Number of Handles 0 |

Flag = 12 – Server to client immediately following flag 11, showing handles stored (1 per handle):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Packet Length 1 | Packet Length 0 | Flag = 0000 1100 | Handle Length | Handle |

Flag = 13 – Server to client, telling client that %L has finished:

|  |  |  |
| --- | --- | --- |
| Packet Length 1 | Packet Length 0 | Flag = 0000 1011 |

**Packet Flow Diagrams:**

1. Connection Setup (sending handle to server)

**Client Server**

Flag = 1

Flag = 2

**Description:** Case where client establishes connection with no error

**Processing on Client:** blocks until receiving a packet from server

**Processing on Server:** establish connection, store handle, and send flag = 2

**Client Server**

Flag = 1

Flag = 3

**Description:** Case where client establishes connection an error

**Processing on Client:** blocks until receiving a packet from server

**Processing on Server:** try to establish connection, send flag = 3

1. Sending a Message (%M) to a Single client

**Client Server**

Flag = 5

**Description:** Case where client sends a message to 1 other client without error

**Processing on Client:** goes back to $: prompt

**Processing on Server:** make sure the message gets forwarded to the dest. handle

**Client Server**

Flag = 5

Flag = 7

**Description:** Case where client sends a message to 1 other client that does not exist

**Processing on Client:** Client displays error message upon receiving flag = 7

**Processing on Server:** send flag = 7

1. Sending a Message to multiple clients

**Client Server**

Flag = 5

**Description:** Case where client sends a message to 1 other client without error

**Processing on Client:** goes back to $: prompt

**Processing on Server:** make sure the message gets forwarded to the dest. Handle

**Client Server**

Flag = 5

Flag = 7

Flag = 7

**Description:** Case where client sends a message to 1 other client that does not exist

**Processing on Client:** displays error message for each handle associated with flag = 7

**Processing on Server:** send flag = 7 for each non-existent handle (this case 2)

1. Sending messages if the entered message text is 450 characters

**Client Server**

Flag = 5, 200 Byte Message

Flag = 5, 200 Byte Message

Flag = 5, 50 Byte Message

**Description:** Case where client sends a message that is 450 bytes long without error

**Processing on Client:** Send multiple messages no longer than 200 bytes until size met

**Processing on Server:** treat each message as its own; forward them to dest handle

**Client Server**

Flag = 5, 200 Byte Message

Flag = 7

Flag = 5, 200 Byte Message

Flag = 7

Flag = 5, 50 Byte Message

Flag = 7

**Description:** Client sends a message that is 450 bytes long to a non-existent handle

**Processing on Client:** Output an error message for each error flag from server

**Processing on Server:** treat each message as its own; respond with flag=7

1. Listing the handles (%L, including flags 10-13)

**Client Server**

Flag = 10

Flag = 11

Flag = 12

Flag = 13

**Description:** Client requesting a %L, where server has 1 handle stored (no errors)

**Processing on Client:** Display the number of handles and then each handle

**Processing on Server:** Send Flag = 11 to respond, then send Flag = 12 for the one client. Once finished, send Flag = 13.

1. Ending a connection (%E)

**Client Server**

Flag = 8

Flag = 9

**Description:** Client exiting

**Processing on Client:** Signals with flag = 8 that it is exiting. Does not exit until ACK from server.

**Processing on Server:** Upon receiving the Flag = 8 signal, server ACKs with Flag = 9.