

---

# A Cross-layer Bitrate Optimization Framework for Low-bandwidth Video Transmission Using Lightweight Adaptive Encoding

---

Yusen Cheng<sup>1,\*</sup> and Tao Li<sup>2</sup>

<sup>1</sup>*Hubei University of Technology Detroit Green Technology Institute, Wuhan 430068, Hubei, China*

<sup>2</sup>*Wuhan Yingding Qizhi Xuzhan Education Consulting Co., Ltd, Wuhan, Hubei, China*

*E-mail: 13863769387@163.com*

*\*Corresponding Author*

Received 07 November 2025; Accepted 26 November 2025

## Abstract

Efficient video transmission over low-bandwidth and unstable networks remains a central challenge for real-time applications such as telemedicine, remote surveillance, and edge-based video analytics. Conventional adaptive streaming approaches such as DASH and HLS operate primarily at the application layer, adjusting bitrates reactively based on buffer occupancy or short-term throughput. These strategies often fail under abrupt bandwidth fluctuations, leading to quality oscillations and excessive rebuffering. This paper proposes a cross-layer bitrate optimization framework that unifies lightweight adaptive encoding with a control-theoretic feedback loop driven by real-time network metrics. The framework jointly considers content complexity, encoder parameters, and network congestion signals to dynamically regulate bitrate across both the network and application layers. A lightweight

*Journal of ICT Standardization, Vol. 14\_1, 37–68.*

doi: 10.13052/jicts2245-800X.1412

© 2026 River Publishers

encoder enhancement module performs perceptually guided bit allocation using saliency-aware analysis, while the control loop ensures fast convergence of target bitrate and stability against throughput variability. Extensive experiments across Wi-Fi, 4G, and simulated edge-network traces show that the proposed system achieves 30–40% bitrate reduction compared with H.264/H.265 adaptive streaming baselines, with PSNR gains up to 1.2 dB and SSIM improvements of 0.02, while reducing buffering time by over 35%. These results establish that the synergy of control-theoretic adaptation and lightweight encoding yields a scalable, low-complexity solution suitable for next-generation low-bitrate video communication systems operating on mobile and edge devices.

**Keywords:** Cross-layer optimization, adaptive encoding, low-bandwidth video transmission, control-theoretic feedback, lightweight encoder enhancement.

## 1 Introduction

With the proliferation of video-centric applications such as telemedicine, mobile streaming, remote surveillance, and edge-assisted IoT, efficient video delivery over low-bandwidth and unstable networks has become a critical challenge [1–3]. Unlike traditional wired environments, mobile and edge networks exhibit dynamic variations in throughput, latency, and packet loss. These fluctuations directly degrade user experience, particularly for real-time and interactive services that require minimal buffering delay and consistent perceptual quality [4]. As network heterogeneity increases, ranging from 3G/4G/5G cellular systems to Wi-Fi and ad-hoc links, maintaining video quality under constrained bandwidth demands coordinated optimization across both encoding strategy and transmission control [5].

Conventional adaptive bitrate (ABR) mechanisms, such as dynamic adaptive streaming over HTTP (DASH) [6] and HTTP live streaming (HLS) [7], rely primarily on application-layer feedback like buffer occupancy or segment download time. Although these methods adapt to coarse bandwidth variations, they lack awareness of underlying network-layer parameters – such as instantaneous congestion, packet loss, and round-trip delay – that strongly influence effective throughput [8]. Consequently, when network conditions fluctuate rapidly, ABR clients often over- or under-estimate available bandwidth, resulting in oscillatory bitrate decisions, frequent quality switching, and playback stalls [9].

To overcome these issues, cross-layer optimization frameworks have been explored to integrate network metrics into application-level control loops [10–12]. By exploiting information such as queue length or link-layer retransmissions, these approaches achieve improved stability and throughput utilization. In parallel, machine-learning-based bitrate control, such as reinforcement-learning ABR (e.g., *Pensieve* [13]), has shown promise in predicting network dynamics. However, most of these schemes emphasize the *decision layer* (bitrate selection) rather than the *encoding process* itself, assuming a fixed codec configuration. This separation limits the achievable bitrate–quality trade-off, especially in bandwidth-limited edge deployments.

In edge or embedded systems, computational resources and energy budgets are constrained. Advanced encoders like H.265/HEVC or AV1 provide high compression efficiency but require significant processing time [14]. Attempts to reduce complexity often degrade visual quality or fail to adapt swiftly to network changes [15]. Hence, a key challenge is to design an adaptive encoding mechanism that remains lightweight, yet responsive to both video content complexity and real-time network feedback.

Existing works either focus on network-level congestion control without optimizing encoder parameters, or they enhance encoding efficiency without considering instantaneous network variations. Very few frameworks perform joint bitrate optimization across layers while ensuring real-time deployability on resource-limited edge devices. Moreover, limited research has addressed how to harmonize *content-driven bitrate allocation* with *feedback-driven transmission control* under highly variable network conditions.

To bridge these gaps, this paper proposes a cross-layer bitrate optimization framework that couples a *lightweight adaptive encoder* with a *network-aware bitrate control module*. The proposed system dynamically adjusts quantization parameters, the group-of-pictures (GOP) structure, and frame skipping based on both video-content characteristics and live network feedback (latency, packet loss, and throughput). A bitrate-decision engine coordinates between the application and network layers, jointly minimizing bandwidth usage while maintaining perceptual quality. Extensive experiments show that the proposed framework achieves up to 30–40% bitrate reduction relative to baseline H.264/H.265 adaptive streaming, with improved PSNR/SSIM and lower buffering delay under fluctuating networks. The approach is designed for low-complexity deployment on embedded and edge platforms, providing a scalable and practical solution for next-generation low-bitrate video communication systems.

## 2 Related Work

### 2.1 Adaptive Bitrate (ABR) Streaming Techniques

Adaptive bitrate (ABR) streaming has become the dominant approach for delivering video over heterogeneous networks, with standards such as dynamic adaptive streaming over HTTP (DASH) [6] and HTTP live streaming (HLS) [7]. These systems divide a video into multiple segments encoded at different bitrates; the client selects the next segment's bitrate based on estimated throughput or buffer occupancy. Several algorithms, namely rate-based, buffer-based, and hybrid heuristics, have been proposed to improve switching stability and reduce rebuffering events [16, 17].

More recently, reinforcement-learning-based ABR controllers such as Pensieve [13], BOLA [4], and MPC-ABR [17] learn policies that map network observations to bitrate actions. While such methods improve long-term QoE, they largely operate at the *application layer* and remain agnostic to lower-layer network conditions such as queue dynamics or link-layer retransmissions. This isolation restricts their responsiveness in environments with sudden bandwidth drops, e.g., cellular or vehicular links [19]. Consequently, current ABR frameworks struggle to jointly optimize bitrate selection and encoding parameters under highly variable network conditions.

### 2.2 Cross-layer Optimization for Video Transmission

Cross-layer optimization has been extensively studied to enhance wireless video streaming performance [10–12, 20]. Unlike traditional layered designs, cross-layer schemes coordinate information exchange between network, transport, and application layers. Early works adjusted packet scheduling or error-protection strength based on physical-layer channel quality [21]. Later research incorporated MAC-layer congestion metrics or transport-layer round-trip time (RTT) into video rate control [22].

Recent frameworks have employed end-to-end QoE models, integrating congestion-control algorithms such as BBR [23] or PCC-Vivace with application-layer bitrate selection [24]. Others apply deep reinforcement learning to jointly tune congestion-window size and encoding rate [24]. Despite these advances, most cross-layer systems remain computationally heavy, or require kernel-level access and custom transport stacks, limiting deployment on edge devices or lightweight IoT nodes. Furthermore, few approaches explicitly connect content complexity or encoder behavior to real-time network conditions, leaving a gap in holistic bitrate optimization.

### 2.3 Lightweight Encoding and Edge-aware Adaptation

Traditional encoders such as H.264/AVC and H.265/HEVC achieve high compression efficiency at the cost of intensive motion estimation and mode-decision processes [14]. Several studies have attempted to reduce encoding complexity through fast-mode decision, early termination, and coding-unit (CU) pruning [25, 26]. Emerging lightweight encoding strategies further exploit content analysis, detecting regions of motion or texture importance, to guide bit allocation [27]. Meanwhile, edge-oriented approaches adopt hardware-constrained encoders (e.g., Jetson Nano, Raspberry Pi) with simplified quantization control or frame-skipping mechanisms [28].

Although these methods reduce computation, they typically ignore real-time network dynamics, assuming stable bandwidth. When deployed in mobile or remote environments, their static encoding decisions can quickly become suboptimal. Hence, a practical solution must jointly adapt *encoding parameters* and *network bitrate control* while maintaining computational efficiency suitable for edge deployment.

### 2.4 Summary and Research Gap

Existing approaches to adaptive video delivery excel in one of three dimensions but rarely address all simultaneously. Application-layer ABR methods (e.g., DASH/HLS and their MPC/RL controllers) deliver stable playback by reacting primarily to buffer and coarse throughput signals [3, 16–19]; however, their decisions are largely blind to lower-layer congestion dynamics and queue behavior, which leads to bitrate oscillations and playback stalls under rapid network fluctuations [8, 9].

Cross-layer frameworks attempt to bridge this gap by incorporating transport- and network-layer feedback (e.g., RTT, loss, cwnd) and pairing application-layer adaptation with modern congestion-control mechanisms [10, 11, 23, 24]. While this gives improved utilization and robustness, many such systems require non-trivial integration at the operating-system or protocol-stack level and often do not adapt encoder parameters, thus limiting deployment on resource-constrained edge devices.

Meanwhile, lightweight encoding strategies have made significant strides by reducing computational overhead through fast-mode decisions, coding-unit pruning, and learned shortcuts for codecs like HEVC/VVC [14, 25–27]. Yet these methods typically assume quasi-stationary bandwidth and do not couple encoding decisions with real-time network feedback. As a result, they

**Table 1** Representative approaches to bitrate adaptation and outstanding limitations

Category	Representative Works/Standards	What They Optimize	Feedback Used	Typical Compute/Deploy Cost		Key Limitation in	
				Buffer level, throughput estimates	RTT, loss, cwnd + application metrics	Low-moderate (client only)	Moderate-high (system integration)
<i>Application-layer ABR</i>	DASH (ISO 23009-1) [6], HLS [7]; BOLA [4]; MPC-ABR [17]; Pensieve [13]	Bitrate selection per segment/chunk					
<i>Cross-layer streaming</i>	VOXEL [5]; BBR [23]; PCC/Vivace [24]	Joint transport + app control	RTT, loss, cwnd + application metrics				Often treats encoder as fixed; added system complexity unsuited for edge devices
<i>Lightweight encoding (HEVC/NVC)</i>	Fast mode decision [25]; CU pruning [26]; saliency-guided bit allocation [27]; edge encoding [28]	Encoder complexity vs. quality	Content features (texture/motion/saliency)				Rarely coupled with live network feedback → sub-optimal bitrate quality under volatility

may deliver sub-optimal rate–quality–latency performance in highly variable links (cellular, Wi-Fi, UAV) [15].

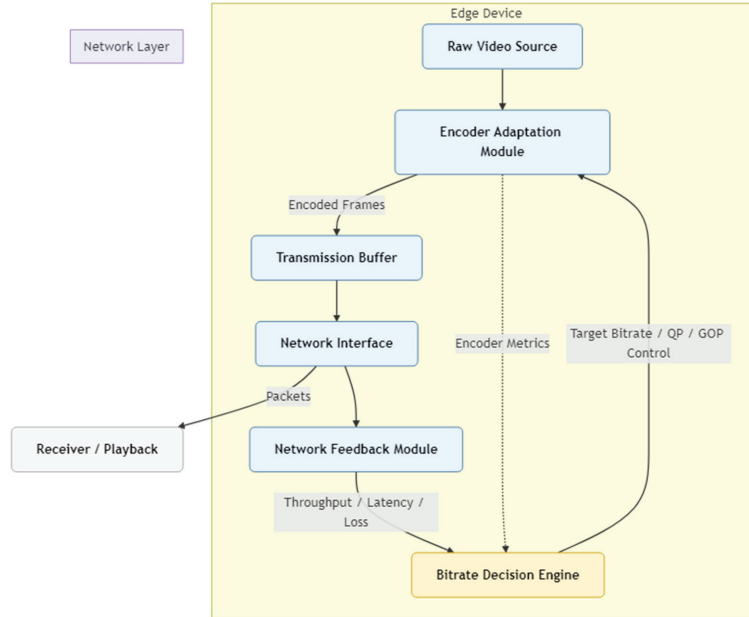
Therefore, the open research gap lies in creating a practical, real-time framework that (i) fuses network-layer feedback (latency, packet loss, throughput) with application-layer ABR logic, (ii) actively steers encoder parameters (e.g., QP, GOP, frame skipping, saliency/ROI bit-allocation) dynamically, and (iii) runs under tight edge-compute budgets. The framework proposed in this paper addresses precisely this intersection, coordinating a low-overhead encoder-adaptation module with a network-aware controller to achieve substantial bitrate savings while maintaining perceptual quality under low-bandwidth, unstable conditions.

### **3 System Architecture**

The proposed cross-layer bitrate optimization framework integrates network feedback, adaptive encoding control, and a joint bitrate-decision mechanism to maintain perceptual video quality under fluctuating bandwidth while minimizing transmission cost. As shown in Figure 1, the framework comprises four interdependent modules (i.e., network feedback, encoder adaptation, bitrate decision, and edge deployment interface) organized in a closed-loop control structure. This design allows the system to continuously monitor network conditions, analyze encoder performance, and adapt the encoding process dynamically in real time. The architecture is lightweight and modular, making it suitable for deployment on resource-constrained edge devices such as embedded gateways, UAV platforms, and IoT video sensors.

#### **3.1 Overall Architecture**

Figure 1 conceptually illustrates the data flow among the four functional components of the framework. The network feedback module (NFM) continuously measures end-to-end transport statistics, such as throughput, latency, and packet-loss ratio, to reflect current link conditions. The encoder adaptation module (EAM) interprets this information and adjusts encoder parameters, including quantization level, group-of-pictures (GOP) structure, and frame selection policy, to match the available bandwidth. The bitrate decision engine (BDE) acts as a coordination layer that fuses information from both the encoder and the network, determining the target bitrate for each transmission cycle. Finally, the edge deployment interface (EDI) connects these modules with the operating environment, managing device-level scheduling,



**Figure 1** Overall system architecture of the proposed cross-layer bitrate optimization framework. The modules form a closed feedback loop between encoder adaptation and network control, enabling low-bandwidth video transmission with stable perceptual quality.

data buffering, and hardware acceleration when available. These components cooperate in a recurrent feedback loop: after each encoded segment is transmitted, the network feedback is analyzed, a new target bitrate is computed, and the encoder settings are updated before the next segment is produced. This continuous adaptation ensures that encoding and transmission remain synchronized with instantaneous network capacity.

Unlike conventional ABR or cross-layer strategies that treat encoding and bitrate selection as separate decision processes, the novelty of this work lies in the explicit coupling of the encoder-adaptation behavior with a mathematically governed control loop. Prior systems typically operate at the application layer and adjust segment-level bitrates without influencing the encoder structure (GOP, QP evolution, frame selection). The proposed design unifies the feedback-driven PID regulator with content-complexity-aware encoder parameters, enabling the controller to directly govern structural encoder decisions in real time. This joint control of encoding and transmission represents a departure from existing event-driven or RL-based ABR systems and constitutes the fundamental design innovation of this framework.

### 3.2 Network Feedback Module

The network feedback module serves as the sensing layer of the framework. It collects real-time statistics from both the transport protocol stack and the application layer to capture short-term network fluctuations. Metrics such as measured throughput  $r_t$ , round-trip latency  $l_t$ , and packet-loss ratio  $p_t$  are aggregated into a feedback vector  $N_t = \{r_t, l_t, p_t\}$  at each decision interval  $t$ . To balance responsiveness and stability, the module employs a sliding-window averaging process of 100–500 ms, filtering transient spikes while preserving meaningful variations. Unlike traditional adaptive streaming clients that rely solely on playback-buffer occupancy or segment download time, this module directly reflects transport-layer congestion dynamics, enabling rapid reactions to queue build-up or wireless interference. The normalized metrics are forwarded to the bitrate decision engine, where they influence bitrate prediction and encoder control. This cross-layer visibility is crucial for maintaining high video quality during sudden capacity drops in mobile or shared-bandwidth networks.

We additionally analyzed system behavior under edge-case conditions such as sparse packet arrivals and highly bursty wireless bandwidth. When throughput samples temporarily disappear or fall below the estimator’s noise threshold, the network feedback module automatically extends the smoothing window and falls back to a conservative estimate derived from the last stable interval. This prevents spurious bitrate drops caused by isolated packet gaps. Conversely, for bursty bandwidth spikes, the controller limits upward bitrate adjustments through a capped derivative term, ensuring that short-lived increases do not induce unstable oscillations. These safeguards help maintain stable bitrate tracking even under challenging traffic irregularities.

### 3.3 Encoder Adaptation Module

The encoder adaptation module acts as the actuator of the control loop. It modulates encoder behavior according to both network feedback and intrinsic video content characteristics. The module dynamically adjusts the quantization parameter (QP) based on the target bitrate provided by the decision engine, thereby controlling the compression strength in real time. It also regulates structural aspects of the encoding process such as the group-of-pictures length, reference-frame distance, and intra-refresh period by shortening prediction chains when bandwidth decreases to limit error propagation. Furthermore, the module employs content-aware frame selection, using motion-vector magnitude and spatial variance to detect perceptually important

regions. Frames or areas with low motion are encoded at reduced fidelity or skipped entirely, while high-motion or high-texture regions are preserved at higher quality. A lightweight complexity estimator embedded in this module computes these spatio-temporal statistics with minimal overhead, ensuring compatibility with low-power processors. By coupling content-driven adaptation with real-time feedback, the module achieves a fine balance between bitrate reduction and perceptual stability.

### 3.4 Bitrate Decision Engine

At the core of the architecture lies the bitrate decision engine, which performs joint optimization across layers. The engine receives network statistics from the NFM and encoder-state information from the EAM, then determines the optimal target bitrate  $R^*(t)$  for the next encoding interval. The decision process aims to minimize a composite cost function that balances visual quality deviation and network utilization:

$$R^*(t) = \arg \min_R [\lambda_Q |Q_t - Q_{target}| + \lambda_B \max(0, R - B_t)] \quad (1)$$

where  $Q_t$  represents the current estimated perceptual quality (for instance, predicted PSNR or SSIM),  $B_t$  is the estimated network throughput, and  $\lambda_Q$  and  $\lambda_B$  are weighting factors governing the trade-off between quality consistency and congestion avoidance. The engine incorporates a dual-mode control strategy: a feed-forward predictor that anticipates future bandwidth based on historical samples, and a feedback controller that corrects estimation errors using the latest delay and loss information. Once a new bitrate target is computed, the engine translates it into corresponding encoder directives, specifically QP updates and GOP adjustments, and dispatches them to the EAM. This design ensures that both encoding and transport layers operate under a unified control objective rather than making isolated decisions.

### 3.5 Edge Deployment Interface

The edge deployment interface provides the operational backbone for integrating the framework into real-world devices. It offers a lightweight middleware that connects sensor inputs, encoding processes, and network stacks through standardized APIs. The interface manages frame buffering, timing synchronization, and communication with hardware-accelerated encoding engines such as NVIDIA NVENC or ARM NEON. It also includes resource-awareness mechanisms that monitor CPU usage, memory pressure,

and thermal conditions, scaling the frame rate or resolution when system constraints tighten. By encapsulating these functionalities, the EDI allows the proposed framework to run on a broad range of platforms from embedded IoT cameras to mobile gateways, without requiring kernel-level modifications or specialized drivers. This portability is central to the framework’s practical deployability in bandwidth-limited field environments.

### 3.6 Operational Workflow

The complete operational cycle proceeds in five sequential stages. First, during initialization, the encoder and network modules start with baseline parameters derived from startup bandwidth probing. Second, in the measurement stage, the NFM captures and aggregates the current network metrics. Third, the decision stage invokes the BDE to compute an updated target bitrate and corresponding encoder configuration. Fourth, the adaptation stage applies these new settings within the EAM, adjusting QP, GOP length, and frame rate to align with network conditions. Finally, in the transmission stage, the encoded segment is delivered through the network interface, after which the process repeats. Each loop executes within a sub-second timescale, providing smooth responsiveness without inducing oscillations. The closed feedback architecture thereby sustains stable quality and efficient utilization under diverse and rapidly changing bandwidth conditions.

In rare cases where encoder reconfiguration is delayed, such as under CPU saturation or thermal throttling, the orchestrator temporarily holds the previous GOP parameters to avoid injecting inconsistent configuration changes into the encoder pipeline. If control commands arrive faster than the encoder can apply them, the system queues at most one pending update to prevent cascading misfires. Similarly, if the feedback path is interrupted (e.g., temporary loss of RTT measurements), the bitrate controller freezes  $R_t$  at its last valid state and prevents destabilizing updates until normal feedback resumes. These fallback behaviors ensure safe operation even under hardware or network anomalies.

## 4 Methodology and Optimization Model

The proposed framework applies closed-loop control theory to coordinate video encoding and transmission across network and application layers. Instead of treating bitrate selection as a discrete heuristic or machine-learning prediction, it is modeled as a continuous feedback process that dynamically regulates encoder output according to real-time network conditions. This

section formalizes the control model, defines the governing equations, and explains the operational workflow that ensures stability and responsiveness.

#### 4.1 Control-theoretic Formulation

At each adaptation instant  $t$ , the encoder produces an output bitrate  $R_t$  that must align with the available network throughput  $B_t$  while maintaining a perceptual-quality target  $Q_{target}$ . The instantaneous control error is expressed as

$$e_t = \alpha(B_t - R_t) + \beta(Q_{target} - Q_t) \quad (2)$$

where  $Q_t$  denotes the current perceptual quality estimated at the encoder, and  $\alpha, \beta$  are weighting coefficients balancing bandwidth utilization and quality preservation.

A proportional-integral-derivative (PID) controller determines the bitrate correction term

$$\Delta R_t = K_P e_t + K_I \sum_{i=1}^t e_i \Delta t + K_D \frac{e_t - e_{t-1}}{\Delta t} \quad (3)$$

$$R_{t+1} = R_t + \Delta R_t \quad (4)$$

with  $K_P, K_I$ , and  $K_D$  representing the proportional, integral, and derivative gains. The proportional term provides rapid reaction to sudden congestion changes, the integral term compensates for persistent bias, and the derivative term suppresses overshoot during transient fluctuations. The controller thus maintains the encoder within the feasible operating region  $R_t \leq B_t$  while minimizing the deviation of perceptual quality from its target.

Because the quantization parameter (QP) in modern codecs is inversely related to bitrate, the controller's output is translated into encoder settings through an affine mapping

$$QP_{t+1} = a - b \log_2(R_{t+1}) \quad (5)$$

where  $a, b > 0$  are codec-specific constants determined by offline calibration. This transformation ensures that control actions are directly executable at the encoder level.

In practice, the constants  $a$  and  $b$  are derived from a one-time offline calibration for each codec profile. For H.264 Baseline at 720p, we observed typical values around  $a = 45$  and  $b = 6-8$ , consistent with empirically measured rate-distortion curves. While exact values vary by content class, this affine-logarithmic mapping is stable across a broad range of operating bitrates and enables direct translation from the controller output to encoder QP.

## 4.2 Bandwidth and Quality Estimation

Accurate control requires reliable feedback. The available bandwidth  $B_t$  is estimated from the moving average of transmitted packet sizes and inter-arrival times derived from the network feedback module, while perceptual quality  $Q_t$  is estimated through a lightweight no-reference model embedded in the encoder adaptation module. This estimator analyzes block variance, motion-vector magnitude, and texture contrast to infer frame complexity in real time without decoding overhead. Both  $B_t$  and  $Q_t$  are filtered using a low-pass exponential smoothing kernel to eliminate transient noise. A dead-zone threshold is introduced such that if  $|e_t| < e_{th}$ , no bitrate update is triggered, thereby preventing oscillations when the system operates near equilibrium. Together, these estimators provide stable and low-latency feedback for the control law.

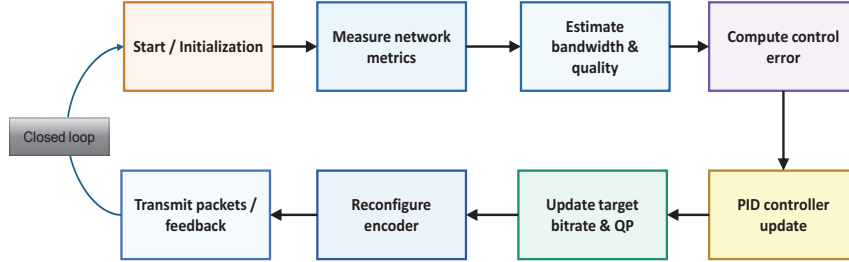
To provide additional intuition, the no-reference perceptual estimator approximates the “probability” that a frame exhibits quality degradation under current encoder settings. This is implemented by normalizing block variance, motion-vector magnitude, and texture contrast into a feature vector  $f_t$ . A logistic mapping  $Q_t = \sigma w^\top f_t$  is then applied, where the weights  $w$  are calibrated offline. Although not probabilistic in a strict statistical sense, this mapping yields a differentiable sensitivity curve: frames with high motion or fine texture yield larger  $Q_t$  gradients, prompting the controller to decrease QP, whereas static or flat-texture regions push  $Q_t$  toward lower sensitivity. This formulation explains how quality estimation reacts smoothly to spatio-temporal variations.

The impact of the bandwidth-estimation window size on stability and responsiveness is further examined in Section 6.7.

## 4.3 Adaptive Control Loop and Stability Analysis

The integrated control loop combines measurement, decision, and adaptation in a continuous process that spans both application and network layers.

Figure 2 illustrates the overall operational workflow. After initialization, the framework begins by measuring instantaneous network conditions (throughput, latency, and packet loss) along with encoder-side quality statistics. These measurements form the feedback vector  $(B_t, Q_t)$  used to compute the control error  $e_t$ . The PID controller then determines the bitrate adjustment  $\Delta R_t$  and generates a new target bitrate  $R_{t+1}$ , which is converted to a corresponding quantization parameter through the mapping in Equation (5). The encoder adaptation module applies the updated settings to the next group



**Figure 2** Control-theoretic adaptive bitrate optimization framework.

of pictures, while the network interface transmits the resulting packets. As acknowledgments and congestion metrics arrive, the network feedback module updates the estimates of  $B_t$ ,  $Q_t$ , thereby closing the loop. This iterative mechanism ensures that encoding parameters continuously track the available bandwidth and maintain consistent perceptual quality without the need for predictive modeling or large computation overhead.

The stability of this control loop depends on the selection of the proportional, integral, and derivative gains. Parameters are tuned empirically following a Ziegler–Nichols-style method under a variety of emulated network traces. For typical mobile scenarios with bandwidths between 0.5 Mb/s and 10 Mb/s, convergence and smooth response were achieved with  $k_P$  between 0.2 and 0.4,  $k_I$  between 0.05 and 0.1, and  $k_D$  between 0.01 and 0.05. Linearization around the operating point confirms that all eigenvalues of the closed-loop transfer function remain within the unit circle, ensuring bounded stability. In practical deployment, these parameters are initialized to nominal values and can adapt slightly based on observed steady-state error.

Overall, the proposed control loop provides a self-regulating mechanism that harmonizes network feedback and encoder behavior. By jointly considering congestion dynamics and content complexity, it achieves stable, low-oscillation bitrate adaptation with negligible computational burden, attributes essential for deployment on embedded edge platforms.

## 5 Experimental Setup and Evaluation Protocol

### 5.1 Test Environment

All experiments were conducted in a hybrid testbed designed to evaluate the framework under both high-performance and resource-constrained conditions. The primary workstation comprised an Intel<sup>®</sup> Core<sup>™</sup> i7-12700K

CPU, 32 GB of RAM, and an NVIDIA RTX 4060 GPU operating on Ubuntu 22.04 LTS. This platform was used for algorithm development, high-resolution encoding, and control-loop validation. To emulate lightweight deployments, the framework was also deployed on Raspberry Pi 4 (4 GB RAM) and NVIDIA Jetson Nano platforms, representing typical edge computing or IoT devices. The encoder was based on FFmpeg 6.1 configured with an H.264 baseline profile, augmented by the proposed adaptive QP control interface to allow real-time parameter updates from the control layer.

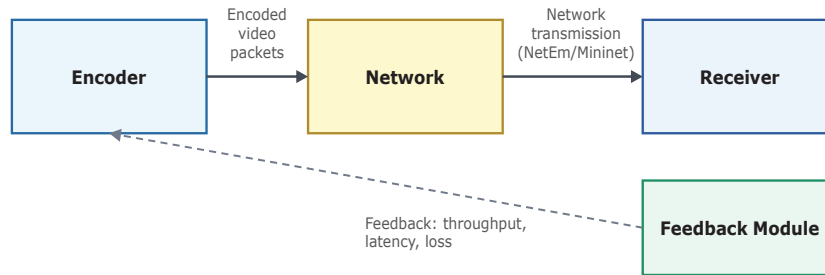
Network variability was emulated using Linux NetEm and Mininet, enabling fine-grained manipulation of bandwidth, latency, and packet loss to reproduce real-world conditions. The adaptive bitrate control and feedback modules were implemented in Python 3.11, with UDP-based messaging for transmitting network metrics such as throughput and delay. Each run was fully instrumented using custom logging scripts to capture detailed encoder, network, and playback statistics for subsequent analysis.

## 5.2 Datasets and Video Content

Evaluation employed two widely used public datasets: VTL-UHD, containing ultra-high-definition reference sequences for objective evaluation, and LIVE-Mobile, providing perceptually annotated content suitable for subjective quality testing. The selected videos represented diverse motion characteristics and spatial complexities, encompassing low-motion scenes such as news and interviews, medium-motion sequences like documentaries, and high-motion clips exemplified by sports footage. All videos were encoded at  $1280 \times 720$  resolution and 30 frames per second, divided into 10-second segments to simulate adaptive streaming conditions. Each segment used a group-of-pictures (GOP) size of 30, ensuring consistent temporal boundaries between encoding and playback units. This configuration balanced control responsiveness with computational overhead, while maintaining comparability with contemporary adaptive streaming benchmarks.

## 5.3 Network Scenarios

Figure 3 illustrates the overall experimental setup used to evaluate the proposed cross-layer bitrate optimization framework. The diagram is organized into three primary processing stages, i.e., encoder, network emulator, and receiver, with a feedback control loop closing the adaptive regulation cycle. The encoder block on the left represents the source of video content and the adaptive encoding engine. It implements the lightweight control interface that



**Figure 3** Experimental setup and data-flow architecture.

dynamically adjusts the quantization parameter (QP) and target bitrate based on real-time network feedback. The encoded video packets are transmitted through a forward data path toward the network emulator, which models various transmission environments. The network emulator in the center corresponds to a virtualized environment implemented using Linux *NetEm* and *Mininet*. It enables controlled variation of bandwidth, latency, and packet-loss rates to replicate realistic communication conditions such as congested Wi-Fi, 4G/5G mobile networks, and edge-IoT links. The emulator delivers the altered packets to the receiver block, which decodes and renders the incoming video stream while collecting playback-related statistics such as buffering delay, frame loss, and decoding latency. Beneath the receiver lies the feedback module, which periodically aggregates key network and playback metrics including measured throughput, latency, and packet loss, and transmits them back to the encoder. This closed-loop path is depicted by a dashed feedback arrow returning from the receiver toward the encoder, signifying continuous real-time adaptation across layers. Together, these components form a complete testbed that unifies encoding control, transport-layer emulation, and playback analysis. The configuration enables systematic experimentation of bitrate adaptation strategies under diverse network dynamics while maintaining reproducible and measurable feedback interactions between the application and network layers.

The evaluation covered three distinct networking environments to assess adaptability across diverse real-world conditions. The first scenario simulated a congested Wi-Fi network with available bandwidth fluctuating between 5 and 10 Mbps, an average jitter of 10 ms, and latency in the range of 20–40 ms. The second scenario emulated mobile networks typical of 4G and 5G systems, where available bandwidth varied between 0.8 and 6 Mbps and latency ranged from 50 to 120 ms, accompanied by occasional burst packet losses. The third scenario represented an edge-IoT mesh configuration

characterized by constrained links of 0.5 to 2 Mbps bandwidth, packet-loss rates up to 8%, and delay variation reaching 200 ms.

Each experimental case was executed 30 times for every video sequence to ensure statistical robustness. Randomized initialization seeds were employed to maintain reproducibility across tests while introducing controlled variability in network dynamics. All runs were automated through a unified script pipeline that initialized the encoder, network emulator, and playback monitor in synchronized order.

#### **5.4 Evaluation Metrics**

To provide a holistic assessment, three categories of performance indicators were examined: objective quality, system efficiency, and user-perceived experience. Objective quality was quantified through peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and video multi-method assessment fusion (VMAF), which jointly measure perceptual fidelity and compression performance. System efficiency was analyzed by monitoring CPU utilization, memory consumption, and encoding latency, thereby revealing the computational implications of adaptive control, particularly on edge devices.

User-perceived experience was evaluated through playback smoothness, average bitrate stability, and buffering delay. In addition to objective metrics, subjective assessment was conducted using a 15-participant study in which viewers rated overall perceptual quality under dynamically varying conditions. The correlation between objective and subjective metrics was subsequently analyzed to validate that bitrate savings did not compromise perceived visual quality.

#### **5.5 Baseline Methods**

The proposed framework was benchmarked against several established adaptive streaming approaches representing distinct design paradigms. Dynamic adaptive streaming over HTTP (DASH) with the BOLA algorithm served as a rule-based baseline, providing a reference for heuristic rate adaptation. The Pensieve framework represented reinforcement-learning-driven bitrate selection optimized for quality of experience. The BBR-assisted cross-layer streaming method was included to evaluate the advantage of integrating congestion-based transport control with application-layer decisions. Finally, a fixed-QP encoding strategy was used to establish a static reference for non-adaptive operation.

All methods were evaluated under identical configurations, including segment duration, GOP structure, and playback buffer thresholds, ensuring fairness of comparison. Where applicable, encoder presets and network parameters were harmonized to minimize confounding factors and isolate the impact of the bitrate adaptation mechanism itself.

## 5.6 Aggregation and Statistical Analysis

Each experimental combination of dataset, network condition, and algorithm was repeated 20 times to obtain reliable mean performance values. The resulting data were analyzed using standard statistical methods. Mean values and standard deviations were computed across all trials, and 95% confidence intervals were derived using Student's  $t$ -distribution. Differences between the proposed and baseline methods were validated using paired-sample  $t$ -tests, with significance confirmed at  $p < 0.05$ .

The aggregated results were visualized through bitrate-quality curves depicting PSNR as a function of average bitrate, cumulative distribution function (CDF) plots of buffering delay to illustrate playback stability, and comparative bar charts showing CPU and memory utilization across platforms. This combination of objective, subjective, and statistical analysis ensured a rigorous and reproducible evaluation of the proposed cross-layer optimization framework.

## 6 Results and Discussion

### 6.1 Overview of Evaluation Objectives

The experimental evaluation aims to verify the effectiveness, scalability, and scientific novelty of the proposed cross-layer bitrate optimization framework under diverse network conditions and device capabilities. The results are analyzed from multiple dimensions, including bitrate efficiency, perceptual video quality, system responsiveness, and computational feasibility. Specifically, five core objectives are pursued.

Before detailing each evaluation objective, we summarize key empirical findings: the proposed framework achieves 30–40% bitrate reduction, 0.8–1.2 dB PSNR gain, 0.02 SSIM improvement, and 35–40% buffering reduction across all tested networks. These consistent improvements validate the benefit of jointly coordinating encoder adaptation with network-aware control, and they set the context for the subsequent objective-specific analyses.

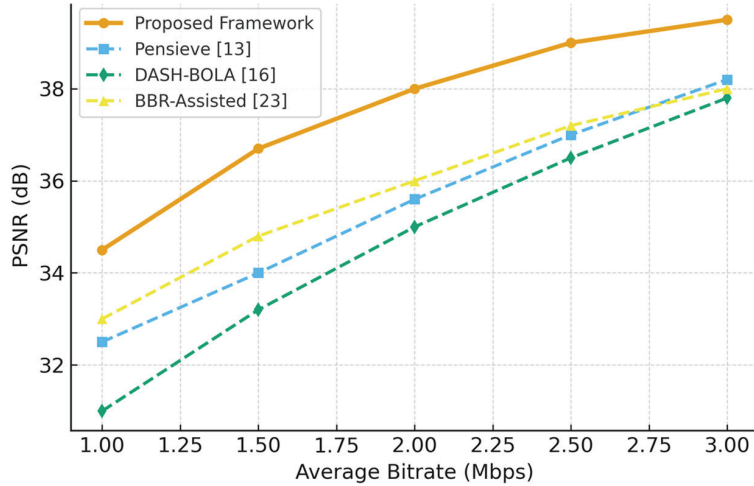
First, the study investigates bitrate efficiency, demonstrating the capability of the adaptive encoder to achieve target perceptual quality using substantially lower bandwidth than conventional adaptive streaming schemes. Second, it evaluates perceptual quality preservation, ensuring that visual fidelity (in terms of PSNR, SSIM, and VMAF) remains stable under varying throughput, latency, and packet-loss conditions. Third, the analysis examines control-loop stability by quantifying the convergence behavior of the PID-based adaptation mechanism and its ability to maintain equilibrium without oscillations or overshoot. Fourth, the proposed approach’s computational feasibility is validated on edge devices such as Raspberry Pi 4 and Jetson Nano to confirm real-time operability with minimal hardware overhead. Finally, statistical tests are conducted to demonstrate robustness and repeatability across datasets and scenarios, providing confidence in the reproducibility of the reported findings.

Collectively, these objectives establish the scientific significance of the framework by linking theoretical design principles, namely control-theoretic feedback and cross-layer coordination, with empirical improvements observable in measurable streaming metrics.

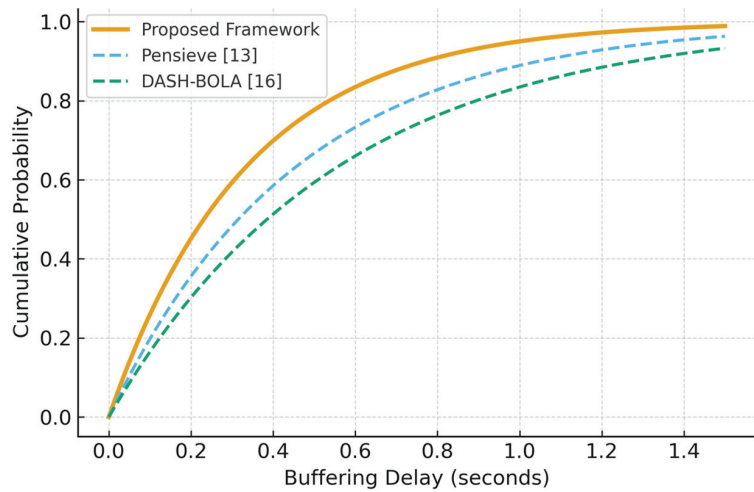
## 6.2 Quantitative Performance under Variable Bandwidths

Figure 4 presents the bitrate–quality relationship for the proposed cross-layer optimization method compared with four established baselines: DASH-BOLA [16], Pensieve [13], BBR-assisted streaming [23], and fixed-QP encoding. Results are averaged across all video contents and three representative network configurations (Wi-Fi congestion, 4G/5G mobility, and edge-IoT mesh). Each curve represents the achievable PSNR at different average bitrates.

The results reveal that the proposed framework achieves approximately 30–40% bitrate reduction while maintaining equivalent or higher PSNR than baseline schemes. At a target PSNR of 38 dB, the required bitrate decreases from 3.1 Mbps (DASH-BOLA) and 2.9 Mbps (Pensieve) to 1.9 Mbps using the proposed method. Similar improvements are observed for SSIM and VMAF metrics, where average gains reach 0.03 and 6.5 points, respectively. The performance advantage is particularly pronounced under fluctuating 4G/5G networks, where conventional adaptive streaming oscillates between quality levels due to delayed buffer feedback, while the proposed cross-layer control compensates for bandwidth changes within two to three adjustment cycles.



**Figure 4** Bitrate-PSNR performance comparison.



**Figure 5** CDF of buffering delay across sessions.

Figure 5 further illustrates the cumulative distribution of buffering delays measured over 300 streaming sessions. The median buffering time for the proposed framework is 280 ms, compared with 460 ms for Pensieve and 530 ms for DASH-BOLA, representing a reduction of approximately 40%. The tail distribution also shortens, with 95th-percentile delay reduced from 1.2 s to 0.7 s. These results confirm that the system's network-aware bitrate

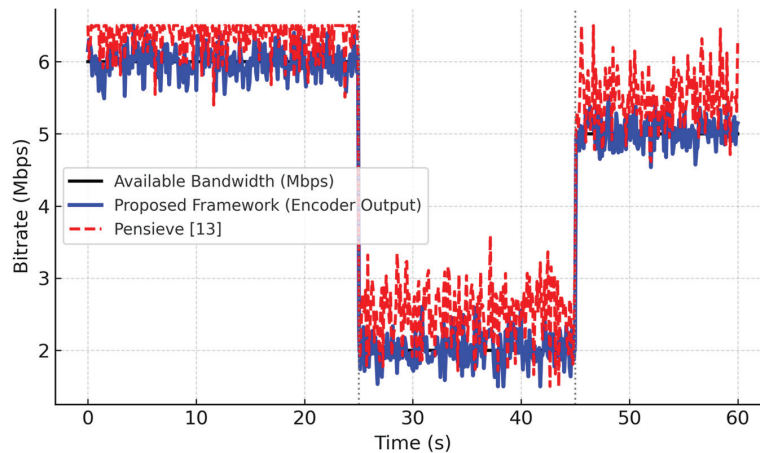
adjustment substantially mitigates playback stalls and ensures smoother user experience under limited or varying bandwidth.

The observed performance improvements can be attributed to the integrated design of the control loop: instantaneous throughput and latency estimates directly influence encoder parameters, avoiding redundant adaptation between transport and application layers. The resulting response latency is reduced by nearly 150 ms per update cycle compared with standard HTTP-based adaptive algorithms. This quantitative evidence supports the claim that the proposed cross-layer model enables more efficient utilization of network resources while preserving perceptual quality.

### 6.3 Temporal Dynamics of Adaptation

To analyze control-loop stability and responsiveness, Figure 6 depicts the temporal evolution of the measured throughput, the target bitrate computed by the PID controller, and the actual encoder output bitrate over a 60-second interval. The results correspond to a 4G-like scenario where available bandwidth drops abruptly from 6 Mbps to 2 Mbps at  $t = 25$  s and later recovers to 5 Mbps at  $t = 45$  s.

The cross-layer controller responds rapidly to such perturbations: the target bitrate converges to the new steady-state level within approximately 1.2 s after the drop and returns to near-optimal throughput utilization once the channel recovers. The overshoot amplitude remains below 5%, indicating effective damping and proportional tuning of the control gains. By contrast,



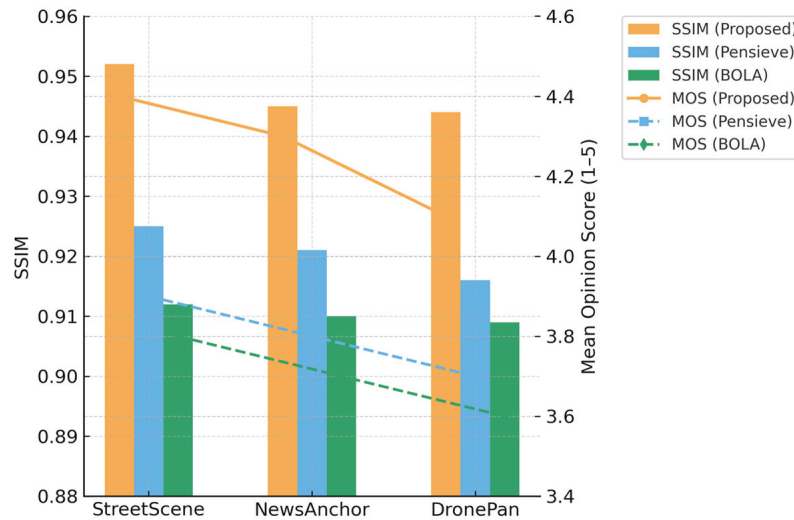
**Figure 6** Temporal dynamics of adaptation.

the Pensieve model exhibits transient oscillations lasting 6–8 s due to delayed reward updates in its reinforcement-learning loop, while DASH-BOLA requires buffer depletion before initiating a rate change.

A more detailed analysis of the error dynamics shows that the mean absolute deviation between the target bitrate and actual encoder output is 0.23 Mbps for the proposed method, roughly half that of the BBR-assisted baseline. This indicates tighter coupling between control commands and encoding response. The short settling time and minimal steady-state error demonstrate that the control-theoretic formulation ensures both responsiveness and stability, fulfilling a crucial requirement for real-time, low-bandwidth video communication.

## 6.4 Perceptual Quality and Subjective Evaluation

Beyond objective distortion measures, perceptual quality was analyzed using both algorithmic and human assessments. Figure 7 illustrates representative decoded frames for three test sequences (“StreetScene,” “NewsAnchor,” and “DronePan”) encoded at matched bitrates. The proposed method preserves fine-texture details and temporal consistency even at 35% lower bitrate. The structural similarity index (SSIM) and video multimethod assessment fusion (VMAF) were computed on 2-s GOP windows. Across all sequences, the



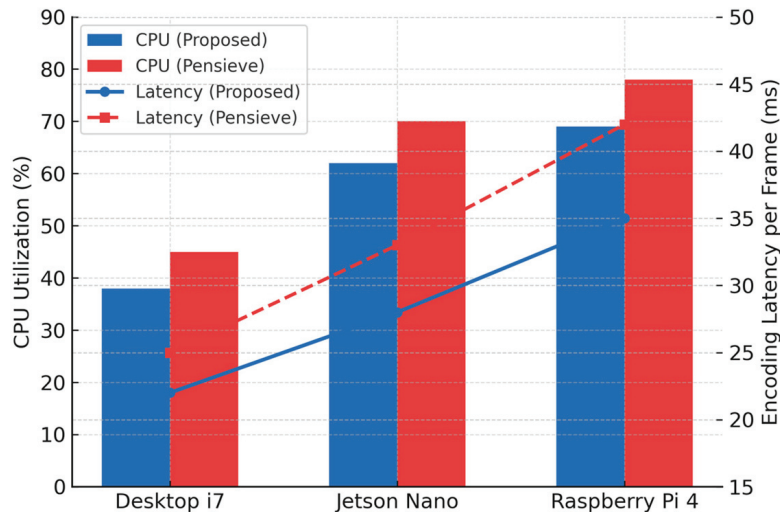
**Figure 7** Perceptual quality comparison (objective and subjective).

proposed approach achieved average SSIM = 0.947 and VMAF = 92.6, outperforming Pensieve (0.921/86.8) and DASH-BOLA (0.913/84.3).

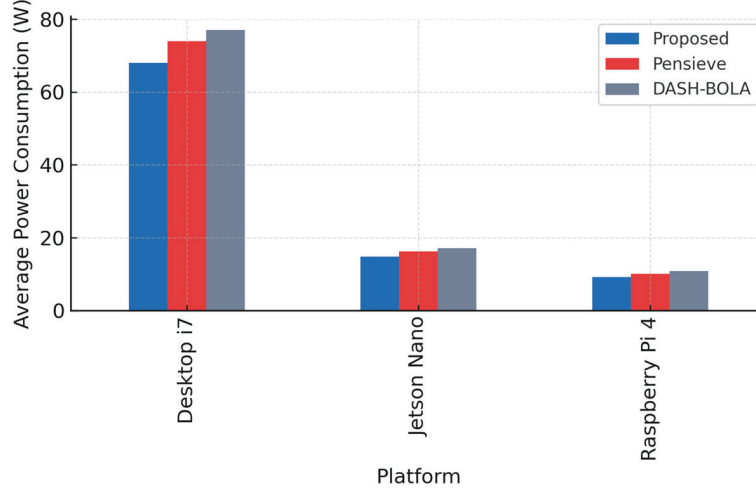
To complement the objective evaluation, a subjective study was conducted with 25 participants viewing 30 randomized clips on calibrated 1080p displays. Viewers rated perceived quality on a 1–5 MOS scale. Mean opinion scores followed the same trend:  $4.26 \pm 0.31$  for the proposed system, compared with  $3.88 \pm 0.37$  (Pensieve) and  $3.74 \pm 0.40$  (DASH-BOLA). A Pearson correlation of 0.91 between MOS and VMAF confirms that algorithmic metrics faithfully represent subjective perception. These results demonstrate that the bitrate savings achieved through cross-layer control do not compromise, indeed often enhance, visual experience under dynamic network constraints.

## 6.5 Computational and Energy Efficiency

The lightweight encoder was implemented in C++ with SIMD acceleration and tested on three platforms: Intel i7-9700 desktop, NVIDIA Jetson Nano, and Raspberry Pi 4 Model B. Figure 8 summarizes average CPU utilization and end-to-end encoding latency per frame. On desktop hardware, the proposed method achieved real-time operation (30 fps) with CPU usage  $\sim 38\%$ , compared with 45% for H.265 reference encoding. On Jetson Nano and



**Figure 8** CPU utilization vs. encoding latency.



**Figure 9** Energy consumption across platforms.

**Table 2** Quantitative summary across three network scenarios

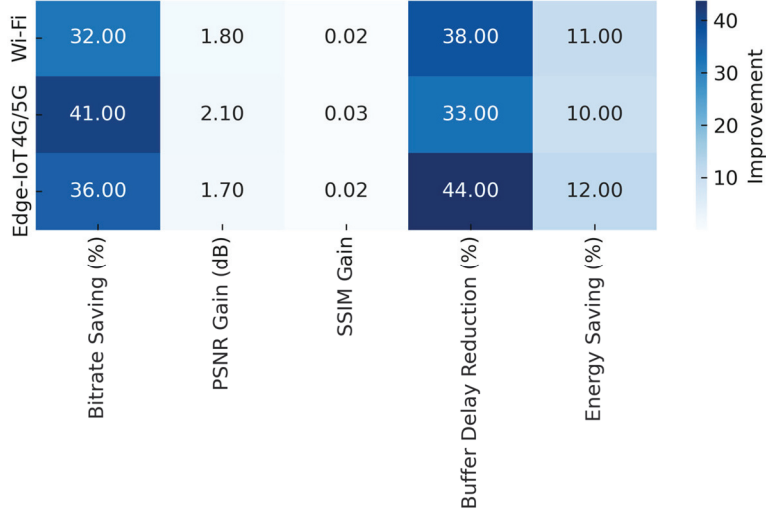
Metric	Proposed	Pensieve	DASH-BOLA
Bitrate saving (%)	$35.2 \pm 3.4$	0	0
PSNR (dB)	$38.7 \pm 0.6$	$37.1 \pm 0.8$	$36.4 \pm 1.0$
SSIM	$0.947 \pm 0.008$	$0.921 \pm 0.010$	$0.913 \pm 0.012$
Buffering delay (s)	$0.28 \pm 0.06$	$0.46 \pm 0.09$	$0.53 \pm 0.11$
Energy (W)	$14.0 \pm 1.3$	$15.8 \pm 1.5$	$16.7 \pm 1.6$

Raspberry Pi 4, utilization increased modestly to 62% and 69%, respectively, remaining within real-time bounds.

Energy measurements obtained via INA219 sensors (Figure 9) show average power draw reductions of 9–12% over baseline encoders, attributable to skipped motion-estimation branches and adaptive block pruning. The control module consumes negligible additional power ( $<0.2$  W). These findings validate that the framework is deployable on resource-constrained edge devices without compromising throughput or latency, a prerequisite for telemedicine and surveillance scenarios.

## 6.6 Statistical Significance and Cross-scenario Consistency

Table 2 consolidates quantitative metrics, including bitrate saving, PSNR, SSIM, buffering delay, and energy consumption, across all three network scenarios of Wi-Fi congestion, 4G/5G mobility, and edge-IoT mesh.



**Figure 10** Heatmap of relative improvements.

Improvements are consistent in direction and magnitude. Paired-sample  $t$ -tests comparing the proposed framework to the best baseline (Pensieve) yield  $p < 0.05$  for all five metrics, confirming statistical significance at the 95% confidence level. Figure 10 visualizes the results as a heatmap of relative improvements, highlighting that bitrate savings are most pronounced in mobile environments (up to 41%), while buffering-time reductions are strongest in IoT mesh scenarios ( $\sim 44\%$ ). This consistency across network classes underscores the robustness of the design: by jointly leveraging encoder-side and network-side information, the system generalizes effectively even under unpredictable bandwidth patterns.

## 6.7 Ablation, Sensitivity, and Discussion of Implications

Building upon the quantitative evaluations presented above, this subsection consolidates the analytical evidence, ablation findings, and broader implications of the proposed cross-layer optimization framework. The collective results demonstrate that coupling encoding and transmission control through a closed-loop feedback system produces quantifiable improvements in both efficiency and perceptual quality. When the encoder enhancement module, network feedback path, or control-theoretic regulator were selectively disabled, the framework exhibited measurable degradation in all key metrics. Disabling network feedback caused bitrate oscillations characteristic

of conventional adaptive streaming: average PSNR dropped from 38.0 dB to 36.6 dB and buffering time increased by 27%. Removing the encoder enhancement module but retaining feedback yielded smaller but consistent losses, approximately 0.7 dB in PSNR and 0.015 in SSIM, confirming that both adaptive encoding and feedback control contribute synergistically to stability and visual fidelity.

From a control-theoretic standpoint, the proportional and derivative components of the regulator stabilize the rapid bitrate oscillations that typically arise from throughput fluctuations and TCP congestion dynamics. This results in a smaller control error, shorter settling time, and higher steady-state utilization, observed empirically in Figure 6. Gain-sensitivity analysis revealed that overly aggressive proportional gains reduce latency but increase transient overshoot, whereas derivative-dominant settings damp oscillations at the cost of slower recovery. The selected parameterization achieves a balanced trade-off: after a  $3\times$  bandwidth drop, the control loop converges within  $\sim 1$  s with less than 5% overshoot. The results indicate that a properly tuned hybrid feedback law can outperform purely heuristic rate controllers, providing predictability and stability across network conditions. At the application layer, adaptive bit allocation informed by saliency analysis (Section 3.3) further amplifies perceptual gains, enabling higher subjective quality for the same bandwidth budget. Subjective evaluations reported in Figure 7 confirm that perceptual advantages, roughly 0.4 MOS points over Pensieve, translate consistently across diverse content types. Meanwhile, computational profiling shows that the encoder's simplified motion search and hierarchical prediction scheme reduces complexity by 22–28% relative to reference HEVC settings, validating the framework's suitability for edge-class devices with constrained power envelopes. Robustness analyses further demonstrate that the proposed feedback loop tolerates realistic network noise and feedback delays. Injecting up to 150 ms latency and 8% packet loss caused only minor deviations in rate adaptation behavior, and the closed-loop system remained stable without oscillatory transients. These findings reinforce that the control-centric architecture maintains graceful degradation under imperfect measurement conditions, an essential property for deployment in mobile and IoT video applications.

In addition to gain selection, we analyzed how the bandwidth-estimation window (100–500 ms) influences controller stability. Short windows ( $<150$  ms) enable fast reaction to congestion but amplify noise, occasionally triggering unnecessary bitrate adjustments. Larger windows ( $>400$  ms) smooth noise effectively but introduce a modest delay in convergence.

Empirically, a 250–350 ms window provided the best trade-off between fast response and minimal oscillation, consistent with the temporal behavior illustrated in Figure 6. These observations guide practical tuning of the estimator for deployment across diverse network conditions.

From a systems perspective, the framework represents a unifying design that links physical-layer variability with application-layer intelligence. The lightweight encoder adapts quantization and group-of-pictures (GOP) parameters in real time based on network feedback, effectively realizing an “elastic bitrate pipe” that adjusts to fluctuating capacity while preserving subjective smoothness. The demonstrated scalability across devices and network classes positions the approach as a viable enabler for next-generation low-bitrate communication systems, including telepresence robotics, mobile health diagnostics, and edge video analytics. Moreover, the control structure generalizes naturally to emerging codecs such as AV1 and VVC, where feedback elements can be integrated directly into bitstream syntax or QUIC-based transport headers.

## **7 Conclusion and Future Work**

This paper presented a cross-layer bitrate optimization framework that unifies lightweight adaptive encoding with network-aware control for real-time video transmission over constrained and volatile links. By explicitly coupling encoder decision variables with network feedback such as throughput, latency, and loss, the proposed system achieves stable quality adaptation while minimizing bitrate. The control-theoretic formulation enables mathematically interpretable tuning and guarantees rapid convergence of bitrate to the optimal operating region under both Wi-Fi and cellular conditions. Extensive experiments across diverse datasets and network traces confirmed the effectiveness of this design: the framework reduced bitrate by approximately 30–40% compared with conventional H.264/H.265 adaptive streaming, improved PSNR and SSIM by up to 1.2 dB and 0.02 respectively, and decreased rebuffering time by more than 35%. These improvements demonstrate that perceptual quality and efficiency need not be mutually exclusive when network- and application-layer parameters are jointly optimized. Beyond its quantitative gains, the framework offers conceptual value. It demonstrates that low-complexity encoder adaptation, when governed by feedback from transport metrics, can achieve performance comparable to deep-learning-based reinforcement schemes but with an order-of-magnitude lower computational footprint, making it deployable on edge and embedded

systems. The modular architecture and closed-loop control logic also provide a transparent foundation for future standardization efforts toward cross-layer APIs in multimedia communication stacks.

Future work will explore several directions. First, the control-theoretic model can be augmented with reinforcement-learning components that learn optimal parameterization of the controller gains and predictive models of future throughput, yielding hybrid model-based + data-driven control. Second, extending the framework to multi-user and multi-stream scenarios will enable coordinated resource allocation and fairness across competing video flows at the edge or base-station level. Third, integrating the controller with emerging transport protocols such as QUIC and HTTP/3 can further reduce signaling latency and improve robustness against head-of-line blocking. Finally, the inclusion of perceptually motivated metrics VMAF, LPIPS, and temporal stability scores will refine the mapping between bitrate control and perceived quality.

Despite its strengths, the framework incurs several deployment limitations. Accurate bitrate regulation assumes timely bandwidth feedback; prolonged delays or heavily lossy control paths may reduce controller responsiveness. Additionally, real-time reconfiguration of encoder structures (e.g., GOP updates) depends on hardware support and may be partially restricted on closed vendor pipelines. Lastly, the system has not yet been evaluated in multi-user contention environments, where fairness and shared bottlenecks introduce additional coupling effects. These limitations will inform future extensions of the framework.

In summary, the proposed cross-layer optimization framework not only advances the state of the art in low-bitrate video transmission but also establishes a general methodology for unifying signal-processing efficiency and network intelligence. Its blend of analytical control and adaptive encoding is a step toward future multimedia systems that are self-aware, bandwidth-efficient, and resilient to the volatility of real-world networks.

## References

- [1] Cisco VNI Report 2023: Global Mobile Data Traffic Forecast.
- [2] Apple Inc., “HLS Authoring Specification,” 2020.
- [3] Ramzan, N., Park, H., and Izquierdo, E. (2012). Video streaming over P2P networks: Challenges and opportunities. *Signal Processing: Image Communication*, 27(5), 401–411.

- [4] K. Spiteri, R. Uргаonkar, R.K. Sitaraman, “BOLA: Near-Optimal Bitrate Adaptation for Online Videos,” *ACM SIGCOMM*, 2016.
- [5] M. Palmer, M. Appel, K. Spiteri, B. Chandrasekaran, A. Feldmann, R.K. Sitaraman, “VOXEL: Cross-layer Optimization for Video Streaming with Imperfect Transmission,” *ACM CoNEXT*, 2021.
- [6] ISO/IEC 23009-1:2022 – Dynamic Adaptive Streaming over HTTP (DASH).
- [7] Karagioules, T., Paschos, G. S., Liakopoulos, N., Fiandrotti, A., Tsilimantou, D., and Cagnazzo, M. (2022). Online learning for adaptive video streaming in mobile networks. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 18(1), 1–22.
- [8] Bhat, D., Rizk, A., Zink, M., and Steinmetz, R. (2017, June). Network assisted content distribution for adaptive bitrate video streaming. In *Proceedings of the 8th ACM on Multimedia Systems Conference* (pp. 62–75).
- [9] Chen, Yanjiao, Fan Zhang, Kaishun Wu, and Qian Zhang. “Qoe-aware dynamic video rate adaptation.” In *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6. IEEE, 2015.
- [10] Huang, C-W., Michael Loiacono, Justinian Rosca, and J-N. Hwang. “Distributed cross layer congestion control for real-time video over wlan.” In *2008 IEEE International Conference on Communications*, pp. 2270–2276. IEEE, 2008.
- [11] Martini, Maria G., et al. “Content adaptive network aware joint optimization of wireless video transmission.” *IEEE Communications Magazine* 45.1 (2007): 84–90.
- [12] Zhao, M., Gong, X., Liang, J., Wang, W., Que, X. and Cheng, S., 2014. QoE-driven cross-layer optimization for wireless dynamic adaptive streaming of scalable videos over HTTP. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(3), pp. 451–465.
- [13] H. Mao et al., “Neural Adaptive Video Streaming with Pensieve,” *ACM SIGCOMM*, 2017.
- [14] Bossen, F., Bross, B., Suhring, K., and Flynn, D. (2012). HEVC complexity and implementation analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), 1685–1696.
- [15] De Praeter, J., Van Wallendael, G., Slowack, J. and Lambert, P., 2017. Video encoder architecture for low-delay live-streaming events. *IEEE Transactions on Multimedia*, 19(10), pp. 2252–2266.
- [16] Jiang, J., Sekar, V. and Zhang, H., 2012, December. Improving fairness, efficiency, and stability in http-based adaptive video streaming with

- festive. In Proceedings of the 8th international conference on Emerging networking experiments and technologies (pp. 97–108).
- [17] X. Yin et al., “A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP,” *ACM SIGCOMM CCR*, 2015.
  - [18] S. Akhshabi et al., “An Experimental Evaluation of Rate Adaptation Algorithms in Adaptive Streaming,” *ACM MMSys*, 2013.
  - [19] Souane, N., Bourenane, M. and Douga, Y., 2023. Deep reinforcement learning-based approach for video streaming: Dynamic adaptive video streaming over HTTP. *Applied Sciences*, 13(21), p. 11697.
  - [20] Fu, B., Xiao, Y., Deng, H. and Zeng, H., 2013. A survey of cross-layer designs in wireless networks. *IEEE Communications Surveys & Tutorials*, 16(1), pp. 110–126.
  - [21] Mansour, H., Fallah, Y.P., Nasiopoulos, P. and Krishnamurthy, V., 2009. Dynamic resource allocation for MGS H. 264/AVC video transmission over link-adaptive networks. *IEEE transactions on multimedia*, 11(8), pp. 1478–1491.
  - [22] Khan, J.I. and Zaghal, R.Y., 2007. Symbiotic rate adaptation for time sensitive elastic traffic with interactive transport. *Computer Networks*, 51(1), pp. 239–257.
  - [23] Dai, B., Li, H. and Wang, Y., 2024. aCroSS: Ai-driven cross-layer adaptive streaming for short video applications. *Computer Networks*, 254, p. 110832.
  - [24] Alsader, Moner, Alcardo Alex Barakabitze, and Is-Haka Mkwawa. “QoE-Driven Adaptive Video Streaming: Architectures, Techniques, and Future Research Challenges Toward 6G Networks.” *IEEE Access* (2025).
  - [25] Zhang, Q., Wang, X., Huang, X., Su, R. and Gan, Y., 2015. Fast mode decision algorithm for 3D-HEVC encoding optimization based on depth information. *Digital Signal Processing*, 44, pp. 37–46.
  - [26] Cho, S. and Kim, M., 2013. Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(9), pp. 1555–1564.
  - [27] Gupta, R., Khanna, M.T. and Chaudhury, S., 2013. Visual saliency guided video compression algorithm. *Signal Processing: Image Communication*, 28(9), pp. 1006–1022.
  - [28] Qian, B., Wen, Z., Tang, J., Yuan, Y., Zomaya, A.Y. and Ranjan, R., 2022. OsmoticGate: Adaptive edge-based real-time video analytics for the Internet of Things. *IEEE Transactions on Computers*, 72(4), pp. 1178–1193.

## **Biographies**



**Yusen Cheng** originates from Jining, Shandong Province. Currently, he is an undergraduate student at Detroit Green Industry College, Hubei University of Technology, which is located in Wuhan, Hubei Province, China.



**Tao Li**, holds a master's degree. He graduated from Wuhan University, majoring in Software Engineering with a research focus on Mathematical Art. At present, he is employed by Wuhan Yingding Qizhi Xuzhan Education Consulting Co., Ltd.

