Journal of Informatics
Electrical and Electronics Engineering, 2025,
Vol. 06, Iss. 01, S. No. 141, pp. 1-18
ISSN (Online): 2582-7006

Survey of Developing Peer-to-Peer (P2P) Multimedia Algorithm for Live Streaming to Enhance QoS in Cloud Environment

Sharf Aldin Musa Idris Elnuor [1], Sudhir Baburao Jagtap[2]

[1 2] SRTM University Nanded, Swami Vivekanand Mahavidyalaya, Udgir, Dist. Latur
Sharfmusa83@gmail.com[1], sudhir.jagtap7@gmail.com[2]

How to cite this paper: S A M I Elnuor, S B Jagtap, "Survey of Developing Peer-to-Peer (P2P) Multimedia Algorithm for Live Streaming to Enhance QoS in Cloud Environment Journal *of Informatics Electrical and Electronics Engineering* (JIEEE), Vol. 06, Iss. 01, S No. 127, pp. 1–6, 2025.

https://doi.org/10.54060/a2zjournals jiee 141

Received: 05/01/2025

Accepted: 31/03/2025
Online First: 18/04/2025
Published: 25/04/2025

Abstract

In this survey, I examined the latest developments in Peer-to-Peer (P2P) multimedia algorithms, particularly how they enhance live streaming performance when integrated with modern technologies like cloud computing, artificial intelligence [AI], block chain, 5G, and edge computing. My focus is on how these technologies work together to address key challenges in Quality of Service (QoS), scalability, latency, and system reliability. I reviewed a range of current system architectures, especially hybrid models that combine P2P with cloud or content delivery networks (CDNs). These setups are evaluated based on how well they manage real-time resources, deliver adaptive video quality, ensure secure content sharing, and make efficient use of peer contributions. I also studied algorithms driven by reinforcement learning, trust-based decentralization, and edge-level computation. From this analysis, I found that hybrid systems offer a strong balance between the stability of cloud services and the flexibility of peer networks. AI plays a valuable role in adapting to changing conditions, while block chain helps secure content delivery although it may sometimes slow the system down. The combination of 5G and edge computing shows real promise in lowering delays and improving real-time responsiveness. However, several challenges remain. These include unpredictable peer behavior, fluctuating network quality, high implementation costs, and persistent security concerns in decentralized environments. In conclusion, I point to future research opportunities focused on building smarter, more reliable, and highly scalable streaming solutions. By connecting multiple emerging technologies, this work provides meaningful guidance for researchers and developers aiming to advance the next generation of decentralized live streaming platforms.

Keywords

P2P live streaming, QoS optimization, hybrid CDN-P2P, AI-driven algorithms, blockchain security, 5G edge computing.

1. Introduction

P2P multimedia algorithms are important for live streaming. They help make sure the stream keeps going smoothly, even when the network is bad or unstable. This makes the stream better. These algorithms make sure we can share the internet well, so we don't need to spend too much money to keep the stream running. When people share video chunks with each other, the system works better, and it's faster too. This means there's no need to use a lot of external servers, and that's good because we can reduce the delay [1]. When we use P2P algorithms, the experience gets better for everyone. This is because it helps with things like how fast the video starts, the lag, and whether the video keeps playing smoothly. These are things that matter a lot when watching a live stream. Also, P2P algorithms help save a lot of money for ISPs because they lower the traffic they send between each other [2]. It's tough to get high Quality of Service (QoS) in P2P live streaming because the network keeps changing. Things like how smooth the video plays, how fast it starts, and any lag need to be checked all the time to make sure the user doesn't get interrupted. Scalability is another big issue when more peers join in. The system must handle everything well to keep the stream going without problems, but that's harder when peers have different upload speeds [3].

Resource optimization becomes trickier when you must balance using upload bandwidth from extra sources like cloud services, while also adjusting how fast the video plays based on the bandwidth available. If this isn't done right, it can cause inefficiencies. Monitoring systems that check bandwidth can also have issues like mistakes or not working properly, which can affect how well the streaming service works. And when network traffic changes suddenly or peers behave unpredictably, it can lead to lower QoS. Traditional methods might not always catch these changes, which could mess up the whole streaming experience [4].

P2P streaming systems today have a lot of trouble with scalability. As more peers join, the system slows down, and that's a big problem. This shows us that we need new algorithms that can adjust quickly as the number of peers or network conditions change. A lot of current algorithms haven't been tested much, so the theories behind them don't always match what really happens. This means we need research that brings theory and real-life applications together for better P2P systems.

Also, block chain systems that could make P2P streaming better haven't been studied enough, especially when it comes to how different numbers of peers affect them. So, we need some fresh ideas to explore how peer behavior can change system performance. There are not enough studies on how well existing algorithms work in practice, and that's a problem. We need more real-world research to test and improve what we have. Finally, many studies don't really look at how advanced hardware could help algorithms perform better. There's a big chance here to create algorithms that use new tech to work even more efficiently [5].

1.1. Objectives and Scope of the Survey: Addressing QoS Enhancements in Cloud-Integrated P2P Systems

This study explores peer-to-peer (P2P) systems integrated with cloud technologies, with a focus on persistent Quality of Service (QoS) issues. It highlights key challenges like playback delays and interruptions in live streaming. The research reviews recent algorithms for efficient cloud resource management under dynamic network conditions. It examines adaptive playback controls and bandwidth monitoring tools, pointing out their limitations in accuracy and fault tolerance. The work identifies hybrid P2P-cloud models as a path to better scalability and resource use. It emphasizes the need for practical testing of these strategies to ensure real-world effectiveness. The goal is to support smoother streaming and consistent QoS for end users [6].

2. Literature Review

The rapid growth of real-time multimedia services has intensified the demand for efficient and scalable streaming solutions. Peer-to-peer (P2P) systems, when integrated with cloud and edge computing, offer a promising pathway to meet these evolving needs. However, maintaining consistent Quality of Service (QoS) across diverse network conditions remains a persistent challenge. Recent studies have proposed hybrid models that blend the strengths of decentralized overlays with centralized infrastructure. These efforts explore strategies for scheduling, resource allocation, and playback control. Security, adaptability, and device heterogeneity also emerge as recurring concerns across the literature. This review critically examines such contributions to highlight trends, gaps, and future directions in P2P cloud streaming systems.

Liem et al. (2018) examine the integration of Adaptive Bitrate Streaming (ABR) into EPON-based P2P systems to enhance live streaming performance. Their approach reduces buffering and latency by leveraging edge nodes to handle processing closer to end users. The study highlights significant improvements in playback stability under constrained network conditions. However, its focus on fiber-optic infrastructure limits the applicability to broader network types, such as wireless or mobile environments. The lack of real-world testing further weakens generalizability. Without validation across varied network scenarios, the framework's full potential remains uncertain. Broader experimentation is needed to assess its scalability and versatility [7].

Farahani et al. (2022) present a hybrid system called RICHTER, which merges peer-to-peer overlays with cloud and CDN support to enhance live streaming performance. Their architecture uses edge computing and NFV to adjust dynamically to latency-sensitive demands. The integration of 5G further strengthens its potential for real-time applications like VR and live sports. Their simulations show strong scalability under peak loads, particularly during high-traffic events. However, the authors acknowledge a lack of real-world validation, especially concerning deployment cost and trust management. Security concerns in decentralized environments also remain insufficiently addressed. These limitations highlight the need for practical testing before large-scale adoption can be considered [8].

Efthymiopoulou et al. (2017) propose a decentralized, cloud-assisted P2P framework designed to manage adaptive media streaming in real time. Their system introduces feedback-driven scheduling and congestion control, supported by cloud monitoring to maintain stream quality. A key strength lies in integrating Adaptive Bitrate Streaming (ABR), enhancing playback stability under changing network conditions. Their layered approach improves fault tolerance and optimizes bandwidth use efficiently. However, the framework assumes uniform peer behavior and neglects differences in user device types. As a result, its performance in mobile or heterogeneous environments remains uncertain. This gap limits its scalability and practical application across diverse network settings [4].

Sina et al. (2019) explore a more algorithmic solution by integrating reinforcement learning (RL) into cloud-supported P2P streaming. Their system, CaR-PLive, adjusts its operation in real time based on peer behavior and available bandwidth, offering both performance gains and better resource distribution. The novelty of their work lies in combining AI-driven decision-making with architectural flexibility. However, their reliance on idealized training data and simplified network models raises concerns about how reliably the system would perform in unpredictable or large-scale real-world settings [3].

Ding et al. (2023) explore security in hybrid P2P-CDN systems by introducing block chain and zero-knowledge proofs to protect content authenticity and user privacy. Their approach addresses a major challenge in decentralized networks—detecting and preventing malicious behavior. By embedding cryptographic proofs, the system ensures trust without exposing user data. However, the added security introduces latency due to block chain consensus, which can conflict with the low-latency needs of live streaming. This reveals a key trade-off between performance and protection. While the model enhances trust, its practical use in time-sensitive applications remains constrained. Further work is needed to balance these opposing demands [5].

In reviewing the work of Dao et al. (2022), I find their approach to live video streaming both progressive and necessary. They present a computational lens through which streaming systems should be evaluated, advocating for the integration of predictive analytics and machine learning across both traditional and peer-to-peer architectures. Their main argument is that future systems must go beyond bandwidth optimization, becoming intelligent enough to manage processing power and memory resources as well. This computation-aware model presents a promising direction for next-generation streaming platforms. However, a key limitation in their work is the absence of empirical support. Their recommendations, while theoretically sound, are not backed by real-world experiments or large-scale trials. This lack of validation limits the immediate applicability of their insights and highlights the need for further testing to establish their feasibility in operational environments [9].

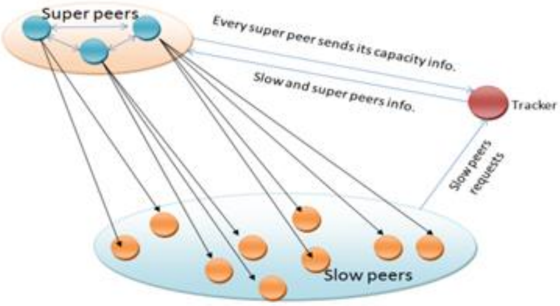
Super peers
Every super peer sends its capacity info.
Slow and super peers info.
Tracker
Slow peers requests
Slow peers

Figure 1. **P2P Overlay Construction**

Taking a different perspective, Hadi et al. (2023) examine the security implications of scaling edge infrastructure in 5G-enabled P2P streaming. While they do not focus directly on latency, they highlight that increasing the number of edge nodes also widens the system's attack surface. Their proposal to integrate lightweight cryptographic techniques and decentralized trust models brings valuable security considerations to the table. However, they fall short of addressing how these protections might affect system responsiveness or energy efficiency — both of which are critical for maintaining a seamless user experience [10].

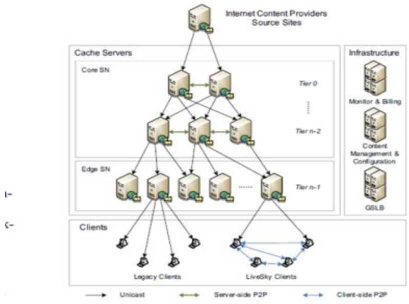

Figure 2. **System Architecture**

3. Cloud Integration and Its Impact on P2P Streaming:

3.1. Advantages of Hybrid P2P-Cloud Models for Scalability and Cost-Effectiveness

Hybrid P2P-cloud models help with scalability because they use the strengths of P2P networks. This means they can handle more users without needing to build a bunch of new infrastructure. It's important for when there are lots of people streaming videos, like during busy times or live events. These models also help save money. By combining P2P and cloud technology, they only use cloud services when it's needed, so you don't have to spend too much on servers but still get good service [5]. With the hybrid method, content is shared among peers, which takes some pressure off the main servers. This makes everything better and cuts down on the cost of running big server farms. Cloud resources paired with P2P networks help improve Quality of Service (QoS). The system adjusts in real-time to meet the demand, ensuring users get a steady, good-quality stream, which keeps people happy.

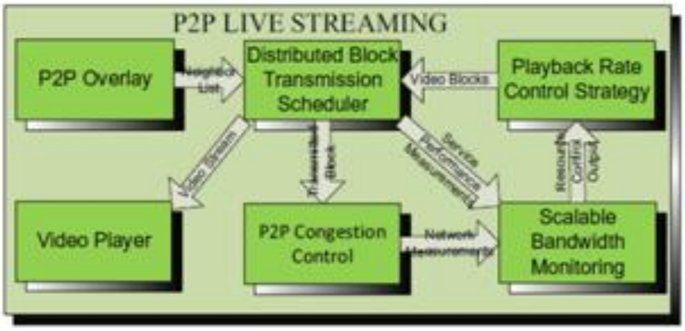

Figure 3. **Proposed P2P live streaming system architecture**

3.2. Balancing Centralized Cloud Resources with Decentralized Peer Contributions

When we combine centralized cloud resources with decentralized peer contributions, it helps us use bandwidth and storage more efficiently. This hybrid method makes sure resources are used well, cutting down costs and making things run smoother. Load balancing can be managed better by using both centralized and decentralized systems [11]. So, when there's a lot of traffic, the system can pull from the cloud and use the bandwidth from peers, which keeps everything running without hiccups. Mixing cloud infrastructure with peer contributions makes the system more reliable. If a cloud server goes down, the decentralized network keeps going, making sure the service stays available. Adding decentralized peers makes it easier to scale up. As more people join, more peers can be added without spending a ton on new cloud infrastructure, which helps handle growth better. Using peer contributions reduces the costs of running big cloud systems. By shifting some of the work to peers, companies don't have to rely so much on expensive cloud services, saving money [12].

3.3. Enhancing QoS through Edge Computing and Dynamic Resource Allocation

Edge computing helps by moving data processing closer to the users, so there's less delay. This is important for real-time things like live streaming, where any delay messes with the experience. When data is processed closer, the system can react faster to what users need, making the quality better. With dynamic resource allocation, the system can adjust things like bandwidth and processing power based on what's happening right then [7]. This helps make sure resources are used well and stops slowdowns during busy times, keeping the stream running smoothly. When you mix edge computing with dynamic resource allocation, bandwidth management gets better. The system can check the upload capacity of peers and adjust how it shares resources, which helps keep the video quality steady for everyone. Edge computing helps with scalability because it can add more resources near users as needed. This means the system can grow without needing a huge central setup, making it easier to handle different numbers of users. By combining edge computing with dynamic resource allocation, fault tolerance improves. If one edge node fails, others can step in and keep everything running, so the service doesn't get interrupted and the quality stays high [4].

4. Algorithmic Advancements for QoS in P2P Multimedia Streaming:

4.1 Key QoS Metrics

Advanced algorithms can reduce latency in P2P multimedia streaming by picking the best peers and routing data more efficiently. For example, dynamic neighbor selection based on things like QoS helps to make sure delays are low, which is important for real-time applications. By adjusting how resources are allocated based on the current network situation, algorithms can boost throughput [13]. They make sure the data flow matches the available bandwidth of peers, so the stream doesn't get interrupted. Bandwidth is used better when algorithms balance the load between peers. By checking upload capacity and moving tasks around, these algorithms help make the most of the bandwidth, which makes streaming better. Reliability gets a boost when P2P networks use self-adaptive methods. These methods can notice when peers leave the network and adjust the setup to keep things stable, even if some peers drop out [14]. When algorithms consider things like user preferences and network conditions, they can really improve the Quality of Experience (QoE). They adjust playback rates and make sure transitions are smooth, which keeps users happy with the stream.

4.2 Algorithmic Techniques for Enhancing QoS:

Using a Shared Block Transmission Scheduler (DBTS) lets peers transmit video blocks in a well-organized way. This helps transmit data better, adjusting to how fast the network and peers are working, which makes the streaming experience better overall [13]. The discussed algorithmic techniques Dynamic Block Transmission Scheduling, Load Balancing through Peer Categorization, Adaptive Bandwidth Allocation, Gossip Protocol for Resource Monitoring, and Hierarchical Clustering for Scheduling collectively contribute to enhancing QoS in P2P streaming systems [4]. Each strategy addresses specific challenges in scheduling and load balancing, offering solutions to improve network efficiency, scalability, and user experience. Future research may focus on integrating these techniques to develop more robust and adaptive P2P streaming architectures.

4.3 AI-Driven Algorithms for Real-Time Optimization

The algorithm checks the system's condition all the time and makes changes based on what it learns. This lets it react quickly when things like network speed or user devices change [3]. By doing this, it makes sure everything works well no matter what happens. AI can also grow and change. It does this by grouping peers based on how well they're working, and then it moves things around if needed. That way, things stay balanced, and the system is always doing its best. Plus, it can predict what will happen next. The system uses data to guess what the network might be like in the future and makes changes ahead of time. This helps avoid problems before they happen and keeps everything running smoothly, even when there are lots of people using it [2].

5. Application for Reinforcement Learning (RL):

5.1. Applications for Reinforcement Learning (RL) in Optimizing QoS and Resource Allocation

Reinforcement Learning is offers a strategic approach to improve a powerful framework for optimizing Quality of Service (QoS) and resource allocation in dynamic and complex environments such as peer to peer (P2P) cloud-based streaming systems .By enabling agents to learn optimal policies through interactions with their environment, RL facilitates intelligent decision-making that adapts to changing network conditions, user behaviors, and system constraints [3]. Dynamic Resource Provisioning: RL algorithms, particularly those based on value iteration and policy gradient methods, can be effectively employed to automate resource provisioning in cloud-assisted streaming systems. By learning the relationship between workload fluctuations and system performance, RL agents can allocate computational and network resources dynamically, ensuring that predefined QoS thresholds (e.g., latency, buffering time) are maintained while minimizing operational costs. Real-Time Decision Making: Algorithms such as Q-learning and Deep Q-Networks (DQNs) enable real-time decisions for bandwidth distribution and video quality selection. These RL agents evaluate the current state of the networks such as congestion levels or node availability and take actions to adjust streaming parameters in response. This facilitates smooth and uninterrupted media delivery, even under fluctuating conditions, thereby improving the end-user Quality of Experience (QoE).

Optimization of Streaming Parameters: RL models can fine-tune system parameters, such as buffer sizes or the urgency threshold in video segment delivery, to achieve an optimal trade-off between service quality and resource usage. By continuously adapting these parameters based on observed network behavior, RL-enhanced systems can reduce latency and improve throughput without overprovisioning [3]. Feedback-Driven Learning: One of the core strengths of RL lies in its ability to learn from experience via reward signals. Feedback-driven learning enables the system to iteratively refine its policies based on the success or failure of past decisions. For instance, if a certain bandwidth allocation strategy results in reduced rebuffing and higher user satisfaction, it is reinforced in the agent’s policy updates, leading to improved future performance. Handling Uncertainty in Environments: In real-world network scenarios, decision-making must often occur under partial observability and uncertainty. RL models formulate are using Markov Decision Processes (MDPs) or Partially Observable MDPs (POMDPs) are well-suited for such contexts. These models allow agents to infer optimal actions despite incomplete knowledge about user behavior, node reliability, or channel quality, thereby enhancing robustness and adaptability.

5.2. Case Studies of RL-Based Algorithmic Improvements

The application of reinforcement learning (RL) in multimedia streaming and peer-to-peer (P2P) networks has been validated through several case studies, each demonstrating the adaptability, robustness, and performance enhancement achieved through RL-based algorithmic frameworks.

Optimization of Video Streaming Parameters

A notable case study employed Q-learning to optimize the urgent sub-window size in a video streaming context. The urgent sub-window determines the segment of video data that must be perfected to avoid playback interruptions. The RL agent dynamically adjusted this parameter to strike a balance between playback continuity and resource expenditure, such as rental or bandwidth costs [3]. Over time, the learning process converged within approximately six hours, indicating that the agent could adapt to evolving system dynamics and user behavior. The results demonstrated significant improvement in resource allocation efficiency playback smoothness.

5.2.1. Peer Selection in P2P Networks

In another study, an RL-based algorithm known as UPDA (Upload Peer Distribution Algorithm) was introduced to improve peer categorization and selection in P2P live streaming. This model classifies nodes into super peers and slow peers based on bandwidth and reliability. Using RL-driven peer selection strategies, the system achieved enhanced Quality of Service (QoS) metrics, including reduced end-to-end delay, improved response times for uploads and downloads, and increased stream stability. Compared to traditional random or static algorithms, UPDA proved more effective in dynamically adapting to network fluctuations [13].

5.2.2. Real-Time Adaptation to Network Conditions

RL has also been applied to manage bandwidth allocation in cloud-based streaming infrastructures. In a representative case, an RL agent continuously monitored key network parameters such as traffic congestion, latency, and server load. Based on this real-time feedback, the agent made informed decisions about bandwidth distribution to maintain QoS, especially during peak usage hours. The system demonstrated strong resilience against unpredictable network conditions, showcasing the viability of RL for real-time, large-scale deployment in cloud environments [16].

5.2.3. Anomaly Detection in Streaming Services

A further application of RL involved anomaly detection in P2P video streaming services. Here, the RL agent learned from historical streaming data and peer behavior to model typical connection patterns [13]. By identifying deviations from these patterns, the agent was able to detect anomalies such as unexpected packet loss or peer churn early and respond with preemptive adjustments in routing or buffering strategies. This predictive capability significantly contributed to maintaining service reliability and user experience.

6. Challenges in P2P Network Dynamics, Scalability and Reliability Issues:

6.1. Issues in content coordination, peer selection, and task assignment.

In P2P networks, keeping content coordinated is hard because peers join and leave all the time, and their upload speeds are different. This can mess up the content delivery and make the quality of the stream go up and down for users. We need good systems to make sure content gets to the right peers, especially when lots of people are using the network [12].

Choosing the right peers for sharing content is super important for using bandwidth well and making streaming smooth. The fact that peers come and go makes it harder to pick the best ones. The algorithms need to change to match those shifts so that the peer-to-peer system keeps running well, which can be tricky in real time [3].

Another issue is giving tasks to peers. Since peers don’t all have the same abilities, some have faster internet and better processors it's important to figure out who gets what tasks. If tasks are assigned wrong, the system slows down, which ruins the experience for everyone [14].

Peer churn, or when peers keep joining and leaving, can mess up the network stability. It makes it hard to keep everything in place, causing delays and drops in quality during content delivery [4].

6.2. Improving Performance Through Algorithmic Innovations

One of the pivotal innovations in P2P streaming involves dynamic peer categorization, where algorithms classify nodes into roles such as super peers and low-capacity peers. Super peers typically those with greater upload bandwidth and processing capabilities—are tasked with redistributing content to slower nodes. This structured hierarchy improves data dissemination, reduces transmission delays, and significantly boosts Quality of Service (QoS) in live streaming applications [13]. Efficient task assignment is another critical area of development. By considering the heterogeneous nature of peer resources such as varying bandwidth capacities and computing power, modern algorithms can distribute tasks in a more balanced and intelligent manner. This reduces the likelihood of congestion, optimizes throughput, and enhances the responsiveness of the network.

To address fluctuating network loads, adaptive bandwidth management techniques based on reinforcement learning have been introduced. These algorithms monitor real-time network states and adjust bandwidth allocation dynamically, ensuring stable and efficient streaming even under high traffic conditions [2]. In scenarios where service quality might degrade, the integration of fallback mechanisms and lock-free data structures provides resilience. These solutions allow the system to rapidly redirect requests to alternative nodes, thereby maintaining playback continuity and improving user satisfaction even in the presence of faults or failures.

6.3. Addressing Peer Churn and Maintaining Consistent Contributions

Dynamic Peer Management plays a crucial role in sustaining service continuity within Peer to Peer (P2P) streaming systems. Given inherently transient nature peers, algorithms capable of dynamically adjusting to the fluctuating availability and performance of nodes are essential. By continuously monitoring network conditions and peer participation, these systems can reassign responsibilities in real-time, thus maintaining uninterrupted content delivery despite ongoing peer churn [4]. To enhance fault tolerance, redundant peer selection mechanisms are employed. These strategies involve maintaining a reserve set of eligible peers that can seamlessly replace departed nodes. This redundancy mitigates service disruption and strengthens overall system resilience, particularly in large-scale deployments where peer turnover is frequent.

Incentive mechanisms further contribute to system stability by motivating consistent participation. Through credit-based systems, reputation scoring, or reward-based models, peers are encouraged to contribute bandwidth and processing resources. Such incentives reduce the rate of abrupt departures and foster a more cooperative and sustainable network environment [14].

Effective load balancing techniques are also imperative. By evenly distributing data forwarding and processing tasks among peers, the system ensures that no single node becomes a bottleneck or point of failure. This not only improves overall performance but also increases fault tolerance by allowing the system to absorb the loss of individual peers without degrading quality of service (QoS) [2].

7. Fault Tolerance Mechanisms in P2P Algorithms:

7.1 Fault Tolerance Mechanisms:

Ensuring fault tolerance is a critical requirement in the design of robust Peer to Peer (P2P) systems, particularly in live streaming and other latency-sensitive applications. Due to the dynamic and decentralized nature of these networks, effective mechanisms must be in place to detect, manage, and recover from node failures without compromising service quality.

Redundancy is one of the fundamental strategies employed to enhance fault tolerance. By replicating data and services across multiple nodes, the system can prevent service disruption in the event of peer failures. When a node becomes unavailable, redundant nodes can assume its responsibilities, thereby ensuring continuous access to content and resources. This distributed replication not only enhances data availability but also supports load balancing during peak demand [17]. Check pointing offers another layer of resilience by enabling periodic saving of a system's operational state. In the event of a fault or crash, the affected node can revert to its most recent checkpoint, reducing downtime and preventing data corruption. In P2P networks, check pointing is particularly useful for long running or tasteful tasks, allowing systems to recover efficiently with minimal loss of progress or data integrity.

To further enhance responsiveness and adaptability, dynamic recovery protocols are integrated into modern P2P systems. These protocols actively monitor network health and detect peer failures in real-time. Upon identification of a fault, tasks and data associated with the failed node are swiftly redistributed among available peers. This rapid reallocation ensures that the system remains operational with minimal performance degradation. Dynamic recovery is essential in highly volatile environments, where peer churn is frequent and unpredictable [15].

7.2 Security Considerations in Algorithm Design:

7.2.1. Cryptographic **Methods for Secure Communication**

Security is a foundational concern in the design of algorithms, particularly within decentralized and peer-to-peer (P2P) systems, where secure communication is critical for maintaining trust, integrity, and confidentiality. A well-structured security framework relies on a combination of cryptographic techniques that ensure data is neither tampered with nor intercepted by unauthorized parties during transmission. Hash Algorithms such as SHA-256 are widely used to ensure data integrity. By producing a fixed-length digest from inputs of arbitrary length, these algorithms allow systems to detect any unauthorized modifications to the data. A single-bit change in the input yields significantly different hash, making tampering easily detectable. Hashing plays a crucial role in both data verification digital signature generation [5]. Commitment Schemes build upon the properties of hash functions to enable secure message commitments. In this approach, a sender can commit to a specific message while keeping it concealed until a designated reveal phase. This method ensures that the sender cannot alter the content once committed, thereby supporting accountability and fairness in secure protocol executions. Elliptic Curve Cryptography (ECC) is increasingly adopted for secure communication due to its ability to offer equivalent security to traditional methods (like RSA) with much smaller key sizes. This efficiency reduces computational overhead and energy consumption, making ECC particularly advantageous in P2P systems and mobile or edge devices where resources are limited. Message Authentication Codes (MACs) are critical for ensuring both authenticity and integrity of messages. By using a generate secret key to a MAC, the recipient can validate the message originated from a legitimate sender that has not been altered in transit. This provides protection against both message forgery and impersonation attacks.

Dynamic Key Management protocols underpin the secure operation of cryptographic systems. These protocols handle key generation, distribution, rotation, and revocation. In dynamic, distributed networks, effective key management ensures that cryptographic keys remain confidential and that communication remains secure even as peers join or leave the system. This is vital for maintaining forward and backward secrecy in long-running sessions [18].

7.2.2 Block Chain-Based Trust Systems to Mitigate Malicious Behaviors

Block chain technology offers a robust and transparent framework for mitigating malicious behaviors in distributed systems by establishing trust without central authority. Through its decentralized architecture and built-in security mechanisms, block chain facilitates secure, tamper-resistant communication and collaboration among peers in P2P networks.

Decentralization is a core strength of block chain. Unlike traditional centralized architectures, block chain distributes control across a network of nodes. This decentralization removes the single point of failure, making it significantly more difficult for adversaries to compromise the system. An attacker would need to gain control over most nodes (commonly more than 51%) to manipulate the ledger, which is economically and technically prohibitive in well-established networks [15].

Transparency and Immutability further reinforce trust by ensuring that all transactions are permanently recorded and publicly verifiable. Once a transaction is appended to the block chain and reaches consensus, it becomes immutable—immune to modification or deletion. This characteristic discourages fraudulent activity, as the entire transaction history remains auditable by all participants, promoting accountability. Smart Contracts provide automated enforcement of rules and agreements. These programmable contracts self-execute when predefined conditions are met, reducing reliance on intermediaries and the potential for human error or manipulation. Smart contracts can enforce access controls, detect anomalies, and prevent double-spending or unauthorized transactions, thus increasing system integrity [18].

Consensus Mechanisms, such as Proof of Work (PoW), Proof of Stake (PoS), and their derivatives, ensure that all nodes agreed on the state of block chain before any update is finalized. These algorithms require significant computational work or stake commitment to influence the network, thus acting as a deterrent to malicious manipulation. By requiring majority approval, consensus protocols protect against attacks such as Sybil attacks and double spending. Cryptographic Security underpins every layer of block chain technology. Techniques such as public-key cryptography, digital signatures, and secure hashing ensure that data is securely encrypted and only accessible to authorized users. These methods safeguard against data breaches, identity spoofing, and unauthorized alterations.

7.2.3. Integrating Security Features into QoS Optimization Algorithms

In peer to peer (P2P) streaming systems, achieving high quality of service (QoS) must be balanced with robust security measures to ensure reliable and secure content delivery. Integrating security features directly into QoS optimization algorithms enables the system to protect against malicious activities while maintaining desirable performance metrics such as latency, jitter, throughput, and packet loss.

Incorporating Cryptographic Techniques is fundamental to securing communications without sacrificing performance. Algorithms such as AES for encryption and SHA-256 for hashing can be seamlessly integrated into QoS workflows. These techniques ensure data confidentiality and integrity during transmission, while computationally efficient implementations can minimize additional latency, thereby preserving user experience. Utilizing Block chain for Trust introduces a transparent and decentralized mechanism for verifying network interactions and service quality metrics. By recording QoS-related data on an immutable ledger, the system can detect anomalies indicative of dishonest behavior or network abuse. Block chain's tamper-proof nature makes it an effective deterrent to falsified QoS reports and promotes accountability among participants in distributed environments [2]. Adaptive Security Measures allow QoS algorithms to dynamically respond to fluctuating threat levels. For instance, an algorithm could monitor for unusual patterns such as increased packet loss or jitter—potential signs of a denial-of-service attack or man-in-the-middle interference—and respond by increasing encryption strength or switching to a more secure protocol. This context-aware adaptability ensures resilience against attacks without permanently degrading performance. Incorporating Authentication Protocols ensures that only verified users can participate in data exchange or consume network resources. Protocols such as Kerberos, Oath, or TLS-based mutual authentication can be embedded into the QoS management layer to authenticate users and prevent unauthorized access, which can degrade system performance or expose vulnerabilities. Dynamic Key Management plays a vital role in maintaining ongoing secure communication sessions. Efficient key generation, distribution, and revocation mechanisms ensure that encryption keys remain protected and valid, even in highly dynamic environments. When embedded within QoS algorithms, key management ensures that encryption overhead does not compromise service quality while still providing a high level of security. Monitoring and Feedback Loops
Security mechanisms may involve continuous tracking of network activity to spot unusual patterns that could signal malicious intent. When such activity is detected, feedback systems can automatically adjust Quality of Service (QoS) settings in real-time. This helps maintain high service quality, even when security threats arise [9].

8. Current Trends in P2P Multimedia Algorithms for Live Streaming:

8.1. AI in Enhancing Hybrid CDN-P2P Streaming Systems

Artificial intelligence plays a key role in improving hybrid CDN and peer to peer (P2P) live streaming systems. One of its main uses is selecting the most suitable peers for data exchange. Unlike traditional methods that rely on location or network provider, machine learning models assess deeper peer relationships to boost data transfer rates and reduce latency, resulting in more efficient streaming.

Machine learning techniques are used to improve peer selection in hybrid CDN–P2P systems. Unlike conventional methods that rely on factors like geographic location or ISP grouping, AI models can uncover deeper patterns in peer relationships. This leads to better data throughput and lower delays, especially in live streaming environments. Reinforcement learning is used in some systems to refine peer selection and resource allocation over time. These algorithms learn from experience, allowing the network to evolve and continuously improve its quality-of-service AI algorithms help distribute bandwidth among peers dynamically, adjusting to real-time network changes. By forecasting performance, they allocate resources more effectively, maintaining service quality while keeping operational costs in check [7] [19].

Predictive analytics powered by AI allow networks to estimate future demand for computing and bandwidth. This enables early intervention before performance issues arise, helping to avoid congestion and ensure smooth streaming. AI systems can detect anomalies in traffic patterns or user behavior, identifying potential threats early. By responding immediately, these mechanisms help safeguard service integrity and maintain high-quality performance [4].

AI also supports real-time video quality adjustment by monitoring user preferences and current network conditions. This reduces buffering and maintains continuous playback, improving overall user satisfaction.

8.2. Hybrid P2P-CDN and P2P-Cloud Architectures for Improved Reliability

Blending the robust infrastructure of Content Delivery Networks (CDNs) with the flexible, distributed nature of Peer-to-Peer (P2P) systems creates a hybrid architecture that improves both efficiency and reliability in content delivery. This integrated model offers reduced latency and high availability, making it especially effective for live streaming. Load balancing also benefits from this dual approach. Traffic is distributed intelligently across both CDN servers and participating peers, reducing the risk of overload on any single node and improving overall system stability [8] [19].

The inclusion of cloud-based P2P components further strengthens fault tolerance. If a CDN server fails, content can still be distributed through active peers in the network, preventing service interruptions and maintaining viewer experience [5]. Such hybrid setups can adjust resources dynamically in response to real-time traffic and network performance. This responsive allocation ensures consistent service quality, even during peak demand periods.

8.3. Advances in Cryptographic Methods and Decentralized Trust Frameworks

Modern cryptographic innovations have led to stronger privacy protections within decentralized systems. Protocols like Crypto Note help conceal transaction details, making it harder to trace or link user activities—an essential feature for preserving anonymity in block chain-based environments [18].

New methods are also advancing how digital identities are managed. These cryptographic techniques support decentralized identity frameworks, giving individuals control over their own data while reducing dependency on centralized institutions. This shift promotes greater confidence in digital interactions. Smart contract technology has similarly benefited from improved cryptographic tools. Enhanced encryption ensures that contracts are executed securely and accurately, limiting the potential for tampering or malicious interference. This strengthens trust among network participants [10].

The reliability of distributed systems has been further boosted by evolving consensus algorithms. Techniques such as Proof of Stake (PoS) and Practical Byzantine Fault Tolerance (PBFT) provide more secure and efficient ways to maintain agreement across the network, minimizing inconsistencies and promoting system integrity.

8.4. Integration of Multi-Layer Algorithms for Dynamic Environments

Multi-layered algorithmic frameworks are increasingly being employed to adapt resource allocation in real-time. This capability is especially vital for dynamic applications like live streaming, where user activity and network conditions can change unpredictably. The system adjusts its resources accordingly to maintain smooth operation. By layering different algorithm types, systems gain the ability to process a wider range of data inputs and apply more sophisticated decision-making strategies. For example, combining reinforcement learning with conventional approaches enables decisions that reflect both current conditions and past patterns, improving accuracy and responsiveness. Quality of Experience (QoE) benefits significantly from this approach. The layered structure allows for real-time tuning of performance factors like latency, buffering, and video clarity. Continuous monitoring of network metrics and user interactions enables systems to maintain high service standards even under changing conditions [9].

Such architecture also enhances scalability and flexibility. As user numbers grow or workloads shift, the system can expand and reconfigure itself without significant drops in performance. This is essential in distributed environments where resource demand fluctuates frequently [3].

Table 1. QoS enhancement in P2P streaming systems

Study (Author, Year)	Objective/Focus	Methodology/Algorithm	Strengths	Limitations	Key Metrics Evaluated	Security Measures
Adaptive Artificial Intelligence Techniques for QoS (2024)	Real-time decision-making for QoS	Reinforcement Learning, Deep Learning (CNN, LSTM)	Dynamic adaptation, significant improvements in QoS metrics	Algorithmic bias, data privacy concerns	Latency, packet loss, throughput	Not directly addressed
Yang et al., 2024	Performance improvements in QoS	AI-based latency and throughput optimization	Reduces latency by up to 2 ms, increases throughput by 8%	Focuses primarily on technical metrics	Latency, throughput	Not addressed
Ghuge, 2024	Anomaly detection in P2P streaming	Hybrid AI Models (CNN + LSTM)	High accuracy in detecting QoS anomalies, low false positive rates	Limited scalability in real-world scenarios	Anomaly detection accuracy	Not addressed
Farahani et al., 2022	Scalability and flexibility in hybrid P2P-CDN systems	Dynamic switching, integration with edge computing and NFV	Enhances scalability, flexibility, and QoE	Security vulnerabilities in P2P networks	Latency, QoE	Blockchain and zero-knowledge proof for copyright protection

Igor, 2024	Fault tolerance in hybrid architectures	Data replication, load balancing, and sharing	Ensures service availability during failures	Limited exploration of integration with advanced technologies	Fault tolerance, service availability	Limited to fault resilience
Minu & Nagarajan, 2024	Trust management through blockchain	Decentralized trust systems, smart contracts	Enhance transparency, automates trust-building processes	Adoption barriers, potential new vulnerabilities	Trust levels, security	Blockchain and decentralized systems
Rajani et al., 2024	Enhancing cryptographic security in IoT	Quantum-Resistant Cryptography (QRC), Elliptic Curve Cryptography (ECC)	Provides robust data protection, reduces computational overhead	Limited real-world deployment	Security, computational efficiency	Advanced cryptographic methods
Majdoubi et al., 2022	Cryptographic security in constrained environments	Advanced cryptographic techniques for IoT	Suitable for IoT, enhances access control	Focusing primarily on technical feasibility	Data security, access control	Quantum-resistant cryptography
Zhang et al., 2016	Develop a QoS-aware peer coordination control mechanism.	Introduced a ring buffer mechanism and self-adaptive adjusting mechanism.	Enhanced streaming synchronization and reduced overhead	Complexity in implementation under varying network conditions.	Startup delay, playout lag, playback continuity	Not specifically addressed
Gao et al., 2013	Measure performance of P2P streaming systems.	Measurement study of various P2P systems.	Provides empirical data on performance metrics	Focused on specific environments, limiting generalizability	Throughput, latency, QoE	Not addressed

This table summarizes various studies on QoS enhancement in P2P streaming systems, focusing on methodologies like AI, blockchain, and cryptography. It highlights strengths such as improved scalability, latency reduction, and security, but also notes limitations like scalability issues, adoption barriers, and the lack of direct security measures in some cases.

9. Case Studies of Successful P2P Algorithm Implementations:

9.1. Real-world applications demonstrating effective QoS and scalability.

A peer-to-peer live streaming system has effectively deployed monitor bandwidth conditions in real time with low computational overhead. This solution maintains a consistent quality of service (QoS) even when network conditions fluctuate, confirming its robustness under practical deployment scenarios. The use of layered algorithms in video transmission across P2P networks has led to a notable increase in throughput. As transmission rates scale, the system enables peers to exchange video streams locally, which not only supports high scalability but also ensures consistent playback quality [7].

An application-aware architecture tailored for P2P streaming significantly improves traffic flow within ISPs by keeping frequently accessed video content closer to end users. This reduces cross-network data transfers, cuts down packet delays, and enhances scalability in dense network environments. Firms like Peer5 and Live peer have demonstrated successful hybrid models that merge P2P streaming with CDN infrastructure. These hybrid systems reduce server load, increase stream reliability, and deliver smoother user experiences, all while maintaining cost-effectiveness [4].

Advanced P2P overlay networks have made it possible to fully utilize available upload bandwidths across devices with varying capabilities. By distributing traffic evenly and minimizing external data routing, the system enhances overall efficiency and sustains high QoS across the network [8].

10. Future Directions in P2P Multimedia Algorithms:

10.1. Integration of 5G and Edge Computing for Ultra-Low Latency

The fusion of 5G technology with peer-to-peer (P2P) multimedia algorithms presents a promising path toward better bandwidth efficiency. With faster data transfer capabilities, this integration supports the demands of high-definition video streaming and real-time services where low latency is critical. As networks grow more complex, adaptive resource management becomes increasingly important. Future P2P algorithms must be capable of reallocating resources on-the-fly, adjusting to shifting network conditions to maintain steady performance. This is especially relevant in 5G environments, where usage can vary sharply throughout the day [8]. Another major benefit lies in improved scalability. The combined strengths of 5G and edge computing will allow P2P networks to support large-scale connectivity, which is essential for applications in smart cities and Internet of Things (IoT), where vast numbers of devices require smooth, continuous multimedia access. Quality of Experience (QoE) also stands to benefit. Faster load times and seamless playback, made possible by real-time edge processing and high-speed 5G connections, will significantly improve user satisfaction—particularly for latency-sensitive tasks like video calls and online gaming.

Edge computing plays a key role in optimizing these systems by processing data closer to users. This local handling of information reduces transmission delays and enhances responsiveness, an advantage particularly valuable in live streaming and interactive media applications [9].

10.2. AI and Reinforcement Learning for Adaptive Streaming and Efficient Resource Management

Artificial intelligence, especially reinforcement learning (RL), is proving to be highly effective in managing resources within P2P streaming systems. These technologies enable real-time adaptation to shifting user demands and fluctuating network conditions, ensuring efficient use of resources while upholding consistent quality of service (QoS). This dynamic provisioning challenge can be framed using a Markov Decision Process (MDP), offering a structured way to make optimal decisions in ever-changing environments [3]. Operational cost reduction is another key advantage. By applying RL methods such as Q-learning, systems can intelligently manage cloud resource rentals, aiming to meet required QoS benchmarks while keeping expenses low—a significant benefit for service providers seeking cost-efficiency without compromising performance.

AI also supports smarter buffer management. Adaptive techniques, such as a two-stage sliding window mechanism, allow systems to modify buffer sizes in response to real-time network feedback. This helps maintain timely delivery of video segments and reduces end-to-end latency, ensuring smooth playback [3] [9]. Additionally, predictive analytics can be used to forecast user behavior and content trends. Machine learning models trained on historical data enable the system to anticipate demand spikes and pre-allocate resources, accordingly, improving overall responsiveness and system readiness.

Perhaps most importantly, the combined use of AI and RL in adaptive streaming leads to a marked improvement in Quality of Experience (QoE). These systems continuously learn from both network performance and user interactions, allowing them to fine-tune streaming quality on the flight. This reduces buffering and delays, providing viewers with a more seamless and engaging experience [3] [4].

The transition to computation-focused models in large-scale P2P multimedia streaming has created valuable opportunities for further research. A particularly promising path lies in harnessing in-network computing to improve system responsiveness, scalability, and overall performance. By embedding these capabilities within P2P algorithms, future designs can more effectively tackle long-standing efficiency issues, particularly in environments with fluctuating network conditions and heavy user loads [20].

10.3. Block chain and Decentralized Systems for Trust-Based Resource Sharing

Block chain technology introduces a distributed trust framework in which no single authority governs the system. Instead, trust is built collectively, as all transactions are transparently recorded on a shared ledger that is open to all participants. This structure reduces the likelihood of fraud and promotes accountability across the network. To maintain consistency in such systems, consensus algorithms play a critical role. These mechanisms ensure that all peers agreed on the validity of each transaction, enabling a unified view of shared resources [18]. This alignment strengthens trust among users and upholds the integrity of the network. Participant verification is also key to maintaining trust. In many decentralized platforms, identity verification protocols are applied before granting access to users. This vetting process helps ensure that only credible actors engage in resource sharing, reinforcing the reliability of the system.

Smart contracts offer further advantages by automating agreements between users. These digital contracts execute predefined conditions without the need for intermediaries, streamlining processes and ensuring that all parties comply with agreed terms, thereby boosting efficiency and mutual trust [5].

Security is another cornerstone of block chain-based systems. Through advanced cryptographic methods, transactions are safeguarded against tampering and unauthorized access—an essential feature when sharing resources in open or semi-open environments [10].

10.4. Balancing Fault Tolerance and Energy Efficiency in Large-Scale Deployments

Achieving a balance between fault tolerance and energy efficiency is a key challenge in large-scale systems. One effective strategy involves dynamically allocating resources in response to real-time demands and workloads. This ensures optimal use of available capacity, reducing energy waste while preserving system resilience in the event of failures. Ongoing performance monitoring is vital to this balance. Load balancing techniques, which help distribute tasks evenly across nodes, prevent any single component from becoming overburdened. This not only mitigates the risk of system slowdowns or breakdowns but also contributes to more efficient energy usage across the network [14].

The strategic use of low-cost infrastructure also offers energy-saving benefits. However, care must be taken to avoid underutilization, which can offset these gains. A well-calibrated approach that ensures consistent, efficient use of inexpensive resources can maintain service quality while reducing overall power consumption. Adaptive algorithms play an increasingly important role in managing this balance. These systems can analyze historical trends and respond to current network conditions, fine-tuning operational parameters in real time. As a result, they help maintain fault tolerance while optimizing energy use based on actual needs rather than static configurations.

Geographical distribution is another important factor. Deploying resources closer to end-users can reduce both latency and energy costs associated with data transmission. At the same time, incorporating redundancies across multiple regions helps safeguard system continuity in the event of localized failures, supporting both reliability and efficiency.

10.5. Exploring Hybrid Algorithmic Models for Scalability and Performance

Hybrid algorithmic frameworks that integrate Peer-to-Peer (P2P) architectures with Content Delivery Networks (CDNs) offer a promising approach to scaling multimedia systems while maintaining high performance. This combination enables more efficient distribution of content by offloading traffic from centralized servers, thereby improving the consistency and quality of streaming services [9] [19]. The adoption of online learning algorithms adds an additional layer of responsiveness. These algorithms enable the system to evolve in real time, adapting its strategies based on shifting network dynamics and user behavior. This capability is essential for maintaining optimal performance in large, dynamic environments.

A key feature in these models is adaptive bitrate control, which adjusts video quality in real time based on user preferences and prevailing network conditions. This ensures a smoother viewing experience by minimizing interruptions during periods of bandwidth fluctuation [8].

To further enhance system efficiency, machine learning techniques can be employed for intelligent peer selection. By analyzing both historical usage patterns and live performance indicators, these models can identify the most reliable and high-performing peers for content delivery, leading to improved data throughput and reduced transmission delays [8] [9]. Resource optimization remains central to the effectiveness of hybrid models. By harnessing distributed and edge computing resources, these systems can allocate tasks more strategically reducing latency and conserving energy while ensuring scalability under increasing demand.

11. Conclusion

This survey explored the current advancements in Peer-to-Peer (P2P) multimedia algorithms used for live streaming, with particular attention to improving Quality of Service (QoS) in systems that incorporate cloud technologies. The study found that hybrid models combining the decentralized nature of P2P networks with the reliability of cloud services and content delivery networks (CDNs) present an effective solution to ongoing issues such as latency, scalability, and overall system dependability. Artificial intelligence, especially in the form of reinforcement learning, plays a vital role in enabling systems to respond in real time. These intelligent algorithms help manage resources efficiently and adapt to changing network conditions, significantly enhancing the end-user's streaming experience. Additionally, the use of block chain and advanced cryptographic methods has proven valuable for securing decentralized platforms. These tools help verify content integrity and foster trust across peer networks. However, they also introduce challenges, such as added processing demands and potential delays. The adoption of 5G and edge computing technologies shows great promise in reducing latency by moving processing closer to users. Yet, the cost and complexity of deploying edge infrastructure remain major obstacles, particularly in resource-limited areas. Algorithmic developments like smart peer selection, adaptive bandwidth use, and flexible scheduling have led to better system performance, especially in handling unpredictable peer behavior and frequent network changes. These methods are critical to sustaining a stable and scalable streaming environment. Despite these advances, key issues persist. The field still requires practical testing frameworks, consistent performance evaluation methods, and enhanced security strategies that do not compromise speed or efficiency. Energy-aware algorithm design is another area in need of further research, particularly for large-scale applications.

References

[1] S. Kotusev, “Enterprise architecture and enterprise architecture artifacts: Questioning the old concept in light of new findings,” *J. Inf. Technol.*, vol. 34, pp. 102–128, 2019, doi: 10.1177/0268396218816273.

[2] F. Tommasi, V. De Luca, and C. Melle, “QoS monitoring in real-time streaming overlays based on lock-free data structures,” *Multimed. Tools Appl.*, vol. 80, pp. 20929–20970, 2021, doi: 10.1007/s11042-020-10198-9.

[3] M. Sina, M. Dehghan, and A. Rahmani, “CaR-PLive: Cloud-assisted reinforcement learning based P2P live video streaming: a hybrid approach,” *Multimed. Tools Appl.*, vol. 78, pp. 34095–34127, 2019, doi: 10.1007/s11042-019-08102-1.

[4] M. Efthymiopoulou, N. Efthymiopoulos, K. Deltouzos, and A. Christakidis, “Robust control in cloud assisted peer to peer live streaming systems,” *Pervasive Mob. Comput.*, vol. 42, pp. 426–443, 2017, doi: 10.1016/j.pmcj.2017.06.005.

[5] Y. Ding, Z. Wu, and L. Xie, “Enabling manageable and secure hybrid P2P-CDN video-on-demand streaming services through coordinating blockchain and zero knowledge,” *IEEE MultiMedia*, vol. 30, pp. 36–51, 2023, doi: 10.1109/MMUL.2022.3191680.

[6] S. Kim, “QoS-aware data forwarding architecture for multimedia streaming services in hybrid peer-to-peer networks,” *Peer-to-Peer Netw. Appl.*, vol. 8, pp. 557–566, 2014, doi: 10.1007/s12083-014-0256-4.

[7] A. Liem, I. Hwang, A. Nikoukar, C. Yang, M. Ab-Rahman, and C. Lu, “P2P live-streaming application-aware architecture for QoS enhancement in the EPON,” *IEEE Syst. J.*, vol. 12, pp. 648–658, 2018, doi: 10.1109/JSYST.2016.2535539.

[8] R. Farahani *et al.*, “RICHTER: Hybrid P2P-CDN architecture for low latency live video streaming,” in *Proc. 1st Mile-High Video Conf.*, 2022, doi: 10.1145/3510450.3517290.

[9] N. Dao, A. Tran, N. Tu, T. Thanh, V. Bao, and S. Cho, “A contemporary survey on live video streaming from a computation-driven perspective,” *ACM Comput. Surv.*, vol. 54, pp. 1–38, 2022, doi: 10.1145/3519552.

[10] A. Hadi *et al.*, “Obscuring transaction information in decentralized P2P wireless networks,” *IEEE Access*, vol. 11, pp. 111053–111067, 2023, doi: 10.1109/ACCESS.2023.3321960.

[11] A. Deshmukh and M. Pund, “Analyzing P2P cloud architectures: A research perspective,” *Int. J. Comput. Sci. Eng.*, vol. 6, no. 1, 2019, doi: 10.14445/23488387/ijcse-v6i1p101.

[12] A. Payberah, H. Kavalionak, V. Kumaresan, A. Montresor, and S. Haridi, “CLive: Cloud-assisted P2P live streaming,” in *Proc. IEEE 12th Int. Conf. Peer-to-Peer Comput. (P2P)*, 2012, pp. 79–90, doi: 10.1109/P2P.2012.6335820.

[13] M. Abadeer, R. Sadek, and G. Selim, “Real-time interactive mechanism for live streaming on peer-to-peer networks,” *Int. J. Comput.*, vol. 19, pp. 208–215, 2020, doi: 10.47839/IJC.19.2.1763.

[14] J. Zhang and Y. Zhang, “QoS-awareness peer coordination control for topology-converging P2P live streaming,” *Multimed. Tools Appl.*, vol. 76, pp. 23835–23858, 2016, doi: 10.1007/s11042-016-4092-9.

[15] A. Ramirez, L. Sardy, and F. Ramirez, “Tikuna: An Ethereum blockchain network security monitoring system,” *arXiv preprint*, arXiv:2310.09193, 2023, doi: 10.48550/arXiv.2310.09193.

[16] O. Jogunola *et al.*, “Comparative analysis of P2P architectures for energy trading and sharing,” *Energies*, vol. 11, no. 62, pp. 1–20, 2017, doi: 10.3390/EN11010062.

[17] B. Choi, J. Sohn, D. Han, and J. Moon, “Scalable network-coded PBFT consensus algorithm,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2019, pp. 857–861, doi: 10.1109/ISIT.2019.8849573.

[18] M. Saad *et al.*, “Exploring the attack surface of blockchain: A comprehensive survey,” *IEEE Commun. Surveys Tuts.*, vol. 22, pp. 1977–2008, 2020, doi: 10.1109/COMST.2020.2975999.

[19] Z., F. Giroire, G. Urvoy-Keller, and S. Rouibia, “When locality is not enough: Boosting peer selection of hybrid CDN-P2P live streaming systems,” 2021. [Online]. Available: (Publisher/DOI not provided).

[20] L. Provensi, F. Eliassen, and R. Vitenberg, “A cloud-assisted tree-based P2P system for low latency streaming,” in *Proc. Int. Conf. Cloud Autonomic Comput. (ICCAC)*, 2017, pp. 172–183, doi: 10.1109/ICCAC.2017.24.