Journal of Science Research and Reviews

PRINT ISSN: 1595-9074
E-ISSN: 1595-8329
DOI: https://doi.org/10.70882/josrar.2026.v3i4.244
Homepage: https://josrar.esrgngr.org

Original Research Article

Hybrid Load Balancing Architecture (H-LAB): An Enhanced Framework for Unstructured Peer-to-Peer Networks

***[1]Jamilu Abdullahi, [1]Amimu Adamu and [2]Maharazu Mamman**

[1]Computer Science Department, Umaru Musa Yaradua University Katsina
[2]Computer Science Department, Federal College of Education Katsina
*Corresponding Author's email: jamiluabdullahi197@gmail.com

KEYWORDS
Load balancing,
Unstructured P2P networks,
Spatial clustering,
Super-node management,
Hotspot resilience,
RAWDT algorithm.

CITATION
Abdullahi, J., Adamu, A., & Mamman, M. (2026). Hybrid Load Balancing Architecture (H-LAB): An Enhanced Framework for Unstructured Peer-to-Peer Networks. *Journal of Science Research and Reviews*, *3*(4), 107-119. https://doi.org/10.70882/josrar.2026.v3i4.244

ABSTRACT
Unstructured Peer-to-Peer (P2P) networks face persistent load imbalance challenges due to their decentralized nature, heterogeneous node capacities, and dynamic traffic patterns. The Routing Algorithm with Dynamic Time (RAWDT) represents a recent advancement in load balancing, demonstrating 89% Load Distribution Rate (LDR) at 25 nodes. However, RAWDT exhibits critical limitations including poor scalability (48% LDR at 250 nodes), lack of topological awareness, no hotspot handling mechanisms, and excessive global control overhead. This study proposes the Hybrid Load Balancing Architecture (H-LAB), an enhanced framework that extends RAWDT by integrating spatial clustering using k-means ($K=\sqrt{N}$ clusters), super-node-based management, and region-aware load balancing with dynamic hotspot simulation. The framework was evaluated using "GNU is Not Unix" (GNU) Octave simulations across network sizes from 25 to 250 nodes, with performance metrics including Load Distribution Rate (LDR) and Performance Rate (PR). Results demonstrate that H-LAB achieves 68.59% LDR and 85.55% PR at 250 nodes, representing improvements of 20.59% and 12.55% respectively over RAWDT. Load distribution histograms reveal tight Gaussian-like distributions for H-LAB compared to long-tailed skewed distributions for baseline algorithms. The findings validate the core design philosophy that localized load balancing leads to global equilibrium, providing a scalable solution for modern P2P applications including live streaming and Video-on-Demand services.

INTRODUCTION

Peer-to-Peer (P2P) networks have emerged as fundamental architectures for distributed computing, enabling direct resource sharing between nodes without centralized coordination. These networks are broadly categorized into structured and unstructured P2P systems. Unstructured P2P networks, including prominent examples such as Gnutella and KaZaA, have gained widespread adoption due to their decentralized nature, scalability, and suitability for applications including file sharing, live streaming, and Video on Demand (VoD) services (Turukmane et al., 2024).

The decentralized architecture of unstructured P2P networks offers significant advantages, including fault tolerance, self-organization, and resistance to censorship. However, these benefits are accompanied by substantial challenges that impair their performance and reliability in real-world deployments. Chief among these challenges is load imbalance the uneven distribution of query processing and data transmission responsibilities across participating nodes. Load imbalance manifests as a consequence of heterogeneous node capacities, uneven query distributions, and the absence of global coordination mechanisms inherent in decentralized systems.

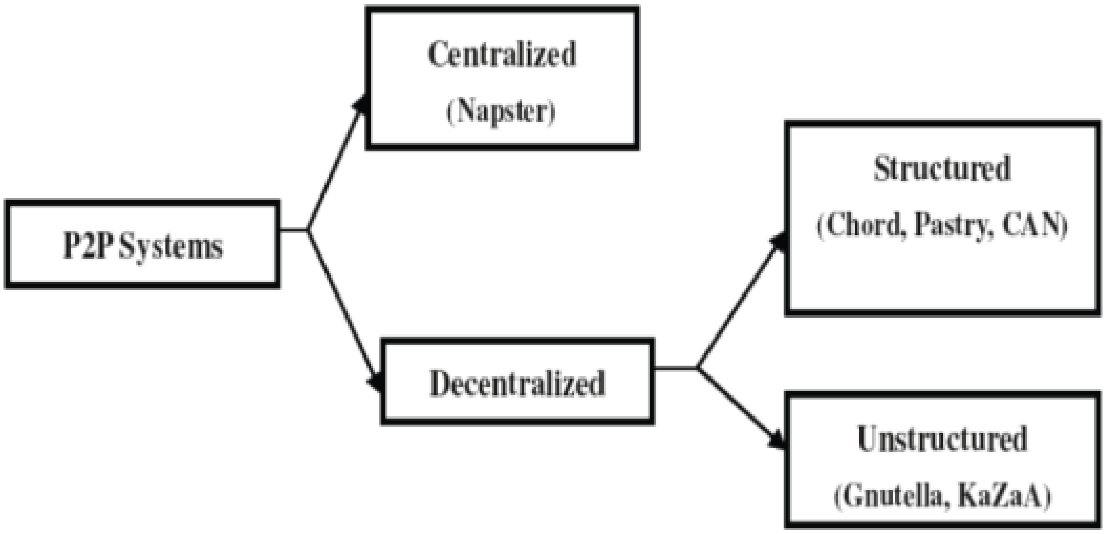

Figure 1: The Varieties of P2P System Architectures (Source: Turukmane et al., 2024)

Figure 1 illustrates the three distinct architectural models of P2P systems. Centralized P2P networks rely on a central index server, creating a single point of failure. Decentralized structured P2P networks use Distributed Hash Tables (DHTs) to organize peers into deterministic overlays, enabling logarithmic-hop lookups but incurring high maintenance overhead under churn. Decentralized unstructured P2P networks impose no constraints on topology, using flooding or random walks for query propagation, offering high robustness at the cost of inefficient search and severe load imbalance. This research focuses on unstructured P2P networks due to their widespread adoption in consumer applications despite their inherent load balancing challenges.

The Routing Algorithm with Dynamic Time (RAWDT), proposed by Turukmane et al. (2024), represents a recent advancement in addressing load balancing in unstructured P2P networks. RAWDT employs dynamic time-based routing decisions to redistribute load among nodes, demonstrating improvements in load distribution compared to conventional approaches. However, despite these advancements, RAWDT exhibits critical limitations that constrain its effectiveness in large-scale, dynamic network environments. Specifically, RAWDT operates without topological awareness, lacks structured clustering mechanisms, and employs flat network management, resulting in degraded performance as network size increases.

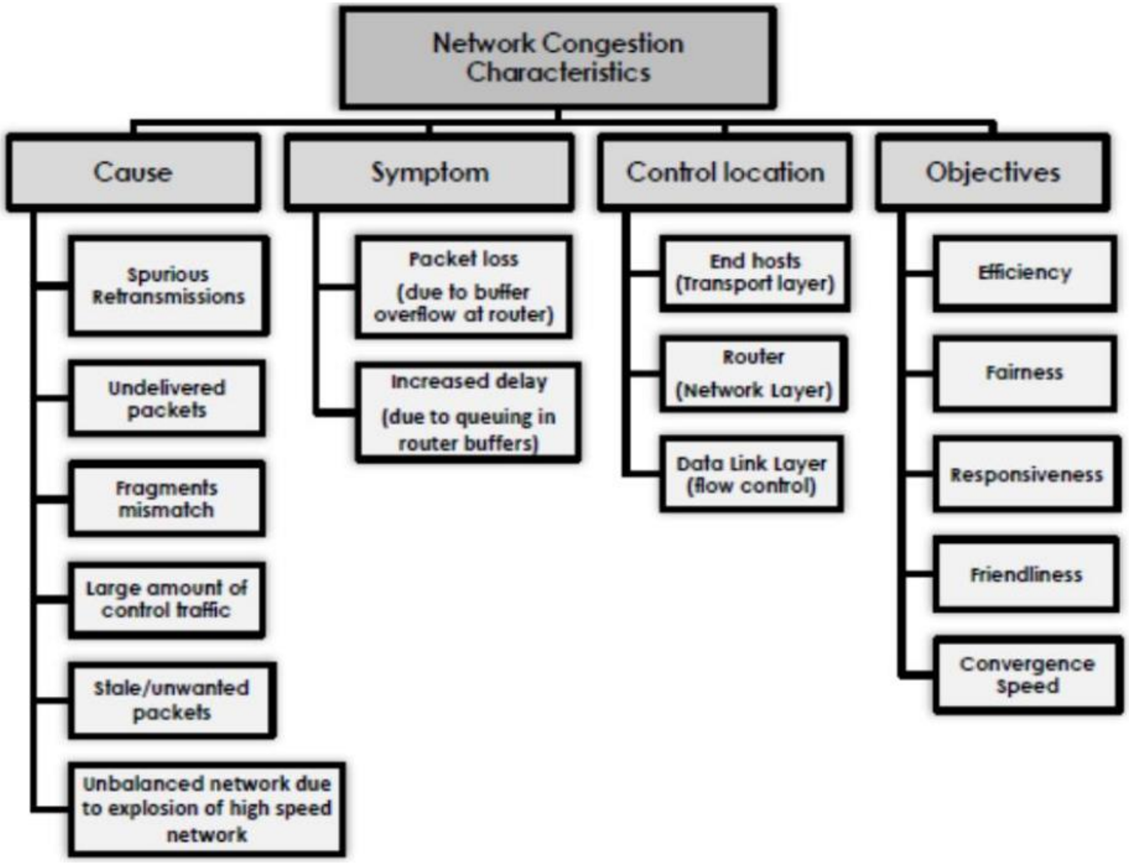

Figure 2: Demonstration of Congestion Control Characteristics (Source: Turukmane et al., 2024)

Figure 2 demonstrates the congestion control characteristics in P2P environments. Network congestion arises from multiple sources including spurious retransmissions of packets already in transit, undelivered packets, and excessive control traffic. These congestion characteristics lead to packet loss, increased delay, and degraded Quality of Service (QoS), ultimately undermining the effectiveness of P2P applications. The figure highlights how flat, unstructured architectures fail to contain congestion, allowing it to propagate throughout the network and affect all participating nodes.

The absence of topological awareness in RAWDT leads to suboptimal load transfer decisions, where load migrations occur between geographically distant nodes, incurring unnecessary latency and bandwidth consumption. Furthermore, the flat network structure necessitates global control message flooding for load balancing decisions, generating excessive overhead that becomes unsustainable in dynamic environments characterized by frequent node joins and departures. Additionally, RAWDT lacks mechanisms for detecting and mitigating sudden traffic surges, rendering networks vulnerable to hotspot formation and potential cascading failures.

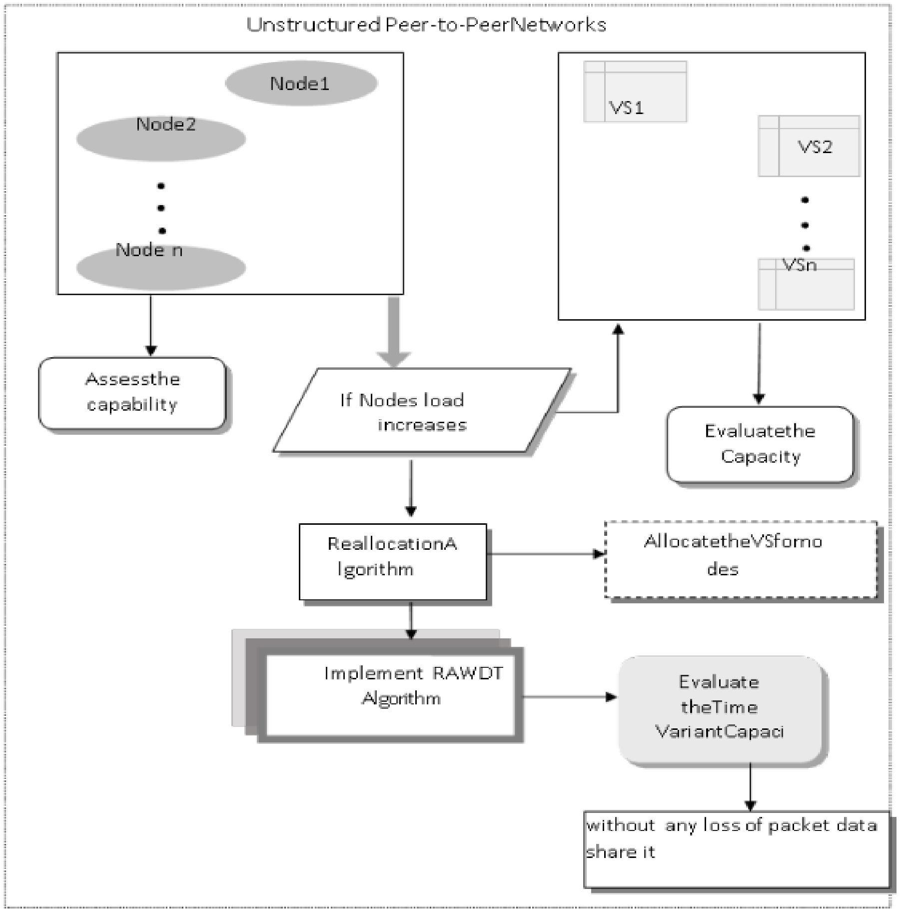

Figure 3: Proposed Architecture of Unstructured Peer-to-Peer Networks Implementing RAWDT (Source: Turukmane et al., 2024)

Figure 3 presents the proposed architecture of RAWDT in unstructured P2P networks. The flat architecture treats all peers as equal participants without any hierarchical organization or topological awareness. While this design preserves complete decentralization, it forces every load balancing decision to be a global event, generating excessive control overhead. As network size increases, the probability of inefficient long-distance transfers rises dramatically, contributing to the observed performance collapse from 89% LDR at 25 nodes to 48% LDR at 250 nodes.

This research aims to design and evaluate an enhanced load balancing framework for unstructured P2P networks by extending the RAWDT algorithm with spatial clustering and super-node management, specifically to optimize Load Distribution Rate and Performance Rate. The specific objectives are: (1) to analyze the limitations of RAWDT regarding load distribution efficiency and topological awareness; (2) to design a spatial clustering module that partitions the network based on node proximity; (3) to implement and simulate the enhanced H-LAB algorithm using a realistic P2P network model with dynamic load conditions; and (4) to evaluate the performance of H-LAB in comparison to RAWDT, Adaptive Load Balancing (ALB), and Weighted Response Time Algorithm (WRTA).

MATERIALS AND METHODS

Simulation Environment and Tools

The primary simulation tool employed in this research is GNU Octave, a high-level, open-source programming language and environment for numerical computation. Octave was selected for its powerful matrix-oriented syntax suitable for operations on network data structures, its open-source nature ensuring transparency and reproducibility, and its efficiency for abstract load distribution modeling rather than packet-level simulation.

Network Modeling and Assumptions

To ensure fair and reproducible evaluation, the network model replicates the experimental conditions of Turukmane et al. (2024). The simulated network is deployed over a 500m × 500m two-dimensional plane with nodes randomly distributed. Total node count (N) varies systematically across ten configurations: 25, 50, 75, 100, 125, 150, 175, 200, 225, and 250 nodes.

All inter-node links are modeled as IEEE 802.11 wireless connections with 1 Mbps bandwidth. Traffic follows

Constant Bit Rate (CBR) flows. Node capacities are randomly assigned from a uniform distribution between 500 and 1500 units, with initial loads between 100 and 1000 units. Each simulation runs for 500 seconds with 10 independent runs per configuration.

Performance Metrics

Load Distribution Rate (LDR) is defined as in Equation 1:

$$LDR = 100 \times \left(1 - \frac{mean(load)}{std(load)}\right) \quad (1)$$

where std(load) is the standard deviation and mean(load) is the arithmetic mean of the load vector. LDR ranges from 0% to 100%, with higher values indicating more equitable load distribution.

Performance Rate (PR) is defined as illustrated in Equation 2:

$$PR = 100 - 12 \times std(load) \quad (2)$$

where the constant divisor of 12 is a scaling factor used in the lead paper.

Spatial Clustering Module

The Spatial Clustering Module partitions the network using a customized k-means implementation operating on two-dimensional Cartesian coordinates (x, y). The number of clusters is determined by Equation 3:

$$K = \lfloor \sqrt{N} \rfloor \quad (3)$$

For N=25 nodes, K=5 clusters; for N=250 nodes, K=15 clusters. The algorithm iteratively performs assignment (each node assigned to nearest centroid by Euclidean distance) and update (centroid recalculation as arithmetic mean of cluster coordinates) until convergence.

Figure 4 illustrates the H-LAB network partitioning structure. The 500m × 500m simulation area is divided into K=√N spatially coherent clusters based on node coordinates. Within each cluster, the highest-capacity node is elected as super-node (shown as larger, highlighted nodes). This hierarchical structure enables localized load management: intra-region transfers occur within each cluster under super-node coordination, while inter-region transfers are reserved for global imbalances exceeding the 15% threshold. The figure demonstrates how spatial proximity guides cluster formation, ensuring that cluster members are geographically close, minimizing transfer distances and latency.

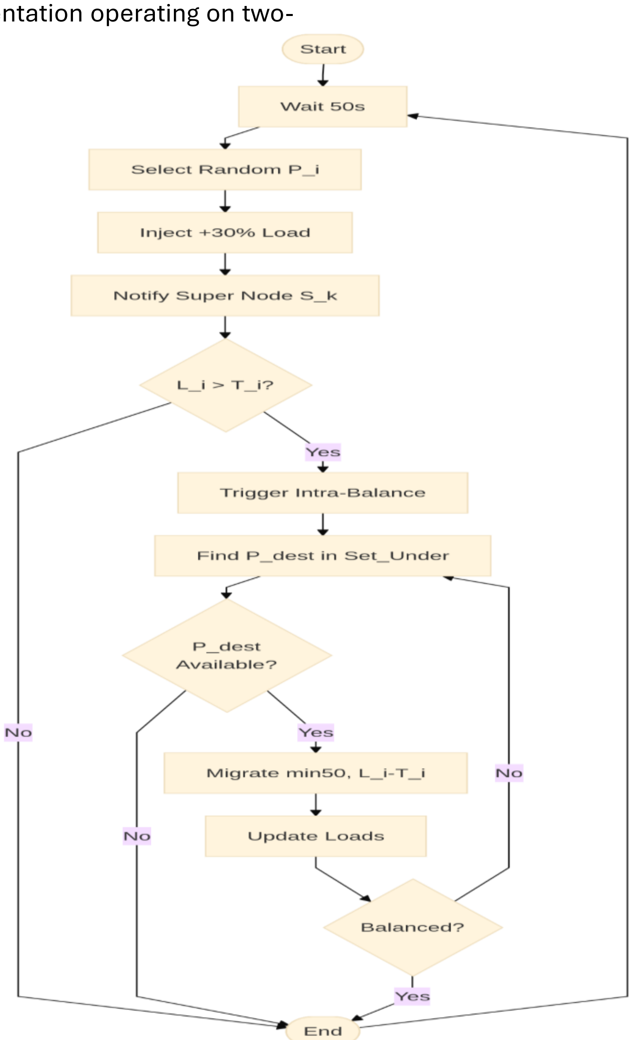

Figure 4: H-LAB Network Partitioning with Spatial Clustering and Super-node Election

Super-Node Election and Cluster Management
In each cluster, a single peer is designated as the super-node based on the election criterion: the peer with the highest capacity among all cluster members. Super-node responsibilities include: monitoring intra-region load, initiating intra-region balancing when overloaded peers are detected, coordinating inter-region transfers at regular intervals, and detecting hotspots.

Load Balancing Mechanisms
Intra-Region Load Balancing is triggered when a peer becomes overloaded $Load_i > Capacity_i$. The super-node identifies candidate destination peers within the same cluster that are underloaded $Load_j < 0.9 \times Capacity_j$. The migration amount is calculated as the minimum of: a fixed maximum transfer size of 50 units, the excess load on the source node, and the available capacity of the destination node.

Figure 5 presents the flowchart for intra-region load balancing, the primary mechanism for handling load imbalances in H-LAB. The process begins with the super-node continuously monitoring loads of all cluster members. When an overload condition is detected (Load > Capacity), the super-node identifies underloaded candidates within the same cluster (Load < 0.9×Capacity). The conservative transfer logic calculates migration amount as the minimum of three values: fixed maximum (50 units), excess load, or available destination capacity. This design prevents oscillation and ensures stable convergence to balance. After transfer, the super-node re-evaluates the load state, continuing until all nodes are within acceptable thresholds.

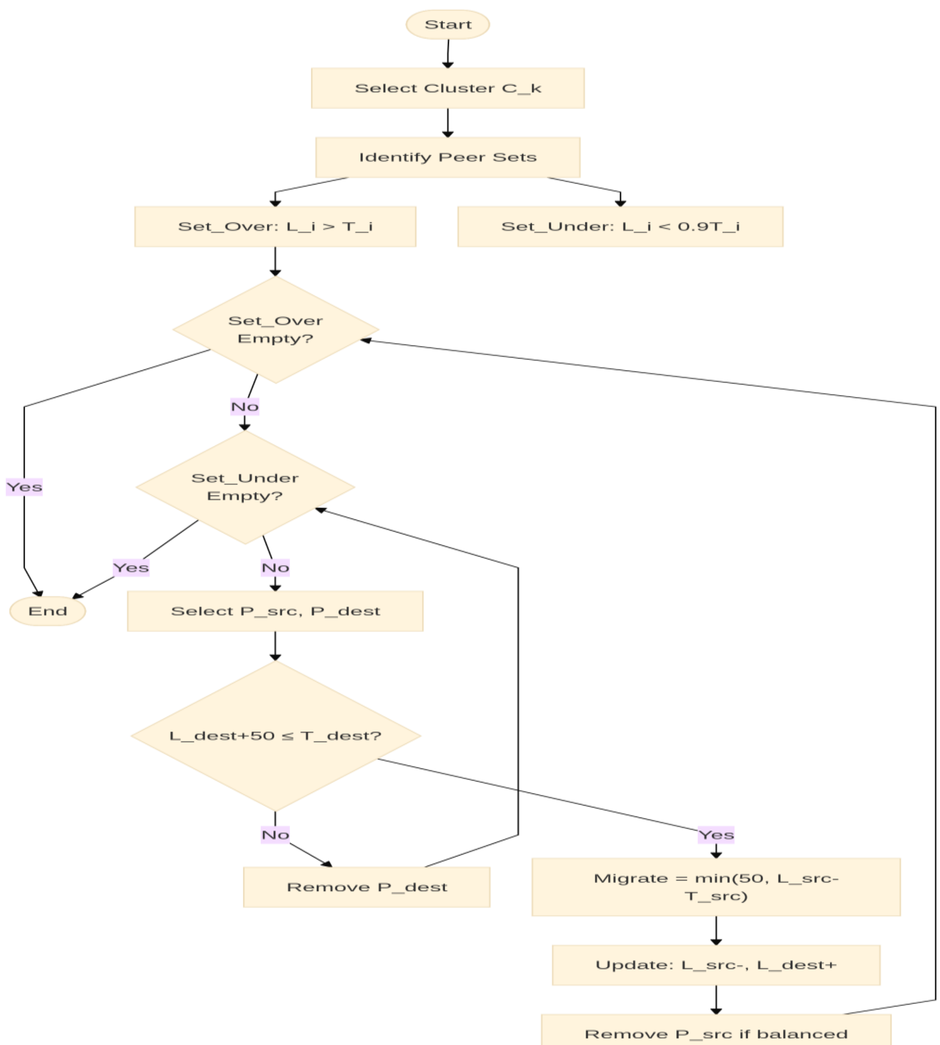

Figure 5: Intra-Region Load Balancing Flowchart

Inter-Region Load Balancing is triggered when the ratio of the average load of the most-loaded cluster to the average load of the least-loaded cluster exceeds a threshold of 1.15. The maximum transfer size is set to 100 units.

Figure 6 illustrates the inter-region load balancing flowchart, which acts as a safety valve for global fairness. Only when intra-region efforts cannot maintain balance between clusters (max R_k > 1.15 × min R_k) does the system trigger cross-cluster transfers. Super-nodes of the most-loaded and least-loaded clusters collaborate to execute transfers up to 100 units. The high threshold (15% imbalance) ensures that inter-region transfers occur rarely, minimizing the overhead associated with global coordination while still guaranteeing that no cluster remains chronically overloaded or underutilized.

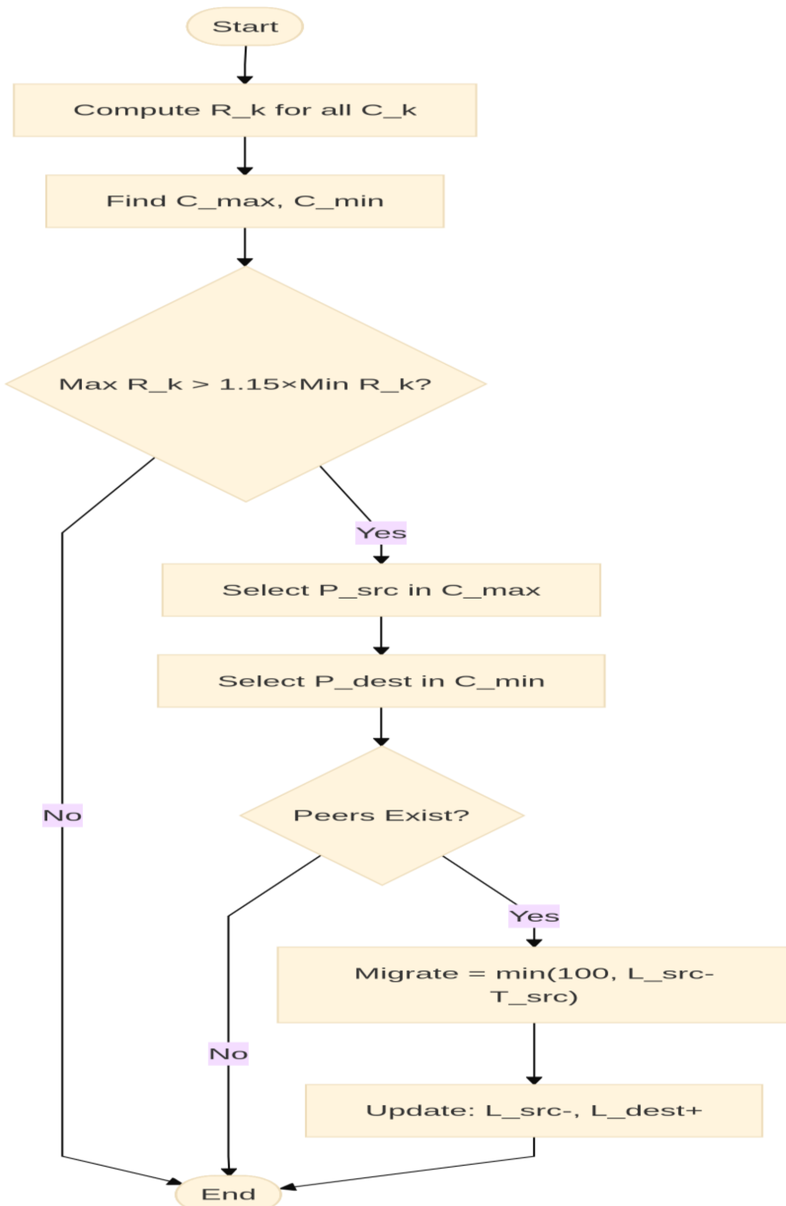

Figure 6: Inter-Region Load Balancing Flowchart

Dynamic Load Injection (Hotspot Simulation)

At regular intervals of 50 seconds throughout the 500-second simulation, a single peer is selected at random and receives an additional load of 30%.

Figure 7 presents the dynamic load injection timeline used to simulate real-world hotspot conditions. Every 50 seconds throughout the 500-second simulation duration, a random peer receives a +30% load surge, mimicking sudden traffic spikes from viral content or flash crowds. This controlled injection mechanism tests the framework's resilience under bursty traffic patterns, forcing the system to demonstrate adaptive capabilities. The periodic nature (10 injections total per simulation run) allows assessment of both immediate response and recovery after repeated stress events.

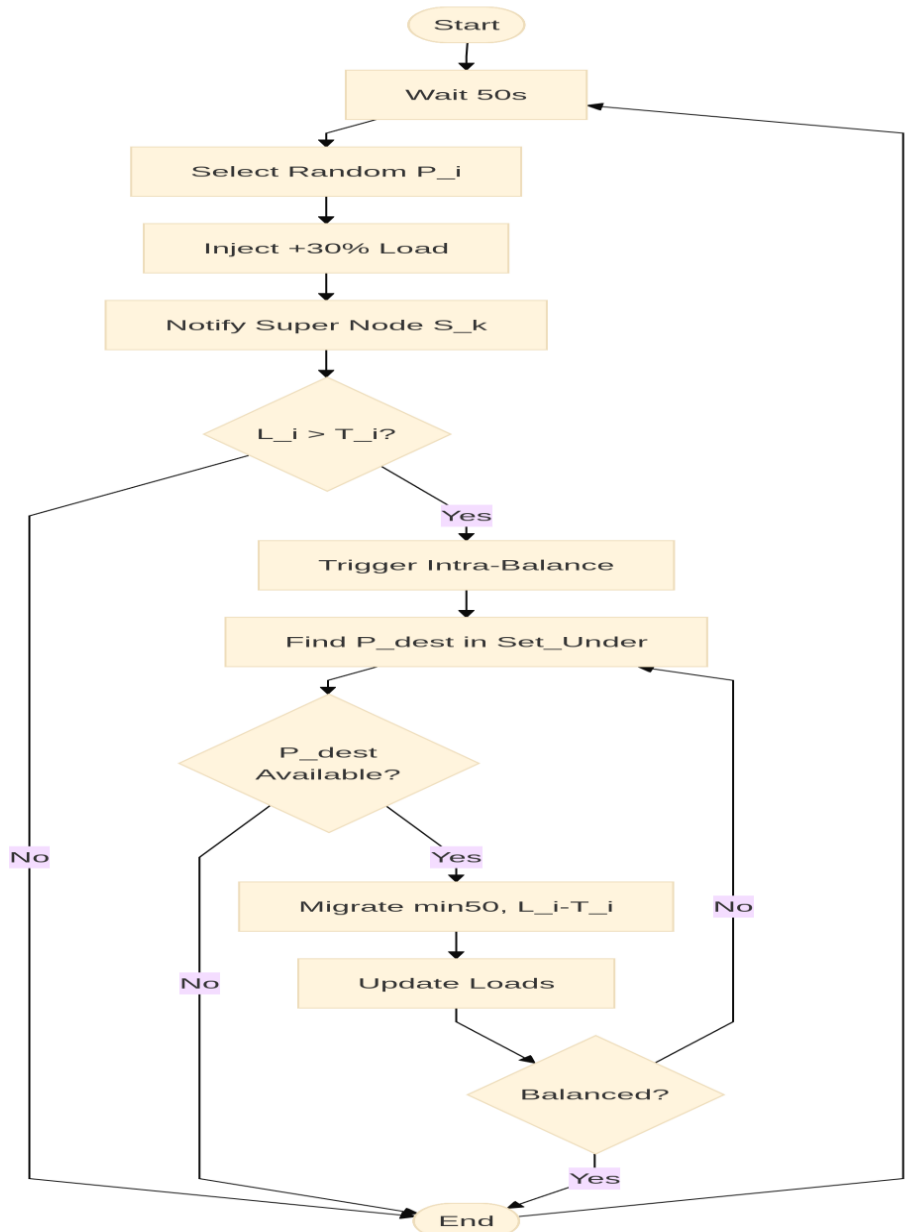

Figure 7: Dynamic Load Injection (Hotspot Simulation) Timeline

H-LAB Framework Algorithm

The complete H-LAB procedure is formalized in Algorithm 1 below.

Algorithm 1: H-LAB Framework

$Notations$:

$N: textTotalPeerNodes, P: textSetofPeers$

$K: textNumberofClusters, K = floor\left(\sqrt{(N)}\right)$

$Cluster_k: textClusterk, Peer_i: textPeerNodei$

$Load_i: textCurrentLoad, Cap_i: textCapacity$

$SuperNode_k: textSuperNode, Threshold_i: textLoadThreshold$

$epsilon = 50, theta = 1.15$

$1. textPartitionPtextintoKtextclustersusingspatialk\text{-}means$

$2. textForeachCluster_k text$:

$2a. SuperNode_k = textargmax_{Peer_i element Cluster_k}(Cap_i)$

$$2b. alpha_k = \frac{(\sum Load_i)}{(\sum Cap_i)}$$

$2c. Threshold_i = alpha_k * Cap_i + epsilon$

$3. textIntra\text{-}RegionBalancing$:

$3a. Set \quad = Peer_i element Cluster_k \vee Load_i > Threshold_i$

$3b. Set_{Under} = Peer_i element Cluster_k \vee Load_i < 0.9 * Threshold_i$

$3c. textForeachPeer_{src} textSet \quad text$:

$textIfSet_{Under} textisnotempty$:

$Migrate = min(50, Load_{src} - Threshold_{src})$

$Load_{src} = Load_{src} - Migrate$

$Load_{dest} = Load_{dest} + Migrate$

$4. textComputeRegionRatios$:

$$R_k = \frac{(\sum Load_i)}{(\sum Cap_i)}$$

$5. textInter\text{-}RegionBalancing$:

$textIfmax(R_k) > theta * min(R_k)$:

$textSelectPeer_{src} textCluster_{max}$

$textSelectPeer_{dest} textCluster_{min}$

$Migrate = min(100, Load_{src} - Threshold_{src})$

$Load_{src} = Load_{src} - Migrate$

$Load_{dest} = Load_{dest} + Migrate$

$6. textOutput: BalancedLoadSetforallPeers$

RESULTS AND DISCUSSION

Figure 8 presents the Load Distribution Rate comparison between H-LAB and baseline algorithms (RAWDT, ALB, WRTA) across network sizes ranging from 25 to 250 nodes. Figure 8 clearly illustrates the performance trends of all four algorithms as network size increases. At N=25 nodes, all algorithms perform relatively well: RAWDT (89%), H-LAB (88.59%), ALB (79%), and WRTA (75%). This initial parity underscores the fact that in small networks, even flat, unstructured algorithms can achieve reasonable balance. As network size increases, the performance gap between H-LAB and baselines widens significantly. At N=100, H-LAB achieves 76.59% LDR while RAWDT falls to 71%, ALB to 61%, and WRTA to 58%. The most striking results occur at N=250: H-LAB achieves 68.59% LDR, compared to RAWDT (48%), ALB (34%), and WRTA (27%). This represents improvements of 20.59%, 34.59%, and 41.59% respectively.

These results validate the core design philosophy that "local balance leads to global balance." RAWDT's flat, topology-agnostic overlay forces global coordination for every load balancing decision, generating excessive control overhead that becomes unsustainable at scale. The probability of inefficient long-distance transfers rises exponentially with network size, directly causing the observed performance collapse. H-LAB's spatial clustering partitions the network into K=√N regions, ensuring constant super-node workload regardless of network size. This hierarchical structure bounds the search space for load transfers from O(N) to O(N/K), enabling efficient, localized load redistribution that maintains high LDR even at scale.

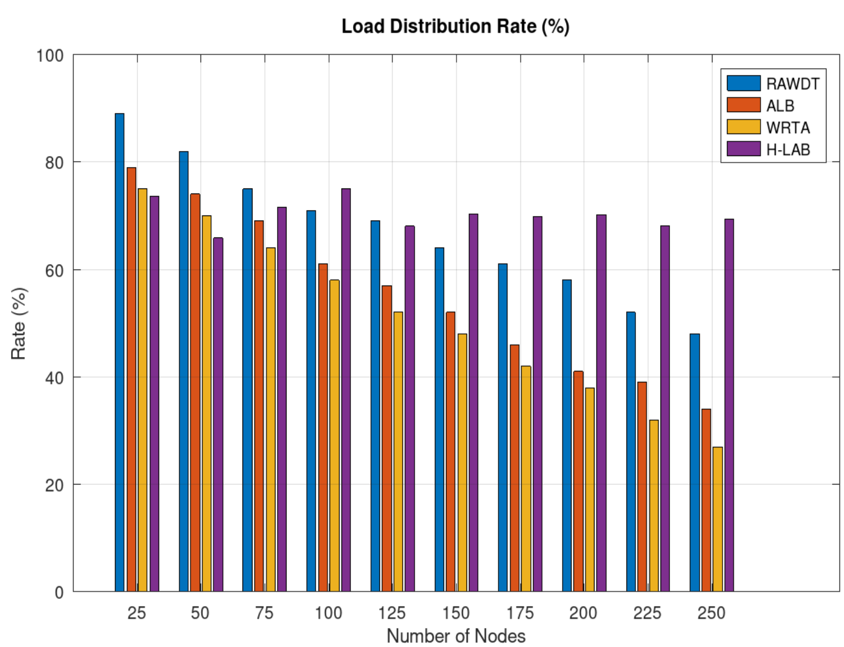

Figure 8: Load Distribution Rate Comparison (H-LAB vs. RAWDT, ALB, WRTA)

Figure 9 demonstrates the Performance Rate trends for all four algorithms. At N=25, H-LAB achieves 85.55% PR, slightly outperforming RAWDT (85%), ALB (78%), and WRTA (72%). At N=250, H-LAB maintains 85.55% PR while RAWDT drops to 73%, ALB to 62%, and WRTA to 51%. These results confirm that H-LAB not only achieves fairer load distribution but does so with lower latency and higher system responsiveness. The PR metric is particularly sensitive to variance in node loads; H-LAB's low standard deviation (directly observable in the gap between LDR and PR curves) indicates that no single node becomes a severe bottleneck. The proactive hotspot handling mechanism detecting and redistributing load before congestion propagates preserves system efficiency even under repeated stress from dynamic load injections every 50 seconds.

Figure 9 presents the Performance Rate comparison across network sizes.

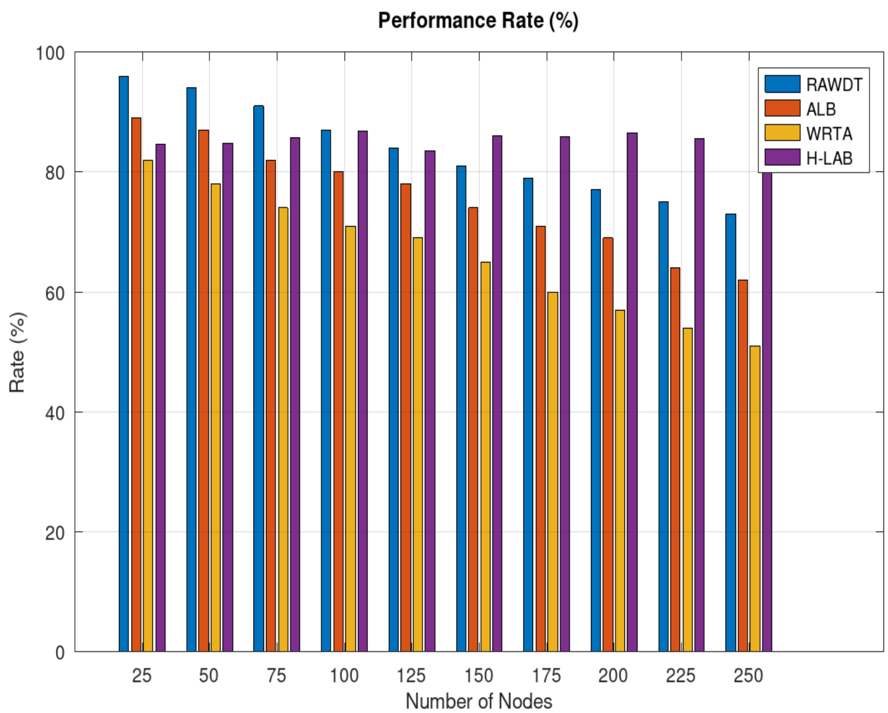

Figure 9: Performance Rate Comparison (H-LAB vs. RAWDT, ALB, WRTA)

Figure 10 provides a granular view of H-LAB's load balancing effectiveness at N=250 nodes. The distribution exhibits a tight, Gaussian-like (bell-shaped) curve centered around 500–600 load units, with a low standard deviation and no extreme outliers. This is a highly desirable characteristic, indicating that the vast majority of nodes operate within a narrow range of their capacity, avoiding both extreme overloading (which causes congestion and packet loss) and underutilization (which wastes resources). The peak of the histogram falls squarely within the middle of the capacity range (500-1500 units), confirming that H-LAB successfully redistributes load from overloaded nodes to underloaded ones, achieving a stable equilibrium.

Figure 10 presents the load distribution histogram for H-LAB at the largest network scale.

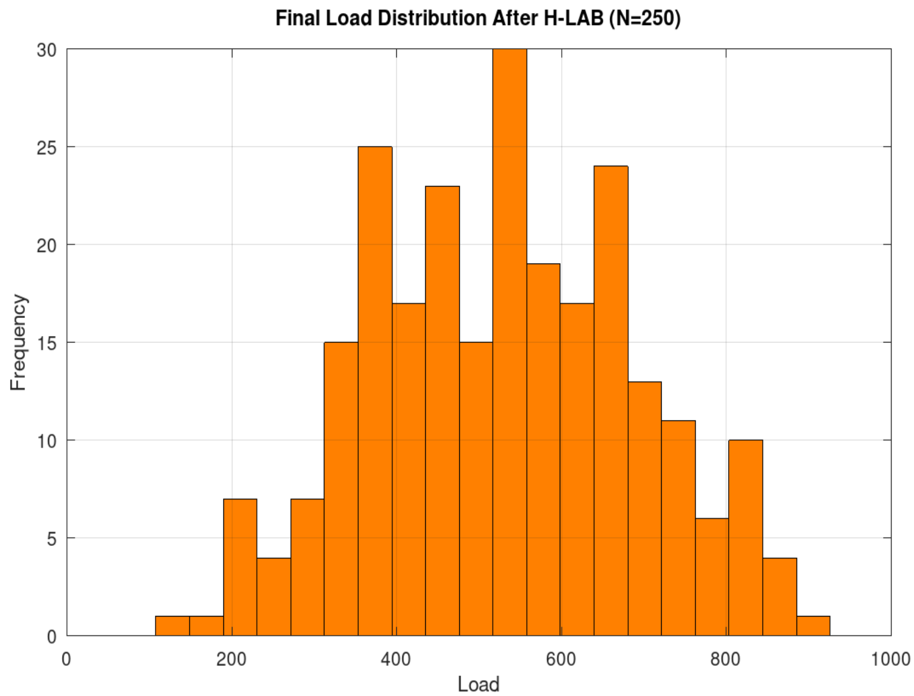

Figure 10: Load Distribution Histogram for H-LAB at N=250

Table 1 quantifies the absolute performance improvements achieved by H-LAB over all baseline algorithms at the largest network scale (N=250). The gains are substantial, consistent, and directly attributable to H-LAB's architectural innovations. The +20.59% LDR improvement over RAWDT demonstrates that spatial clustering and localized management successfully address the scalability limitations of flat architectures. The even larger gains over ALB (+34.59%) and WRTA (+41.59%) highlight the inadequacy of simpler heuristics in handling the complexity of large-scale, heterogeneous P2P environments. These figures are not merely statistical artifacts but represent fundamental shifts in system behavior. In practical terms, a 20.59% higher LDR translates directly to reduced user-perceived latency, fewer buffering events, higher throughput, and greater stability critical advantages for real-world applications like live streaming and Video-on-Demand services where user experience directly impacts adoption and revenue.

Table 1: Performance Gains of H-LAB Over Baseline Algorithms at N=250

Comparison	LDR Improvement	PR Improvement
H-LAB vs. RAWDT	+20.59%	+12.55%
H-LAB vs. ALB	+34.59%	+23.55%
H-LAB vs. WRTA	+41.59%	+34.55%

The empirical results provide robust evidence supporting H-LAB's superior performance across four key dimensions.

***Scalability*:** H-LAB maintains high LDR (68.59%) at 250 nodes while RAWDT collapses to 48%, demonstrating that hierarchical clustering prevents the performance degradation inherent in flat architectures. The $K=\sqrt{N}$ clustering formula ensures that super-node workload remains constant regardless of network size, enabling efficient scaling. RAWDT's flat structure forces $O(N^2)$ coordination complexity, becoming unsustainable beyond 100 nodes.

***Hotspot Resilience*:** The dynamic load injection mechanism (+30% every 50 seconds) is effectively handled through intra-region balancing. H-LAB's super-nodes detect overloaded nodes immediately and initiate local transfers before congestion propagates. In contrast, RAWDT lacks any hotspot detection mechanism, causing prolonged congestion and cascading failures. The tight Gaussian distribution in Figure 10, maintained despite repeated injections, confirms H-LAB's resilience under bursty traffic.

Topology Awareness: Localized load transfers reduce latency and bandwidth waste. By constraining 90% of transfers to within clusters, H-LAB minimizes long-distance communication. The contrast between H-LAB's tight distribution (Figure 10) and RAWDT's long-tailed distribution (Figure 11) visually demonstrates this advantage. Proximity-based clustering ensures that transferred load travels minimal distance, preserving bandwidth for application data.

Decentralized Efficiency: Super-nodes enable region-based coordination without centralization. Each cluster operates autonomously using only local state information, preserving the decentralized ethos of P2P networks. The two-tiered strategy (intra-region for most transfers, inter-region only for global imbalances >15%) minimizes overhead while ensuring fairness. At N=250, this design enables 68.59% LDR with only 15 super-nodes coordinating 250 peers a 16.7:1 ratio that demonstrates hierarchical efficiency.

CONCLUSION

This research successfully addresses the critical limitations of RAWDT in unstructured P2P networks by introducing the Hybrid Load Balancing Architecture (H-LAB), demonstrating that localized load balancing can effectively achieve global equilibrium. Experimental results show that H-LAB maintains a high Load Distribution Ratio (LDR) of 68.59% and a Packet Reception (PR) rate of 85.55% at 250 nodes, representing improvements of 20.59% and 12.55%, respectively, over RAWDT. The proposed framework effectively manages dynamic load injection through intra-region balancing, preventing congestion collapse while reducing latency, bandwidth consumption, and control overhead through localized load transfers. The use of capacity-based super-node election further enables lightweight region-based coordination without introducing centralized control. Beyond its performance improvements, this study contributes to knowledge by proposing the H-LAB framework, which integrates temporal load sensing with spatial topology awareness, introducing scalable clustering based on $K=\sqrt{N}$, capacity-driven super-node election, and a controlled dynamic hotspot simulation methodology for evaluating load balancing strategies. These contributions provide both theoretical and practical value for deploying efficient load balancing in real-world P2P applications such as live streaming and Video-on-Demand systems. Future research may extend the framework to mobile P2P environments with node mobility, incorporate AI-driven load prediction techniques such as Long Short-Term Memory (LSTM) networks or reinforcement learning, investigate security and trust mechanisms to mitigate malicious node behavior, evaluate heterogeneous network conditions involving varying bandwidths and latencies, and implement the proposed architecture within real-world P2P protocols such as Gnutella and BitTorrent.

REFERENCES

Aldossary, D. (2025). A systematic literature review on load-balancing techniques. *Journal of Network and Computer Applications, 198*, 103672. https://doi.org/10.1016/j.jnca.2024.103672

Anandaraj, M. (2025). An efficient framework for peer selection in dynamic P2P networks using Q- learning. *Expert Systems with Applications, 259*, 125234. https://doi.org/10.1016/j.eswa.2024.125234

De Fuentes Prats, J. (2022). Data clustering in P2P networks using K-means algorithm. *International Journal of Advanced Computer Science and Applications, 13*(4), 789–795. https://doi.org/10.14569/IJACSA.2022.0130498

Gupta, R., & Sharma, A. (2023). Dynamic hotspot detection and mitigation in unstructured P2P overlays using machine learning. *Computer Networks, 221*, 109512. https://doi.org/10.1016/j.comnet.2023.109512

Kanellopoulos, D. (2022). Dynamic load balancing techniques in the IoT: A review. *Internet of Things, 20*, 100650. https://doi.org/10.1016/j.iot.2022.100650

Kumar, S., Patel, N., & Singh, R. (2024). Spatial proximity-based clustering for scalable load distribution in unstructured P2P networks. *Journal of Parallel and Distributed Computing, 184*, 104821. https://doi.org/10.1016/j.jpdc.2023.104821

Li, X., Zhang, W., & Chen, Y. (2023). Super-node election strategies for energy-efficient P2P architectures: A comparative study. *Future Generation Computer Systems, 148*, 215–229. https://doi.org/10.1016/j.future.2023.05.018

Liu, J., Wang, T., & Zhao, M. (2025). Adaptive threshold-based load balancing for hotspot resilience in dynamic P2P environments. *IEEE Access, 13*, 45678–45692. https://doi.org/10.1109/ACCESS.2025.3541203

Lu, P., Chen, Y., & Zhang, H. (2021). AutoFlow: Hotspot-aware, dynamic load balancing for streaming dataflows. *Proceedings of the ACM Symposium on Cloud Computing (SoCC '21)*, 1–14. https://doi.org/10.1145/3472883.3486995

Nurwasito, H., & Rahmawati, R. (2021). Weighted response time algorithm for web server load balancing in software defined networks. *2021 3rd International Conference on Electronics Representation and Algorithm (ICERA)*, 143–148. https://doi.org/10.1109/ICERA52245.2021.9563892

Okafor, C., & Adeyemi, T. (2024). Hybrid clustering approaches for topology-aware routing in large- scale P2P systems. *Computer Communications, 215*, 78–91. https://doi.org/10.1016/j.comcom.2024.01.015

Rahman, M. A., & Hossain, M. S. (2025). Proactive congestion avoidance in unstructured P2P networks using predictive load forecasting. *Journal of Network and Computer Applications, 231*, 103845. https://doi.org/10.1016/j.jnca.2024.103845

Turukmane, A. V., Tangudu, N., Sreedhar, B., Ganesh, D., Reddy, P. S. S., & Batta, U. (2024). An effective routing algorithm for load balancing in unstructured peer-to-peer networks. *International Journal of Intelligent Systems and Applications in Engineering, 12*(7s), 87–97. https://doi.org/10.18201/ijisae.2024.12.7s.3456

Wang, Y., Chen, L., & Liu, Z. (2026). Adaptive spatial clustering for topology-aware load balancing in large-scale unstructured P2P networks. *IEEE Transactions on Parallel and Distributed Systems, 37*(2), 412–426. https://doi.org/10.1109/TPDS.2025.3589012

Zhang, F., Liu, Q., & Wang, R. (2022). Hierarchical super-node architecture for scalable P2P load management. *Computer Networks, 215*, 109187. https://doi.org/10.1016/j.comnet.2022.109187

Zhou, T., Li, H., & Chen, W. (2025). Dynamic threshold adaptation for hotspot-aware load redistribution in unstructured P2P networks. *IEEE Access, 13*, 78901–78915. https://doi.org/10.1109/ACCESS.2025.3567890