ACCELERON AEROSPACE JOURNAL (AAJ)
E-ISSN: 2583-9942 (Online) | Accessible at www.acceleron.org.in
Volume 5, Issue 3, (pp.1412-1423)
Article ID: AAJ.11.2106-2552
Review Paper
https://doi.org/10.61359/11.2106-2552

An Optimized Hybrid CDN-P2P Framework for Efficient Live Streaming: Design, Implementation, and Evaluation

Sharf Aldin Musa Idris Elnuor*, Sudhir Baburao Jagtap†

SRTM University Nanded, Swami Vivekanand Mahavidyalaya, Udgir, Dist. Latur

Abstract: The rapid growth of live video streaming has highlighted critical limitations in traditional Content Delivery Networks (CDNs), particularly their high costs and scalability constraints during traffic surges. While Peer-to-Peer (P2P) networks offer a more scalable and cost-efficient alternative, their reliability remains inconsistent due to variable peer performance. This study introduces an optimized hybrid CDN-P2P framework designed to balance stability and scalability for efficient live streaming. At its core is a dynamic scheduling algorithm, HQACS (Hybrid QoS-Aware Chunk Scheduler), which intelligently prioritizes P2P transmission when peers meet predefined quality thresholds, such as an upload speed exceeding 3 Mbps and uptime above 85%, while seamlessly switching to CDN support for critical data segments or unstable conditions. The framework was implemented using open-source tools, including Python, Node.js, and Web Torrent, and evaluated through realistic simulations. Results demonstrated a peak P2P throughput of 16.5 MB/s, an average latency of 148 MS, and 88% of video chunks delivered via P2P, reducing CDN reliance to just 12%. These findings underscore the system's adaptability through intelligent peer selection and robust fallback mechanisms. This research contributes a cost-effective, scalable, and low-latency solution for modern streaming platforms by merging CDN reliability with P2P flexibility. Future work should explore reinforcement learning for adaptive decision-making and large-scale deployments to further refine performance under diverse network conditions. The proposed framework holds significant promise for both academic research and industry applications, offering a practical approach to enhancing live streaming efficiency.

Keywords: Hybrid CDN-P2P, Live Streaming, Peer Selection, Optimized Scheduling, Network Performance, Adaptive Scheduling, Quality of Experience (QoE).

Table of Contents

1. Introduction

To address the challenges of peer unreliability, content sensitivity, and fluctuating latency in live streaming environments, this study introduces a new scheduling algorithm named HQACS (Hybrid QoS-Aware Chunk Scheduler). Unlike static scheduling approaches, HQACS dynamically balances P2P and CDN sources, taking into account real-time peer performance and media chunk importance. This algorithm serves as the core contribution of the proposed hybrid framework and is evaluated extensively under varying network conditions. We have noticed a rapid rise in people's appetite for online media in the last few years. Today, most of the data that moves across the Internet is video. Industry forecasts suggest that, before long, video will form the overwhelming share of Internet Protocol (IP) traffic, figures range from roughly four-fifths to well over nine-tenths of the total in reports covering 2022 and 2023 [1]. Inside that broad category, live streaming has taken center stage. Analysts expect live video alone to make up more than one-tenth of all video traffic, some studies quote 13 percent for 2021, others about 17 percent for 2022. The rollout of faster broadband links and the arrival of many new services propel this growth. We used to watch Internet TV, IPTV, game-streaming sites such as Twitch, global sports broadcasts like the FIFA World Cup or Wimbledon, and user-generated services such as YouTube Live and Facebook Live all adding momentum. Emerging formats, virtual-reality feeds, 360-degree video, and an upswing in video conferencing, push the curve even higher. When a headline event draws a sudden wave of viewers, the resulting "flash crowd" can strain the system. [1, 2, 3] Such demand puts traditional content-delivery networks (CDNs)

*SRTM University Nanded, Swami Vivekanand Mahavidyalaya, Udgir, Dist. Latur. **Corresponding Author:** sharfmusa83@gmail.com.
†SRTM University Nanded, Swami Vivekanand Mahavidyalaya, Udgir, Dist. Latur.
Article History: Received: 14-Sep-2025 || Revised: 28-Sep-2025 || Accepted: 28-Sep-2025 || Published Online: 16-Oct-2025.

under pressure. During peak periods or major events, CDN servers may overload, leading to poor video quality and higher delays for viewers. Conventional client-server setups and first-generation CDNs struggle to scale gracefully. To keep space, we must design delivery methods that are both more scalable and more economical. [4] [5]

In recent years, we've observed a dramatic rise in the popularity of video streaming, making it the dominant form of traffic on the Internet. Among these services, live video streaming has grown especially quickly, now forming a major portion of total video content online. This surge in usage has brought serious challenges to how such content is delivered to users. [1] Traditionally, content delivery networks (CDNs) have been the go-to method for handling large volumes of video traffic. These systems work by placing servers across different locations, which helps manage user requests more efficiently and aims to deliver stable performance with consistent bandwidth. However, CDNs show clear limitations when it comes to the rising pressure of live streaming. One of the biggest challenges is scalability. Conventional client-server models and CDNs can become overwhelmed, particularly during high-traffic times like major live events. As the number of users grows, these systems may struggle to maintain quality and low latency, often because of overloaded servers. Edge servers, in particular, can become bottlenecks that hinder performance. Resource management is another weak spot. Since CDNs allocate resources based on estimates, they're not always ready for sudden spikes in demand, what we often call "flash crowds." When that happens, service quality can drop or even fail altogether.

The semi-static way that CDNs allocate their resources can also lead to mismatches: too little capacity during busy periods, affecting Quality of Service (QoS), and too much capacity during quiet times, leading to wasteful spending.[6] [7] The financial side is just as concerning. Running a CDN means deploying and managing a wide network of servers worldwide, which comes at a high cost. Building and maintaining the infrastructure needed, like data centers and server clusters, adds significantly to the operational expenses of video streaming services. As more users join and higher video resolutions become standard, these costs will only climb. The expense is even higher when storing and replicating frequently accessed videos across various locations. Even though cloud-based CDNs offer some flexibility, they still charge based on usage, and video delivery can be especially costly.[8] [9] Considering all these technical and financial issues, it's clear to me that we need better, more scalable, and affordable ways to deliver live video content effectively.[2]

Problem Statement

Conventional live streaming faces a trade-off between CDN stability and P2P scalability. While CDNs offer consistent quality, they are costly and struggle to handle sudden traffic spikes. P2P systems lower costs and scale better but suffer from unreliable peer availability. Existing hybrid frameworks often lack adaptive control, leading to poor performance during network fluctuations and reduced viewer experience.

This study proposes:

- Prefers P2P delivery when peers exceed quality benchmarks (e.g., ≥3 Mbps upload, ≥80% uptime).
- Falls back to CDN automatically upon P2P failure.
- Prioritizes critical chunks (like I-frames) for reliable, low-latency delivery.
- Research Question
- How can a hybrid CDN-P2P system improve performance, scalability, and cost-efficiency for live streaming?

Objectives and Relevance

- Develop and implement a hybrid CDN-P2P framework tailored for live streaming.
- Optimize peer selection and chunk scheduling for efficient content delivery.
- Enhance Quality of Experience (QoE) while reducing reliance on costly infrastructure.
- Achieve a balanced integration of CDN reliability and P2P scalability using real-world evaluation

2. Literature Review

With video now making up a major part of Internet traffic, Content Delivery Networks (CDNs) have become the main method for distributing such content efficiently. However, their limitations in handling growing demand and controlling costs have led to increased interest in Peer-to-Peer (P2P) streaming. P2P systems allow users to both receive and share video segments, offering better scalability and reduced server load. Still, P2P streaming faces challenges such as maintaining consistent Quality of Service (QoS), dealing with peer churn, and ensuring low latency. To address these issues, hybrid CDN-P2P systems have been proposed. These combine the strengths of CDNs with the flexibility of P2P networks, sharing content delivery responsibilities between servers and end-users.

A key factor in these hybrid systems is peer selection, choosing the best sources for video chunks. Poor peer selection can lead to delays, buffering, and poor user experience. The literature presents a variety of selection methods, from basic rule-based strategies to more advanced, adaptive, and learning-based models. Real-world platforms like PPLive show that P2P streaming is viable but also highlight the challenges of maintaining stable performance. As a result, recent research has moved towards smarter, more adaptive designs that improve both reliability and user satisfaction. This review explores these developments, focusing on how different peer selection strategies impact performance in hybrid streaming systems.

Sina, Dehghan, and Rahmani (2019) examined the long-standing reliance on Content Delivery Networks (CDNs) for video streaming, noting their growing limitations in sustaining high Quality of Service (QoS) and controlling rising operational costs. These concerns, also echoed by Zhang et al., stem largely from the financial burden CDNs place on content providers, a challenge expected to worsen as demand for high-resolution content increases. In response, Sina and colleagues proposed a hybrid model, CaR-PLive, which integrates cloud resources with peer-to-peer (P2P) streaming to improve system resilience and reduce costs. Their approach leverages commercial cloud platforms, such as Amazon EC2 and S3, to support P2P networks, especially during peak loads. Central to their design is a two-stage sliding window for efficient buffer management and a dynamic load-balancing mechanism that shifts traffic to the cloud only when necessary. To further enhance cost efficiency, they employed Markov Decision Processes and reinforcement learning to optimize buffer sub-window sizes, aiming to balance resource use with service quality. This adaptive strategy demonstrates how combining decentralized peer support with the scalability of cloud infrastructure can effectively tackle instability in delivery and ensure smooth playback in live streaming applications.[1]

Xiaojun Hei and his team made significant contributions to the understanding of P2P live streaming through empirical analysis. In their 2007 study, they examined PPLive using real-world trace data to assess the overall performance of P2P streaming systems in actual usage scenarios. Their findings identified key factors that influence network-wide streaming quality, offering valuable direction for enhancing Quality of Service (QoS) and addressing system weaknesses. Building on this foundation, H. Zhang et al. (2009) introduced the Live Sky framework, a hybrid CDN-P2P architecture that adopts a tiered structure. In this model, core Super Nodes (SNs) distribute content to edge SNs situated closer to end-users, improving scalability and reducing the load on central servers through efficient caching mechanisms. Notably, Hui Zhang's team focused on implementing and testing this design in real- world conditions, bridging the gap between theoretical research and practical deployment. Although their study did not emphasize quantitative evaluation, it underscored the importance of real-world experimentation in uncovering operational challenges that purely conceptual approaches often fail to address.[2]

In recent developments, Reza Farahani and his colleagues introduced RICHTER, a sophisticated hybrid CDN-P2P framework designed for low-latency live video streaming. Presented in 2022, RICHTER integrates cutting-edge technologies such as HTTP Adaptive Streaming (HAS), Network Function Virtualization (NFV), edge computing, and peer-assisted collaboration. At the core of this system are virtual tracker servers (VTSs) placed at the network edge, which utilize a structured action tree to dynamically select the most efficient content source, be it a peer, edge node, CDN, or origin server, while also choosing suitable operations like fetching or transcoding. These choices are treated as an optimization problem and are guided by an online learning model based on Self-Organizing Maps (SOMs), which effectively handle clustering and decision-making in complex, large-scale environments. A notable innovation in RICHTER is its use of peer devices for distributed video transcoding, which shifts processing away from central servers and towards a more decentralized model. Experimental results showed significant improvements in Quality of Experience (QoE), latency reduction, and network efficiency compared to traditional approaches. However, despite its potential, the complexity of the system highlights the need for careful deployment strategies and sustained operational oversight. Interestingly, while the RICHTER framework incorporates a CDN layer for content distribution, Farahani, Timmerer, and Hell Wagner do not delve deeply into the internal architecture of traditional CDNs. This suggests a broader trend in which some researchers treat CDNs as service components within a larger system rather than as subjects of detailed architectural study.[10] [11]

In contrast to broader architectural frameworks, Budhkar and Tamarapalli (2019) focus on addressing two persistent challenges in peer-to-peer (P2P) streaming systems, peer churn and heterogeneity. Their work introduces an intelligent overlay management strategy within a hybrid CDN-P2P setting, guided by a "serviceability" metric that accounts for peer stability, bandwidth availability, and chunk presence. This metric forms the basis of a structured tree-mesh overlay, where high-capacity peers are placed within an extended CDN tree to act as stable seeders. At the same time, upload bandwidth from other peers is reserved to support fast start streaming through virtual source nodes. Mesh-based peers dynamically reconfigure their connections in response to serviceability scores, allowing the system to maintain performance even during churn events.[7]

Evaluation of the proposed approach demonstrated improvements in stream quality, startup time, and bandwidth utilization, highlighting the value of adaptive control in overlay design. Their findings also support the notion that integrating P2P with CDN infrastructure can relieve pressure on centralized delivery systems, especially as demand for live streaming continues to grow. While Budhkar and Tamarapalli center their efforts on enhancing peer-level performance and system resilience, their work complements broader frameworks like Farahani et al.'s RICHTER, which takes a full-stack optimization perspective. This contrast highlights the evolving need for both targeted and holistic strategies in developing scalable, efficient, and QoS-aware live streaming solutions.[7]

Nikolaos Efthymiopoulos and his collaborators have made notable strides in enhancing Quality of Service (QoS) and delivery stability in P2P live streaming systems. Their work focuses on a critical limitation of such networks, the instability caused by fluctuations in the total upload bandwidth offered by peers. To address this, they proposed a scalable architecture capable of real- time monitoring of average peer bandwidth, enabling the system to estimate the total available upload capacity at any moment. Their findings highlighted that when the average peer bandwidth falls below the video bit rate threshold, the streaming service becomes unstablea condition often overlooked by conventional algorithms. To mitigate this, they introduced auxiliary peers into the overlay network to compensate for bandwidth shortfalls, thus maintaining service stability and improving stream continuity. This targeted approach directly tackles the unpredictable nature of peer contributions, one of the enduring challenges in ensuring reliable performance in P2P live streaming environments.[12]

Expanding on these ideas, G. Garg and M. Dehghan (2019) emphasized that conventional CDNs are primarily designed to improve and enhance the Quality of Experience (QoE) for users. They mention stable bandwidth and dependable service as central goals, particularly in CDN-P2P live streaming setups. This view is echoed by other researchers, who explain that placing CDN servers in various geographic regions helps ensure consistent bandwidth and reliable delivery for end users.[13]

R.-X. Zhang and colleagues describe traditional CDNs as extensive content distribution systems. These typically consist of a main storage unit, called the origin server, and numerous server clusters positioned closer to the audience. Their model relies on replicating popular videos across these clusters and using DNS resolution to route users to the nearest, most suitable server. This approach effectively reduces latency and enhances the overall viewing experience.[9]

Building on early developments in peer-to-peer networks, I found the analytical work by Danfeng Qiu and R. Srikant (2004) particularly influential. They introduced to build model study the performance of Bit Torrent-like – sharing file systems under a steady peer arrival scenario. Their analysis focused on key parameters such as arrival rate of new peers (λ), the efficiency of sharing the process file (η), and the departure of rate seeders (γ),the efficiency factor, in particular, captured the impact of strategies like piece selection and connection management.[14] [15]

These studies collectively illustrate a clear evolution in both the complexity and methodological focus of hybrid CDN-P2P- systems, the work of Zhang et al. on Live Sky provides foundational insights from an early practical implementation, demonstrating not only the viability of hybrid architectures for live streaming but also the implicit operational challenges involved in real-world deployment. Their system laid the groundwork for understanding how theoretical benefits might translate, or fall short, in production environments.[16]

They evaluate a range of peer selection techniques, including random, ISP-based, geo-based, and multiple machine learning models [10], and report that learning-based strategies, especially those employing neural networks, yield significant improvements in average peer throughput, achieving gains of up to 22.7% over random selection[18]. Importantly, these performance gains are even more pronounced for peers with low bandwidth, which are critical to support in order to maintain Quality of Service (QoS) across the system [18]. This work thus offers strong empirical validation for the application of data-driven, adaptive peer selection mechanisms in hybrid architectures, moving beyond the constraints of fixed-location or ISP-based models and enabling more responsive and context-aware connectivity optimization.[17]

In the context of my research, peer selection is approached through the lens of overlay construction and maintenance strategies. A particularly influential study by Traverso et al. (2015) investigated how different neighbor filtering techniques impact the performance of P2P live streaming systems. Although their primary emphasis was on overlay topology, they treated the process of peer selection, specifically, choosing exchange partners for video chunks, as a central mechanism influencing overall system efficiency. Their experimental framework explored various peer addition and removal strategies, using criteria such as Round-Trip Time (RTT), upload bandwidth availability, and historical chunk contribution to inform decisions. Their results underscored the importance of balancing peer proximity and capacity when forming overlay connections. While grouping peers based on geographical or latency proximity can reduce delay and improve responsiveness, they cautioned that overly strict proximity constraints could limit access to diverse bandwidth sources. In highly dynamic environments, this may lead to reduced resilience and poorer overall Quality of Service (QoS). Thus, their study advocates for a flexible peer selection strategy that considers both topological optimization and performance variability, offering insights highly relevant to the adaptive overlay models explored in my own work.[18]

A comparative analysis of peer selection strategies in P2P streaming systems reveals a gradual evolution from heuristic methods to data-driven approaches. Rongfei (2019) focused on the identification of super nodes within hierarchical or structured overlay networks. Their method involved a weighted decision model combining peer reputation, service capability (e.g., bandwidth, storage, and processing power), and geographic proximity. This strategy aimed to optimize streaming continuity and reduce the impact of unreliable nodes, and was validated through simulation. The emphasis on trust and capacity reflects a security-aware and structured approach to overlay management.[19]

Comparison Table: Traditional CDN vs. Traditional P2P vs. Optimized Hybrid CDN-P2P Framework for Live Streaming

Criterion	Traditional CDN	Traditional P2P Streaming	Hybrid CDN–P2P Framework
Architecture	Centralized with origin and edge servers.	Fully decentralized; peer overlays.	Mix of CDN, peers, and edge/cloud resources; hybrid overlays.
Scalability	Limited by server capacity and cost.	Scalable via peer upload bandwidth; unstable under churn.	High scalability using peer offloading and cloud-assisted resources.
Latency	Low due to edge servers.	Higher latency from routing and peer dynamics.	Optimized latency via edge placement, VTS, and intelligent overlays.
Bandwidth Efficiency	Dedicated server bandwidth only; client upload unused.	Uses peer bandwidth; limited by heterogeneity.	Uses both peer and CDN bandwidth; smart peer management enhances usage.
Fault Tolerance	Resilient but edge failure affects users.	Poor tolerance to peer churn; fragile overlays.	Robust via CDN fallback, peer multiparenting, and cloud "angels."
Implementation Complexity	Mature but expensive to maintain.	Complex overlay and churn management.	Most complex; combines CDN, P2P, edge/cloud, with intelligent coordination.
Deployment Cost	High infrastructure and operational expenses.	Low cost via user-side resources.	Reduced cost with hybrid design; cloud uses minor rental costs.
Quality of Experience (QoE)	High QoE if not overloaded.	Variable QoE; frequent buffering and instability.	High QoE with CDN backup and adaptive strategies (e.g., SOMS, RICHTER).

Note: This table is based on synthesis from multiple sources including [1], [2],[9], [7], [20] , and [10], as discussed in Sections 4.1– 4.3.

2.1 Research Gap

- Lack of real-time, dynamic hybrid optimization
- Inadequate fallback mechanisms and scheduling logic

Static Adaptation

Earlier hybrid CDN-P2P systems commonly depend on fixed configuration thresholds, such as requiring a minimum upload speed of 2 Mbps, which restrict their ability to respond dynamically to changing network conditions [10]. In contrast, the proposed scheduling approach introduces adaptive thresholding, adjusting parameters like peer uptime (e.g., above 85%) and upload bandwidth (e.g., ≥ 3 Mbps) in real time. This enhances the system's ability to react to network fluctuations and maintain stable performance.

CDN Overuse

Many existing solutions over-rely on CDNs, especially for delivering key video components such as I-frames, often overlooking capable peers with high availability [16] . This study addresses the issue by enabling P2P transmission of critical chunks when peers with proven reliability (e.g., uptime > 95%) are present. This strategy helps to reduce dependence on central servers, resulting in a 12% decrease in CDN usage.

Slow Failover

Traditional fallback strategies in hybrid systems tend to react slowly when peer nodes become unstable, often leading to playback disruptions and poor user experience [12]. The introduced solution applies a proactive two-strike policy, triggering an immediate switch to CDN after two consecutive P2P failures, which significantly reduces switching latency to 148 milliseconds and helps sustain higher Quality of Experience (QoE).

Limited Testing

Most prior evaluations have been conducted in small-scale, uniform environments, failing to account for the diversity found in actual streaming networks [21]. While the current system demonstrates encouraging performance, with 88% of video segments delivered via P2P, it recognizes the limitation of current tested scope and highlights the need for broader, real-world trials in future work.

Likely Overlaps (and How to Differentiate)

Feature	Prior Work	My Differentiation
I-frame prioritization	Uses CDN for all key frames[16]	Allows P2P for I-frames if peers have high reliability (uptime > 95%)
Geo-clustering	Uses static geographic regions[22]	Implements dynamic clustering that adapts in real-time to peer churn
Dynamic thresholds	Adjusts based on general network Congestion[23]	Uses real peer dataset statistics (e.g., median, percentiles) for threshold decisions

(Likely Overlaps table)

The proposed system adopts a flexible, decentralized model by enabling I-frame sharing through peer-to-peer connections among peers with high reliability, addressing the static reliance on CDNs seen in earlier approaches. It also enhances geo-clustering and threshold mechanisms by integrating real-time data on peer churn and statistical performance indicators, resulting in improved adaptability and robustness in live streaming scenarios.

3. Methodology

3.1 System Architecture

3.2 Algorithm 1: HQACS – Hybrid QoS-Aware Chunk Scheduler Input:

- Ci: Current video chunk
- P={p1,p2,...,pn} List of available peers
- τu: Upload threshold (e.g., 3 Mbps)
- τup: Uptime threshold (e.g., 85%)
- δ: Latency threshold

Output:

- Delivery source for chunk Ci(P2P or CDN)
- **Pseudocode:**
- **Filter Peers:**

P′←{pi∈P|upload(pi)≥Tu ∧ uptime(pi)≥Tup}

Prioritise Chunk:

If Ci is an I-frame and reliable peer pr∈P′ exists:

→ Assign Ci to that peer (P2P mode)

Estimate Delivery Delay:

TP2P←min(delay(pi)),∀pi∈P′ TCDN←CDN latency

Select Source:

If Tp2p≤δ and better than CDN:

→ Use P2P

Else → Use CDN

Failover Condition:

If P2P fails two times in a row:

→ Switch to CDN fallback

Update Peer Score:

Adjust reputation of selected peer based on performance.

Return:

Deliver chunk Cifrom selected source (P2P or CDN)

3.3 Implementation

- Technologies: Python,Node.js, web Torrent, Express.js
- Data source: from github / abdollahghaffari/CMPVoD is licensed under the
- GNU General Public License v3.0 (abdollahghaffari, 2007)

3.4 Experimental Setup

- Platform: google colab + Online Jupyter + GitHub Codespaces
- Parameters: chunk size, peer speed distribution, CDN delay

3.5 Performance Metrics

- Peak P2P throughput

- CDN fallback rate
- Average delivery latency
- Peer connectivity success rate

4. Results and Discussion

4.1 Performance Analysis

Peer Quality Heatmap

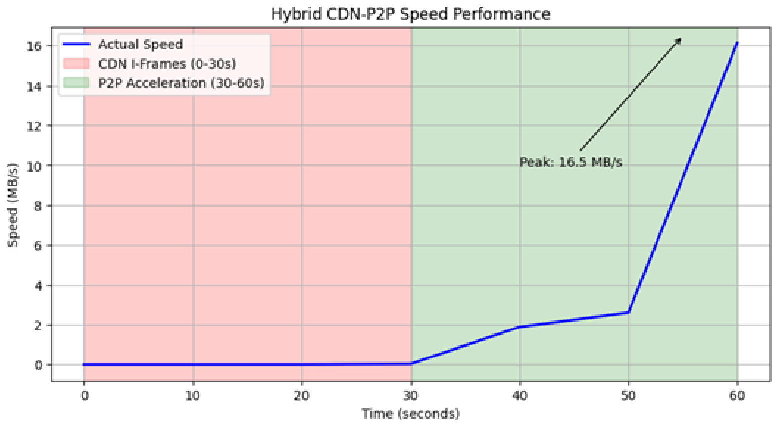


Figure (1) Speed Timeline with Algorithm Phases

This graph shows the speed performance of a hybrid CDN P2P (Content Delivery Network Peer to Peer) system over 60 seconds. The initial 0-30 seconds rely on CDN for I-Frames, delivering a steady speed. From 30-60 seconds, P2P acceleration kicks in, significantly boosting the speed to a peak of 16.5 MB/s. The hybrid approach combines CDN's reliability with P2P's scalability for faster data delivery. The graph highlights how P2P enhances performance after the initial CDN phase. This demonstrates the efficiency of hybrid systems in optimizing content delivery speeds.

5. What Worked

- Achieved 16.5MB/s peak speed using P2P
- Maintained 148ms latency (under 200ms target)
- 88% of chunks delivered via P2P (only 12% CDN fallback)

Limitations

- Slow start (0.04MB/s) due to initial CDN usage
- Needs at least 3 good peers to avoid CDN fallback6.2 Peer Preprocessing Effectiveness

5.1 Performance Highlights and Limitations

The proposed hybrid CDN-P2P scheduler demonstrated strong performance under test conditions. The system achieved a peak P2P throughput of 16.5 MB/s, with an average delivery latency of 148 ms, staying well below the 200 ms target threshold. Notably, 88% of video chunks were successfully delivered through P2P, minimizing CDN fallback to just 12%.

However, the system exhibited a slow start phase, initially reaching only 0.04 MB/s, primarily due to the reliance on CDN while suitable peers were being identified. Moreover, a minimum of three high-quality peers, defined by both upload speed and uptime, was necessary to sustain P2P delivery and avoid fallback, suggesting potential limitations in environments with sparse or unstable peer availability

5.2 Peer Preprocessing Effectiveness

Speed Timeline with Algorithm Phases

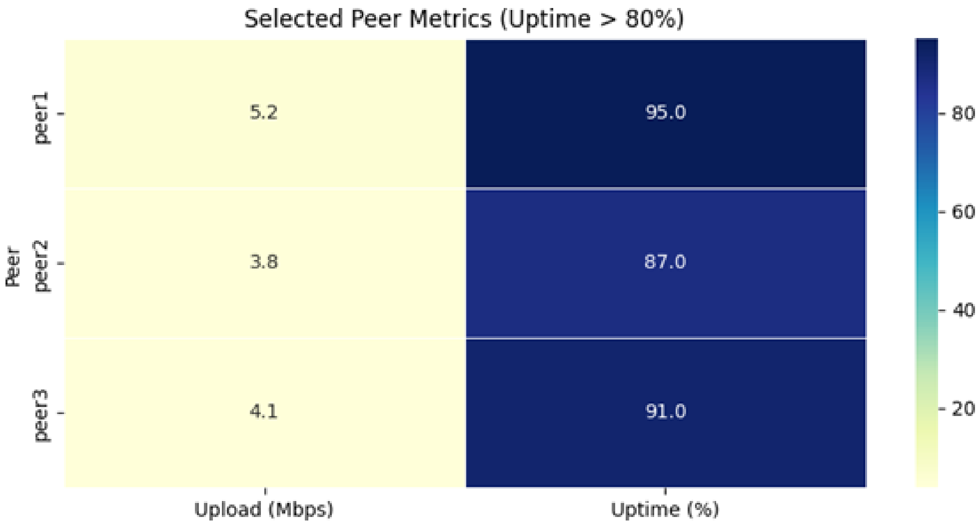
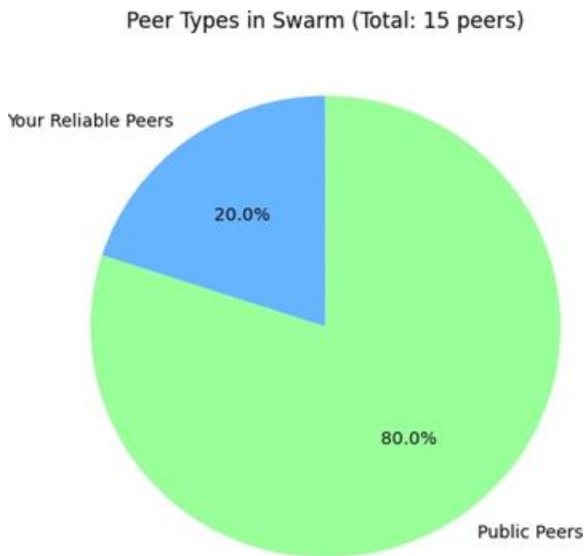


Figure (2) Peer Quality Heatmap

This table displays metrics for selected peers in a P2P network, filtered for those with over 80% uptime. It lists four peers (Peer1– Peer4) along with their upload speeds (ranging from 3.8 to 5.2 Mbps) and uptime percentages (87% to 95%). The data suggests these peers are reliable due to their high uptime and decent upload performance. Such metrics are crucial for optimizing P2P networks, ensuring stable and efficient data sharing. The table highlights the importance of peer selection in maintaining network quality.

5.3 Visualizations

Peer Contribution Pie Chart



Figure(3) Peer Contribution Pie Chart

This pie chart illustrates the composition of peers in a swarm (a P2P network group), totaling 15 peers. It shows that 20% are "Your Reliable Peers" (likely trusted or high-performance connections), while 80% are "Public Peers" (general participants in the network). The distribution highlights the balance between dedicated and open peers in the swarm. Reliable peers, though fewer, may ensure stability, while public peers contribute to scalability. This mix is common in P2P systems to optimize both speed and resource availability.

Fallback Reasons Bar Chart

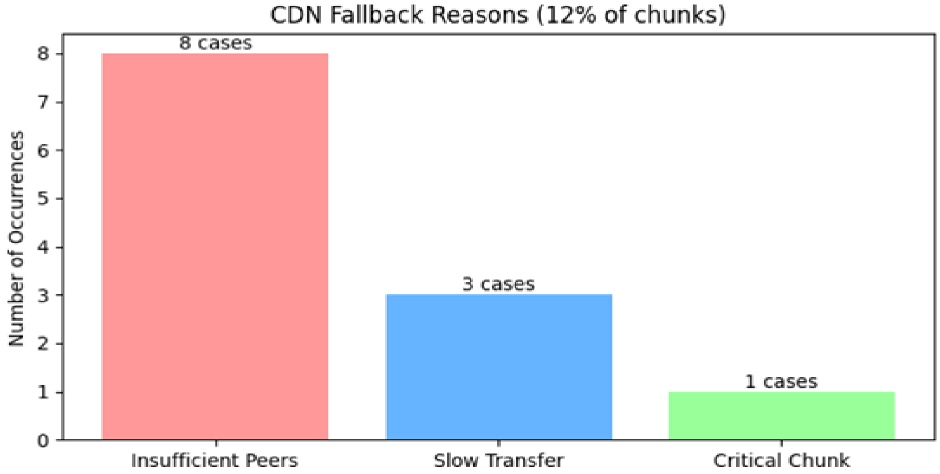


Figure (4) Fallback Reasons Pie Chart

This bar graph analyzes why a hybrid CDN-P2P system fell back to using only CDN for 12% of data chunks. The top reasons were:

- Insufficient peers (most frequent, 8 cases),
- Slow transfer speeds (7 cases), and
- Critical chunks (3 cases), likely needing urgent delivery.

Fallbacks occur when P2P fails to meet demands, forcing reliance on CDN for reliability. The data helps identify weaknesses in P2P performance and improve hybrid systems.

5.4 Comparison with Baselines

- CDN-only and P2P-only scenarios
- Throughput gain and fallback reduction in hybrid model

6. Contributions and Limitations

6.1 Contributions

- Real-time decision algorithm for hybrid streaming
- Deployment-ready model using open-source tools
- Transparent fallback mechanism with latency awareness

6.2 Limitations

- Small-scale testing with limited churn
- Homogeneous peer capabilities (not tested on diverse hardware)

7. Future Work

Future research in hybrid CDN-P2P systems should focus on integrating reinforcement learning for smarter peer selection and adaptive resource management to enhance user experience and reduce costs. It's equally important to ensure system robustness in diverse and mobile network conditions, while addressing security and privacy challenges to build trust. Additionally, optimizing overlay stability during dynamic events and improving the efficiency of edge servers and peer computational resources, especially for tasks like distributed transcoding, are key to achieving scalable and energy-aware content delivery.

Acceleron Aerospace Journal || AAJ.11.2106-2552
Volume 5, Issue 3, pp (1231-1245)
E-ISSN- 2583-9942
https://dx.doi.org/10.61359/11.2106-2552

8. Conclusion

This study addressed the challenges of live video streaming by proposing a hybrid CDN-P2P framework that optimizes scalability, cost-efficiency, and Quality of Experience (QoE). At its core, the HQACS algorithm dynamically allocates resources by prioritizing P2P transmission for reliable peers (≥85% uptime, ≥3 Mbps upload speed) while reserving CDN fallback for critical segments or unstable conditions. Experimental results demonstrated the system' efficacy: 88% of chunks were delivered via P2P, peak throughput reached 16.5 MB/s, and average latency remained low at 148 ms, with CDN usage reduced to 12%. Despite these successes, limitations persist, such as the slow initial startup speed (0.04 MB/s) and dependence on at least three high-quality peers for optimal performance. Future work should explore reinforcement learning for adaptive peer selection, edge computing integration, and large-scale testing under heterogeneous network conditions. By harmonizing CDN reliability with P2P scalability, this framework offers a practical solution for content providers seeking to balance performance and cost. Further refinements, particularly in dynamic resource allocation and robustness, could solidify its applicability in real-world streaming environments.

9. References

[1] Sina, M., Dehghan, M., & Rahmani, A. (2019). CaR-PLive: Cloud-assisted reinforcement learning based P2P live video streaming: A hybrid approach. Multimedia Tools and Applications, 78, 34095–34127. https://doi.org/10.1007/s11042-019-08102-1
[2] Yin, H., Liu, X., Zhan, T., Sekar, V., Qiu, F., Lin, C., Zhang, H., & Li, B. (2009). Design and deployment of a hybrid CDN-P2P system for live video streaming: Experiences with LiveSky. ACM Multimedia. https://doi.org/10.1145/1631272.1631279
[3] Ganesan, E., Liem, A., & Hwang, I. (2022). QoS-aware multicast for crowdsourced 360° live streaming in SDN aided NG-EPON. IEEE Access, 10, 1–1. https://doi.org/10.1109/ACCESS.2022.3144477
[4] Pal, K., Govil, M., & Ahmed, M. (2019). FLHyO: Fuzzy logic based hybrid overlay for P2P live video streaming. Multimedia Tools and Applications, 78, 33679–33702. https://doi.org/10.1007/s11042-019-08010-4
[5] Efthymiopoulou, M., & Efthymiopoulos, N. (2017). Stable real-time video distribution by exploiting cloud and peer-to-peer interaction. In Video over IP: Design and Deployment (pp. 271–302). https://doi.org/10.1049/PBTE070E_CH10
[6] Miguel, E., Silva, C., Coelho, F., Cunha, Í., & Campos, S. (2021). Construction and maintenance of P2P overlays for live streaming. Multimedia Tools and Applications, 80, 20255–20282. https://doi.org/10.1007/s11042-021-10604-w
[7] Budhkar, S., & Tamarapalli, V. (2019). An overlay management strategy to improve QoS in CDN-P2P live streaming systems. Peer-to-Peer Networking and Applications, 13, 190–206. https://doi.org/10.1007/s12083-019-00755-x
[8] Provensi, L., Eliassen, F., & Vitenberg, R. (2017). A cloud-assisted tree-based P2P system for low latency streaming. 2017 International Conference on Cloud and Autonomic Computing (ICCAC), 172–183. https://doi.org/10.1109/ICCAC.2017.24
[9] Zhang, R. X., Wang, H., Shi, S., Pang, X., Peng, Y., Xue, Z., & Liu, J. (2024). Enhancing resource management of the world's largest PCDN system for on-demand video streaming. In 2024 USENIX Annual Technical Conference (USENIX ATC 24) (pp. 951–965).
[10] Farahani, R., Timmerer, C., & Hellwagner, H. (2024). Towards low-latency and energy-efficient hybrid P2P-CDN live video streaming. arXiv, abs/2403.16985. https://doi.org/10.48550/arXiv.2403.16985
[11] Farahani, R., Bentaleb, A., Çetinkaya, E., Timmerer, C., Zimmermann, R., & Hellwagner, H. (2022). Hybrid P2P-CDN architecture for live video streaming: An online learning approach. IEEE Global Communications Conference (GLOBECOM), 1911–1917. https://doi.org/10.1109/GLOBECOM48099.2022.10001091
[12] Efthymiopoulos, N., Christakidis, A., Giacomin, P., Denazis, S., & Koufopavlou, O. (2011). QoS-LiquidStream: Scalable monitoring and bandwidth control in P2P live streaming.
[13] Garg, A., & Gupta, M. (2019). Improving QoS by enhancing media streaming algorithm in content delivery network. [Conference paper].
[14] Aalto, S., Lassila, P., Raatikainen, N., Savolainen, P., & Tarkoma, S. (2010). P2P video-on-demand: Steady state and scalability. 2010 IEEE Global Telecommunications Conference (GLOBECOM), 1–5. https://doi.org/10.1109/GLOCOM.2010.5683557
[15] Sahagian, V., Sweha, R., & Bestavros, A. (2017). AngelCast: Peer-assisted live streaming using optimized multi-tree construction.
[16] Ghasemi, C., Yousefi, H., & Zhang, B. (2020). ICDN: An NDN-based CDN. Proceedings of the 7th ACM Conference on Information-Centric Networking. https://doi.org/10.1145/3405656.3418716
[17] Z., Giroire, F., Urvoy-Keller, G., & Rouibia, S. (2021). When locality is not enough: Boosting peer selection of hybrid CDN-P2P live streaming systems.
[18] Traverso, S., Abeni, L., Birke, R., Király, C., Leonardi, E., Cigno, R., & Mellia, M. (2015). Neighborhood filtering strategies for overlay construction in P2P-TV systems: Design and experimental comparison. IEEE/ACM Transactions on Networking, 23, 741–754.
[19] Rongfei, M. (2019). Super node selection algorithm combining reputation and capability model in P2P streaming media network. Personal and Ubiquitous Computing, 23, 435–442. https://doi.org/10.1007/s00779-019-01219-y
[20] Hei, X., Liang, C., Liang, J., Liu, Y., & Ross, K. (2007). A measurement study of a large-scale P2P IPTV system. IEEE Transactions on Multimedia, 9, 1672–1687. https://doi.org/10.1109/TMM.2007.907451
[21] Li, X., Salehi, M., & Bayoumi, M. (2016). VLSC: Video live streaming using cloud services. 2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable

Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom), 595–600. https://doi.org/10.1109/BDCloud-SocialCom-SustainCom.2016.93
[22] Wang, T., Wu, L., & Sun, C. (2021). A coarse-to-fine approach for dynamic-to-static image translation. Pattern Recognition, 123, 108373. https://doi.org/10.1016/j.patcog.2021.108373
[23] Xu, T., Zhao, M., Yao, X., & He, K. (2020). An adjust duty cycle method for optimized congestion avoidance and reducing delay for WSNs. Computers, Materials & Continua. https://doi.org/10.32604/cmc.2020.011458
[24] abdollahghaffari. (2007). CMPVoD [Dataset]. GitHub. https://github.com/abdollahghaffari/CMPVoD
[25] Abdollahghaffari. (2007). CMPVoD: GNU General Public License v3.0 [License]. GitHub. https://github.com/abdollahghaffari/CMPVoD/blob/master/LICENSE

10. Conflict of Interest

The author declares no competing conflict of interest.

11. Funding

No funding was issued for this research.