Peer to Peer Transmission of Data using Versatchat Approach

Manjunatha Swamy. C[1*], *Senthilnathan.* S[2], *Shivananda.* S[3], *Manu* Elappila[4], *Naveen* Kumar. C[5], and *Vigneshwaran.* T[1]

[1]Department of Artificial Intelligence, Machine Learning & Data Science, Christ University, Kengeri Campus, Bengaluru, India
[2]Department of Electronics and Communication Engineering, Christ University, Kengeri Campus, Bengaluru, India
[3]Department of Information Science and Engineering, M. S. Ramaiah Institute of Technology, Bengaluru, India
[4]Department of Computer Science and Engineering, Christ University, Kengeri Campus, Bengaluru, India
[5]Department of Computer Science and Engineering, Aditya College of Engineering and Technology, Bengaluru, India

Abstract. World is moving online physical transportation of information from place to place and person to person communication is becoming an important thing and Information transfer through fibre optical cables connecting continents and countries. versatchat is an online chat application that converts your conversation into virtual data and transports it to any corner of the world having an active internet connection giving you the ability to communicate with anyone at any remote location from the comfort of your own home, This application supports has an light weight single threaded server which can be run by anyone turning an client into an server without the need for high end hardware and provides support for both Peer To Peer and One to many. Many existing applications and the competition from whatsapp, messenger in the market and hence proposed work is introduced with new features such that no other app has in the market. Searching for messages based on time, type, whom has sent the message, grouping of messages into layouts, uploading of individual media to Google drive and re download them as new messages with very simple and easy use of GUI. Keywords: Multiuser, Client and Server, ChatUser Communication, TCP.

1 Introduction

As businesses grow and reaches out to a wider customer base a need arises to transfer important transactions and deals reliably to large groups of people over varying geographical distances. Networking can help with this. Using TCP/IP protocols can make our application wait for a secure link to be established between our server and the target machine in which doing communicating with by waiting for an acknowledgement message from the target host system. Once communication has been established messages are passed quickly and securely over the underlying network layer.

Versatchat is a virtual chat application[1] that uses TCP/ID protocol to transfer data means text messages, videos, documents, pictures as multimedia over large distances all the while making sure data packets are resembled in the correct order and all the bytes are received before decoding the message. Some challenges here face with networking are handling partially received data, how to store this data until the full message has been received. How to handle network failures? What precautions to take while sending large files such as video and audio to ensure they arrive correctly at the target machine.

The purpose of building this application is facilitating data sharing between two or more remote systems connected over a network. How to transfer data over long distances while making sure it arrives as whole packets and how to re-assemble these packets in the correct order is the purpose are the challenges of networking which are proposed to be tackled by this application. Although a lot of similar applications already exist today in the open market, here proposing better security and infrastructure and more user friendly interactive GUI. Configuring of the application is as simple as changing the data of few text files and all messages are transported over secure TCP network which ensures message delivery by using an acknowledgement system. Media is transported over an password encrypted Apache FTP client and server and media is not re uploaded again when the message associated with it is forwarded therefore reducing overall bandwidth

2 Related Work

Peer to peer data transfer is a systematic way of structuring the movement of data across nodes through a medium in given environment context. It involves proper planning, constructing and usage of online chat application for the organization, including administration, tracking events using associated strategies. Ease of finding the target points with nearby connection or network as local area net work (LAN), sometime base o the organization type it includes wide area network as well. These strategies help organizations to utilize resources effectively and efficiently. Web of things is evolved to greater extent to

* Corresponding author: manjunath.nhce5@gmail.com

utilize the information in transforming the data as and when it is required.

Domain knowledge to work with point to point network (P2P) for [2,3] an organization is associates with technical support in the working environment such that if huge number of nodes in the transmission is involved then security aspects also should be addressed. Encryption of the data is mandatory to ensure and manage in certain levels of risk where ranges also matter to do basic checking through wireless fidelity network. Two mode of structuring the communication such as wired and remote authorization, in wired systems acquiring real mode of transmission across different nodes. Ethernet mode of cabling is known for low cost and long life which is normally used in computer at home and working place. Fiber optic cable[4] is used to transfer the data more trust basis over long distance at high speeds. Long distance communication use radio waves to carry the data through air medium which enhance different devices[5] to be used with any organization without using cable system. Long distance communication uses microwave, satellite, cell and Bluetooth devices with others. Cable oriented communication ensures speed, dependability and security aspects, also provides greater adaptability and long time availability. Managing of network based systems is grouped with predetermined plan of execution. This process of management is affected by the wireless[6] environment approaches like LAN, WAN with various capacities.

3 Proposed System

Proposed systems to use a single threaded server and client which can accept connections from multiple sources while still maintaining performance. Our application is platform independent and operates on a decentralized server where a client can become a server by directing network traffic by running a small server instance on their device. Our app programmed in java is backward compatible with many old operating systems as JDK versions date back to 1.3 and below and is architecture independent.

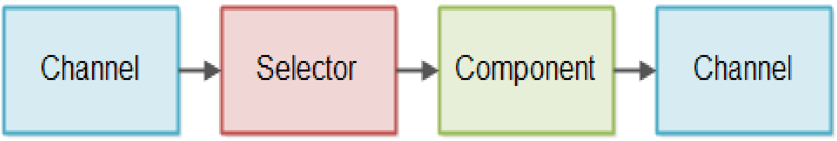

Fig. 1. Selector channel model.

In Figure 1, a part utilizes suitable selector to verify proper link to process information. Then, at that point, part of message creates some result in view of the info. The result is kept in touch with a Channel once more. A non-obstructing Input Output(IO) pipeline doesn't have to both perused and compose information. A few pipelines may just understand message[7], and a few pipelines may just compose information. A non-obstructing IO pipeline may likewise be perusing from different Channels simultaneously. For example, perusing information from various Socket Channels. The progression of control in the above chart is likewise streamlined. This starts the perusing of information from the Channel through the Selector. Link drives the information into link and from that point, regardless of whether that is what the above chart proposes.

Typically, when a server has to handle multiple clients, the server creates a separate for each thread which runs parallelly with the server. These threads[8,9] check the client socket at frequent intervals for new data and sends the data if any to the server for processing. This works fine for high end computers with heavy computation power but as number of clients increases the threads start to consume more system resources and slow down the whole server and even crashing it when space runs out. To take this introducing a server that uses just a single thread for processing thousands of clients while keeping resource message minimum.

The greatest distinction between a non-hindering and an obstructing IO pipeline is the manner by which information is perused from the basic Channel (attachment or document). In Figure 2, IO parallel links read information from flow (an attachment document) and partition of information as intelligent messages. Process of segmentation of information as sub partitions for utilizing a tokenizer. All things being equal, you break the flood of information into greater messages, Dividing the flow as information's to Message Reader. Here is a representation of reduce of flow to required messages.

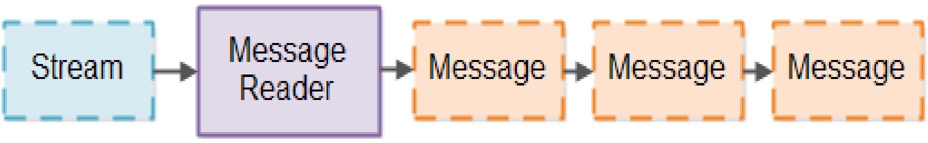

Fig. 2. Blocking server pipeline.

An obstructing IO pipeline can utilize an Input Stream like connection point where each byte in turn can be perused from the basic Channel, and where the Input Stream like connection point blocks until there is information prepared to peruse. These outcomes in an obstructing Message Reader execution. Essentially, an obstructing Message_writer is a piece of a message was framed, and message framing must be continued later. At this point read a block of information from a Selectable Channel couldn't say whether that information block contains less or in excess of a message. An information block might actually contain a fractional message (under a message), a full message, or in excess of a message, for example 1.5 or 2.5 messages.

The proposed message processing tree in shown below in Figure 3 explains on data block received is segregated into message and further message is divided into meta partitions in order to make communication better. Message Reader is essential in obstructing information framing.

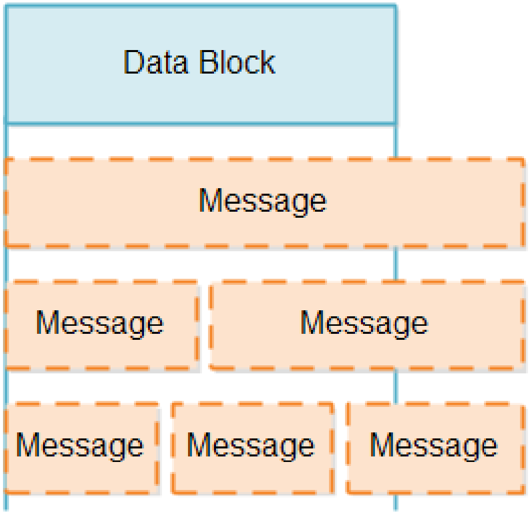

Fig. 3. Partial message processing tree.

Key concepts such as finding complete information in block of data. Message arrives in the identification of full messages to check whether the information contains less than one complete message. In the event that the information block contains atleast one full message, these messages can be sent down the pipeline for parallel handling. At the point when there is a fractional message in an information block, either without anyone else or after at least one full message, that halfway message should be put away. Recognizing full messages and adding away fractional leads to obligation. Various links are used to utilize one Message Reader for each Channel.

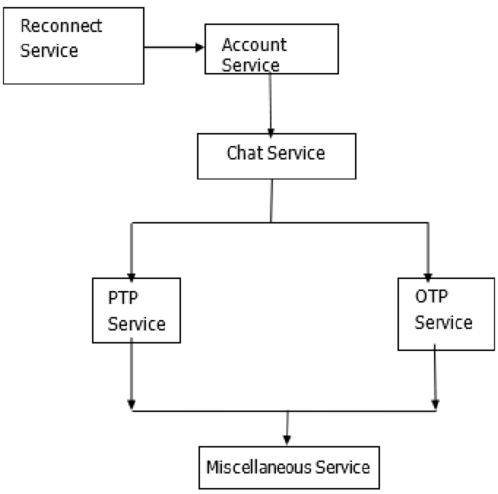

Fig. 4. Architecture for peer to peer data transfer system.

In Figure 4, the Reconnect Service it is responsible for restoring state of the application in an event of network /server failure. It relog's in to the server and resends any pending messages or important event messages. In Account Service stage handling Logging in to the server or signing out from the server. Also, for first time registration or re-registration into the server in creates the necessary folders and files for first time use.

PTP Service is also known as Peer to Peer[10,11] Or 1 V 1 service is used to handle chat messages between two people on a chat. The main resources managed by this class are the ChatMetaData, TelChatMsg, TelChatNotification Class.

Chat MetaData Stores the person information of each chat directory such as the person with whom to are talking to and has handlers which update the person information when he updates his profile information. TelChatMsg this is message class for all text/media messages associated with one to chat. It also has the receipientID, the senderID, and the timestamp it was sent on. TelChat Notification is a sub classed notification class of the MsgNotify Class for handling notifications specific to TelChatMsgs. Later OTPService[12] also known as One to Many Or 1 V Many services are used to handle chat messages between two people on a chat. It inherits from ChatService Class to handle custom notifications bandwidth [13] uses specific message classes to distinguish chat messages from group messages. It makes sure that the chat message is sent to the right directory and only viewed if the user has opened the chat. It spawns download threads for managing media downloads and loads/updates metadata of chats. The main resources managed by this class are the ChatGroup MetaData, TelGroupMsg, GroupChat Notification Class.

OTP Service is known as One to Many Or 1 to Many services are used to handle chat messages between two people on a chat. ChatGroup MetaData: Stores all the memberID of all the people involved in the group chat as an HashMap of Strings. Contains The owner who created the group, the timestamp it was created on and handlers for listening for member updates such as new member added, member removed or member changed to admin. GroupChatMsg. The message class for all text/media messages associated with one to chat It Is Further It contains the receipientID, the groupID, and the time stamp it was sent onGroupChat Notification: The sub classed notification class of the MsgNotify Class for handling notifications specific to GroupChat Messages.

Miscellaneous Service is responsible for metadata changes to chats and such groups. These involve account changes such as someone changing their phone number, deleting their account, resending pending messages when network is back online. It loops through all chats in a single list and propagates these messages to everyone as these are not specific messages.

3.1 Algorithm for Proposed System

Step 1: SignUp the registered User and verify the account exist or not and select the channel model using selector().
Step 2: Check User Actions as Open Chat and its associated actions.
Step 3: Verify the Message queue and Socket services to send, read and close the sessions.
Step 4: Collect the message stream.

Step 5: Read and Load message Dataset.
Step 6: Divide the data blocks to convert into Proper messages with suitable bandwidth of the channel.
Step 7: Transform the messages and connect to server.
Step 8: Find suitable services associated with model and use pipeline read cycle.
Step 9: Categorize base message as Chat holder, Telchat message and Group chat message.
Step 10: Use Notify function mechanism and file selection type and do communication

4 Result Discussion

In this work, a practical prerequisite characterizes an element of a product framework or its part. A capability is portrayed as a bunch of information sources, the way of behaving, and yields. Practical necessities might be computations, specialized subtleties, information transmission control and handling and other explicit usefulness that characterize what a framework should achieve. Conduct prerequisites portraying every one of the situations where the framework utilizes the practical necessities are caught being used cases. Mainly system do important functions by considering sessions[14], unique Uid, password with secret key later compare in repository information. Check for pattern match then it progress otherwise exception will occur. Transform the document to new abstracted document by suitable steps. On other end it should be able to decrypt it using other pseudo code. Various screen shots are ensemble here to discuss as result part.

In Figures 5 and 6 message encapsulates all media files who have their extensions enlisted in the MME Table of the protocols jar. Media Messages actions presented to the user will differ depending on the state the message is in. If the files who file is still being downloaded/uploaded then the user will be unable to open the file, if the file is a zip file, then an in built ZipViewer will be presented to the user. Right clicking on the media messages will show the full information of the message.

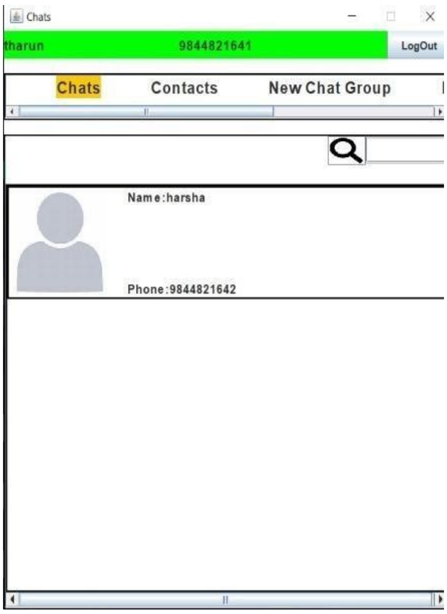

Fig. 5. Chat panel of the application.

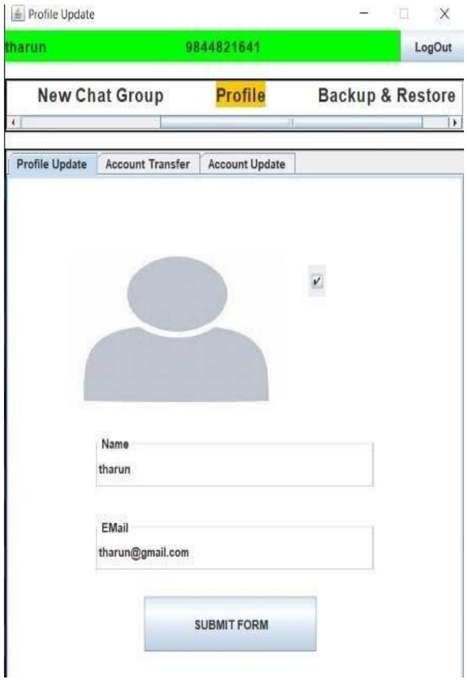

Fig. 6. Phone transfer and profile update.

In Figure 7, Network of the authorization[15] can be framed with available options profile update, account transfer and account update. Ordered flow starts if authorization reverify with Google link, which includes metrics to retrieve with specific access permission. Authorization, time allocation, explicit permission is verified by application. As a result of proper verification code user can exchange new unique identifier both at beginning and at the end of the process.

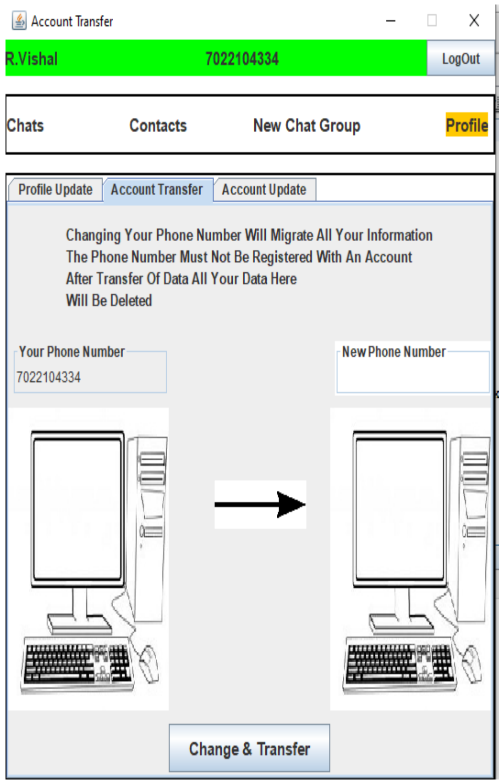

Fig. 7. Account transfer.

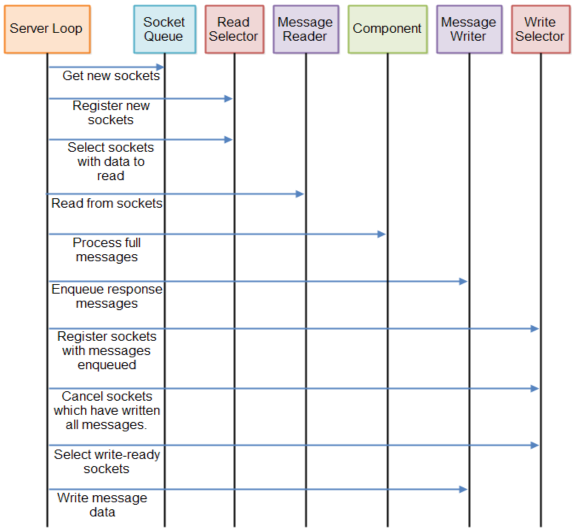

Fig. 8. Server interaction model.

In Figure 8, server interaction model to check so that approaching information occasionally might check whether fresh information is obtained or not. System verifies on various occasions until at least complete message is found. It isn't sufficient to check one time. Essentially, a non-hindering system verifies occasionally in the event that there is any information to compose. In the event that indeed, the server needs to check by comparing associations kept in touch with them. Messages are kept in queue when time is short to transfer, non impeding system terminates with parallel Requirements to execute. Execution of multiple messages in parallel leads system performance.

5 Conclusion

In this digital age where the world is connected via the internet, a world where leading business entrepreneur's and the average middle-class men need to communicate with each other across the globe by focusing less on how and more on when and why, an environment needs to be created where only relevant individuals can share important media such as documents, pictures and, videos among themselves securely and efficiently. Network transmission is very promising and feasible in the society compared to use of cell network, automobiles and delivering the messages. User experience the communication across other users with chat applications. This is a variation of Internet approach to enable high reliable transmission of information. Chat App's provide connection oriented anytime as required.

Future enhancements for the work can integrate with AI approach to predict the load balance and congestion control mechanism to handle the transmission effectively across the nodes or system.

References

1. E. Forsythand, C. Martell, Lexical and discourse analysis of online chat dialog, in Proceedings of the International Conference on Semantic Computing (ICSC), IEEE, Irvine, CA, USA, October 08 (2007), 19-26
2. B. Xue, Y. Mao, S.B. Venkatakrishnan, S. Kannan, Goldfish: peer selection using matrix completion in unstructured P2P network, in Proceedings of the 2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), IEEE, Dubai, United Arab, July 12 (2023), 1-9
3. H. Yao, Y. Xiang, J. Liu, Virtual prosumers P2P transaction based distribution network expansion planning. IEEE Trans. Power Syst. **39**, 1044-1057 (2024). https://doi.org/10.1109/TPWRS.2023.3240830
4. M. Aziz, M. Jarke, Mobile-to-mobile multimedia service provisioning in the IMS using REST-based mobile services. Int. J. Comput. Commun. Eng. **2**, 455 (2013)
5. H. Naveen, H. Chetan, B. Kulkarni, S. Mohanty, S. Druva Kumar, G.M. Sreerama Reddy, The effective transmission of acquired sensor data with FFT, DWT and DTCWT in Different Channel Environment. Int. J. Recent Technol. Eng. **8**, 737-742 (2019). https://doi.org/10.35940/ijrte.B1136.0782S319
6. H.Y. Hsieh, R. Sivakumar, On using peer-to-peer communication in cellular wireless data networks. IEEE Trans. Mob. Comput. **3**, 57-72 (2004). https://doi.org/10.1109/TMC.2004.1261817
7. M.T. Kimour, D. Meslati, Deriving objects from use cases in real-time embedded systems to provide message transmission. Inf. Softw. Technol. **47**, 533-541 (2005). https://doi.org/10.1016/j.infsof.2004.10.003
8. Y. Kawahara, T. Aoyama, H. Morikawa, A peer-to-peer message exchange scheme for large-scale networked virtual environments. Telecommun. Syst. **25**, 353-370 (2004). https://doi.org/10.1023/B:TELS.0000014789.70171.fd
9. D. Hajder, S. Bhushan, Distributed chat application for secure P2P messaging. **20**, 323-333 (2020)
10. V.N. Padmanabhan, H.J. Wang, P.A. Chou, Resilient peer-to-peer streaming. Proc. ICNP, IEEE, Atlanta, GA, USA, November 04-07 (2003), 16-27.
11. M. Ripeanu, I. Foster, A. Iamnitchi, Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. IEEE Internet Comput. **6**, 50-57 (2002).
12. G. Brambilla, M. Picone, S. Cirani, M. Amoretti, F. Zanichelli, An adaptive peer-to-peer overlay scheme for location-based services, in Proceedings of the 2014 IEEE 13th International Symposium on Network Computing and Applications (NCA), IEEE, Cambridge, MA, USA, August 21-23 (2014), 181-188
13. M. Jain, C. Dovrolis, End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput, ACM SIGCOMM Computer Communication Review,

Association for Computing Machinery, New York, NY, United States, August 19 (2002), 295-308

14. Z. Chen, A. Delis, P. Wei, Identification and management of sessions generated by instant messaging and peer-to-peer systems. Int. J. Coop. Inf. Syst. **17,** 1-51 (2008). https://doi.org/10.1142/S0218843008001750
15. S. Sen, J. Wang, Analyzing peer-to-peer traffic across large networks, in Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement, Association for Computing Machinery, New York, NY, United States, November 06 (2002), 137-150