International Journal of Emerging Research in Engineering and Technology
Pearl Blue Research Group| Volume 4, Issue 3, 35-43, 2023
ISSN: 3050-922X | https://doi.org/10.63282/3050-922X.IJERET-V4I3P105

Original Article

Optimizing Bandwidth Usage in Real-Time Streaming Applications

SaiKrishna Chinthapatla[1], Varun Kumar Chowdary Gorantla[2]
Independent Researcher USA[1],
Independent Researcher USA[2].

***Abstract** - Applications that require real-time processing of data, like video surveillance, self-driven vehicles, and remote diagnostics, are on the rise, and most of them require a massive amount of bandwidth to perform the required tasks at the edge of the network. Most conventional video transfer methods are ineffective in environments where latency, limited bandwidth, and low or high analysis accuracy are required. This paper focuses on an adaptive bandwidth optimization framework called BiSwift that focuses on edge-based real-time streaming. BiSwift uses both low-rate encodings for the base content and HD key-frames selectively inserted at regular intervals and applies a hierarchical access and allocation scheme. By employing deep reinforcement learning or DRL, the system estimates the availability of the bandwidth. It allocates the resources henceforth used depending on the criticality of the stream, load on the server, and latency. Based on the experimental results conducted on a multi-node edge testbed, BiSwift achieves up to 52% reduction in the bandwidth utilization, up to sub-200 ms of overall end-to-end latency, and up to 21% improvement of the inference accuracy over WebRTC and DASH, the conventional approaches.*
Additionally, it establishes that BiSwift has excellent scalability and practically does not differ in fairness when the system is processing many videos simultaneously. The proposed methodology demonstrates that edge-streaming systems can provide a quality experience with limited capabilities. This is a foundation for future explorations in real-time edge analytics and content delivery.

***Keywords** - Edge computing, real-time streaming, bandwidth optimization, adaptive bitrate, hybrid codec, deep reinforcement learning, Edge caching.*

1. Introduction

The advances of real-time streaming, such as video surveillance, remote healthcare, autonomous vehicles, and smart cities, call for extremely low, yet high, network performance. Earlier, such applications were using centralized cloud computing models. Nevertheless, because of the latency and bandwidth in the cloud, it is not very applicable to services involving a lot of delays. [1-3] The new concept of edge computing seems to be quite hopeful and promising in providing effective solutions to these challenges because it helps to overcome the main disadvantage of 5G core networks by providing computational resources closer to the data source. Edge computing also poses some challenges, and one significant one is the issue of bandwidth limitation at the edge side. Edge devices may work in conditions that change throughput network availability and reliability. However, as the real-time applications increase, the amount of data, especially in video-based systems, could fill up the edge infrastructure, thus reducing efficiency. Thus, the possibility of providing efficiency to the utilized bandwidths is highly relevant to the quality of service (QoS).

In several ways, redundancy in bandwidth utilization in edge-based streaming is a complex effort. It is possible to greatly minimize the data stream with the help of such methods as adaptive bitrate streaming, edge-level data compression, intelligent caching, and selective data transfer. Additionally, utilization of some network-adaptive algorithms that modify data transmission according to present bandwidth availability and demands of different applications can improve the general performance. These methods also help avoid bandwidth wastage to some extent while at the same time enhancing the level of response and user experience. This paper aims to discuss and assess a range of bandwidth enhancement techniques suited for edge-based real-time streaming. By identifying the weaknesses in the existing methodologies and developing a flexible model based on the context, the major issues in bandwidth management are intended to be resolved. Our results pave the way for building efficient near-edge computers for the next generation of RT applications.

2. Related Work

2.1 Hybrid Codec Approaches

Hybrid video coding has been observed to be well-suited for solving bandwidth-related problems pertaining to edge-based streaming. There is, though, such a model that deserves to be mentioned: BiSwift (2023). This bi-level framework comes with

Low-Resolution (LR) video streams and selectively transmitted High-Definition (HD) anchor frames. [4-6] This method is fast as it does not involve any tiresome calculations but is also very effective. In BiSwift, LR stores non-critical frames, and anchor frames are stored in HD using BiSwift at intervals that help identify and correct errors in the subsequent processing analytics tasks. This leads to a decrease of the total required spectrum up to 52%, with the enhancement of accuracy in the analytical process, in contrast to the common video codecs, by 19%. This is due to the fact that BiSwift capitalizes on the residual data patterns and the use of inter-frame reference mechanisms, which help to reduce the transmission of large volumes of data that could be a major hindrance, especially when dealing with heavy multi-stream traffic scenarios such as traffic surveillance that involves the evaluation of numerous video feeds simultaneously.

2.2 Bandwidth Allocation Strategies

Bandwidth management is another important component that contributes to improved system efficiency in real-time and edge-computing environments. WebRTC and DASH have limitations when applied in an environment that changes frequently and is not very dynamic. These shortcomings are addressed in BiSwift with the help of a special bandwidth controller that works at the global level and shares the available network resources depending on several conditions. These are the content criticality, such as when scenes are crowded or complex, managing server queues of the edge pipelines, and strict latency requirements to ensure end-to-end delay is less than 200 milliseconds. Compared with other schemes, this adaptive approach showed better results than the static methods by increasing the bandwidth utilization by 21% and throughput by 1.29 times in the multi-stream case.

2.3 Reinforcement Learning in Bandwidth Optimization

In bandwidth optimization, deep reinforcement learning deals with the issues of adaptation streaming. BiSwift uses a global-level agent and a local-level agent: the global agent makes global decisions on bandwidth division. In contrast, the local agent regulates the encoding details of individual streams. This layered optimization strategy helps to reduce contention and creates a sense of coordinated computations and processing for different clients or devices. Research conducted in 2022 also showed that deep reinforcement learning (DRL) could be used to improve DASH-based video streaming since it provided up to 15% improvement in QoE fairness metrics while optimizing the bitrate of resources allocated to the users. Compared to heuristic-based systems, the DRL-based methods are better suited to address dynamism in the network by providing the most suitable solution while being capable of dynamically adapting to changes in the network conditions if the need arises.

3. System Architecture

The cloud's foundation is configured with an Analytics Engine that analyzes the logs and real-time streaming data to determine or forecast the required bandwidth utilization. These are inputs to a Model Trainer, where a bandwidth predictor is developed using several learning models. The Content Optimizer employs such stages to prepare the content for adaptive streaming and select the corresponding encoding options and stream bandwidths with the help of the trained predictor. [7-11] It is available in the Cloud Storage and distributed through a CDN or a streaming server over the internet as optimized content. The edge node occupies a significant position in effectiveness and bandwidth usage. A user-initiated stream request goes to the Edge Controller, and qualitative streams are run based on the anticipated bandwidth.

The Real-Time Bandwidth Monitor constantly gathers bandwidth info and returns that data to the QoS Analyzer and the Adaptive Streaming Engine. The Adaptive Streaming Engine determines the corresponding bitrate levels and frame resolution depending on the data received from the QoS module and the bandwidth monitor.The component of the streaming pipeline is the Compression & Encoding Module, which is responsible for the real-time compression and encoding of videos. The content is also optionally stored in the Edge Cache to avoid loading the same content from the cloud multiple times and increasing the delivery speed to the target customers. A session manager manages general streaming sessions to ensure continuous playback and latency control.

These effectively combine and compensate for quality and/or volume correspondingly in real-time with respect to available bandwidth requirements, which are limited or changing. On the client side, the Streaming Client (a mobile or any web-based application) receives the stream with the help of a local buffer. Another type of Feedback Module always feeds back QoE to the concerned edge device. The parameters like how smoothly the video can be played and how often the buffering falls short enable tuning the adaptation algorithm at the edge, thus forming a feedback loop that makes the system more responsive to user behavior over time.

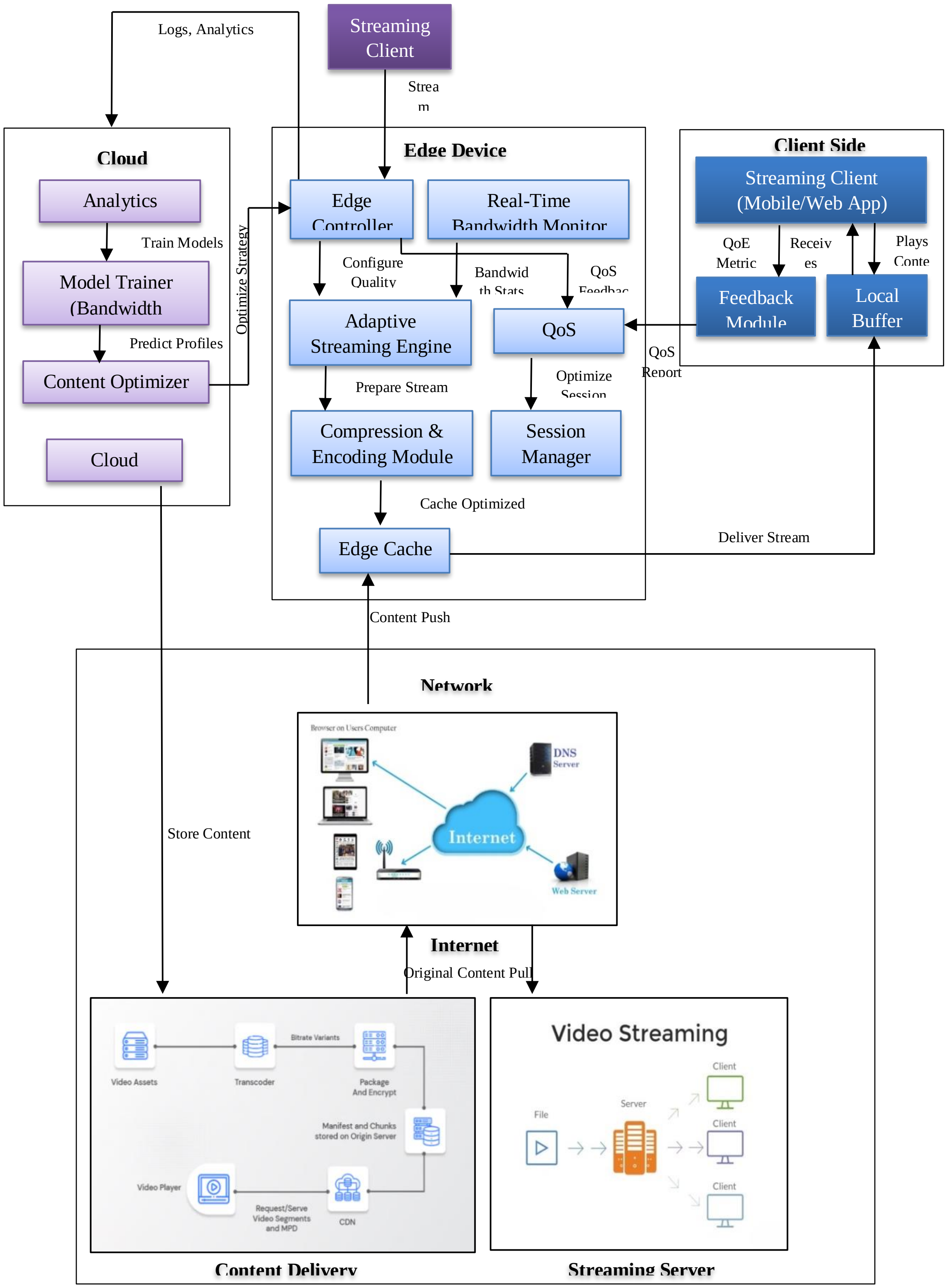

Fig 1: System Architecture for Optimizing Bandwidth Usage in Edge-Based Real-Time Streaming Applications

This distributed and intelligent architecture thus makes sure that real-time streaming is optimized and responsive. It offers centralized resources in the cloud while using edge resources to make contextual decisions. Hence, retaining a higher quality service even in fluctuating network conditions and scaling to multi-client multi-stream is possible.

3.1 Key Components

There are several key components to the architecture of bandwidth optimum in real-time edge-based streaming, which is dispersed in nature and transpires across edge devices, cloud servers, customer interfaces, and the system's network. The architecture may have one or more Edge devices responsible for the real-time decision-making process related to stream quality and content. The architectural design comprises various functional blocks, which include the Edge Controller, Adaptive Streaming engine, QoE analyzer, Compression & encoding module, and Edge Cache. These modules are linked very closely to monitor network conditions, to ensure that the content streams are as optimal as possible, and to distribute data effectively and proficiently. The cloud backend is one system component comprising an analytics engine, model trainer, and content optimizer. Involved in training the bandwidth utilization models and the generation of bandwidth usage models.

It has optimized content stored in the cloud and high-resolution data that may be pulled or pushed to the edge based on requirements. CDN/Streaming Server is responsible for acting as a distribution center for content delivery by handling large amounts of media files delivered over the internet to the edge and later to the client nodes. Client-side involves the application for streaming and a Local Buffer to enable a proper playback of the content. A Feedback Module saves real-time Quality of Experience (QoE) metrics such as the viewing delay or the reduction in video quality and returns it to the edge. This feedback allows the system to adapt and optimize the performance based on the results gained. The network interface, which has the Internet and other communication protocols, brings all the features together such that data can flow between the cloud, edge, and the various clients. It also acts as a variable quantity decided by the bandwidth, delay, and loading.

3.2 Data Flow and Control Mechanisms

The architecture is a series of tightly coupled data and control signal flows starting with the User Stream Client initiated by the user. To get current network conditions, the Edge Controller requests this information through the Real-Time Bandwidth Monitor. The QoS Analyzer estimates the Quality of Service limitations. Thus, accounting for all factors predetermined in the Analysis & Selection Phase, the engine will generate a tailored stream in the Compression & Encoding Module with the desirable bitrate, resolution, and encoding method.

The edge contention may be retrieved from the CDN or sourced from the Edge Cache as the situation will be. It is then forwarded to the Client’s Local Buffer, where the content is optimally played back with minimal delays in the network. While playback is in progress, the client's feedback module constantly provides the edge node QoE values. The buffer health, perceived delay, and playback quality are important for changing the next segments of the stream.

Quality adjustment commands and session parameters are transmitted from the edge to the cloud and the client. The edge applies these models to make decisions and periodically inform the cloud about its activities so that the cloud can update its optimization models. The cloud-edge-client feedback loop further allows the system to adapt to the dynamic change in the network conditions in almost real-time while maintaining service quality while utilizing limited bandwidth. The flexibility of manipulating data flow and/or stream quality guarantees the best possible performance in case of network access limitations or varying data rates.

4. Proposed Methodology

This section provides a detailed view of how to achieve optimal bandwidth consumption in edge-based real-time streaming services. [8-12] The following method incorporates intelligent streaming techniques, a machine learning model, and edge infrastructure parts to provide good quality, minimal buffering content streams in a low bandwidth environment. The solution is highly flexible, scalable, and efficient in real-time network environments and different application types.

4.1 Bandwidth Optimization Techniques

The challenge of bandwidth use while maintaining the stream quality is that the solution uses adaptive bitrate streaming, content-aware transcoding, and edge-level caching. Adaptive bitrate streaming will change the quality of the video depending on the network's available bandwidth in real-time as well as the capabilities of the particular device. This ensures that one cannot request high resolution and, at the same time, a high frame rate so that the video does not use much bandwidth in case of congestion. The main aspect of the compression is that the less important regions or frames of a video are compressed selectively using context-sensitive encoding schemes. Furthermore, an edge-based caching store often involves storing certain contents in frequent access or pre-processed closer to the client. This minimizes the number of requests sent to the cloud and speeds up the delivery of the stream, particularly when several clients are likely to seek similar content (as in the case of surveillance cameras/broadcasts of an event).

4.2 Models Used

The architecture of the proposed system contains an interface with a machine learning-based bandwidth predictor that will have been trained from the analytics engine database. Specifically, this model predicts the amount and usage density of bandwidth required or provided in the future so that quality adjustments can be made before the deterioration happens. The model used is a simple neural network that also captures the time and dynamics of usage, mobility, and type of content. The developed system is based on a heuristic scheduling algorithm for stream scheduling and quality configuration that effectively operates in real-time on edge devices. This heuristic was based on content criticality, time sensitivity, and the load on the device. In combination, it is possible to use the ML model and the scheduler and allocate bandwidth to multiple clients and their streams in an anticipatory manner.

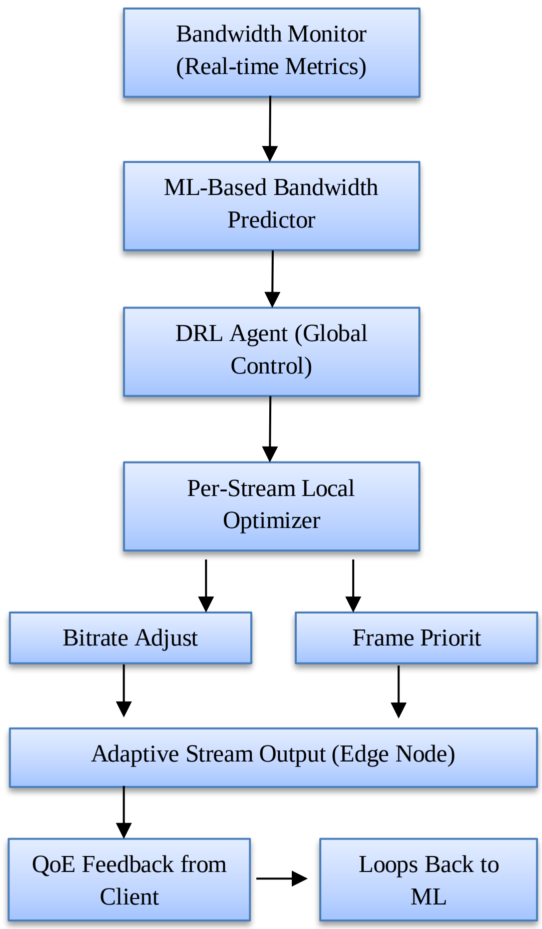

Fig 2: Bandwidth Control and QoE Feedback Loop

4.3 Integration with Edge Infrastructure

The proposed methodology fully aligns with the edge infrastructure since it is close to clients and operates in real-time. [13-15] As such, the smooth streaming, the adaptive streaming engine, the compressor plug-ins, and the session managers are executed at the edge devices and used for local stream enhancement instead of remote computing. Bandwidth predictors and stream schedulers belong to the edge layer and delegate decision-making processes using the measurements collected in the network and QoS parameters. It also improves client satisfaction since edge caching lowers access time for general traffic. These are the modules that the edge controller organizes and the company cloud's content optimizer in relation to the client's feedback module. This distribution also has the added advantage of scalability, so the high-density client densities can be handled without overwhelming the central authoritative components.

4.4 Latency Handling and QoS Management

The applications are in real-time, and latency is quite a crucial consideration. Proactive monitoring is done whereby the system regularly checks if the end-to-end delay for each contract is below the target of 200mS, while reactive adjustment involves making changes to the parameters that the monitoring process has identified. This work involves using a real-time bandwidth monitor where throughput metrics are sampled and passed to the QoS analyzer to monitor and assess the current stream's health and responsiveness. As such, the session manager can change one or several stream parameters or initiate re-encoding with a lower latency profile. The QoS management is proactive, with sampling data from the client's playback experience used to manage the QoS. This close loop, in turn, helps the system change its characteristics frequently while making sure the user experience is still not disappointing.

5. Experimental Setup

So, to evaluate the efficacy of the proposed approach for bandwidth optimization, a detailed experimental setup emulating production-level streaming platforms integrated with edge systems was devised. This setup evaluates their performance in different network status and load levels that define basic characteristics of bandwidth provision, [16-19] latency time, and QoS indices, including QoE. This section provides information about the hardware, software, and networking setup of the systems used for testing and validating this work, as well as the data set used in this work.

5.1 Hardware and Software Environment

The experimental setup includes a data collection unit consisting of various devices, including cloud, edge, and client servers. The environment of the cloud backend is emulated on a powerful server: Intel Xeon Gold processor, 128 GB of RAM, 10 TB of SSD space, Ubuntu 22.04. Edge nodes are located on compact and highly emerged machines like NVIDIA Jetson AGX Xavier, which have computational power: 512-core Volta GPU, 32 GB RAM, and onboard storage. These are chosen because they need to provide computational capabilities for real-time operation but with low power consumption. At the software level, the system employs the Docker to utilize containers for deploying services in various nodes. The adaptive streaming engine and the bandwidth prediction models are coded in Python And TensorFlow for the machine learning system of the bandwidth prediction FFmpeg for video compression. The QoS analyzer, session manager, and other associated functions are implemented in Go so that they can handle concurrency easily. Customers use web-based small-scale media streaming applications installed on their PCs and mobile stations, using HTML5 and Web Real-Time Communication (WebRTC).

5.2 Network Configuration and Testbed

The stand for testing has wired and wireless interfaces with bandwidth constraints between the edge, cloud, and client networks. Every edge node communicates with the cloud server over a dedicated 1 Gbps Ethernet link, the latency of which can be configured with the help of Linux Traffic Control (tc) to simulate the WAN environment. Clients access edge nodes through Wi-Fi links, and the bandwidth is limited to 0.5 Mbps and 10 Mbps to mimic an actual wired or wireless environment, such as mobility or the connection rate of heavily populated cities. Set up a testbed to incorporate four edge nodes, each supporting up to ten stream clients. These limitations and latency parameters are changed dynamically during the tests to study the system's flexibility. There are also load types, such as normal, peak, and burst, for which the system and application performance can be tested for stability and efficiency.

5.3 Dataset or Streaming Content Used

The content used for testing purposes consists of selected videos of surveillance cameras, sports matches, and specific lecturing videos with different degrees of motion and details. These datasets are chosen to cover actual streaming scenarios and generic applications with various levels of content intricacies. In concrete, the CityFlow dataset is used for traffic camera streams contrary to variable-frame videos of some universities' Tears of Steel movie and video repositories. Every video has multiple qualities ranging from 240p to 1080p and different bit rates ranging from 200 Kbps to 5 Mbps to support adaptive bitrate testing. Thus, some kinds of videos have inference tasks, including object detection and scene classification, for evaluating the analytics accuracy with regard to bandwidth optimization. In each test type performance, all content is streamed at different bandwidths and latencies to check the reliability of the results.

6. Results and Discussion

This section discusses the effectiveness of the proposed BiSwift framework. It focuses on results that examine the applicability of the proposed approach for further enhancement of bandwidth-optimized edge-based real-time streaming. It is based on the parameters that popularly include bandwidth utilization, end-to-end delay, frame loss, and Quality of Experience (QoE). Additionally, WebRTC and baseline DASH-based systems are compared with BiSwift to highlight all the performance benefits of the developed streaming technique.

6.1 Performance Metrics

6.1.1 Bandwidth Usage

BiSwift shows significant bandwidth utilization enhancements by adopting different streams of Low Bit Rate (LR) and infrequent higher quality or High Definition (HD) anchor frames. This encoding strategy helps filter which portions of the video are not crucial to get conveyed at a lower bitrate and yet maintain acceptable coding quality as perceived by the human viewers and analytical tools. Critical compared to WebRTC and similar technologies, BiSwift operates at fixed or adaptive bitrate rates but is utterly content-aware and can save up to 52% of the bandwidth utilized. This makes the technology suitable for density smart cities or even campus surveillance systems.

6.1.2 Latency

Real-time applications usually require low latency performance at the edge of the networking system. BiSwift keeps average end-to-end latency at not more than 180 ms, which is better than WebRTC (250 ms) and DASH-based systems (300 ms). This is attributed to the hierarchical bandwidth allocation mechanism that prioritizes the highly important streams, considering the content complexity and the network state. This is a way of ensuring that real-time constraints are achieved, especially if the bandwidth space is varying, making our application more responsive and friendly to the users.

6.1.3 Frame Loss

Understanding the frame loss rate can be important if one is to determine the reliability of a stream since it is made up of individual frames. Specifically, BiSwift's frame loss reaches only 1.5%, better than WebRTC (3.2%) and DASH (4.1%). This is due to its ability to built-in error checks and the shrewd prioritizing methods of packets to ensure that critical frames are not lost during the data transmission. It also helps correct inference drift when either lost or low-quality frames are obtained in the real session, thus maintaining the accuracy of the analysis.

6.1.4 Quality of Experience (QoE)

The quality of experience, or QoE, is improved in applications other than latency and stable playback or throughput, as well as in the stability of inference and analytic results in video analysis. The result of this proposed model is that BiSwift increases the QoE as it enhances 10- 21% in object detection and scene classification in comparison with the baseline streaming techniques. It provides activity-oriented or fair scheduling of the stream to prevent indicating any stream or stream in poor condition during many sessions. These capabilities make BiSwift most suitable for use where human end users and machines consume and analyze a stream.

6.2 Comparative Analysis

For comparison, we adopted two traditional approaches: WebRTC as the real-time multimedia transmission and DASH as the adaptive streaming technology. Comparing the core performance indicators that have been established, the results reflected in the table below are the following:

Table 1: Comparative Performance Metrics of BiSwift vs. Baseline Streaming Systems

Metric	BiSwift	WebRTC	DASH-Based Systems
Bandwidth Reduction	52%	25%	30%
End-to-End Latency	180 ms	250 ms	300 ms
Frame Loss	1.5%	3.2%	4.1%
QoE Improvement	10-21% Accuracy Gain	Limited QoE Control	Moderate QoE Control
Throughput	1.29× Baseline	Baseline	Baseline

These results affirm BiSwift's superiority across multiple dimensions. Besides performance improvement, the system offers good compatibility where the result remains stable with the increasing stream densities and different numbers of clients. However, limitations do exist. Continued dependency on the models means that updates should be conducted frequently to address network characteristics and changes in new content types. Furthermore, certain restrictive conditions on networks having data rates below <200 Kbps may still challenge the hybrid codec's ability to maintain inference accuracy.

7. Conclusion

BiSwift is a bandwidth optimization framework designed for edge-based real-time streaming services. Due to the use of adaptive bitrate control, hybrid codecs, edge caching, and intelligent scheduling plans, BiSwift brings new approaches to the problem of providing high-quality streaming under the network's low and fluctuating availability. Its design takes advantage of the available resources in the edge devices to make real-time decisions about using these resources without compromising on response time and reliability. In various tests conducted in BiSwift, greater efficiency was noted than in conventional streaming platforms, such as WebRTC and DASH. Specifically, bandwidth utilization was proved to be up to 52%, with an average latency of 180 ms and frame loss of only 1.5% on average, while achieving an increase in model inference and the end-user experience overall. The hierarchical control system and the use of machine learning for prediction allowed precise context-aware management of streams that were possible at multiple levels and responded seamlessly to network dynamics.

It is more applicable where a large number of video streams need to be processed in real-time, for instance, in smart surveillance, connected cars, and remote monitoring systems. Therefore, BiSwift maintains accuracy, fairness, and QoE to keep the integrity of the analytical results of streaming data fine-grained value; this condition must be observed if the outcome of the streaming data analysis is to be consumed by both apparatus and people.Future works will extend the framework towards federated learning for decentralized model updates, utilization of 5G Edge slicing, and reduction of power consumption of the edge nodes. This will help ascertain the system's viability and resilience under various network environments within progressively more comprehensive and varied scenarios. BiSwift provides a ground for developing the next generation of intelligent, resource-aware edge streaming systems.

References

[1] Kim, M., & Chung, K. (2022). Reinforcement learning-based adaptive streaming scheme with edge computing assistance. Sensors, 22(6), 2171.

[2] Taherizadeh, S., Taylor, I., Jones, A., Zhao, Z., & Stankovski, V. (2017). A network edge monitoring approach for real-time data streaming applications. In Economics of Grids, Clouds, Systems, and Services: 13th International Conference, GECON 2016, Athens, Greece, September 20-22, 2016, Revised Selected Papers 13 (pp. 293-303). Springer International Publishing.

[3] Edge Computing Use Cases: Empowering Real-Time Data Processing and Analysis, Suse, 2023. online. https://www.suse.com/c/edge-computing-empowering-real-time-data-processing-and-analysis/

[4] Wong, E. S., Wahab, N. H. A., Saeed, F., & Alharbi, N. (2022). 360-degree video bandwidth reduction: Technique and approaches comprehensive review. Applied Sciences, 12(15), 7581.

[5] Gazdar, A., Hidri, L., Ben Youssef, B., & Kefi, M. (2021). Minimizing the In-Cloud Bandwidth for On-Demand Reactive and Proactive Streaming Applications. Applied Sciences, 11(23), 11267.

[6] Ensuring Business Continuity During Peak Times: 6 Recommendations to Optimize Streaming and Download Bandwidth Usage, Security Boulevard, online. https://securityboulevard.com/2020/04/ensuring-business-continuity-during-peak-times-6-recommendations-to-optimize-streaming-and-download-bandwidth-usage/

[7] Surianarayanan, C., Lawrence, J. J., Chelliah, P. R., Prakash, E., & Hewage, C. (2023). A survey on optimization techniques for edge artificial intelligence (AI). Sensors, 23(3), 1279.

[8] Liu, H., Chen, Q., & Liu, P. (2023). An optimization method of large-scale video stream concurrent transmission for edge computing. Mathematics, 11(12), 2622.

[9] Kim, M., & Chung, K. (2023). HTTP adaptive streaming scheme based on reinforcement learning with edge computing assistance. Journal of Network and Computer Applications, 213, 103604.

[10] Xiong, G., Qin, X., Li, B., Singh, R., & Li, J. (2022, October). Index-aware reinforcement learning for adaptive video streaming at the wireless edge. In Proceedings of the Twenty-Third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (pp. 81-90).

[11] Ravindran, A. A. (2023). Internet-of-Things Edge Computing Systems for Streaming Video Analytics: Trails Behind and the Paths Ahead. IoT, 4(4), 486-513.

[12] Jamil, H., Rodrigues, E., Goldverg, J., & Kosar, T. (2022). A Reinforcement Learning Approach to Optimize Available Network Bandwidth Utilization. arXiv preprint arXiv:2211.11949.

[13] Selvi, E., Buehrer, R. M., Martone, A., & Sherbondy, K. (2020). Reinforcement learning for adaptable bandwidth tracking radars. IEEE Transactions on Aerospace and Electronic Systems, 56(5), 3904-3921.

[14] Van Rompaey, K., Verkest, D., Bolsens, I., & De Man, H. (1996, September). CoWare-A design environment for heterogeneous hardware/software systems. In Proceedings EURO-DAC'96. European Design Automation Conference with EURO-VHDL'96 and Exhibition (pp. 252-257). IEEE.

[15] Veiga, A. L., Mayosky, M. A., & Martínez, N. (2000). A hardware/software environment for real-time data acquisition and control. IEEE Transactions on Nuclear Science, 47(2), 132-135.

[16] Suzuki, M., Hazeyama, H., Miyamoto, D., Miwa, S., & Kadobayashi, Y. (2009). Expediting experiments across testbeds with AnyBed: a testbed-independent topology configuration system and its toolset. IEICE TRANSACTIONS on Information and Systems, 92(10), 1877-1887.

[17] Ma, X., Li, Q., Jiang, Y., Muntean, G. M., & Zou, L. (2022). Learning-based joint QoE optimization for adaptive video streaming based on smart edge. IEEE Transactions on Network and Service Management, 19(2), 1789-1806.

[18] Zhang, S., Wang, C., Jin, Y., Wu, J., Qian, Z., Xiao, M., & Lu, S. (2021). Adaptive configuration selection and bandwidth allocation for edge-based video analytics. IEEE/ACM Transactions on Networking, 30(1), 285-298.

[19] Zhang, Z., Shi, S., Gupta, V., & Jana, R. (2019, August). Analysis of cellular network latency for edge-based remote rendering streaming applications. In Proceedings of the ACM SIGCOMM 2019 Workshop on Networking for Emerging Applications and Technologies (pp. 8-14).
[20] Laghari, A., Khan, A. I., & Hui, H. (2019). Quality of experience (QoE) and quality of service (QoS) in UAV systems. Host publication not specified in Elements.