World Journal of Advanced Research and Reviews

eISSN: 2581-9615 CODEN (USA): WJARAI

Cross Ref DOI: 10.30574/wjarr

Journal homepage: https://wjarr.com/

(REVIEW ARTICLE)

Artificial Intelligence in HTTP Adaptive Streaming: A review within the evolution of cable television distribution

Daniel Abel Santos Bernal * and Alfredo Salado De Gracia

Independent Researcher, Panama City, Republic of Panama.

World Journal of Advanced Research and Reviews, 2026, 31(02), 028–040

Publication history: Received on 20 June 2026; revised on 29 July 2026; accepted on 01 August 2026

Article DOI: https://doi.org/10.30574/wjarr.2026.31.2.2037

Abstract

This article reviews the evolution of cable television distribution, from coaxial networks to virtualized architectures based on the internet protocol and the cloud, and examines how artificial intelligence has transformed the bitrate adaptation (ABR) algorithms that underpin HTTP Adaptive Streaming (HAS) throughout that transition. Peer-reviewed academic literature and industry documentation were reviewed, and the managed and best-effort distribution models on which these algorithms operate were characterized, alongside the ongoing transport-layer shift from TCP to QUIC. The analysis identifies four fronts where artificial intelligence intervenes at different points of the distribution chain, bitrate decision-making through deep reinforcement learning, content-aware encoding, client-side super-resolution, and energy-aware adaptation, each backed by quantitative gains of up to 43% in quality of experience over a strong reinforcement-learning baseline and up to 57% in energy consumption relative to state-of-the-art algorithms. The article concludes that the sophistication of these models is systematically in tension with the real-time decision requirement of live streaming, which positions coordination across fronts, rather than the isolated improvement of each one, as the central challenge facing the industry.

Keywords: Artificial Intelligence; Adaptive Streaming; Deep Reinforcement Learning; Content-Aware Encoding; Quality of Experience; Energy Efficiency.

1. Introduction

Content distribution in television has come a long way since the first cable systems of 1949 [1], evolving from coaxial networks and hybrid fiber-coaxial (HFC) architectures toward today's IP-based distribution, up to the incorporation of cloud-virtualized headend functions [2]; nonetheless, this transition from physical infrastructure toward virtualized architectures raises scalability and quality-of-service demands that motivate the analysis developed in this article.

Video is the application that generates the most internet traffic, a share that has become even more pronounced over the last decade, in which its traffic quadrupled between 2017 and 2022 to reach an 82% share of the total, driven largely by an explosion in live streaming whose traffic multiplied fifteenfold over that same period [3], compounded by a 14% increase in global viewing time between 2021 and 2022 [3], which together has intensified the pressure on distribution architectures to scale without degrading the user experience, a challenge that drove the adoption of new content delivery paradigms.

Around 2005, Move Networks introduced HTTP Adaptive Streaming (HAS) [4], a paradigm that made it possible to treat multimedia content as standard web content and thereby facilitate its delivery in small fragments capable of transparently traversing firewalls and NAT devices. Despite these scalability advantages, the operation of HAS, underpinned by adaptive bitrate (ABR) algorithms, faces critical challenges on "best-effort" networks, as evidenced by a commercial deployment case that documented degradations of up to 300% in Quality of Experience (QoE) indicators,

* Corresponding author: Daniel Abel Santos Bernal: ORCID: 0009-0001-8795-8347

such as playback start time and the stall (buffering) rate, when comparing distribution over a managed network against pure OTT distribution over the public internet [5].

Peroni and Gorinsky [3] adopt an end-to-end pipeline perspective to analyze video distribution, spanning ingest (capture and encoding) and cloud processing (transcoding and segmentation) through to final distribution via content delivery networks (CDN). From this perspective, the optimization of adaptation decisions is shifting from heuristic methods toward the use of Artificial Intelligence and Deep Reinforcement Learning (DRL), which allows systems to autonomously maximize user satisfaction under highly volatile networks. Emerging technologies such as Super-Resolution (SR), which reconstructs high-definition video at the client from lower-resolution versions, and the use of next-generation codecs such as VVC, further offer additional room to optimize bandwidth and the energy efficiency of television distribution [3].

To support this analysis, both peer-reviewed academic literature and industry documents (CableLabs, ETSI, vendor white papers) were reviewed, given that the applied nature of the topic requires contrasting it with current industry practice.

The article is organized as follows. Sections 2 and 3 reconstruct the evolution of cable distribution infrastructure, from its coaxial networks to the migration of headend functions to the cloud; Section 4 develops the operation of HAS and classifies ABR algorithms according to the network model and their problem-solving methodology, while Section 5 describes the functional architecture and the distribution models on which they operate; Section 6 explores the artificial intelligence trends aimed at improving QoE; and Section 7 closes with the conclusions and the industry's future trends.

2. Evolution of Television Distribution

The technological transition in video distribution has been gradual, starting in 1949 in Astoria, Oregon, where L.E. "Ed" Parsons built the first documented Community Antenna Television (CATV) system. In this first stage, distribution was entirely over coaxial cable; it relied on capturing television signals via an antenna placed at an elevated point and then amplifying and distributing them to homes. Technically, this model operated by transmitting analog radio-frequency (RF) signals, subject to high attenuation over long distances, which required cascaded amplifiers to preserve signal integrity before it reached subscriber receivers.

Subsequently, the cable industry moved to distributing its signals via Hybrid Fiber-Coaxial (HFC) networks, which would become the basis of modern digital cable television. In this scheme, optical fiber is used to carry signals from the Headend to optical nodes, which then convert back to RF for final distribution over coaxial cable. In this digital distribution model, quadrature amplitude modulation (QAM) is used to carry video streams in defined spectrum channels (e.g., 6 MHz under J.83 Annex B [6] or 8 MHz under DVB-C [7]).

The development of video distribution over the IP protocol marked the transition from radio-frequency carrier distribution toward IPTV, whose distribution chain, according to [8], is made up of four functional components: the video headend, the core network, the access network, and the home network.

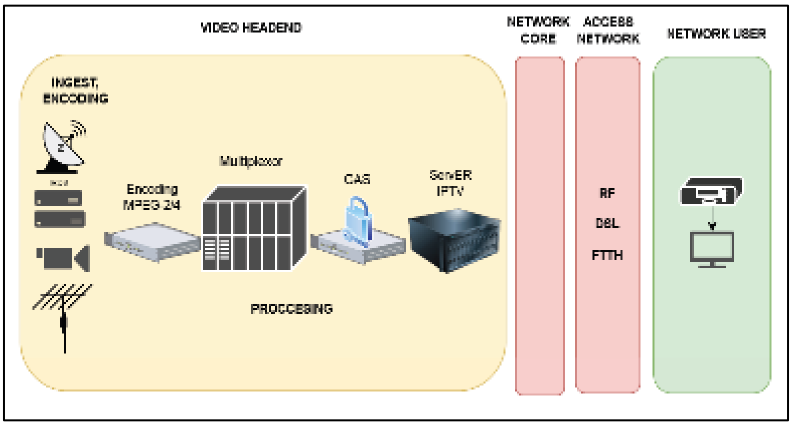

Figure 1 Functional components of IPTV.

At this stage, channels are encoded and distributed via multicast or unicast streams, over which Quality of Service (QoS) policies are applied that separate video traffic from other non-critical data, ensuring fluctuation-free delivery.

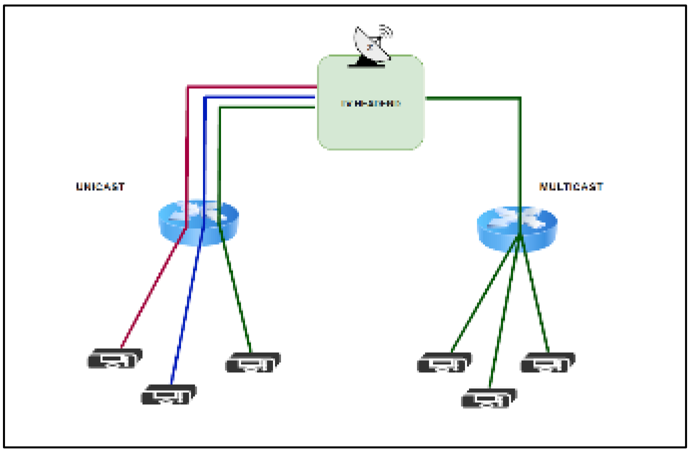

Figure 2 Unicast and multicast distribution architectures.

In parallel with the managed IPTV model, the industry adopted unmanaged streaming solutions based on HTTP Adaptive Streaming (HAS), capable of operating over public best-effort networks and offering scalability and global-reach advantages over the managed model based on RTP/RTCP [9].

This coexistence between managed distribution (IPTV/QAM) and HTTP-based distribution gave rise to a hybrid television model, in which linear content is complemented by on-demand content delivered via broadband technologies. In the cable television space, this model materializes in initiatives commercially known as TV Everywhere (TVE) [46], under which the operator or Multichannel Video Programming Distributor (MVPD) offers a free online aggregation of its programming, conditioned on authenticating that the user is also its service subscriber. The technical specification that standardizes this subscriber authentication and authorization, enabling access to linear and on-demand content on internet-connected devices, was developed by the cable industry through the Online Content Access (OLCA) interface [38].

3. Migration of Headend Functions to the Cloud

Within the evolution of cable television systems is also the transition of video headend functions from physical equipment located on operators' premises toward cloud computing environments, made possible by Network Function Virtualization (NFV) [43]. Thanks to NFV, it is possible to decouple functions that used to be performed at the Headend and run them virtually on servers located in data centers [10]. For video specifically, the applicable virtualized functions span transcoding, content distribution, and caching [11], which can be orchestrated and dynamically scaled according to demand. In practice, this transition does not imply the immediate replacement of infrastructure, since operators have adopted hybrid models in which virtualized and physical functions coexist during the migration process [10].

The ingest of signals into the virtualized headend has also been affected by technological evolution; Reznik et al. [12] document the adoption of protocols such as SRT (Secure Reliable Transport) and RIST (Reliable Internet Stream Transport), which make it possible to transport signals over the internet with reliability levels comparable to those of dedicated links.

3.1. Headend as a Cloud Service

The network function virtualization described above has enabled the cable television industry to look toward what is known as TV Headend as a Service (TVHEaaS), in which the entire headend operation, signal acquisition, hybrid cloud processing, and delivery to satellite, IPTV, OTT, and mobile devices, can be contracted as a SaaS (Software as a Service) solution [2]. This virtualized computing capacity is, precisely, what sustains the execution of the ABR algorithms detailed below.

4. HTTP Adaptive Streaming (HAS) and ABR Algorithms

According to Bentaleb et al. [4], bitrate adaptation algorithms in HTTP Adaptive Streaming (HAS) are characterized by operating in a distributed manner, being executed mainly by each individual client. This distributed operation presupposes prior preparation of the content on the server: the original media stream is fragmented into segments or "chunks" of equal playback duration, each of which is encoded into multiple representations that vary in bitrate, resolution, and quality. To coordinate access to these versions, the server also generates an index file, the Media Presentation Description (MPD) or manifest, listing the URLs and metadata of all available representations.

Once the session starts, the client first downloads this manifest to identify the quality options and, from there, activates an application control loop that repeats throughout playback: at each iteration it measures critical parameters such as available network bandwidth and buffer fill state, and uses that data as input to iteratively select and request the most suitable bitrate segment among the available representations. The goal is to obtain the highest possible quality without compromising the buffer level needed to avoid interruptions or stalls.

This dynamic is articulated, from an operational perspective, in two fundamental states: the buffer-filling state and the steady state. The first occurs at the start of playback or after a rebuffering event, when the client requests segments consecutively and immediately, as soon as the previous download finishes, until reaching a predefined safety threshold. Once that threshold is reached, the system transitions to the steady state, governed by a pattern of active (ON) and idle (OFF) periods: during the ON period the client downloads the current segment and records the throughput obtained, information it uses before entering an OFF period of temporary inactivity prior to requesting the next fragment. This dynamic mechanism is, ultimately, what allows the client to adapt to network fluctuations through continuous monitoring of the system state, thereby optimizing the end user's Quality of Experience (QoE) [4].

4.1. Classification According to Network Control

The academic literature allows ABR algorithms to be classified into two broad families according to the degree of network participation in the adaptation decision: managed schemes, in which the operator's infrastructure actively or fully intervenes in quality selection, and best-effort schemes, in which that decision rests exclusively with the client. It should be noted that this same literature admits complementary taxonomies, for example, according to the point where the adaptation logic runs (client, server, network, or hybrid schemes) [4], which largely overlap with the classification adopted here.

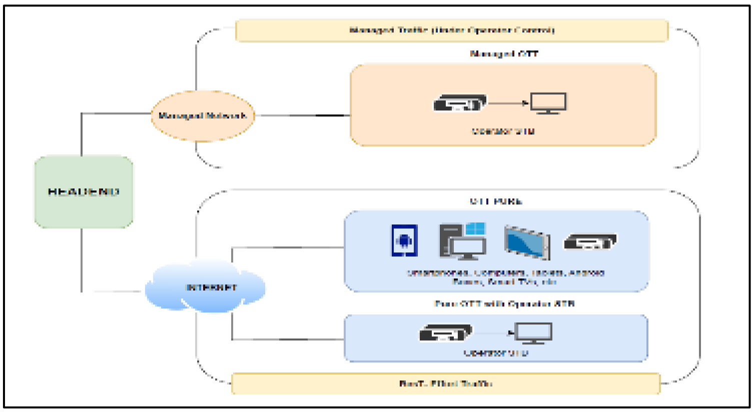

Figure 3 ABR classification according to network control.

4.1.1. Managed ABR

Managed ABR algorithms incorporate network components that have direct visibility into the infrastructure's state to assist, or in some cases replace, the client terminal's decisions. Current research groups these solutions under the concept of Network-Assisted Video Streaming (NAVS) and classifies them into three main families according to the level of network participation.

The first, called SDN-Assisted ABR, employs a software-defined network controller to dynamically allocate resources and notify the client about available transport capacity. In this scheme, the terminal retains its own adaptation algorithm but adjusts its behavior according to guidance received from the network, as proposed by works such as SDNDASH [13], SDNHAS [14], and SABR [15].

A particular case is CANE [16]; although it operates within a network managed by the operator (through traffic shaping applied invisibly at the router or access point), the bitrate adaptation decision remains entirely on the client side, with the client receiving no signal or notification from the network. The operator merely predicts, via machine learning models, the client algorithm's behavior in order to shape the available bandwidth and thereby induce the desired quality. CANE therefore does not constitute a managed ABR scheme in the strict sense, but rather a form of network management that operates around a best-effort client without modifying it.

On the other hand, the Edge/MEC-Assisted ABR approach moves algorithm execution to the operator's edge servers. By having a comprehensive view of the network and the state of the caches, the server takes over the decision process while the client remains unaware of that logic. This modality stands out in implementations such as ES-HAS [17], CSDN [18], LEADER [19], ECAS-ML [20], and ARARAT [21].

Finally, Multicast ABR (mABR) is based on delivering content via IP multicast to a proxy located at the edge or on the subscriber's premises, which handles the conversion to unicast without requiring modifications to the player. This model has a solid regulatory framework, represented by the ETSI TS 103 769 standard [22] and CableLabs technical specifications [23]. Likewise, Farshad et al. [24] offer one of the most recent analytical evaluations of this architecture's efficiency in operator networks.

4.1.2. Best-Effort ABR

Recent research classifies best-effort network algorithms according to their problem-solving methodology. The following presents in detail the three major fields distinguished by the academic literature.

Intuition-Based Methods

This category of algorithms is grounded in heuristic rules derived from expert knowledge, and is usually classified into three main strands. Buffer-based approaches, such as BBA [25] and SARA [26], establish a direct correspondence between buffer occupancy level and bitrate. Bandwidth- or throughput-centered methods, by contrast, rely on predicting historical network performance, as seen in LOLYPOP [27] and STALLION [28]. Between both extremes, hybrid algorithms emerge that integrate both parameters with the aim of optimizing system stability and efficiency; FESTIVE [29] and PANDA [30] are representative examples of this approach.

Theory-Based Methods

Unlike heuristic approaches, these methods formulate the problem within a formal theoretical framework and apply systematic logic to derive solutions with performance guarantees. Model Predictive Control (MPC) stands out for its ability to anticipate future system states and thereby optimize decision-making, as seen in RobustMPC and FastMPC [31]. Proportional-Integral-Derivative (PID) controllers, in turn, employ classical feedback mechanisms for system adjustment; PIA [32] is a representative example of this technique. Lyapunov Optimization (LO), aimed at maximizing utility under strict stability constraints, underpins algorithms such as BOLA [33]. Bayesian Optimization (BO), finally, is used for the automated and efficient configuration of controller parameters, as proposed by ERUDITE [34].

Machine Learning-Based Methods

Machine Learning-based methods are grounded, lastly, in training models from data to generalize decision-making in previously unobserved scenarios. Deep Reinforcement Learning (DRL) has established itself as the predominant trend in this area, relying on trial-and-error processes within simulated environments to optimize its control policies; Pensieve [35], which implements the A3C algorithm, is a leading example. Imitation Learning (IL), by contrast, seeks to replicate the behavior of expert policies through cloning techniques, as described in the development of PiTree [36]. Supervised Learning (SL), for its part, relies on classifiers or sequential models to predict the system's optimal actions, as seen in the Karma proposal [37].

According to [3], this classification applies both to video-on-demand (VoD) environments and to real-time transmissions, though with different implications. In VoD, the algorithm has greater leeway to manage the buffer since the content is pre-stored, whereas in real-time transmissions the ultra-low latency constraints force the use of segments of only a few seconds, which significantly reduces the capacity to react to network fluctuations. This distinction is especially relevant for DRL-based algorithms, whose computational demand can conflict with the real-time requirements of live transmissions.

5. HAS Distribution Architecture

5.1. Functional Blocks of the Distribution Logic

Following the end-to-end QoE contribution scheme proposed by Fautier [5], the distribution logic can be broken down into five functional blocks.

Transcoding System: This block converts the ingested content into multiple representations of different resolution and bitrate, allowing the system to adapt video quality to each user's network conditions.

Packaging and Origin: Segments the profiles into fixed-duration fragments and generates the manifests that the client downloads to learn the available representations and the location of each segment.

Content Delivery Network (CDN): Its function is to deliver the service to a large mass of concurrent users through caching servers placed as close as possible to the end client.

Client Device: The player on the end device is responsible for real-time adaptation; it parses the manifest, estimates network conditions, and dynamically selects the optimal quality profile to ensure playback continuity.

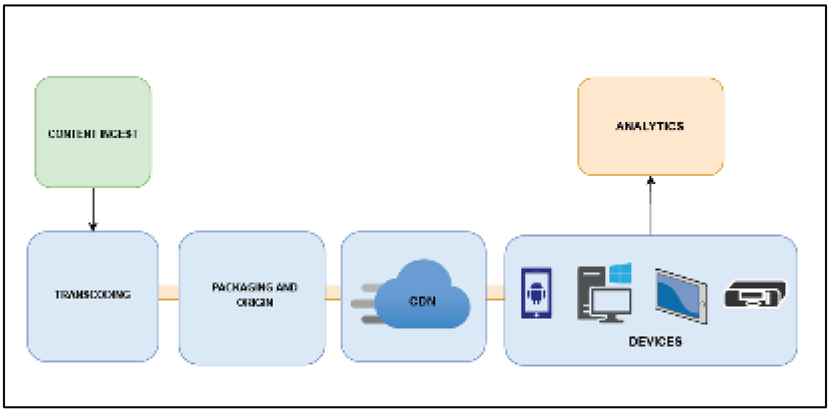

Figure 4 Functional blocks for HAS distribution.

Analytics: Monitors end-to-end key service indicators, start time, profile stability, and stall events, and feeds back into the system to dynamically optimize resources according to the actual state of the infrastructure.

5.2. Distribution Architectures

According to Fautier [5], content distribution via ABR protocols is articulated mainly through two distinct models, differentiated by the level of control the operator exercises over the network infrastructure. This same dichotomy, already introduced in Section 4.1 at the algorithmic level, is carried over here to the network architecture on which these algorithms operate.

The **managed model** is deployed over private infrastructures where video traffic is carried through dedicated virtual networks (VLANs) with delivery priority. This configuration makes it possible to implement deterministic Quality of Service (QoS) mechanisms from the headend to the terminal, guaranteeing a user experience fully controlled by the operator. This model is also referred to as IPTV 2.0.

By contrast, the **best-effort model** (*best effort*) uses the public internet as the primary transport medium, also known as *Pure OTT*. In this scenario, the operator lacks control over the underlying network and the subscriber's device, generally operating under a "Bring Your Own Device" (BYOD) scheme. To mitigate the performance fluctuations inherent to this modality, providers usually implement caching strategies by placing servers at access providers' points of presence (PoPs), seeking to reduce delivery latency.

Fautier also addresses an intermediate scheme based on operator-managed devices within the best-effort model. In this variant, although the content is distributed without network control, the operator supplies the receiving terminal (e.g., a *Set-Top Box*), which grants it a level of control over the client's application and software stack, enabling a significant improvement in Quality of Experience (QoE) compared to a Pure OTT solution, even though the underlying traffic still behaves in a best-effort manner.

5.3. From TCP to QUIC

The transport layer emerges, in turn, as a new frontier for optimization. QUIC is being increasingly adopted in place of TCP, the protocol around which existing ABR algorithms were designed, and it introduces a different network dynamic through features such as 0-RTT connection establishment and the absence of head-of-line blocking at the flow level. Mondal and Chakraborty [42] empirically document that these ABR algorithms are, in their current form, less compatible with QUIC than with TCP, in part because multiplexing audio and video over the same UDP socket introduces additional latency into audio segments, which raises the need for adaptation algorithms specifically designed to exploit this protocol's characteristics. More recent research already responds to this observation, with algorithms that exploit stream multiplexing and request cancellation native to HTTP/3 to significantly reduce playback interruption events, gradually closing that compatibility gap [44].

6. AI in Improving QoE

Beyond the classification of ABR algorithms presented in Section 4, recent research converges on a set of complementary trends that broaden the role of artificial intelligence in the video distribution chain.

6.1. Deep Reinforcement Learning (DRL) in Bitrate Adaptation

Unlike the heuristic and theory-based methods described in Section 4.1.2, which explicitly encode their decision rules through expert-designed conditions or a formal mathematical model, DRL formulates bitrate selection as a reinforcement learning problem in which an agent observes a system state at each step, chooses an action, and receives a reward from the environment, so that the decision policy is not manually programmed but is instead derived from training a neural network [35]. Pensieve [35], as a representative case of this approach, defines the state through six variables (the throughput measured over the last eight chunks, the corresponding download time, the available sizes of the next chunk at each bitrate level, the buffer occupancy, the remaining playback chunks, and the bitrate of the last downloaded chunk) and solves the resulting policy through the actor-critic A3C algorithm, in which an actor network of convolutional and dense layers outputs a softmax policy over the available bitrate levels, while a critic network of analogous architecture estimates the expected value of each state to guide the parameter update via the policy gradient.

Training does not take place over real downloads but over a chunk-level simulator fed with real network traces, drawn from FCC broadband datasets and Norwegian 3G/HSDPA mobile network datasets, which allows Pensieve to experience the equivalent of 100 hours of video playback in just 10 minutes and to generalize its policy even under synthetic traces not observed during training [35]. Against robustMPC, the best reference algorithm among those evaluated, Pensieve improves QoE by 15.5%, 18.9%, and 24.6% across the linear, logarithmic, and high-definition-biased utility functions, respectively, and reduces rebuffering events by between 10.6% and 32.8%, coming within just 9.6% to 14.3% of the offline optimum computed via dynamic programming with perfect knowledge of the future [35]. These results show that a policy learned from data, without explicitly coded decision rules, can consistently outperform both the heuristic methods and those based on model predictive control described in Section 4.1.2.

Pensieve's performance gave rise to an extensive family of subsequent DRL algorithms that build on its chunk-level simulator training scheme, among which are variants using federated learning, meta-learning, or adversarial training, a family within which the already-introduced PiTree [36] and Karma [37] are located, replacing the DRL policy with decision trees and a supervised learning model respectively [3]. Peroni and Gorinsky [3] warn, however, that the adoption of these algorithms in live transmissions faces a practical limitation, given that their high computational demand conflicts with the real-time decision requirements inherent to live streaming, which matches the tension between DRL model complexity and latency constraints already noted in Section 4.1.2.3.

6.2. Content-Aware Encoding

Compression efficiency, in parallel with adaptation logic, continues to be a determining factor for QoE. Although H.264 retains the largest adoption share, the industry is progressively migrating toward next-generation codecs such as HEVC, VP9, AV1, and VVC; recent comparisons report average bitrate savings of 63% (AV1) and 78% (VVC) versus H.264 for 8K content, measured via PSNR-based rate-distortion comparisons [41]. This efficiency gain widens the margin available for ABR algorithms to optimize quality within the same bandwidth budget. However, it does not come free, since each codec generation substantially increases the computational complexity and energy consumption of the encoding process (VVC, for instance, is up to eight times more complex than HEVC and quadruples its energy consumption), which introduces a new optimization dimension, beyond bandwidth, for distribution systems [44].

Facing this same compression-efficiency problem, the industry has identified that encoding an entire catalog under a fixed bitrate ladder is inefficient in the face of content of varying complexity, as illustrated by Timmerer et al. [44] with the case of content uniformly encoded at 8,100 kbps at 1080p under HEVC, where a low-complexity video could reach perceptually lossless quality at just 2,000 kbps while a high-complexity one could require a rate above the assigned one. One response to this problem is per-title encoding: De Cock et al. (2016), as reported in [44], encode the content into multiple bitrate-resolution combinations, evaluate the resulting quality in each, and select the combination that maximizes perceived quality for a given bitrate, rather than applying the same set of profiles to the entire catalog.

Given that exhaustively evaluating all encoding combinations demands high computation, poorly compatible with the real-time decision-making that live streaming requires, recent research is oriented toward methods that predict optimal encoding parameters rather than computing them by brute force [44]. Katsenou et al., as reported in [44], train a machine learning model that estimates the "crossover bitrate" between resolutions from the VMAF perceptual quality metric, while OPTE (Menon et al., cited in [44]) predicts in real time the optimal resolution for each bitrate level during live transmissions, drawing on video complexity extractors such as VCA, EVCA, and DeepVCA. Telili et al., also reported by [44], benchmark hand-crafted methods against deep learning methods for predicting the optimal bitrate ladder, which the authors present as evidence of the potential of AI-driven approaches as a viable alternative to computational brute force in live encoding scenarios.

Peroni and Gorinsky [3] document a complementary approach at the transcoding stage, in which DeepLadder solves the construction of the encoding ladder through reinforcement learning with the PPO algorithm, combining content features, available bandwidth, and storage costs to decide each chunk's profile, while MAMUT relies on multi-agent Q-learning to jointly adjust encoding threads, the quantization parameter, and processor frequency through a reward that integrates quality, bitrate, and energy consumption. These same authors note that, unlike the ingest stage where deep neural networks predominate, machine learning methods applied at the processing stage tend to rely on simpler models, such as decision trees or random forests, to speed up the selection of encoding parameters without incurring the computational cost of a deep network [3]. In live streaming deployments, this orientation toward content-aware encoding already reports measurable benefits, since ARTEMIS reduces encoding compute by 25% and end-to-end latency by 18% versus static ladders, while also improving QoE by 11%, and its extension ALPHAS for multiple simultaneous streams achieves QoE gains of up to 20% in dynamic scenarios [44].

6.3. Content-Aware Super-Resolution at the Edge

At the client end, integrating Super-Resolution (SR) with ABR algorithms allows high-definition video to be reconstructed from lower-resolution segments via neural networks, reducing the required bandwidth without sacrificing perceived quality. Proposals such as NAS [39] and SRAVS [40] illustrate this approach, in which the client takes on part of the computational load traditionally borne by the network.

Unlike the DRL methods described in Section 6.1, which train a single decision policy valid for any content, NAS [39] adopts a content-aware approach that trains a distinct super-resolution network for each video, deliberately exploiting neural networks' propensity to overfit, since the generalization error is unpredictable while the error on the training content itself is practically nil. The server hosts, offline, a family of networks based on a variant of the MDSR architecture, one per input resolution and per quality level (from "Low" to "Ultra-high"), organized so the client can apply a partially downloaded version of the network for an incremental quality improvement rather than waiting for the full download. At playback time, the client decodes each chunk, locally applies the corresponding super-resolution network, and replaces the decoded frame with the reconstructed version before inserting it into the playback buffer, so that the quality improvement occurs entirely on the client side without requiring a higher bitrate from the server.

NAS's decision component extends the A3C algorithm already described in Section 6.1, such that at each step the client decides whether to download a video chunk or a super-resolution network chunk, both competing for the same available bandwidth, and defines an effective bitrate that translates the quality gain obtained via SR into an equivalent nominal bitrate to compare its performance against conventional streaming. Evaluated on 27 YouTube videos under 3G and broadband network traces, NAS improves QoE by 43.08% over Pensieve and achieves up to 17.13% bandwidth savings for equivalent QoE, although it requires training a network per video at an approximate cost of 10 minutes of training per minute of content, which the authors estimate is amortized after 30 accumulated hours of viewing over a CDN, in addition to a considerable client-side compute cost, since the super-resolution stage dominates the processing time of each chunk (3.69 of 4.88 total seconds on a desktop GPU) [39].

SRAVS [40] instead adopts a generic rather than per-video strategy, training a single SRCNN model with only three convolutional layers on a broad set of videos and applying it indiscriminately to any content, on the grounds that a more complex model, though it may offer better quality, would not be viable in real time on the client. Unlike NAS, which serializes download and reconstruction, SRAVS keeps separate download and playback buffers that allow both processes to be parallelized, and its decision module, also based on A3C, widens the action space of traditional ABR to jointly choose the resolution to download and the target reconstruction resolution, incorporating into its state an estimate of the client device's compute capacity relative to a standard reconstruction time published by the server. Evaluated on heterogeneous devices, from GPU-equipped PCs to low-end mobile phones, SRAVS improves QoE by between 8% and 15% over Pensieve and reduces rebuffering by up to 50%, though it faces the same core conflict as NAS between downloading more content or spending that time reconstructing it, and it learns to avoid super-resolution on lower-capacity devices so as not to jeopardize the playback buffer [40].

Peroni and Gorinsky [3], in their review of the full distribution pipeline, further document edge-computing-assisted super-resolution systems such as VISCA, which caches low-resolution chunks at the network edge and upscales them via SR before serving them to the client, and note that, despite these advances, the use of super-resolution at the distribution stage toward the end user remains comparatively less common than at the ingest stage, which positions NAS and SRAVS as still relatively scarce references within an expanding line of research.

6.4. Sustainability: "Green ABR"

Deep reinforcement learning, already presented in Section 6.1 as the engine of bitrate adaptation, also extends toward a different objective, that of energy efficiency. GreenABR [45] incorporates energy consumption directly into its reward function, together with perceptual quality (VMAF) and rebuffering events, and thereby achieves savings of up to 57% in energy consumption and 60% in data consumption compared with state-of-the-art ABR algorithms, without sacrificing perceived QoE (which even improves by up to 22% in some scenarios).

GreenABR [45] breaks down energy consumption into two separately measurable components, local playback (processing and screen) and data acquisition over the network, and for the former trains a regression neural network on real power measurements taken with a precision power meter on a Samsung Galaxy S4 phone, using as input variables the bitrate, resolution, file size, scene-change rate (motion rate), and the chunk's VMAF, after empirically finding that energy consumption does not depend linearly on a single parameter such as bitrate, unlike what earlier work assumed. For data acquisition they adopt a throughput-based model already validated in the literature, with a reported accuracy of close to 90%. The reward function combines these two components with perceived quality through an exponential quality amplifier, which only activates above a VMAF threshold beyond which the perceptual improvement is no longer noticeable to the human eye, and an equivalent exponential amplifier for the energy penalty, so that the system avoids both an excessively quality-greedy and an excessively energy-greedy behavior [45].

Unlike Pensieve and the DRL algorithm family described in Section 6.1, which rely on policy-gradient methods such as A3C, GreenABR [45] opts for a four-layer DQN, justified by the authors on the grounds that this type of architecture is less sensitive to hyperparameter configuration and action-space size, in addition to converging in fewer training iterations. Evaluated against six reference algorithms, including BOLA and Pensieve itself, under 3G, 4G, and broadband network traces, and using a reference QoE model derived independently from the SQoE-III subjective dataset to avoid biasing the comparison in favor of its own reward function, GreenABR achieves energy savings of between 11% and 57% depending on the set of representations evaluated, with a QoE degradation of just 1% versus the best competitor in the most conservative scenario. When the authors themselves retrain Pensieve with the same energy-oriented reward function (Pensieve-E), it achieves 4% higher QoE than GreenABR, but at the cost of between 27% and 45% more energy consumption, evidence the authors cite for their design's better quality-energy balance [45].

Timmerer et al. [44] place GreenABR within a broader line of energy-aware ABR algorithms, which includes later extensions from the same group of authors aimed at generalizing the model to different sets of representations without retraining, as well as alternative systems such as E-WISH, which incorporates energy cost directly into a scheme based on bandwidth, buffer, and video quality. These same authors warn, however, that GreenABR's reliance on deep reinforcement learning adds complexity and demands substantial computational resources during training, which can hinder its implementation on low-power devices, a limitation that GreenABR partly mitigates by running only inference, not training, on the client, with its own energy overhead of just about 1% relative to that of video streaming [45]. Timmerer et al. [44] conclude that the energy efficiency of adaptive streaming remains an open problem that transcends the optimization of the individual ABR algorithm, since it also depends on decisions made in encoding, storage, and CDN distribution, which positions GreenABR as a first step within a broader design challenge for the entire distribution chain described in Sections 3 and 5 of this article.

Table 1 Summary of artificial intelligence techniques applied to adaptive video streaming

Technique	Representative algorithm	Point of intervention	Main quantitative result
Deep Reinforcement Learning (DRL)	Pensieve [35]	Client, bitrate decision	+15.5% to +24.6% QoE and -10.6% to -32.8% rebuffering versus robustMPC
Content-aware encoding	ARTEMIS / ALPHAS [44]	Server, encoding ladder construction	-25% encoding compute, -18% end-to-end latency, and +11% to +20% QoE versus static ladders

Content-aware super-resolution	NAS [39]	Client, quality reconstruction	+43.08% QoE and -17.13% bandwidth versus Pensieve
Generic super-resolution	SRAVS [40]	Client, quality reconstruction	+8% to +15% QoE and up to -50% rebuffering versus Pensieve
Energy-aware ABR	GreenABR [45]	Client, bitrate decision with energy-aware reward function	-11% to -57% energy consumption and up to +22% QoE versus state-of-the-art ABR algorithms

Note. Percentages represent the improvements reported by each study relative to its own baseline, which differs between works (see Sections 6.1 to 6.4). QoE = Quality of Experience. Source: own elaboration based on [35,39,40,44,45].

7. Conclusion

The review of recent literature places artificial intelligence in four complementary fronts that act on different points of the distribution chain described in Section 5, since deep reinforcement learning decides the bitrate at the client from a learned rather than explicitly programmed policy (Section 6.1), content-aware encoding predicts, on the server side, the optimal encoding parameters for each title instead of applying a fixed ladder (Section 6.2), super-resolution shifts part of the quality burden to the client device itself through networks trained per video or generically (Section 6.3), and energy-aware ABR introduces the device's power consumption as an additional variable in the reward function (Section 6.4). This diversity of fronts confirms that the heuristic, theoretical, and machine-learning classification presented in Section 4.1.2 for bitrate selection has extended toward other stages of the distribution pipeline, from encoding to client-side reconstruction.

From these four fronts emerges a cross-cutting finding that runs through much of the evidence reviewed, namely that the sophistication of the AI model is systematically in tension with the real-time decision requirement that live streaming demands: DRL algorithms face a computational demand that limits their live deployment [3], brute-force encoding is poorly compatible with live decision-making and research is therefore shifting toward predictive methods [44], NAS requires training a distinct super-resolution network for each video before it can be streamed [39], and GreenABR requires hours of offline training before it can be applied [45]. The QoE gained through AI therefore depends on there being room to absorb that computational cost before or during playback, a condition more easily met in video-on-demand than in live transmissions, as already noted in Section 4.1.2.3 regarding the general classification of ABR algorithms.

That said, the literature reviewed does not support the conclusion that artificial intelligence replaces the fundamentals of adaptive streaming, since its performance remains tied to the availability of contextual information, the quality of the metrics used to estimate QoE, and the constraints inherent to the transport network, such that an intelligent algorithm, however well trained, performs only as well as the distribution architecture in which it operates allows. This same transport constraint shows up in the migration from TCP to QUIC described in Section 5.3, where ABR algorithms designed around TCP still do not fully exploit the new protocol's characteristics [42], which confirms that the underlying network architecture continues to condition the margin for improvement available both to conventional adaptation algorithms and to the AI techniques that operate on top of them.

There is, however, an underlying challenge worth keeping in view, namely that the commercial deployment case where QoE degraded by as much as 300% when comparing a managed network with pure OTT distribution [5] shows what traditionally guaranteed quality in cable, control over its own network; today there are already cable operators distributing their services using HAS and under pure OTT models, where that control is no longer guaranteed by the infrastructure, and QoE comes to depend on the ABR algorithm's ability to anticipate and adapt to the network.

The four fronts reviewed in Section 6 and the shift toward QUIC described in Section 5.3 point in the same direction, that of a bitrate adaptation that is no longer thought of as an isolated function of the client player but instead integrates with server-side encoding, device-side reconstruction, energy consumption, and the underlying transport layer. The challenge this review leaves open is how to coordinate these fronts with one another, given that the quantitative evidence gathered in Sections 6.1 to 6.4 already consistently supports the QoE improvement that artificial intelligence delivers, without the aggregate computational cost of training and running several AI models within the same distribution chain ending up eroding the bandwidth and energy savings that each promises individually.

Compliance with ethical standards

Disclosure of conflict of interest

The authors declare that there exists no conflict of interest in this work.

Statement of informed consent

Informed consent is not applicable, as this study is a literature review that does not involve human participants.

References

[1] Smith ES. The emergence of CATV: a look at the evolution of a revolution. Proc IEEE. 1970;58(7):967-82.

[2] Globecast. Understanding the role and impact of TV headend as a service. Paris: Globecast/Orange; 2023.

[3] Peroni L, Gorinsky S. An end-to-end pipeline perspective on video streaming in best-effort networks: a survey and tutorial. ACM Comput Surv. 2024.

[4] Bentaleb A, Taani B, Begen AC, Timmerer C, Zimmermann R. A survey on bitrate adaptation schemes for streaming media over HTTP. IEEE Commun Surv Tutor. 2019;21(1):562-85.

[5] Fautier T. How OTT services can match the quality of broadcast. SMPTE Motion Imaging J. 2020;129(3):16-25.

[6] International Telecommunication Union. Digital multi-programme systems for television, sound and data services for cable distribution. ITU-T Recommendation J.83 Annex B. Geneva: ITU; 2007.

[7] European Telecommunications Standards Institute. Digital Video Broadcasting (DVB); framing structure, channel coding and modulation for cable systems. ETSI EN 300 429 V1.2.1. Sophia Antipolis: ETSI; 1998.

[8] Shihab E, Cai L. IPTV distribution technologies in broadband home networks. In: Canadian Conference on Electrical and Computer Engineering; 2007. p. 765-8.

[9] Boronat F, Montagud M, Marfil D, Luzon C. Hybrid broadcast/broadband TV services and media synchronization: demands, preferences and expectations of Spanish consumers. IEEE Trans Broadcast. 2018;64(1):52-69.

[10] Mijumbi R, Serrat J, Gorricho JL, Bouten N, De Turck F, Boutaba R. Network function virtualization: state-of-the-art and research challenges. IEEE Commun Surv Tutor. 2016;18(1):236-62.

[11] Viola R, Martin A, Zorrilla M, Montalban J. A survey on virtual network functions for media streaming solutions. ACM Comput Surv. 2022;54(10s):1-38.

[12] Reznik Y, Cenzano J, Zhang B. Transitioning broadcast to cloud. SMPTE Motion Imaging J. 2021;130(9):18-32.

[13] Bentaleb A, Begen AC, Zimmermann R. SDNDASH: improving QoE of HTTP adaptive streaming using software defined networking. In: Proceedings of ACM Multimedia 2016; 2016. p. 1296-305.

[14] Bentaleb A, Begen AC, Zimmermann R, Harous S. SDNHAS: an SDN-enabled architecture to optimize QoE in HTTP adaptive video streaming. IEEE Trans Multimedia. 2019;21(8):2136-53.

[15] Bhat D, Rizk A, Zink M. SABR: network-assisted content distribution for QoE-driven ABR video streaming. ACM Trans Multimed Comput Commun Appl. 2018;14(2s):57.

[16] Hosseinzadeh M, et al. CANE: a cascade control approach for network-assisted video QoE management. IEEE Trans Control Syst Technol. 2023;31:2543-54.

[17] Farahani R, Tashtarian F, Erfanian A, Timmerer C, Ghanbari M, Hellwagner H. ES-HAS: an edge- and SDN-assisted framework for HTTP adaptive video streaming. In: Proceedings of ACM NOSSDAV 2021; 2021.

[18] Farahani R, Tashtarian F, Amirpour H, Timmerer C, Ghanbari M, Hellwagner H. CSDN: CDN-aware QoE optimization in SDN-assisted HTTP adaptive video streaming. In: Proceedings of IEEE LCN 2021; 2021.

[19] Farahani R, Tashtarian F, Timmerer C, Ghanbari M, Hellwagner H. LEADER: a collaborative edge- and SDN-assisted framework for HTTP adaptive video streaming. In: Proceedings of IEEE ICC 2022; 2022.

[20] Aguilar-Armijo J, Çetinkaya E, Timmerer C, Hellwagner H. ECAS-ML: edge computing assisted adaptation scheme with machine learning for HTTP adaptive streaming. arXiv:2201.04488 [Preprint]. 2022.

[21] Farahani R, Shojafar M, Timmerer C, Tashtarian F, Ghanbari M, Hellwagner H. ARARAT: a collaborative edge-assisted framework for HTTP adaptive video streaming. IEEE Trans Netw Serv Manag. 2023;20(1):625-43.

[22] European Telecommunications Standards Institute. Digital Video Broadcasting (DVB); adaptive media streaming over IP multicast. Sophia Antipolis: ETSI; 2020.

[23] CableLabs. IP multicast adaptive bit rate architecture technical report [Internet]. 2016 [cited 2026 Jul 30]. Available from: https://apps.cablelabs.com/specification/ip-multicast-adaptive-bit-rate-architecture-technical-report/

[24] Farshad A, Nilsson M, Appleby S. Adaptive media streaming over IP multicast (mABR): a data analytic evaluation. In: Proceedings of the 2nd International Workshop on Design, Deployment, and Evaluation of Network-Assisted Video Streaming (ViSNext); 2022. p. 14-20.

[25] Huang TY, Johari R, McKeown N, Trunnell M, Watson M. A buffer-based approach to rate adaptation: evidence from a large video streaming service. In: Proceedings of ACM SIGCOMM 2014; 2014. p. 187-98.

[26] Juluri P, Tamarapalli V, Medhi D. SARA: segment aware rate adaptation algorithm for dynamic adaptive streaming over HTTP. In: Proceedings of IEEE ICCW 2015; 2015. p. 1765-70.

[27] Miller K, Al-Tamimi AK, Wolisz A. QoE-based low-delay live streaming using throughput predictions. ACM Trans Multimed Comput Commun Appl. 2016;13(1):4.

[28] Gutterman C, Fridman B, Gilliland T, Hu Y, Zussman G. STALLION: video adaptation algorithm for low-latency video streaming. In: Proceedings of ACM MMSys 2020; 2020. p. 327-32.

[29] Jiang J, Sekar V, Zhang H. Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE. IEEE/ACM Trans Netw. 2014;22(1):326-40.

[30] Li Z, et al. Probe and adapt: rate adaptation for HTTP video streaming at scale. IEEE J Sel Areas Commun. 2014;32(4):719-33.

[31] Yin X, Jindal A, Sekar V, Sinopoli B. A control-theoretic approach for dynamic adaptive video streaming over HTTP. In: Proceedings of ACM SIGCOMM 2015; 2015. p. 325-38.

[32] Qin Y, et al. A control theoretic approach to ABR video streaming: a fresh look at PID-based rate adaptation. In: Proceedings of IEEE INFOCOM 2017; 2017. p. 1-9.

[33] Spiteri K, Urgaonkar R, Sitaraman RK. BOLA: near-optimal bitrate adaptation for online videos. IEEE/ACM Trans Netw. 2020;28(4):1698-711.

[34] De Cicco L, Cilli G, Mascolo S. ERUDITE: a deep neural network for optimal tuning of adaptive video streaming controllers. In: Proceedings of ACM MMSys 2019; 2019. p. 13-24.

[35] Mao H, Netravali R, Alizadeh M. Neural adaptive video streaming with Pensieve. In: Proceedings of ACM SIGCOMM 2017; 2017. p. 197-210.

[36] Meng Z, Chen J, Guo Y, Sun C, Hu H, Xu M. PiTree: practical implementation of ABR algorithms using decision trees. In: Proceedings of ACM Multimedia 2019; 2019. p. 2431-9.

[37] Xu B, Chen H, Ma Z. Karma: adaptive video streaming via causal sequence modeling. In: Proceedings of ACM Multimedia 2023; 2023. p. 1527-35.

[38] CableLabs. Online Content Access (OLCA), authentication and authorization interface specification. CL-SP-AUTH1.0-C01-160616. Louisville: Cable Television Laboratories; 2016.

[39] Yeo H, Jung Y, Kim J, Shin J, Han D. Neural adaptive content-aware internet video delivery. In: Proceedings of USENIX OSDI 2018; 2018. p. 645-61.

[40] Zhang Y, et al. Improving quality of experience by adaptive video streaming with super-resolution. In: Proceedings of IEEE INFOCOM 2020; 2020. p. 1957-66.

[41] Uhrina M, Sevcik L, Bienik J, Smatanova L. Performance comparison of VVC, AV1, HEVC, and AVC for high resolutions. Electronics. 2024;13(5):953.

[42] Mondal A, Chakraborty S. Does QUIC suit well with modern adaptive bitrate streaming techniques? IEEE Netw Lett. 2020;2(2):85-9.

[43] ETSI NFV Industry Specification Group. Network Functions Virtualisation (NFV); architectural framework [Internet]. [cited 2026 May 24]. Available from: https://www.etsi.org/deliver/etsi_gs/nfv/001_099/002/01.02.01_60/gs_nfv002v010201p.pdf

[44] Timmerer C, Amirpour H, Tashtarian F, Afzal S, Rizk A, Zink M, et al. HTTP adaptive streaming: a review on current advances and future challenges. ACM Trans Multimed Comput Commun Appl. 2025;21(7):198.

[45] Turkkan BO, Dai T, Raman A, Kosar T, Chen C, Bulut MF, et al. GreenABR: energy-aware adaptive bitrate streaming with deep reinforcement learning. In: Proceedings of the 13th ACM Multimedia Systems Conference (MMSys '22); 2022 Jun; Athlone, Ireland. p. 150-63.

[46] Waterman D, Sherman R, Ji SW. The economics of online television: industry development, aggregation, and "TV Everywhere". Telecomm Policy. 2013;37(9):762-83.