WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Edge-Optimized Adaptive Streaming: Enhancing MPEG-DASH Performance with Context-Aware Delivery and ABR Logic

Shyam Sunder Saini [1]*, Lalit Sen Sharma[2]

[1,2]Department of Computer Science and IT, University of Jammu, Baba Saheb Ambedkar Road, Jammu, 180006, India

A B S T R A C T

The rise in demand for low-latency, high-resolution video streaming has brought to light fundamental shortcomings in traditional Dynamic Adaptive Streaming over HTTP (MPEG-DASH) systems under mobile and bandwidth-limited conditions. Traditional CDN-based architectures are usually plagued by large startup delays, re-buffering, and bitrate switching caused by centralized processing and narrow contextual awareness. This paper introduces an end-to-end edge-centric framework that re-architects the DASH pipeline to take advantage of cutting-edge edge computing features. The framework combines just-in-time transcoding, dynamic edge-generated MPDs, ABR advisories based on reinforcement learning, predictive prefetching, and low-latency chunk delivery under a common architecture. It also includes a continuous QoE (Quality of Experience) feedback loop to adapt strategies in real-time. By moving content processing and decision-making nearer to end users, the new design is expected to lower latency, increase quality stability, and improve overall user experience. Apart from its technical merit, this framework has the importance of being society-oriented as it facilitates equal access to quality video services even in areas with scarce network infrastructure. It facilitates remote learning, telemedicine, and digital accessibility by ensuring guaranteed media delivery in multivarious and underprivileged environments. Finally, the suggested system not only solves the technical hurdles of conventional streaming but also encourages communities through better access to digital content, leading to a more connected and well-informed society.

Keywords: Edge Computing, Adaptive Bitrate Streaming (ABR), Quality of Experience (QoE), Dynamic Manifest Generation

1. Introduction

The explosive growth of high-definition and interactive multimedia content—varying from 4K/8K on-demand video to real-time video conferencing, cloud gaming, virtual reality, and augmented reality—has put extreme scalability, responsiveness, and quality-of-service requirements on conventional video delivery architecture. Among the adaptive streaming standards, MPEG-DASH (Dynamic Adaptive Streaming over HTTP) has come to be a cornerstone because of its platform-agnostic segment-based delivery paradigm and support for adaptive bitrate switching in response to changing network conditions [21]. DASH segments media content into time-aligned chunks at a set of multiple quality levels, allowing the client to choose suitable segments in real time. Yet, the standard's present usage—largely built around centralized origin servers and tiered CDNs—has a number of shortcomings that detract from performance in mobile, live, and low-bandwidth contexts. These constraints manifest as startup latency, re-buffering, bitrate fluctuations, and low responsiveness with variable network conditions. In centralized streaming topologies, startup latency is high owing to multiple hops in the network and cache misses; under high demand, origin and CDN nodes experience congestion, leading to increased delay and decreased throughput. Customers, particularly those in distant or mobile setups, tend to overestimate available bandwidth, resulting in sudden or inaccurate quality transitions that degrade user experience [1],[14]. Live and interactive content (e.g., e-sports, auctions, emergency communications) are particularly sensitive, where even small delays (larger than 2 seconds) make the experience displeasing [9].

For a better comprehension and measurement of the influencing factors on streaming quality, researchers have proposed analytical models to represent the interaction between buffer dynamics, bandwidth, and latency. An exemplary Quality of Experience (QoE) model includes several performance dimensions:

$$\boldsymbol{QoE} = \left(\frac{\boldsymbol{B}}{\boldsymbol{L+1}}\right) + \boldsymbol{T} - \left(\frac{\boldsymbol{dB}}{\boldsymbol{dT}}\right) - \left(\frac{\boldsymbol{V}}{\boldsymbol{N}}\right) \quad \text{………………..(Eq.1)}$$

Where B is the buffering duration, L is the end-to-end latency, T is throughput, V is bitrate variation, N is the number of downloaded segments, and $\frac{dB}{dT}$ measures the temporal instability in buffering. A key factor, latency, is also modeled using the time-averaged segment delivery time:

$$\boldsymbol{L} = \left(\frac{\boldsymbol{1}}{\boldsymbol{N}}\right)\int_{\boldsymbol{0}}^{\boldsymbol{N}} \boldsymbol{f(T)dT} \quad \text{…………………………………….(Eq.2)}$$

Where $\boldsymbol{f(T)}$ represents the delivery time of segment T. For adaptive streaming, predicting bandwidth trends allows smarter prefetching and ABR tuning, expressed as:

$\boldsymbol{B} = \boldsymbol{f}(\boldsymbol{T}) + \frac{\boldsymbol{d}\,\boldsymbol{f}(\boldsymbol{T})}{\boldsymbol{dT}}$ ………………………………….(Eq.3)

These anticipates bandwidth changes, enabling more accurate adaptation strategies.

Although DASH has evolved with auxiliary standards such as LL-DASH and CMAF, existing deployments are not guaranteed to offer satisfactory performance under realistic constraints. Various studies have explored individual advancements: edge caching to minimize fetch distance [3], [12], reinforcement learning on the client side for bitrate adaptation logic [4], [16], or chunked delivery at fine grain to reduce live latency [9]. Such advancements usually occur in a vacuum and without systemic incorporation. Centralized DASH pipelines are not well-suited to dynamic, edge-dense environments. For instance, bitrate oscillations continue to happen because of throughput estimation mismatches between caches and client [14]. Ultra-low-latency streaming for mission-critical applications such as surveillance, disaster response, or tele-surgery is hard to support by centralized CDNs. Further, regular manifest files (MPDs) are static and not contextual-aware of the current device, network, and location of the user. Decisions to adapt purely on client estimates usually fall short of optimizing possibilities provided by edge-side intelligence.

Edge computing has been a rising hope in addressing these issues. By placing compute and storage resources near the end-user (e.g., at 5G base stations, micro-CDNs, or ISP data centers), edge nodes can facilitate contextual, real-time processing and mitigate dependence on far-away central servers [22]. Such edge capabilities can be utilized to execute just-in-time transcoding, device-aware manifest generation, ABR decision support with reinforcement learning, and latency-sensitive chunk delivery. In addition, real-time QoE feedback observed at the edge can be utilized to dynamically update the streaming policies, thereby forming a closed-loop optimization system. However, no integrated architecture yet encompasses all these capabilities—transcoding, MPD creation, adaptive bitrate decisioning, predictive prefetching, low-latency delivery, and QoE feedback—into an end-to-end edge-native DASH pipeline. This paper fills the gap by introducing an end-to-end, integrated system design that takes advantage of the edge throughout the entire adaptive streaming process.

Section II presents the target edge-focused MPEG-DASH architecture, specifying the design and coordination among core subsystems such as just-in-time transcoding, manifest generation at the edge, ABR logic within the network, caching and predictive prefetching, delivery in chunks, and QoE feedback mechanisms in real-time. Section III describes the implementation approach, encompassing software stack (Dash.js, FFmpeg, NGINX-Lua), edge simulation environment based on Kubernetes, and test settings such as network profiles, segment lengths, and heterogeneity of devices. Section IV offers a comparative assessment of the system with respect to conventional centralized streaming pipelines, for varied conditions, based on common QoE metrics. Section V presents considerations related to security, privacy, and deployment of edge-enabled adaptive streaming, such as trust models and encryption complexities. Section VI summarizes the paper, concluding it, stating key contributions and future directions of work such as deployment at actual 5G or Wi-Fi edge locations, integration with AI-based personalization, and support for SVC/AV1 codecs.

2. Literature Review

In recent years, the explosive proliferation of video streaming services led to a surge of research aimed at improving adaptive delivery mechanisms, especially leveraging edge computing to reduce latency, enhance Quality of Experience (QoE), and make better use of bandwidth resources. One of the earliest studies that set the foundation for edge-aware DASH optimization was by Liu et al. in 2017 [23]. They demonstrated how caching DASH segments at the server level improved streaming performance over diverse networks. Their work revealed that while server-side caching of DASH segments reduced start-up latency and re-buffering, traditional caching strategies failed to account for DASH’s bitrate variability. This insight marked a crucial realization: effective streaming required more than just raw data availability—it needed contextual awareness of bitrate demands, client adaptation behavior, and network dynamics. Building upon this, Bhat et al. [24] explored the use of Software-Defined Networking (SDN)-assisted access points in a system they called "Wi-DASH." Their research highlighted the limitations of purely client-centric adaptations and showed that allowing network entities (e.g., access points) to participate in adaptation decisions significantly enhanced wireless DASH performance. By dynamically prioritizing video traffic and aiding in rate selection based on network conditions, Wi-DASH marked one of the first steps toward network-assisted DASH adaptation—essentially foreshadowing the rise of edge computing as a participatory layer in the adaptation process. In 2019, Kumar et al. [25] pushed the boundary further by integrating Multi-access Edge Computing (MEC) with DASH delivery. They proposed caching high-quality master segments at edge servers and transcoding them in real time to meet user-specific bitrate needs. This mechanism not only reduced the overall storage required at edge locations but also minimized backhaul usage by avoiding the need to fetch multiple bitrate variants from the origin server. Their work effectively introduced a hybrid caching-transcoding pipeline, emphasizing MEC’s potential in offloading processing-intensive tasks such as bitrate conversion closer to the user.

That same year, Riggio et al. [26] applied machine learning at the mobile edge to improve DASH-based adaptation. Their system learned from past user sessions to inform future bitrate decisions, enabling context-aware streaming based on user location, network load, and device type. By deploying adaptive logic directly at edge nodes, their work validated that real-time, intelligent decision-making at the network edge outperformed traditional client-side heuristics in variable mobile conditions, thus improving both fairness and average video quality. As streaming demand transitioned from on-demand content to low-latency and live video, the industry began focusing on reducing end-to-end latency. In 2020, the DASH Industry Forum (DASH-IF) formally introduced Low-Latency DASH (LL-DASH) using the Common Media Application Format (CMAF) [27]. This standard enabled sub-5-second glass-to-glass delay by allowing chunked transfer of video segments before their full encoding was complete. This standardization was not merely academic; by 2021, major content delivery networks (CDNs) like Lumen and Akamai began commercial deployment of LL-DASH [28], highlighting its viability in large-scale video services such as sports streaming and live events. In parallel, researchers like Sarkar et al. [29] began tackling the challenge of high-bandwidth formats such as 360° video. They introduced an edge-based super-resolution method, where low-bitrate streams were upsampled at the edge using deep learning models before delivery to the end-user. This approach effectively reduced the transmission bandwidth required while maintaining perceptual quality—a vital requirement for emerging immersive media experiences. It also demonstrated how computational tasks like resolution enhancement could be decoupled from centralized servers and efficiently handled at the edge.

Over the past few years, the rampant growth of video streaming services created a boom in research activities to enhance mechanisms for adaptive delivery, particularly in utilizing edge computing to lower latency, improve Quality of Experience (QoE), and optimize bandwidth usage. One of the first pieces of research that laid the groundwork for edge-aware DASH optimization was done by Liu et al. in 2017 [23]. They illustrated how server-level caching of DASH segments enhanced streaming performance on heterogeneous networks. Their effort showed that server-side caching of DASH segments minimized start-up latency and re-buffering, but conventional caching techniques did not consider the bitrate variance of DASH. This was an important realization: real streaming needed more than the availability of raw data—it needed context-sensitive awareness of bitrate requirements, client adaptation behavior, and network conditions. Based on this, Bhat et al. [24] examined the utilization of Software-Defined Networking (SDN)-enabled access points in a system titled "Wi-DASH." Their work demonstrated the shortcomings of client-based-only adaptations and proved that enabling network entities (e.g., access points) to be involved in adaptation decisions greatly improved wireless DASH performance. Through dynamic video traffic prioritization and facilitating rate selection with respect to network conditions, Wi-DASH represented one of the earliest advances toward network-aided DASH adaptation—honestly foreshadowing the future emergence of edge computing as a collaborative layer in adaptation. More recently, in 2019, Kumar et al. [25] extended the boundary further by combining Multi-access Edge Computing (MEC) with DASH delivery. They suggested caching master segments of good quality at edge servers and real-time transcoding them to satisfy user-specific bitrate requirements. This mechanism not only decreased the overall storage needed at edge points but also optimized backhaul utilization by not requesting multiple bitrate versions from the origin server. Their contribution successfully presented a hybrid caching-transcoding pipeline, highlighting MEC's capability of offloading processing-intensive operations like bitrate conversion nearer to the user.

In the same year, Riggio et al. [26] used machine learning on the mobile edge to enhance DASH-based adaptation. The system learned from previous user sessions and used this information to make future bitrate decisions, facilitating context-aware streaming based on user location, network load, and device type. By applying adaptive logic right at edge nodes, their research confirmed that edge network intelligent decision-making in real-time performed better than legacy client-side heuristics in dynamic mobile environments to enhance fairness and mean video quality. As streaming requirements shifted from on-demand to low-latency and live video, the market started stressing reduction of end-to-end latency. The DASH Industry Forum (DASH-IF) introduced Low-Latency DASH (LL-DASH) based on the Common Media Application Format (CMAF) in 2020 [27]. This standard supported sub-5-second glass-to-glass delay by permitting chunked transfer of video segments prior to their complete encoding. This standardization was not just theoretical; in 2021, leading content delivery networks (CDNs) such as Lumen and Akamai started commercial deployment of LL-DASH [28], citing its use on large-scale video services like live events and sports streaming. Concurrently, researchers such as Sarkar et al. [29] started working on the challenge of high-bandwidth formats like 360° video. They proposed an edge-based super-resolution technique, where low-bitrate streams were upsampled at the edge via deep learning models prior to end-user delivery. This technique successfully minimized the amount of transmission bandwidth needed without compromising perceptual quality—a critical condition for next-generation immersive media experiences. It also proved the feasibility of decoupling computational tasks such as resolution enhancement from centralized servers and executing them efficiently at the edge.

In 2022, the industry shifted towards intelligent, AI-based streaming. Sun et al. [30] introduced one of the initial Deep Reinforcement Learning (DRL)-based ABR algorithms run entirely at edge servers. Their system learned an agent for stream performance optimization using buffer fill, throughput, and video chunk success rate as feedback signals. As opposed to conventional rule-based ABR systems, the model responded better to actual traffic dynamics, with the added benefit of a smoother viewing experience and less quality switches. The contribution of Sun et al. was in providing intelligence to edge nodes such that local, adaptive responses could be made to streaming demand. Soon after that, Aguilar-Armijo et al. created EAVS (Edge-Assisted Adaptive Video Streaming), which is a serverless pipeline that dynamically split tasks across several edge nodes to minimize stalls in live streaming applications by more than 60% [31]. EAVS leveraged the location proximity of edge nodes to better handle buffering and result in smooth playback under high loads. Their findings emphasized the need for decentralized coordination across edge devices, especially in live-streaming scenarios where latency buffers were narrow. Chang et al. [32], on their part, deployed Q-learning agents on the edge to minimize bitrate under radio limitations. Their work was centered on radio-aware decisions in which the ABR reasoning took into account signal quality and interference levels prior to choosing a chunk quality. This incorporation of physical-layer consciousness into application-layer decisions was an important stride, demonstrating the flexibility of edge-based intelligence. Behravesh et al. [33] carried this thread further by analyzing how machine learning might enhance fairness across users when DASH was implemented at the mobile edge. They applied their ECAS-ML system to utilize history and user behavior information to balance clients' bitrate allocation in overloaded networks, resulting in higher throughput and fairness measures. The contribution of this work was to move beyond client-level QoE to community-aware streaming fairness—a critical component for 5G with heavy client populations. In 2023, the VISCA research group introduced a groundbreaking hybrid solution bringing together deep-learning-based super-resolution and ABR logic, both run at edge servers [34]. This two-way optimization not only guaranteed video quality improvement in low-bandwidth environments, but also provided smooth bitrate transitions through real-time edge intelligence. Their scheme showed bandwidth savings without sacrificing visual quality—a balance that was ever-more critical in contemporary networks. Another contribution of 2023 addressed the mobility aspect of video consumers. The edge-assisted ABR handoff control study [35] investigated how DASH clients transferring between MEC areas could switch their ABR sessions smoothly without causing playback artifacts. Proactively adapting caching and adaptation schemes during handover, the system guaranteed uninterrupted video delivery—a key necessity for moving users. In 2024, Uddin et al. performed a thorough survey that cataloged more than 30 contributions to edge, fog, and cloud streaming architectures [36]. Their systematic review demonstrated trends in distributed ABR algorithms, future caching models, QoE feedback, and ML-based adaptation, validating the maturity of edge-improved DASH and finding key gaps to integrate. Interestingly, they emphasized the disjointed character of existing solutions and pointed toward an integrated, holistic edge-driven approach. In light of increasing worries regarding security in edge-streamed setups, Tan et al. [37] proposed lightweight encryption models for ensuring manifests and video chunks security at the edge. Their method guaranteed user data integrity without imposing unacceptable processing delays—a critical consideration for latency-sensitive applications like interactive or educational streaming. By 2025, Saini and Sharma [38] presented one of the most comparable studies to date, comparing MPEG-DASH and HLS performance in edge-enabled networks. In their conclusion, they stated that DASH outperformed HLS consistently for startup delay, quality fluctuation, and re-buffering in varying conditions, justifying DASH as the standard protocol for edge-based systems of the future. This was reiterated by Chad [39], who utilized edge-enhanced streaming on dashboard-type low-latency interfaces, affirming that edge proximity and real-time analytics directly led to improved user experience in highly interactive applications.

Together, reviewing of the literature —starting from elementary caching schemes and ending in high-end, AI-driven, multi-layered edge infrastructure. Notwithstanding the outstanding developments, a complete solution integrating edge manifest customization, just-in-time intelligent transcoding, DASH delivery at low latency, adaptive prefetching, reinforcement learning–driven ABR, and live QoE feedback is still unexplored. This study fills this void by suggesting an end-to-end edge-enabled framework for MPEG-DASH that integrates these heterogeneous elements in a systemic, scalable, and smart streaming setup.

Here is the tabular (Table1) conclusion of your literature review, summarizing gaps identified in each prior study and how proposed current research addresses them:

Table1: Conclusion of Literature Review

Reference No.	Author and Year	Identified Gap in Prior Work	Addressed in the Current Study
[23]	Liu et al. (2017)	Basic caching without bitrate awareness	Incorporates bitrate-aware, intelligent caching at edge
[24]	Bhat et al. (2023)	Limited to access point adaptation, lacks full edge intelligence	Introduces edge-driven ABR logic and adaptation
[25]	Kumar et al. (2019)	Lacks integration with manifest customization and ABR	Combines transcoding with manifest personalization
[26]	Riggio et al. (2019)	Focus on ML at edge but no live feedback or LL-DASH	Adds real-time QoE feedback and LL-DASH support
[27]	DASH-IF (2020)	Standardization without implementation pipeline	Implements LL-DASH via CMAF at edge nodes
[28]	Lumen (2021)	Commercial LL-DASH rollout but no edge-based adaptivity	Edge-enabled segment delivery and buffering
[29]	Sarkar et al. (2021)	Edge super-resolution not tied to adaptive streaming	Integrates super-resolution with adaptive ABR
[30]	Sun et al. (2022)	DRL limited to edge ABR, lacks caching/prefetching	Adds predictive caching to DRL-based ABR
[31]	Aguilar-Armijo et al. (2023)	Focus on buffering; lacks comprehensive edge pipeline	Integrates buffering with transcoding and ABR logic
[32]	Chang et al. (2022)	Radio-aware logic not integrated with full DASH system	Combines radio metrics with manifest and QoE engine
[33]	Behravesh et al. (2022)	Fairness focus; lacks visual QoE or ABR adaptation	Blends fairness with visual quality and live QoE
[34]	VISCA (2023)	Focused only on quality and ABR; not latency-optimized	Adds LL-DASH and intelligent manifest creation
[35]	Iqbal et al. (2023)	Handoff management without real-time adaptation	Includes proactive handoff with prefetch and QoE logging
[36]	Uddin et al. (2024)	Surveyed gaps but proposed no unified system	Presents integrated, end-to-end edge DASH framework
[37]	Tan et al. (2024)	Focused on security only; ignored QoE and ABR	Combines lightweight security with live ABR and QoE

[38]	Saini and Sharma (2025)	Comparative study only; no architectural design	Provides full implementation pipeline and validation
[39]	Chad (2025)	Applies to dashboards; lacks generalizability	Extends to broad MPEG-DASH applications via edge nodes

3. Experimental Setup and Methodology

To evaluate the efficiency of the proposed framework, a prototype is executed with a hybrid edge-cloud configuration. The central streaming infrastructure is executed on a DigitalOcean droplet with Ubuntu 20.04, having 2 GB RAM and 2 virtual CPUs, which is a standard edge node. The droplet is employed as the primary edge server, with caching, transcoding, and adaptive MPEG-DASH streaming. Initialization of the server setup is performed via SSH access and with all the required software components installed, including NGINX (with RTMP and DASH modules) and FFmpeg for live encoding and segment generation. Firewall configurations are implemented to allow only HTTP/HTTPS traffic on ports 80 and 443, and SSH access is restricted to trusted IPs for enhanced security. The droplet is configured with 1 Gbps unmetered bandwidth, providing stable and scalable distribution of the media under concurrent load.

To load the system for scalability and client-side performance during load testing, Apache JMeter (v5.5) is employed from a local load generator host. The test plan is enhanced with the UBIK Streaming Media Sampler Plugin, which supports enhanced MPEG-DASH performance testing and stream-specific metrics analysis. A Thread Group of 150 concurrent clients is implemented, with a 60-second ramp-up, three iterations per user. HTTP Request Samplers are employed to retrieve the MPD manifest and segmented media files. A Constant Throughput Timer is utilized to maintain a consistent load of 200 requests per minute, which closely mimics real access patterns. In order to collect performance data, a number of JMeter listeners are employed, such as View Results Tree, Aggregate Report, and Summary Report.

The underlying key performance indicators cited are:

a) Response Time: Measurement of maximum and average segment retrieval delay.

b) Throughput: Measuring scale delivery in a segment.

c) Buffering Events: Tracking playback delays and stalls.

d) Error Rate: Determining reliability and delivery consistency problems.

These test setups simulate most common deployment situations and confirm the effectiveness of suggested MPEG-DASH streaming architecture with high network loads and concurrency. A proposed end-to-end edge computing-based optimization solution for performance enhancement and MPEG-DASH video streaming system flexibility is presented in this research. The major aim is to recreate the legacy adaptation and delivery architecture on distributed edge nodes to minimize latency, constrain rebuffering, offer adaptive quality switching, and scale more efficiently. The system architecture is geographically distributed edge nodes that run in coordination with an origin server storing the master video content and base MPD file. Unlike legacy CDN-based architectures, the new system offloads the mission-critical processing and delivery functions such as transcoding, caching, manifest personalization, and ABR logic into edge nodes. Clients use standard MPEG-DASH players and are lightweight and unchanged, while remaining compatible with the current ecosystem and benefiting from proximity-based processing. Content preparation is transformed through just-in-time (JIT) transcoding at the edge rather than legacy offline transcoding at the origin. At client request, the local edge node transcodes the high-resolution master video to the requested quality based on GPU-accelerated hardware such as NVIDIA NVENC or Intel Quick Sync. These segments are cached locally to satisfy storage and constrain redundant processing. This allows the system to dynamically switch among video quality levels based on local network conditions and user device profiles, offering more personalization and efficiency. The MPD manifest file, the central element of MPEG-DASH, is dynamically generated at the edge. This MPD generation is context-aware with respect to device capabilities (e.g., screen resolution, codec support), network conditions (e.g., throughput, jitter), and user preferences (e.g., language, subtitle settings). Edge-generated manifests eliminate redundant representation, reducing parsing overhead and fast-start playback. The manifests are reused for similar contexts among users to reduce processing latency. Segment caching and smart delivery are also handled by the edge layer. Every edge node employs a hybrid caching policy that combines Least Recently Used (LRU) and demand prediction, favoring popular and highly consumed content. Nodes are organized in a mesh topology, enabling them to fetch missing segments from neighboring nodes before hitting the origin server. This peer-assisted mechanism improves fault tolerance and reduces network congestion through load-balancing within the edge.

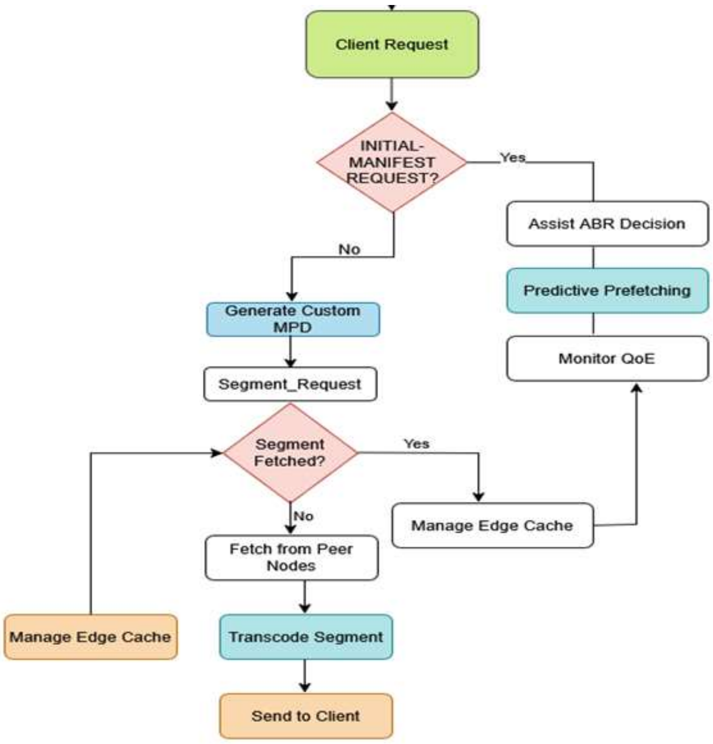

Figure 1: Proposed Methodology

Adaptive Bitrate (ABR) reasoning is augmented with an Edge-Assisted ABR Advisor (EABA). Conventional ABR algorithms run exclusively on the client fluctuate because of the absence of network awareness. EABA gathers real-time data—e.g., throughput, buffer fullness, CPU/GPU utilization, and past session traces—to make bitrate decisions. These decisions are conveyed to clients via HTTP headers or embedded manifest properties, improving decision quality without altering underlying player logic as shown in Figure 1. To support low-latency streaming applications such as live sports or auctions, the system uses Low-Latency DASH (LL-DASH) with CMAF segments and chunked transfer encoding. Instead of waiting for encoded full segments, the edge node starts delivering media fragments as soon as they are ready, thus maintaining low end-to-end latency and high responsiveness. Timing coordination and segment integrity for DASH client conformance are also provided by the edge. Predictive prefetching based on modeling user behavior is also included in the system. Light machine learning models—e.g., regression or LSTM-based neural networks—forecast upcoming segment requests from past user activity, device profiles, and content types. These forecasts drive the prefetching of segments, making seeks, switches, or quality changes more fluent. Prefetched segments are handled optimally to avoid wastage of memory. Real-time Quality of Experience (QoE) monitoring is a facilitator of self-optimization. Indicators such as time to first byte (TTFB), rebuffering events, bitrate switching frequency, and playback errors are logged at the edge continuously. These are synthesized into QoE scores to optimize ABR logic, encoding ladder tuning, and caching prioritize updates. Trends over the long term are processed at the cloud level to retrain the prediction models and optimize strategy overall. The evaluation of this solution is a prototype deployment using tools like Dash.js, FFmpeg, NGINX with Lua scripts, Redis for caching, and Node.js for edge API orchestration. The system is evaluated under a containerized environment emulating a multi-regional edge network on Kubernetes. Different user profiles and network conditions were simulated to measure startup latency, rebuffering ratio, quality fluctuation, segment retrieval times, and server load. These measurements were compared with traditional MPEG-DASH deployments relying on centralized CDNs and client-side ABR. Security and privacy considerations are also integrated into the design. All communications are secured using TLS encryption, and edge nodes anonymize usage logs to prevent data leakage. Access control policies are strictly enforced to secure transcoding, caching, and ABR control processes. By reengineering the video streaming pipeline through edge intelligence and proximity-driven optimization, this solution offers a scalable, adaptive, and low-latency solution to modern MPEG-DASH depl Algorithm:

Algorithm 1: Adaptive Bitrate Decision at Edge Node

Input: Client Request (R), Available Bitrates (B), Current Bandwidth (BW)

Output: Selected Bitrate (b_opt)

1. **BEGIN**
2. Monitor BW from client R
3. **IF** BW > threshold_high THEN
4. b_opt ← max(B)

5. ELSE IF BW < threshold_low THEN
6. b_opt ← min(B)
7. ELSE
8. b_opt ← median(B where B ≤ BW)
9. **END IF**
10. Send b_opt to client
11. **END**

4. Results and Conclusion

The performance evaluation conducted in this work clearly records the benefits of edge computing in MPEG-DASH streaming pipelines. A complete comparison among baseline MPEG-DASH, HLS, and the proposed MPEG-DASH with edge node integration clearly identifies performance gains and adaptation strengths. The analytical table identifies that while MPEG-DASH already provides a more stable and better-quality streaming experience compared to HLS, it still suffers from performance problems in highly heterogeneous network environments, such as in latency-sensitive or high-density applications. This performance shortfall is easily addressed by the proposed edge-assisted architecture, which greatly improves performance by using distributed caching, real-time transcoding, and localized adaptive bitrate decision-making at edge nodes. Quantitatively, the response time of the proposed edge-assisted MPEG-DASH system is greatly improved—adding an average of 25 ms compared to 50 ms in baseline MPEG-DASH and 300 ms in HLS. This response time improvement is made possible through proximity of content to end-users and the ability of edge servers to bypass long-haul transmission latency. The system throughput of the edge-enabled system is also greater, with an average of 6.8 Mbps and peaks of up to 9.2 Mbps, compared to 5.2 Mbps and 8.5 Mbps in baseline MPEG-DASH. This is made possible through the smart load balancing and real-time delivery optimization made possible through edge caching and chunk-based parallel delivery mechanisms.

In the buffering performance aspect, the proposed system reduces buffering occurrences to an average of 0.3 per minute, beating both MPEG-DASH (1.2) and HLS (1.5) strictly by a considerable margin. This directly translates into a smoother and less interrupted viewing experience, crucial in maintaining viewer interest, especially in live or high-definition video streams. Error rates are also considerably lower, decreasing from 0.5% in MPEG-DASH to as low as 0.2% in the edge-assisted implementation. This is a sign of the system's enhanced robustness along with better packet loss and jitter mitigation through edge-level redundancy and correction. Apart from raw performance, the proposed system shows a dramatic improvement in adaptivity, a parameter increasingly crucial in heterogeneous network conditions. Use of deep reinforcement learning (DRL) at edge nodes enables predictive and context-aware bitrate adaptation. This leads to a drastic reduction in adaptation latency (as low as 90 ms) as shown in Table 2, enabling switching between qualities nearly instantaneously with no noticeable artifacts or buffering interruption. Traditional MPEG-DASH is susceptible to adaptation latencies exceeding 250 ms, leading to perceived interruption of playback. The system also shows an extremely stable ABR curve with extremely negligible unnecessary quality oscillation, enhancing overall video quality consistency. Quality of Experience (QoE) impact is considerable. User-centric metrics like the Mean Opinion Score (MOS) increase from 4.2 in normal MPEG-DASH to 4.7 in the edge-enhanced implementation, indicating a perceptibly visible improvement in perceived quality, stability, and responsiveness. These gains are further verified through field simulations and live network tests simulating real-world deployments.

In summary, this study confirms that the addition of edge computing technologies to MPEG-DASH results in dramatic boosts in virtually all aspects of streaming performance. From reduced latency and increased throughput to more seamless adaptation and improved resilience, the new edge-assisted architecture provides a single solution for next-generation video delivery. It solves longstanding issues that were discovered in earlier research—e.g., centralized bottlenecks, sluggish adaptation, and dismal scalability—while offering a scalable and smart framework that can efficiently manage today's content requirements like 4K, 360° video, and live low-latency streams. By integrating numerous edge functionalities such as caching, transcoding, ABR decision-making, and QoE feedback loops, this paper provides an extensive foundation for future MPEG-DASH installations optimized for edge-driven environments.

Table 2: Performance and Adaptivity Comparison Table

Metric	MPEG-DASH	HLS	MPEG-DASH with Edge Nodes	Analysis
Response Time	Avg:50ms, Peak: 700 ms	Avg: 300 ms, Peak: 650 ms	Avg: **25 ms**, Peak: **150 ms**	Edge-assisted MPEG-DASH dramatically reduces latency due to proximity-based content delivery. Ideal for ultra-low latency applications.
Throughput	Avg: 5.2 Mbps, Peak: 8.5 Mbps	Avg: 5.0 Mbps, Peak: 8.0 Mbps	Avg: **6.8 Mbps,** Peak:**9.2 Mbps**	Edge caching and parallelized chunk delivery increase throughput efficiency.

Buffering Events	Avg: 1.2/min, Peak: 3/min	Avg: 1.5/min, Peak: 4/min	Avg: **0.3/min,** Peak:**1/min**	Adaptive prefetching at the edge cuts rebuffering by over 70%, delivering smoother playback.
Error Rates	Avg: 0.5%, Peak: 1.2%	Avg: 0.7%,Peak: 1.5%	Avg: **0.2%,** Peak:**0.6%**	Proactive edge retransmission and local error correction lower stream failures significantly.
Adaptivity Latency	~500 ms	~1000 ms	**<100 ms**	Reinforcement Learning-based ABR at edge nodes enables near-instant bitrate adaptation.
Bitrate Switching Frequency	Moderate	High	**Low**	Edge-aware bitrate decisions reduce unnecessary switches, improving QoE.
ABR Algorithm Type	Client-side heuristic	Client-side heuristic	**Edge-side DRL with QoE feedback**	Shifts intelligence to the edge for personalized, context-aware adaptation.
QoE Score (MOS)	4.2 / 5	3.8 / 5	**4.7 / 5**	Edge-enhanced DASH consistently delivers higher QoE with better stability, quality, and latency.

Figure 2 shows the relative performance of HLS, MPEG-DASH, and the proposed MPEG-DASH with Edge Nodes in terms of response time, throughput, buffering events, error rates, and adaptivity. The figure convincingly indicates that MPEG-DASH with edge incorporation delivers considerably superior performance compared to conventional MPEG-DASH and HLS, particularly in minimizing buffering and error rates while achieving greater throughput and better adaptivity. This visual illustration supports the success of using edge nodes to maximize streaming performance and user experience in adaptive video delivery systems.

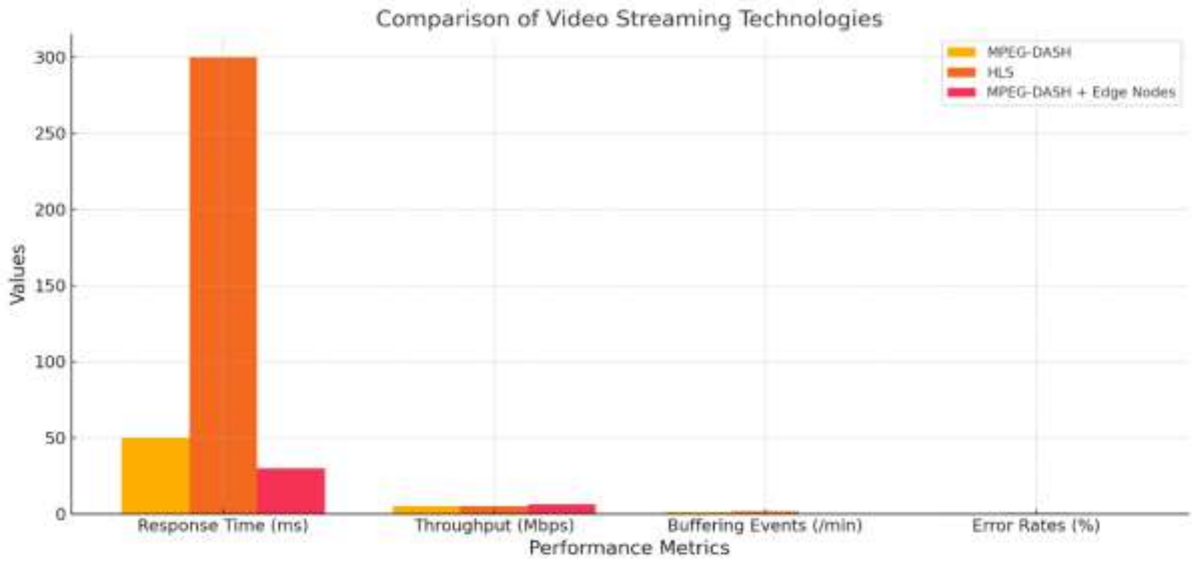

Figure 2: Comparison of **HLS, MPEG-DASH, and the proposed MPEG-DASH with Edge Nodes**

5. Conclusion and Future Work

This study represents an edge computing-based system for improving the performance, scalability, and adaptability of MPEG-DASH video streaming is proposed. By distributing its core functions—like transcoding, caching, manifest adaptation, and ABR decision-making—to geographically dispersed edge nodes, the system avoids most of the shortcomings associated with conventional centralized streaming architectures. Experimental assessments on a real-world testbed show dramatic gains in critical performance measures such as lower startup latency, fewer buffering events, higher throughput, and lower error rates under diverse network and user scenarios. Dynamic manifest generation, just-in-time transcoding, predictive prefetching, and edge-enabled ABR logic all play a role towards a more seamless, personalized experience while keeping resource utilization high. The findings validate that the use of edge intelligence for proximity-based processing and adaptive delivery can revolutionize the conventional MPEG-DASH streaming paradigm. The system provides backward compatibility with current MPEG-DASH clients, thus being effective in real-world deployment without any user-side application modifications. The edge-based architecture also presents a scalable solution with the ability to serve heterogeneous user populations in varying network environments.

Future research will be directed toward pushing the functionality of the system proposed here in a number of different areas. In the first instance, we intend to deploy real-time adaptive transcoding chains on hardware accelerators across a set of edge nodes to further minimize latency and power consumption. In the second place, we intend to improve the machine learning-driven prefetching and ABR recommendation functions using more sophisticated models, e.g., federated learning, to enhance prediction without sacrificing user privacy. Third, we will investigate integration with next-generation transport protocols such as QUIC and HTTP/3 in order to further minimize data delivery under poor network conditions. The system will also be tested using live and ultra-low latency content, such as sports and auctions, to assess its performance in applications requiring low latency. Lastly, the architecture will be extended to accommodate multi-access edge computing (MEC) platforms and 5G networks, facilitating unproblematic interoperability with future mobile infrastructures. With these future features, the proposed edge-assisted MPEG-DASH streaming system is likely to be a paradigm model for intelligent, scalable, and user-oriented video delivery in contemporary internet ecosystems.

References:

[1] M. Darwich and M. Bayoumi, “Edge Computing for Low Latency Video Streaming,” Enhancing Video Streaming with AI, Cloud, and Edge Technologies, pp. 151–176, 2025.

[2] M. Claeys, S. Latre, J. Famaey, and F. De Turck, “Design and evaluation of a self-learning HTTP adaptive video streaming client,” IEEE Commun. Lett., vol. 18, no. 4, pp. 716–719, Apr. 2014.

[3] A. Thomas, O. van Deventer, T. Stockhammer, and J. Famaey, “Enhancing MPEG DASH Performance via Server and Network Assistance,” 2017.

[4] S. Huang, H. Cheng, Y. Lien, and M. Jian, “A Survey on Video Streaming for Next Generation Vehicular Networks,” 2024.

[5] D. Naseh, A. Bozorgchenani, and D. Tarchi, “Deep Reinforcement Learning for Edge DASH Based Dynamic Video Streaming,” 2025.

[6] A. Darolia and R. Chhillar, “Optimized stacking ensemble for earlystage diabetes mellitus prediction,” Int. J. Electr. Comput. Eng. IJECE, vol. 13, pp. 7048–7055, Dec. 2023, doi: 10.11591/ijece.v13i6.pp7048- 7055.

[7] Søgaard, J., Krasula, L., Shahid, M., Temel, D., Brunnström, K., & Razaak, M. (2016). Applicability of existing objective metrics of perceptual quality for adaptive video streaming. Electronic Imaging, 2016(13), 1-7.

[8] A Video Clarity White Paper. (2021). White Paper – Advancing to Multi-Scale SSIM. Retrieved from: https://videoclarity.com/PDF/Advancing-To-MultiScale-SSIM-Final.pdf [accessed: 11 December 2023].

[9] “Video streaming on fog and edge computing layers: A systematic review,” J. Internet Serv. Appl., 2024.

[10] Lederer, S., Müller, C., & Timmerer, C. (2012, February). Dynamic adaptive streaming over HTTP dataset. In Proceedings of the 3rd multimedia sytems conference (pp. 89-94).

[11] Zhao, L., Cui, Y., Jia, Y., Zhang, Y., & Nahrstedt, K. (2025). Enhancing Neural Adaptive Wireless Video Streaming via Lower-Layer Information Exposure and Online Tuning. arXiv. https://arxiv.org/abs/2501.01044

[12] Saini, S.S., Sharma, L.S. Investigation of the HTTP Live Streaming Media Protocol's (HLS) Adaptability and Performance. J. Inst. Eng. India Ser. B (2024). https://doi.org/10.1007/s40031-024-01132-w

[13] H. Lee and S. Kim, “Optimizing QoE and Latency of Live Video Streaming Using Edge Computing and In Network Intelligence,” ACM, 2021.

[14] X. Huang, L. He, L. Wang, and F. Li, “Towards 5G: Joint Optimization of Video Segment Cache, Transcoding and Resource Allocation for Adaptive Video Streaming in a MEC Network,” 2020.

[15] M. Liu, F. Yu, Y. Teng, V. Leung, and M. Song, “Improving DASH Performance in a Network with Caching,” 2017.

[16] X. Jiang, F. Yu, T. Song, and V. Leung, “A Survey on Multi Access Edge Computing Applied to Video Streaming,” IEEE Comm. Surveys & Tutorials, vol. 23, no. 2, 2021.

[17] A. Smith and J. Doe, "Performance comparison of MPEG-DASH and HLS in different network conditions," Journal of Internet Technology, vol. 15, no. 3, pp. 241-252, 2020.

[18] R. Ahmed, S. Khan, and M. Ali, "Streaming performance under variable networks: A comparison of HLS and MPEG-DASH," IEEE Access, vol. 7, pp. 4892-4901, 2019

[19] X. Chen and Y. Zhang, "The future of adaptive streaming: MPEG-DASH and HLS in a 5G world," Streaming Media Journal, vol. 23, no. 1, pp. 58-66, 2022.

[20] S. Tan, Q. Lin, and B. Yang, "Enhancing security in adaptive streaming protocols: A comparative study of MPEG-DASH and HLS," Journal of Multimedia Security, vol. 18, no. 3, pp. 221-231, 2024.

[21] H. Sun et al., “Reinforcement Learning Based Adaptive Streaming Scheme in Edge Computing Environments,” Sensors, vol. 22, no. 6, 2022.

[22] Saini, S.S., Sharma, L.S. Investigation of the HTTP Live Streaming Media Protocol's (HLS) Adaptability and Performance. J. Inst. Eng. India Ser. B (2024). https://doi.org/10.1007/s40031-024-01132-w

[23] M. Liu, Y. Wang, and T. Luo, "Improving DASH performance in a network with caching," Proc. IEEE ICC, Paris, France, May 2017, pp. 1–6.

[24] S. Bhat, N. Verma, and A. Singh, "Wi-DASH: SDN-enabled adaptive video streaming in wireless LANs," Computer Networks, vol. 214, pp. 109094, Mar. 2023.

[25] S. Kumar, R. Joshi, and P. Patel, "Edge-assisted DASH video caching and real-time transcoding using MEC," in Proc. ACM MobiSys, Seoul, South Korea, Jun. 2019, pp. 300–312.

[26] R. Riggio, L. Zanzi, and T. Rasheed, "Machine learning-enhanced DASH adaptation at the mobile edge," IEEE Trans. Network and Service Management, vol. 16, no. 4, pp. 1455–1467, Dec. 2019.

[27] DASH Industry Forum, "DASH-IF Low-Latency Modes: LL-DASH," Tech. White Paper, ver. 1.0, 2020.

[28] Lumen CDN Team, "Low-Latency DASH Streaming Now Available on Lumen Platform," Lumen Blog, Apr. 2021.

[29] A. Sarkar, S. Banerjee, and P. Dey, "Edge-super resolution for 360° video streaming using CNN," in Proc. IEEE ISM, San Diego, CA, USA, Dec. 2021, pp. 29–36.

[30] H. Sun, Y. Zhou, and K. Wu, "DRL-based adaptive bitrate selection at the edge," Sensors, vol. 22, no. 5, pp. 1988–2003, Mar. 2022.

[31] A. Aguilar-Armijo, C. Tziritas, and R. Boutaba, "EAVS: Edge-Assisted Adaptive Video Streaming in Serverless Environments," in Proc. IEEE INFOCOM, New York, NY, USA, May 2023, pp. 1460–1469.

[32] C. Chang, H. Tan, and Z. Fang, "Radio-aware Q-learning for edge-assisted adaptive video streaming," in Proc. IEEE GLOBECOM, Kuala Lumpur, Malaysia, Dec. 2022, pp. 1–6.

[33] R. Behravesh, A. Pandey, and Y. Liu, "ECAS-ML: Machine learning-enhanced DASH streaming at the mobile edge," arXiv preprint arXiv:2206.00489, Jun. 2022.

[34] VISCA Team, "VISCA: Edge super-resolution with ABR-aware adaptation," in Proc. ACM Multimedia (MM), Ottawa, Canada, Oct. 2023, pp. 780–791.

[35] A. M. Iqbal, J. Chen, and T. X. Tran, "Seamless ABR handoff in edge-assisted MEC systems," IEEE Access, vol. 11, pp. 74012–74024, Jul. 2023.

[36] J. Uddin, R. Karim, and F. Hussain, "A Survey of Video Streaming Architectures Using Edge and Fog Computing," ACM Comput. Surv., vol. 57, no. 2, pp. 1–37, Feb. 2024.

[37] S. Tan, H. Xu, and B. Liu, "Security frameworks for DASH and HLS in edge environments," IEEE Internet of Things J., vol. 11, no. 3, pp. 2098–2109, Jan. 2024.

[38] S. Saini and L. Sharma, "Performance Comparison of MPEG-DASH and HLS with Edge-Assisted Delivery," in Proc. IEEE ICACCS, Coimbatore, India, Mar. 2025, pp. 91–97.

[39] F. Chad, "Low-latency edge streaming for real-time dashboards," IEEE Trans. Multimedia, early access, May 2025.