IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 3, March 2023

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.423

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 3, March 2023 ||

| DOI:10.15680/IJIRSET.2023.1203104 |

Peer-to-Peer File Streaming Using Web Sockets Protocol

Geetha Rani E[1], Roshan Jose S[2], Joel Thomas Chacko[3], Jeanette Krizelda K[4], Joshua Paul C[5]

Associate Professor, Department of Computer Engineering, MVJ College of Engineering, Kadugodi, Karnataka, India[1]

U.G. Student, Department of Computer Engineering, MVJ College of Engineering, Kadugodi, Karnataka, India[2]

U.G. Student, Department of Computer Engineering, MVJ College of Engineering, Kadugodi, Karnataka, India[3]

U.G. Student, Department of Computer Engineering, MVJ College of Engineering, Kadugodi, Karnataka, India[4]

U.G. Student, Department of Computer Engineering, MVJ College of Engineering, Kadugodi, Karnataka, India[5]

ABSTRACT: This work studies a peer-to-peer file streaming system with the following characteristics. The file streaming capacity is dynamic and depends on the storage capacity of the individual peers' devices. The large file is split into multiple chunks to stream it across peers. These chunks are not stored on the cloud or any intermediate channel, eliminating the need for a cloud storage bucket. To achieve this real-time streaming among peers, the Secure WebSocket (WSS) protocol is used, which has a minimal buffering delay. Depending on the internet bandwidth of the sender, the chunk size is dynamically modified to achieve a faster streaming rate. Before the chunks are received, the system allocates the required storage space for the incoming file stream. As these chunks arrive, they are written directly to the native storage of the device. Each room's name is unique. The creator of the room shares the name with his peers to start a streaming session. From this point, any peer in the room can act as a sender or a receiver.

KEYWORDS: Peer to Peer, File Streaming, Web Sockets, Data Transfer (Blobs)

I. INTRODUCTION

The current society is highly dependent on data. Being able to possess data and share data seems to be the invisible fabric on which society not just lives but thrives. Sharing large files requires cloud storage. Axiomatically, it implies that one's information is stored on the Network. By sending the files through Peer-to-Peer Network Communication the online cloud storage mandate is negated; thus, able to send it to the recipient directly. The efficient WebSocket (WSS) protocol as the Peer-to-Peer Network Communication Framework is utilised.

This work explores a peer-to-peer file streaming system with the following properties: -
File streaming capacity is dynamic and depends on the storage capacity of each peer's device. Large files are split into multiple chunks for streaming between peers. These blocks are not stored in the cloud or intermediate channels, so no cloud storage bucket is required. To achieve this real-time streaming between peers, the Secure Web Socket (WSS) protocol with minimal buffering latency is used. Chunk sizes are dynamically adjusted for faster streaming rates, depending on the sender's internet bandwidth. Before starting to receive chunks, the system allocates the required disk space for the incoming file stream. As these chunks arrive, they are written directly to the device's native memory. Each room has a unique name. Room creators share their names with their peers to start a streaming session. From this point on, each peer in the room can act as a sender or receiver.

The gap between rate-distortion analysis and content delivery research is bridged using this work [1]. A robust premier service on the internet for the past few years has been Peer-to-Peer data streaming services. The Mesh-based structure is used by most working systems where data scheduling is one of the most important problems. A new scheduling algorithm aiming at the optimal throughput is proposed: Bipartite-matching based Block Scheduling

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 3, March 2023 ||

| DOI:10.15680/IJIRSET.2023.1203104 |

algorithm [2]. The purpose of this file streaming system is to describe state-of-the-art strategies that enable P2P streaming systems to use efficient scheduling of streaming data [3].

Peer-to-peer data streaming is a scalable service and this service is applicable for encrypted and unknown network traffic. The distinction between P2P file-sharing traffic and P2P media streaming are classified and understood in this work. Each packet is sent at high frequencies in a P2P media streaming traffic when compared to the file sharing traffic

4. - This is an understanding that portrays P2P file sharing as a more efficient file-sharing service. This is a P2P streaming system which is gossip-based called Continuity Streaming. This system brings a new and essential property (complete coverage of data propagation) while maintaining high resilience and low overhead. DHT supports on-demand data retrieval which enables quick recovery of data segments, which have a higher probability of being lost or missed due to gossip-based data scheduling. This service of DHT enables quick and seamless continuous playback during streaming [5]. Live peer-to-peer streaming has developed into a potent and adaptable means for distributing multimedia material. The primary service for media streaming is the media content transmission scheme, which is in charge of data assignment and availability of information exchange. To handle changing network conditions, the differential media delivery technique suggested in this research uses redundant data receiving and optimum buffer management [6]

II. RELATED WORKS

A few works closely related to this field of study are mentioned below. In the first work [7], an attempt to understand P2P file sharing through a survey of network coding was made. Protocols that enable large file distributions over untrusted or unreliable networks are Peer-to-peer (P2P) distribution of content and file-transfer systems; these protocols are commonly used to transmit large files through the network. Transmission technique in Network Coding is of interest to researchers because it can improve network throughput and robustness and reduce download times. This review work evaluates, compares, and categorises the techniques most recently used to improve the performance of his P2P content distribution system using Networks coding. This work attempts to understand the P2P file-sharing system's network encoding performance; this study has been a comprehensive attempt to understand the same and has been a first of its kind. In the following work [8], a survey of the various file-sharing methods in the P2P networks is made. This is a trust-based model that enables users to have reliable communications with other peer devices. Essentially, it sheds on the concerns peers have while establishing a safe connection and the various actions of the Peer-to-Peer systems. There can exist collusion behaviour which occurs in the P2P File Sharing System, and an empirical study is conducted in this work [9].

In this reference [10], the authors try to maintain the user's network anonymity by providing a semi-distributed, peer-to-peer file-sharing system as a service that allows users to securely stream any file and never leaves a traced copy of the file anywhere. The sender's and receiver's file systems are saved, but the receiver can use a digital signature to verify the integrity of the file. Some protocols prevent third parties from keeping copies of one's data, and one of them is used in this work: WebSockets implementing the Secured Web Sockets Protocol. Reference [11] also suggests using Peer-to-Peer connectivity for the sharing of highly sensitive personal data, as in this methodology the storage is decentralised and the individual copies of data aren't available for exploitation. In this reference [12], communications between P2P-enabled intranet devices for transferring secure content are studied. In the third reference [13], the authors use a hybrid clustering technique to apply the K-Means algorithm to large, distributed datasets. An algorithm that can be used on large data sets is K-Means; it is a simple, scalable and robust algorithm. The k-Means algorithm isn't apt for extracting information from given unstructured data from structured data. K-Means is also an unsupervised algorithm used for clustering. Computing power, network bandwidth, and network participants are managed by Peer-to-Peer Services to facilitate the sharing of data and resources among peers. Accuracy, computational complexity, and accuracy of distributed clustering become important issues in the data delivery process in P2P environments, as the overall system performance suffers. The peer-to-peer live streaming resilience is understood in this work [14].

In this reference [15], the unstructured mobile networks' mobility methodology of distributed data updates is understood. In peer-to-peer (MP2P) systems which are unstructured and mobile, the high mobility of nodes causes frequent connection interruptions, leading to frequent topology mismatch problems and data transmission errors.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 3, March 2023 ||

| DOI:10.15680/IJIRSET.2023.1203104 |

Data transmission and synchronisation overhead should not be ignored. Flooding is a basic and simple data synchronisation mechanism to maintain data consistency. This is because mobile nodes are unaware of other users who have the same shared data item.

In this reference [16], peer-to-peer (P2P) networks are widely used in a variety of applications such as distributed storage, cloud computing, and social network applications. In P2P networks, impartiality promotes incentives to motivate users to contribute resources to the network. The main idea of this work is to provide impartial bandwidth allocation of channels in P2P file-sharing networks and presents an effective maximisation model across coupled networks aimed at providing varieties of fairness among requesting users.

Using this reference [17], an attempt to estimate the uplink and downlink's access link capacity division is made. The summation of the upload and the download is constant for a given interval. Minimization of the uplink is done in such a manner where the individual requirements of the individual peers are also satisfied, and this is implemented using the linear programming approach. The implementations of P2P file streaming are numerous. For example, blockchain technology uses P2P decentralisation services for the sharing of data of the sensitive kind [18], and another example of this is NVMe transactions using P2P and PCIe switch networks [9].

III. METHODOLOGY

To stream files using StreamIt every user is required to join a room. These rooms do not have any member limit. StreamIt has two main segments in its algorithm: the sender side and the receiver side. When the sender selects a file to stream, StreamIt splits the file data into multiple BLOBs (Binary Large Objects) that are 1 MegaByte in size.

These BLOBs are transferred to all the users in that room, through WSS (Secure Web Sockets) protocol. The WSS protocol is real-time and peer-to-peer in nature, i.e. it does not require any mediator to communicate between clients. Once the receiver receives its first stream Blob, an empty file of that file type is created, and the stream is directly piped into the empty file. Once the stream is complete, the stream is closed, and the file is sealed.

The objectives of the following work are: -

- Fast transfer speeds
- reasonable chunk size of ~ 1MB
- multi peer transfers
- no temporary storage
- secure (encrypted) mode of transfer
- real-time communication (WebSockets)

The various existing systems have the following characteristics, and they are: the existing system uses HTTP or FTP protocols, that use the polling technique; it uses HTTP or FTP protocol is not real-time in nature; this system uses bytes as the data type to transfer chunks; the availability of certain data packets among the peers has maintained a structure called as Bitmaps. The corresponding demerits of the following systems are: Protocols like HTTP and FTP send data over unencrypted channels; The risk of data breaching is prevalent; they use the polling technique, which isn't suitable for real-time communication; Multi-peer streaming of data is bandwidth expensive, and not simultaneous; User privacy isn't maintained when sensitive data is stored in an online cloud server.

The comparison between the existing system and the proposed one is:

- The existing system uses FTP/HTTP but the proposed system uses Secure WebSockets
- FTP/HTTP is not real-time, and uses a polling technique, whereas in the proposed system WebSockets offers real-time updates.
- The existing system uses FTP/HTTP that does not use any encryption standards, which means data is transferred as plain text making it vulnerable to hacking. In contrast, in the proposed system WSS uses an AES-256 algorithm which encrypts and decrypts the data transferred making it more secure than the existing system.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 3, March 2023 ||

| DOI:10.15680/IJIRSET.2023.1203104 |

The advantages of the proposed system are: Web sockets are encrypted protocol which transfers files over a secured channel, unlike FTP; The WSS or Web Sockets protocol doesn't use the polling technique; it's in real-time, which facilitates spontaneous acknowledgement of data sent through messages; Network bandwidth is saved, as this work facilitates multi-peer streaming of data through the WSS protocol standard; Since the data isn't stored in the online cloud service, users' privacy is maintained; there are no redundant copies of sensitive information available for exploitation. The Scope of this work is to give a better, more accurate model which tries to overcome the limitations of Existing Systems. The following are the methods used for the same:

A. Secure Protocol Transmission

Using encrypted protocols, data security breaches are minimised and data safety is ensured. In this work, Secure WebSockets are used, a secure and encrypted protocol to forward packets to multiple peers. Data security breaches are minimised and data safety is ensured by using encrypted protocols. In this work, packets are forwarded to multiple peers using Secure WebSockets, an encrypted and secure protocol.

B. Online Anonymity Ensured

In this work, peer-to-peer file streaming using the WSS protocol is utilised. Since the files are sent directly to fellow peers, the online presence of the data is avoided; Thereby, ensuring online anonymity. The WSS protocol is used for peer-to-peer file streaming in this work. The data are not made public online because they are sent directly to peers; thereby guaranteeing anonymity online.

C. Multi-peer Streaming

A single isolated file can be transferred to multiple peers simultaneously using the WSS protocol deployed in this work. All the users in the same room will have access to the streamed files; Thereby saving data. The WSS protocol used in this work allows for the simultaneous transfer of a single isolated file to multiple peers. The streamed files will be accessible to all users in the same room; Consequently, saving data.

D. Distributed Streaming

The WSS protocol used in this work allows for the simultaneous transfer of a single isolated file to multiple peers. The streamed files will be accessible to all users in the same room; Consequently, saving data. This work makes use of the WSS protocol, which lets a single isolated file be transferred to multiple peers simultaneously. All users in the same room will be able to access the streamed files.

This work takes the user data and splits the file data into multiple BLOBs (Binary Large Objects) that are 1 MegaByte in size. These BLOBs are transferred to all the users in that room, through WSS (Secure Web Sockets) protocol. The WSS protocol is real-time and peer-to-peer in nature, i.e. it does not require any mediator to communicate between clients. Once the receiver receives its first stream Blob, an empty file of that file type is created, and the stream is directly piped into the empty file. Once the stream is complete, the stream is closed, and the file is sealed.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 3, March 2023 ||

| DOI:10.15680/IJIRSET.2023.1203104 |

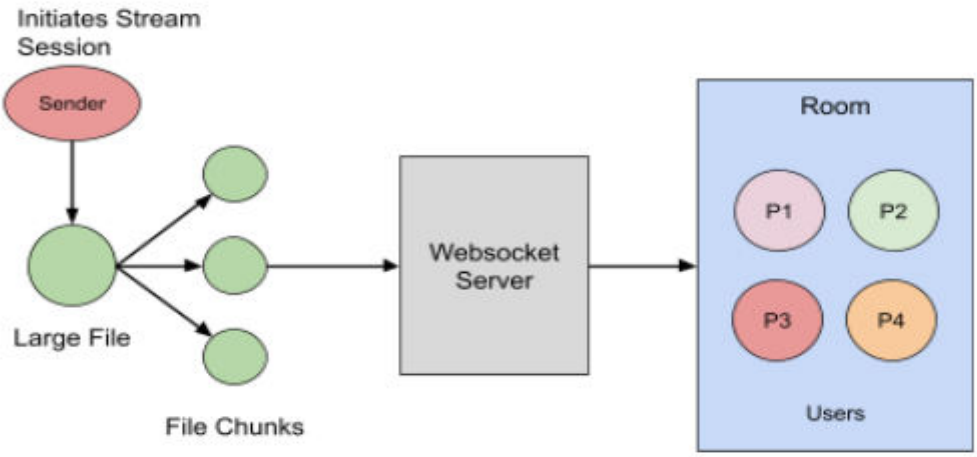

Fig. 1. Methodology

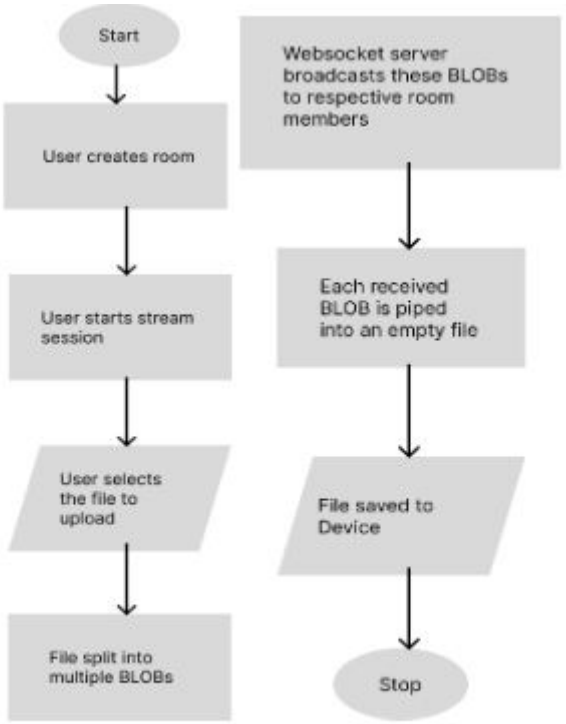

Fig. 2. StreamIt Algorithm Flowchart

A. WSS Protocol (Secure WebSocket)

WebSocket is a computer networking protocol that uses a TCP connection to provide bi-directional channels. In 2011, the IETF standardized the WebSocket protocol with RFC 6455. Web applications use WebSockets as the primary protocol as it is the latest API specification. It's maintained by the WHATWG and is a living standard that replaces the W3C's WebSocket API. WebSocket is distinct from HTTP. The WebSocket protocol works on the application layer for both the OSI and TCP models. RFC 6455 states that WebSocket is backwards compatible with HTTP as it "is designed to work over HTTP ports 443 and 80 as well as to support HTTP proxies and intermediaries." The WebSocket protocol handshake switches from the HTTP protocol to the WebSocket protocol using the HTTP Upgrade header to ensure compatibility.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 3, March 2023 ||

| DOI:10.15680/IJIRSET.2023.1203104 |

B. SocketIO

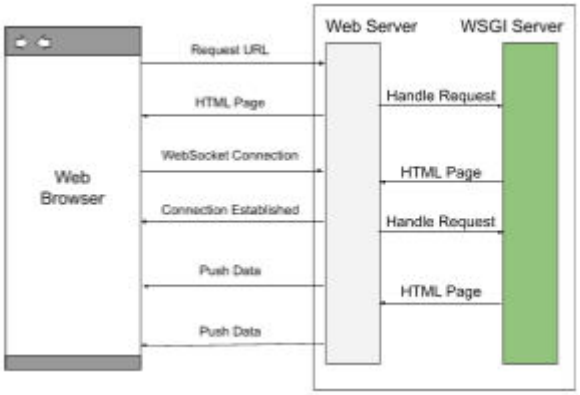

Fig. 3. WSS Protocols

Socket.IO is a WebSocket library that helps us to establish a low-latency, full-duplex and event-based connection between a client and a server. It provides additional warranties like a fallback to HTTP long-polling or automated reconnection. The WebSocket Protocol provides a bi-directional and low-latency communication channel between the browser and server.

This WebSocket library has a duplex channel between the server and the client, which is established with a Websocket connection as and when possible and uses HTTP as a backup protocol. The library is divided into two: plumbing (low-level) and the API: Socket. IO (high-level).

C. Flutter

Flutter is a free, open-source and multi-platform application development framework. It means that using the same codebase and programming language, he can create two different apps for iOS and Android. It uses the Dart programming language. It is a User Interface development tool that can be used to create multi-platform applications. Dart is similar to JavaScript, but it supports data types unlike the latter.

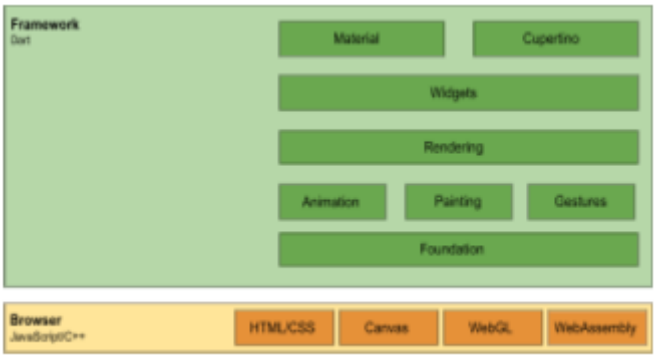

Fig. 4. Flutter Framework Architecture

There are two important parts to Flutter: -

- An SDK (Programming Advancement Pack) that contains all the tools required that help in creating applications and compiling your code to native machine code.
- A Framework that supports an assortment of reusable UI components (buttons, text data sources, sliders, etc) that one can customise for one's requirements.

D. NestJS

NestJS is a framework used to develop server-side applications that come with scalability in mind. It combines elements of Object Oriented Programming, Functional Programming, and Functional Reactive Programming, is built with and comes with TypeScript support out of the box, yet still enables developers to code in pure JavaScript.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 3, March 2023 ||

| DOI:10.15680/IJIRSET.2023.1203104 |

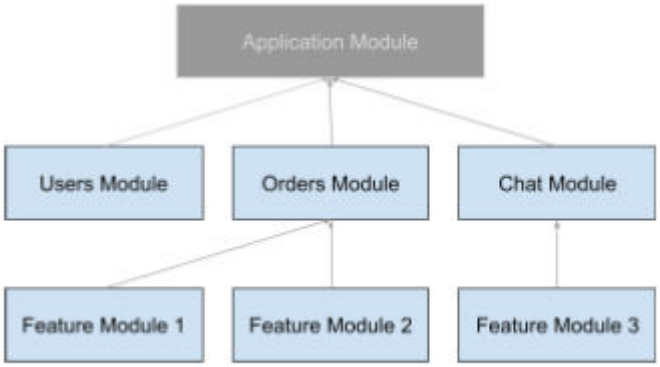

Fig. 5. NestJS Architecture

IV. EXPERIMENTAL RESULTS

This paper implemented using this Web Sockets protocol has a bi-faceted benefit. Primarily, the transaction speeds. The transaction speeds that were encountered using this protocol is summarised in Fig. 6. The streaming rate seems compatible in comparison with other contemporary protocols. The secondary benefit is that this protocol leaves no online traces of the data it's transmitting, thereby maintaining online anonymity. As suggested in the Performance Graph shown below, the File size versus time relation is fairly linear; this protocol suggests reliable network speeds in addition to network anonymity. The implementation of the work was carried out where a private room is created with a unique key by an user that is shared to other peers to join the room to stream the file. The user can select the file that has to be streamed to the peers present in the room; the peers receive the file based on their network bandwidth. The observed time duration for streaming files (with one sender and receiver) of varying sizes is portrayed in the graph below.

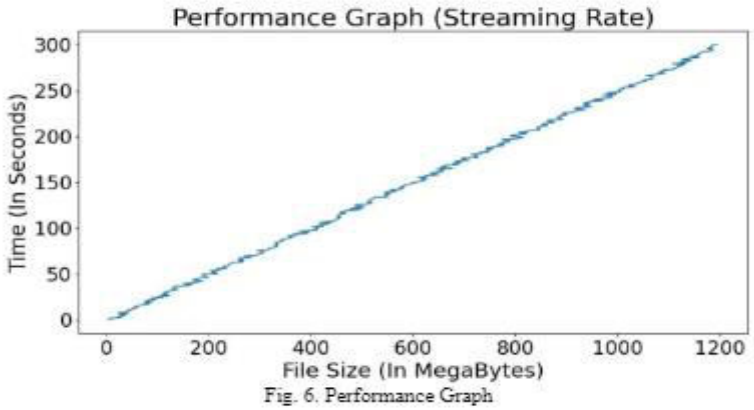

Fig. 6. Performance Graph

V. CONCLUSION

Quick and robust file sharing has become imperative given global businesses continue to rely on web applications for data collaborations and data exchange. Faster file sharing using intelligent cloud-optimised routing is preferred over other contemporaries by vendors as they provide quicker connectivity. In this work, an attempt to develop a similar system using Peer Peer Data Streaming is made using WSS Protocol to facilitate the same. The highest download and upload speeds, and network anonymity are guaranteed under this protocol, i.e. Web Sockets Protocol. Semi-Decentralized network connectivity for the users can be provided - this work attempts to store the user's network information in the cloud, and the user information is sent anonymously.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 3, March 2023 ||

| DOI:10.15680/IJIRSET.2023.1203104 |

A notable future scope of improvement is where the peers can be divided again to multiple sub-rooms to ensure high data transfer speeds. Currently the lowest network bandwidth of a peer is incorporated as the transfer bandwidth for all of the peers present in a room resulting in an overall increase in data transfer rates.

There are numerous applications for this work. A demerit of this work and a scope of improvement is that web browsers have limited storage facilities; the cache ranges between 50Mb and 100Mb; the browsers aren't authorised to access the native device storage as it possesses a security threat. This threat is negated when this work's methodology is deployed on cellular devices.

REFERENCES

1. Chun-Yuan Chang, Cheng-Fu Chou, Kwang-Cheng Chen, "Content-Priority-Aware Chunk Scheduling Over Swarm-Based P2P Live Streaming System: From Theoretical Analysis to Practical Design", IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol.4, no.1, pp.57-69, 2014.
2. Jiqing Wu, Yuxing Peng, Feng Liu, "Transmission Scheduling in Data-Driven Peer-to-Peer Streaming towards Optimal Throughput", 2009 IEEE International Conference on Networking, Architecture, and Storage, pp.277-280, 2009.
3. Jian Feng, "A Research on Scheduling Strategy in Peer-to-Peer Streaming Media", 2009 Pacific-Asia Conference on Circuits, Communications and Systems, pp.439-442, 2009.
4. Li-Juan Zhou, Zhi-Tang Li, Bin Liu, "The Proposition and Certification of P2P Media Streaming Traffic Features", 2008 Fourth International Conference on Natural Computation, vol.5, pp.281-285, 2008. Zhenhua Li, Jiannong Cao, Guihai Chen, "ContinueStreaming: Achieving high playback continuity of Gossip-based Peer-to-Peer streaming", 2008 IEEE International Symposium on Parallel and Distributed Processing, pp.1-12, 2008.
5. Shengsheng Yu, Rongtao Liao, Jingli Zhou, Jing Yu, "Differential Media Delivering Strategy for Loss Tolerant Streaming", 2008 ISECS International Colloquium on Computing, Communication, Control, and Management, vol.1, pp.290-294, 2008.
6. AbuDaqa, A.A.; Mahmoud, A.; Abu-Amara, M.; Sheltami, T. Survey of Network Coding Based P2P File Sharing in Large Scale Networks. Appl. Sci. 2020, 10, 2206
7. S. Vimal and S. K. Srivatsa, "A survey on various file sharing methods in P2P networks," 2017 Third International Conference on Science Technology Engineering & Management (ICONSTEM), Chennai, India, 2017, pp. 305-310.
8. X. Xiong et al., "High-performance radar data recorder architecture based on NVMe P2P transaction and PCIe switch network," IET International Radar Conference IET IRC 2020, Online Conference, 2020, pp. 624-628.
9. V. Argyropoulos, E. Alepis and C. Patsakis, "Semi-Decentralized File Sharing as a Service," 2022 13th International Conference on Information, Intelligence, Systems & Applications (IISA), 2022, pp. 1-8.
10. M. Conoscenti, A. Vetrò and J. C. De Martin, "Peer to Peer for Privacy and Decentralization in the Internet of Things," 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), Buenos Aires, Argentina, 2017, pp. 288-290.
11. Saroliya, J. Mondal and M. Agrawal, "A Solution for Secured Content Transferring in between Multiple Hosts within P2P Enabled Intranet," 2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3), Lakshmangarh, India, 2020, pp. 1-3.
12. Raju, S., Chandrasekaran, M. Performance analysis of efficient data distribution in P2P environment using hybrid clustering techniques. Soft Comput 23, 9253–9263, 2019.
13. V. Fodor and G. Dan, "Resilience in live peer-to-peer streaming [Peer-to-Peer Multimedia Streaming]," in IEEE Communications Magazine, vol. 45, no. 6, pp. 116-123, June 2007.
14. Chuan-Chi Lai, Chuan-Ming Liu, A mobility-aware approach for distributed data update on unstructured mobile P2P networks, Journal of Parallel and Distributed Computing, Volume 123, 2019, pp. 168-179, ISSN 0743-7315.
15. Shiyong Li, Wei Sun, Quan-Lin Li. Utility maximisation for bandwidth allocation in peer-to-peer file-sharing networks. Journal of Industrial and Management Optimization, 2020, 16(3): 1099-1117.
16. R. Kumar and S. K. Awasthi, "Link Capacity Division in P2P File Sharing System," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021, pp. 1-5.
17. Demir, O., Kocak, B. 2019. A Decentralised File Sharing Framework for Sensitive Data. In: Younas, M., Awan, I., Benbernou, S. (eds) Big Data Innovations and Applications. Innovate-Data 2019. Communications in Computer and Information Science, vol 1054. Springer, Cham.
18. -Y. Chang, C. -F. Chou and K. -C. Chen, "Content-Priority-Aware Chunk Scheduling Over Swarm-Based P2P Live Streaming System: From Theoretical Analysis to Practical Design," in IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 4, no. 1, pp. 57-69, March 2014.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 3, March 2023 ||

| DOI:10.15680/IJIRSET.2023.1203104 |

19. Y. Gotoh, T. Yoshihisa, H. Taniguchi and M. Kanazawa, "A method to reduce interruption time considering several clients on broadcast and communications integration environments," 12th IFIP/IEEE International Symposium on Integrated Network Management IM 2011 and Workshops, 2011, pp. 989-996.
20. J. Feng, "A Research on Scheduling Strategy in Peer-to-Peer Streaming Media," 2009 Pacific-Asia Conference on Circuits, Communications and Systems, 2009, pp. 439-442.
21. L. -J. Zhou, Z. -T. Li and B. Liu, "The Proposition and Certification of P2P Media Streaming Traffic Features," 2008 Fourth International Conference on Natural Computation, 2008, pp. 281-285.
22. Jiang, Zeng Liang, et al. “A Media Streaming Data Scheduling Algorithm Based on P2P.” Applied Mechanics and Materials, vol. 552, Trans Tech Publications, Ltd., June 2014, pp. 377–380.
23. Q. Lian, Z. Zhang, M. Yang, B. Y. Zhao, Y. Dai and X. Li, "An Empirical Study of Collusion Behavior in the Maze P2P File-Sharing System," 27th International Conference on Distributed Computing Systems ICDCS 2007, Toronto, ON, Canada, 2007, pp. 56-56, doi: 10.1109/ICDCS.2007.84.
24. N. Magharei and R. Rejaie, "PRIME: Peer-to-Peer Receiver-drIven MEsh-Based Streaming," IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications, Anchorage, AK, USA, 2007, pp. 1415-1423, doi: 10.1109/INFCOM.2007.167.

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.118

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

Scan to save the contact details