



Optimizing Structured Peer-to-Peer Networks Using Load Separation for Efficient Data Replication

¹S Venu Gopal, ²MDN Akash, ³Guntupalli Greeshma, ⁴K Ravikanth

¹Assistant Professor, ²PG Research Scholar, ³UG Research Scholar, ⁴Assistant Professor

^{1,2,4}Department of CSE, ³Department of CSE (Data Science)

¹Vardhaman College of Engineering, Hyderabad, India

²Aurora University, Hyderabad, India

³Vardhaman College of Engineering, Hyderabad, India

⁴Aurora University, Hyderabad, India

Abstract: Peer-to-Peer (P2P) networks enable decentralized communication where each node functions both as a client and a server, facilitating direct data exchange without the need for a central coordinator. However, as the number of nodes increases, the uneven distribution of load among peers often leads to congestion, packet loss, and reduced system efficiency. To address this issue, this paper proposes the Load Separation Method (LSM) for structured P2P overlay networks. The proposed approach identifies and isolates heavily loaded peers from lightly loaded ones, thereby improving network reliability and optimizing data transfer performance. Simulations conducted using the NS2 network simulator demonstrate that applying LSM effectively minimizes packet loss and enhances throughput by balancing the network load. The study further discusses the significance of load-aware routing strategies in achieving scalability and stability in structured P2P systems.

Index Terms - Peer-to-Peer Networks, Structured Overlay, Load Balancing, Packet Loss, NS2 Simulation, Load Separation Method

I. INTRODUCTION

Peer-to-Peer (P2P) networking represents a distributed communication model in which nodes, or peers, share resources such as bandwidth, storage, and data without depending on a centralized server. Each peer plays a dual role as both a client and a server, promoting scalability, fault tolerance, and decentralized control. P2P systems are widely utilized in applications such as file sharing, distributed storage, and content delivery networks (CDNs). P2P architectures are broadly categorized into structured and unstructured networks. In structured P2P systems, node connections and data placement follow deterministic rules, often implemented through Distributed Hash Tables (DHTs) such as Chord, Pastry, or CAN. This ensures efficient lookup and routing. In contrast, unstructured networks rely on random connections and flooding mechanisms for resource discovery, which can lead to high overhead and inefficiency as the network scales.

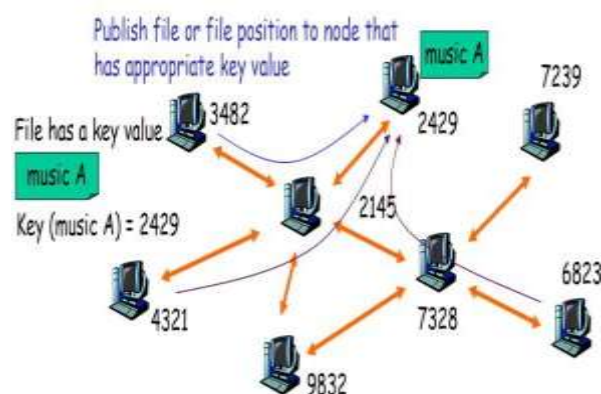


Fig 1: Example for Structured P2P representation



Fig 2: Example for Unstructured P2P representation

A major challenge in both architectures, particularly in structured overlays is load imbalance, where certain nodes become heavily burdened with data or communication requests. Such imbalances result in packet loss, delays, and reduced throughput. To mitigate these issues, this study introduces the Load Separation Method (LSM), a lightweight yet effective mechanism for identifying and isolating heavily loaded peers from lightly loaded ones. The objective is to enhance reliability and optimize network performance by minimizing traffic congestion and avoiding overloaded communication paths. In a P2P environment each of the node has same vantage, capabilities and responsibilities. This type of environment is different from client/server environment. In c/s architectures, computers store info and can access to resources, which other systems in network can access through the network. In c/s not required for full time administration. Every individual user act as administrator to his system. User can easily control his machine and resources. Building and maintaining of this type of structured P2P network is very less cost. Here I am taking the one peer to peer network to show the simulation of load separation method.

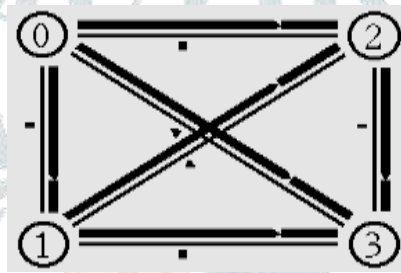


Fig 3: Example for Simulation

II. SIMULATION AND RESULTS

In this simulation, the average number of nodes (Avgn) represents the total number of active peers participating in the network. The load at each node (In) is calculated based on the total packets transmitted and received, reflecting the node's communication activity. The average load (avgIn) denotes the mean value of all individual node loads, providing an overall measure of network balance. Each node is characterized by two key parameters: from_node, representing the number of packets sent, and end_node, representing the number of packets received. By analysing these parameters, the simulation evaluates load distribution, identifies overloaded nodes, and measures packet loss across various network events such as transmission, queuing, and link failure.

Table 1: Node-to-Node Packet Flow and Transmission Statistics in the P2P Network

From Node	To Node	No. of Pkts			
		Dropped (d)	In Queue(+)	Dequeue (-)	Sent
0	1	45	29306	29306	0
0	2	28	14688	14688	17
1	2	34	4684	4679	34

1	3	45	29511	29511	0
2	3	45	29511	29511	0
2	0	42	14671	14671	3
3	2	45	29511	29511	0
3	1	45	29511	29511	0

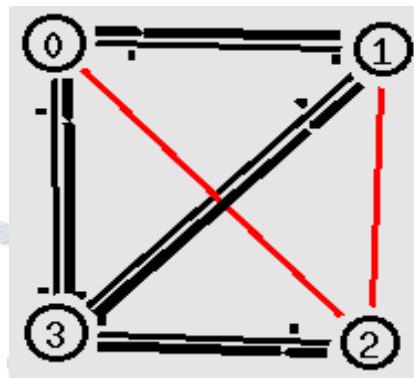


Fig 4: 2 links fail

In this simulation, each packet is considered to have a size of 1024 bits. To analyse network reliability and load behaviour, link failures were intentionally introduced between node(1)–node(2) and node(2)–node(1), simulating real-world packet drop conditions. Following the simulation, the data presented in Table 1 was obtained, showing the number of packets sent, queued, dequeued, and dropped between nodes. Based on these results, simulation graphs were generated to illustrate the relationship between packets sent and packets dropped across all nodes in the network.

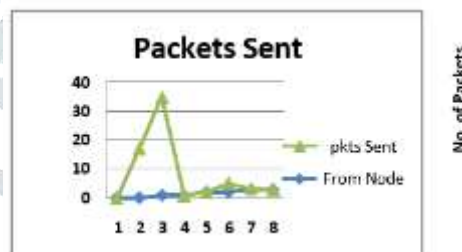


Fig 5: Number of Packets Sent

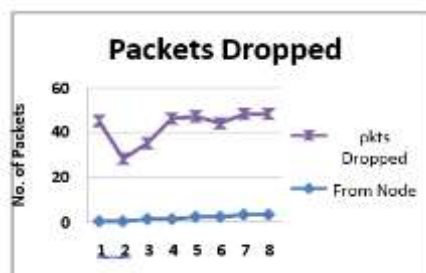


Fig 6: Number of Packets Dropped

Based on the trace file analysis, it is observed that the packet drop rate is significantly high due to network congestion along certain links. To mitigate this issue, the proposed method identifies heavy traffic links and the corresponding overloaded nodes within the network. These high-traffic connections are then isolated from the main data transmission paths, and heavily loaded nodes are separated from lightly loaded ones. This segregation helps in redistributing the traffic load more evenly across the network, thereby minimizing packet loss and improving overall communication efficiency.

Pseudocode for load classification in a peer-to-peer (P2P) network:

n = number requests, enqueue, dequeue

for all MyPeers

if (enqueue > max)

queryHit(n) is very high, heavy load at node

if (enqueue < min)

queueHits is low, queryHit(n) is low, low load at node in the network

III. SUMMARY

In this paper the main focus is replication of packets. while replication data may lose due to the above-mentioned reason. For that purpose, we will be applying load separation method by finding heavy loaded nodes from low loaded peers.

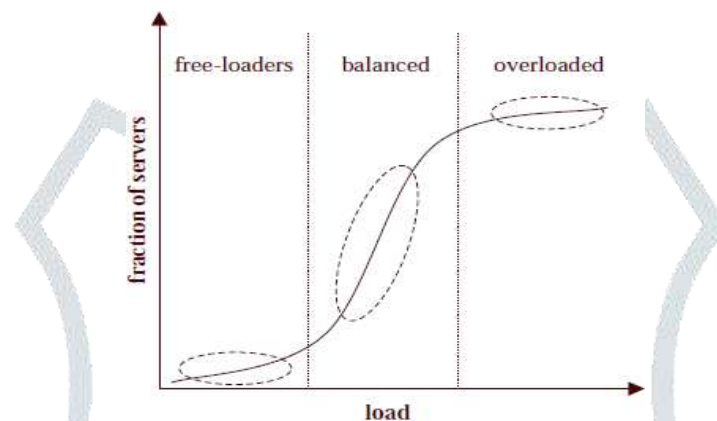


Fig 7: Load Separation Visualization

Here heavy loaded peer nodes are grouped and low loaded peer nodes are grouped. so, separating low loaded peer nodes from heavy loaded peers. Figure 7 illustrates the overall outcome of the proposed method, showing the separation of free-loaded, balanced, and overloaded nodes within the structured peer-to-peer network. This classification enables efficient routing decisions, allowing data to be transmitted and received with minimal loss while maintaining balanced network utilization.

IV. CONCLUSION

This paper introduced a novel Load Separation Method (LSM) to enhance the efficiency and reliability of structured peer-to-peer (P2P) overlay networks. By classifying peers according to their current load and isolating heavily loaded nodes, LSM reduces packet loss, improves throughput, and ensures uniform load distribution. Simulation results demonstrate significant gains in packet delivery ratio and latency reduction. The method's lightweight and scalable design allows seamless integration with existing overlay protocols such as Chord and Pastry. Future work will focus on extending LSM to dynamic and large-scale environments through machine learning based load prediction and adaptive routing, further improving fault tolerance and self-optimization in decentralized networks.

REFERENCES

- [1] D. R. Karger, E. Lehman, F. T. Leighton, R. Panigrahy, M. S. Levine, and D. Lewin, "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web," *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, pp. 654–663, May 1997.
- [2] P. Keleher, S. Bhattacharjee, and B. Silaghi, "Are virtualized overlay networks too much of a good thing?" *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*, 2002.
- [3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content addressable network," *Proceedings of the ACM SIGCOMM 2001 Technical Conference*, 2001.

- [4] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," *Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*, 2001.
- [5] Schiller, Eryk & Felber, Pascal & Kropf, Peter & Serbu, Sabina. (2014). Survey on Load Balancing in Peer-to-Peer Distributed Hash Tables. *IEEE Communications Surveys & Tutorials*. 16. 473-492. 10.1109/SURV.2013.060313.00157.
- [6] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "A measurement study of peer-to-peer file sharing systems," *Proceedings of Multimedia Computing and Networking 2002 (MMCN '02)*, San Jose, CA, USA, Jan. 2002.
- [7] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications," *Proceedings of the ACM SIGCOMM '01 Conference*, San Diego, CA, USA, Aug. 2001.
- [8] B. Y. Zhao, K. Kubiatowicz, and A. Joseph, "Tapestry: An infrastructure for fault-resilient wide-area location and routing," *University of California at Berkeley Technical Report*, Tech. Rep. UCB/CSD-01-1141, 2001.
- [9] A. Rowstron and P. Druschel, "Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility," *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP'01)*, 2001.
- [10] Farahi, Rasoul. "A comprehensive overview of load balancing methods in software-defined networks." *Discover Internet of Things* 5.1 (2025): 6.
- [11] S Venu Gopal, N Sambasiva Rao. "Dynamic Sharing of Files from Disconnected Nodes in Peer to Peer Systems." (ICEEOT) – 2016, *IEEE Digital Explorer*.
- [12] K Kalpana, Natesh S Rao, Sadanand S Venu Gopal "Effect of addition of dexamethasone to ropivacaine in supraclavicular brachial plexus block: A prospective, randomized, double-blind study." *J Evid Based Med Healthc*, 2015, Vol.2, pg:5016-22.

