Service Description

The Chat Protocol (CP) is intended to give users of the ability to send basic text messages between two hosts. The connection will be between the local host and a remote host to send messages. This simple computer network protocol is at the application layer because it is intended to be utilized by users and applications directly. Of the two types of application layer paradigms, this protocol uses the client-server, see Figure 1.

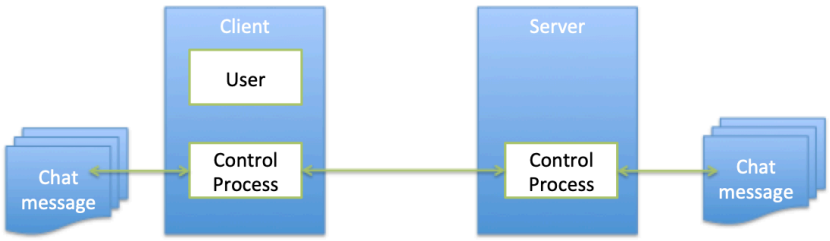

Figure 1

Message Definition (PDUs)

Addressing

This protocol will listen, retrieve and send messages through port 5035. All of the messages will be UTF-8 string format there is no need to operate on a second port to download and retrieve messages for better parsing of the data. This is because all messages will be in the same format.

Chat Messages

The chat messages will be in ASCII format and using the UTF-8 notifications for indicating beginning of message and end of line, CRLF indicating that message is complete.

Protocol Response messages

Each response has a 3 digit number in the PDU.

3 bytes	1-1014 bytes
Response code	Variable: response message

Response Code	Definition
220	received username successfully
230	user authenticated successfully
240	message successfully downloaded
250	message successfully downloaded
255	end of messages
256	no new messages

Protocol Error messages

Each error message has a 3 digit number in the PDU.

3 bytes	1-1014 bytes
Response code	Variable: response message

Error Code	Definition
500	Bad Command
540	error sending message
545	receiver user does not exist
546	too many : characters found in PDU from client
550	error downloading message
560	error sending username
565	error authenticating user

Note: There are more than 2 - as noted in the project description only 2 of these will be counted towards the project design requirements, however it was necessary to add these additional error messages to complete the design of this protocol.

Commands

Each command is a 4 ASCII character followed by variable data in the PDU.

4 bytes	1-10 bytes
USER	Variable: username

4 bytes	1-15 bytes
PASS	Variable: password

4 bytes	1-10 bytes	Delimiter	Message 1-1009 bytes
SMSG	Variable: username of receiver of message	:	Message

4 bytes
TERM

4 bytes
RMSG

4 bytes	1-10 bytes	Delimiter	Message 1-1019 bytes
RMSG	Variable: username of sender of message	:	Message

Chat Protocol Commands	
Command	**Description**
USER	Username used for login credentials to the system. **Error Response:** 560 - error sending username Notice that the error code available only indicates that the username was not successfully transmitted and not related to the validity of the username. This is for security purposes. Once the username is sent, the next message expected should be the password. This strategy will limit knowledge to hackers; the goal is to prevent explicitly identifying if the password or the username is correct. **Success Response:** 220 - received username successfully As noted in the Error response section above, this command will not give an indication if the username is valid or not. If the username was successfully received the state of the protocol is ready to receive the password and will return a success status code indicating that the protocol was successfully received.

Chat Protocol Commands	
Command	**Description**
PASS	User Password used for login credentials. **Error Response:** 565 - error authenticating user Notice that the error code available for this command is only to indicate that authentication failed. For security purposes, stated in the USER command description, information regarding whether the authentication issue is related to username or password is withheld. **Success Response:** 230 - user authenticated successfully For the protocol to move to a state where it can process commands beyond authentication the protocol must receive 230 indicating that the user has been successfully authenticated.
RMSG	This command is to tell the protocol to Receive new messages sent by another user. **Error Response:** 550 - error downloading message If the client reaches timeout before receiving and end of line symbol, then there was an error downloading the message. If there is an error downloading the message then the protocol should terminate all processes and return to idle to ensure that this protocol will not crash any system. This error response will turn the protocol state to ‘error processing’ which will tell the protocol to terminate and return to idle. **Success Response:** 250 - message successfully downloaded When downloading a Chat message, when the message reaches <CRLF> on the client side, this means that the entire message has been downloaded and the protocol state should be ready to process the next command.

Chat Protocol Commands	
Command	**Description**
SMSG	This command is to tell the protocol to send the chat message sent by another user. **Error Response:** 540 - error sending message If the server side did not get to <CRLF> notification before reaching timeout, there was an error in sending the message. If there is an error sending the chat message then the protocol should terminate all processes and return to idle to ensure that this protocol will not crash any system. This error response will turn the protocol state to 'error processing' which will tell the protocol to terminate and return to idle. **Success Response:** 240 - message successfully sent When sending a Chat message, when the server reaches <CRLF> that means that the entire message has been downloaded and the protocol state should be ready to process the next command.
TERM	This command is to tell the protocol to return to Idle state and terminate any process in progress.

Client-Server Example Interaction

Figure 2 represents an example of the client server message Interaction for the Chat Protocol.

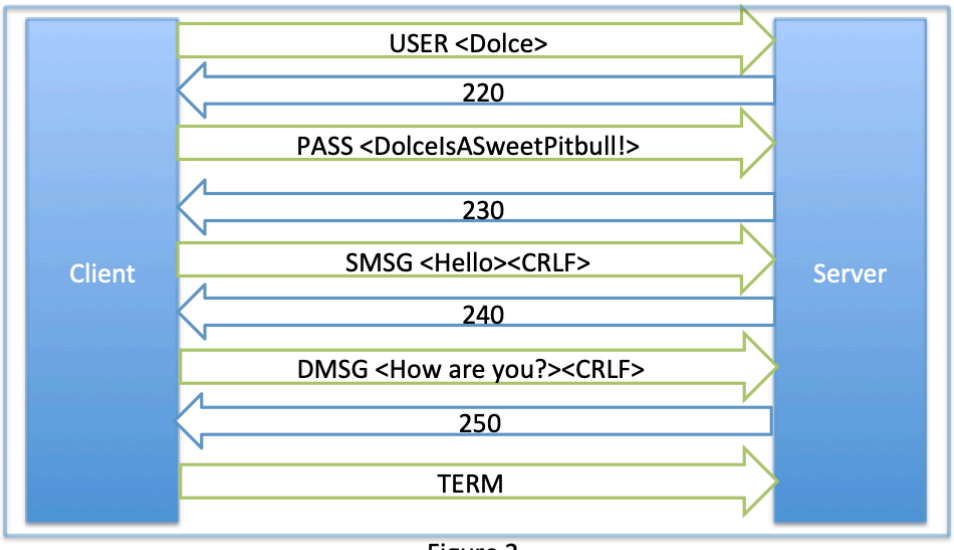

Figure 2

DFA

The Chat protocol States table describes the purpose for each state and the messages that are received and sent for each state. See Figure 3 for DFA, for more information about response code and their definitions, please see section PDU - Protocol Error messages and Protocol Response codes.

Chat Protocol States	
State	**Description**
Idle	This state is intended to represent the protocol listening on port 5035 to initiate the Chat Protocol. **Incoming Messages:** 560, 565 If there was an issue with authentication or not receiving the username then the protocol will be brought to idle state. TERM If the protocol sends TERM this indicates that the user should be logged out and all processes should be terminated and the protocol state should return to idle. this command can be sent from process command state when user wishes to end a session or the error processing state. **Outgoing Messages:** USER To be brought out of the idle state the protocol must receive USER command with the username used to authenticate the user.
User Validated	This state is when the username has been successfully received, validation of username will not happen until the 'Authentication Validated' state. **Incoming:** USER The protocol should receive the username that will be used for authentication **Outgoing:** 560 This is an error message intended to bring the protocol back to idle state when there was an error retrieving the username PASS This is the next command that should be received and it is the password that will be used for authentication.

Chat Protocol States	
State	**Description**
Authentication validated	This state is when the Password and the username has been received. It will check that the username exists and the password received is correct for the given user. **Incoming Messages:** PASS This command will be received along with the data variable containing the password for the given user. **Outgoing Messages:** 565 This is an error message if there was an issue with authentication. This message will be received for any of the following issues: 1. Username does not exist 2. Password is not correct for given user 3. Issue receiving password
Process Command	This state is when user has been authenticated and the protocol is ready to process commands to send or download messages or to log user out. **Incoming Messages:** 230 This is the response code from the Authentication Validated state. 240 This is the response code from sending message state indicating that SMSG command has been completed successfully. 250 This is the response code from downloading message state indicating that DSMG command has been completed successfully. **Outgoing Messages:** TERM This is the command to be used if the user is logged out and an indication to stop all processes and move to an idle state. RMSG This is the command used if there is a message that needs to be dowloaded. SMSG This is the command used if there is a message that needs to be sent.

Chat Protocol States	
State	**Description**
Receiving Message	This state is to represent the protocol downloading messages. **Incoming Messages:** RMSG This message is a command indicating for the protocol to download a chat message. **Outgoing Messages:** 250 This message is a response code indicating that downloading of the chat message has completed successfully. 550 This message is an error response code indicating that there was an issue downloading the chat message.
Sending Message	This state is to represent the protocol sending a message. **Incoming Messages:** SMSG This message is a command indicating for the protocol to send a chat message. **Outgoing Messages:** 240 This message is a response code indicating that the message was successfully send and the protocol should return to process command state to wait for next command. 540 This message is an error response code indicating that there was an issue downloading the chat message and the next state should be error processing.
Error Processing	This state is to represent the protocol processing an error from either sending a message or downloading a message. **Incoming Messages:** 550 This message is an error response code from the downloading message state. 540 This message is an error response code from the sending message state. **Outgoing Messages:** TERM This message is a command which will log out user and terminate all processes returning the protocol to idle state.

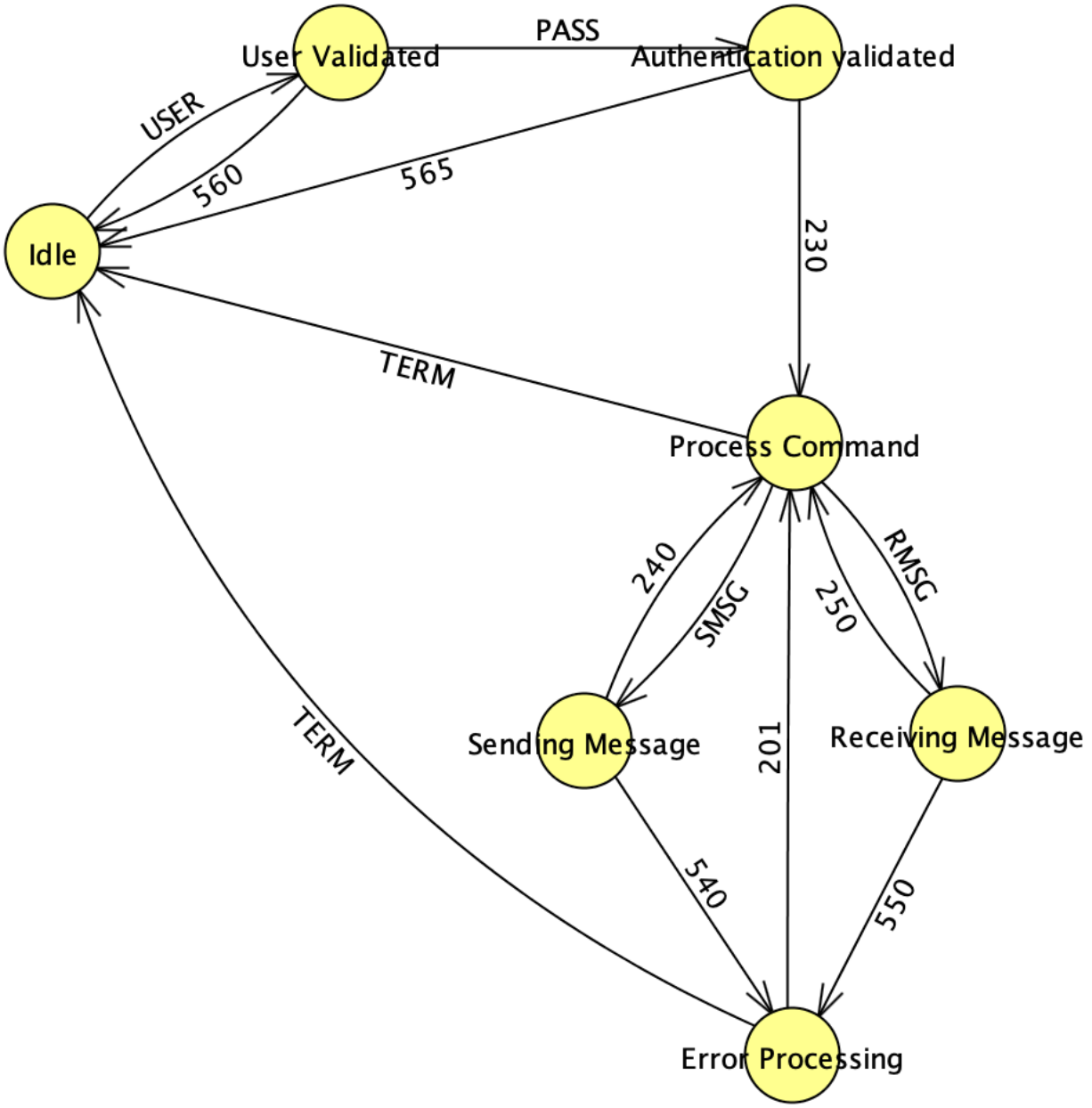

Figure 3

Note: The messages represented in the DFA (Figure 3) only represent messages that change the state of the protocol.

Extensibility

Because this protocol is simple and leverages other stable network protocols to perform certain functions, there is the ability to add commands/services to this protocol. More importantly there is the ability to change out a protocol leveraged at a layer underneath the application layer. As seen in the CP Architecture Layering chart.

CP Architecture Layering	
Application Layer	CP
Presentation Layer	UTF-8
Transport Layer	TCP
Data Link Layer	IPv4

To make this protocol more widely usable with many different use cases would be to add encryption when sending the password and the chat message data over the network. This could be done by adding a protocol that encrypts in transit over the transport layer. An example would be to use SSH instead of TCP in the transport layer. This is just one example of many potential changes that can be made to CP.

Security implications

There is a level of security in this protocol and that is authentication. This protocol has to verify that the user has access to the host and confirm their password. This protocol will wait to verify user and password together and only respond if there is a problem with authentication and will not identify if the issue is with the user or password. The reason for this is to not let a potential hacker know if they have retrieved a correct user which can make the protocol vulnerable.

However, there is much to be desired with security. All communication is sent in plain text. This means that anyone who is watching the network will clearly see your password and messages being sent back and forth. A good way to update this protocol would be to use SSH instead of TCP to ensure data is encrypted in transit.

List of Changes during Implementation

Added Response Error Messages:
- 500: If command is out of order, or command does not exist
- 545: error sending message - receiver user does not exist
- 546: too many : characters found in PDU from client

Added Response Success Messages:
- 255: end of new messages
- 256: no new messages
- 201: Recovered from error

Other Changes:
- not using telnet string, just using utf-8

- PDU length will be 1024 bytes - first 4 will be the command or error code
- change name of downloading command to receive message. So instead of DMSG it will be RMSG
- Updated CP Architecture Layering to represent what was implemented.

Design Elements Not implemented

Recovery from error processing state in the server has not been implemented.

References

1. Reynolds, Joyce, and Jon Postel, "Assigned Numbers", RFC 943, ISI, April 1985.
2. Postel, Jon, and Joyce Reynolds, "Telnet Protocol Specification", RFC 854, ISI, May 1983.
3. Postel, Jon, "Transmission Control Protocol - DARPA Internet Program Protocol Specification", RFC 793, DARPA, September 1981.
4. https://docs.python.org/3/library/sqlite3.html
5. https://docs.python.org/3.5/library/socket.html
6. https://docs.python.org/3/library/threading.html