

e-ISSN: 3048-4766

Volume 08 Issue 03 Sep-
Dec 2026

*Corresponding Author:

V. U. Ene,

*P.G Student, Department of
Electrical and Electronic
Engineering, Rivers State
University, Port Harcourt,
Nigeria*

Submission Date: Apr 07,
2026

Copyright Received Date:
Apr 13, 2026

Minimization of Zapping Delay in Internet Protocol Television (IPTV) Networks using Predictive Burst-Assisted Channel Switching Scheme

V. U. Ene^{1*}, B. I. Bakare², P. Elechi³

*¹P.G Student, Department of Electrical and Electronic
Engineering, Rivers State University, Port Harcourt, Nigeria*

*²Senior Lecturer, Department of Electrical and Electronic
Engineering, Rivers State University, Port Harcourt, Nigeria*

*³Reader, Department of Electrical and Electronic
Engineering, Rivers State University, Port Harcourt, Nigeria*

ABSTRACT

One of the significant factors affecting customer satisfaction with Internet Protocol Television (IPTV) systems is zapping delay when changing to another channel. This paper describes a new framework, called the Predictive Burst-Assisted Channel Switching (PBACS) framework, intended to reduce the delay associated with changing channels. The PBACS framework combines the predictive ability of machine learning to predict user viewing behavior and multicast burst streaming capability to pre-load probable next-viewed channels prior to changing to the new channel. Using simulation tools such as MATLAB/Simulink, results showed that the PBACS framework reduces zapping delay time from a base of 150-210 milliseconds to 90-130 milliseconds; an overall average improvement of 36.7%, exceeding improvements made by the hybrid method by 11.4%. PBACS improved the best-case zapping delay from 150 to 90 milliseconds (40%) and the worst-case zapping delay from 210 to 130 milliseconds (38.1%). Additionally, resource costs declined by 34.2% and prediction accuracy improved to 85-95%, an improvement of 13.3% when compared to the baseline. Moreover, according to results, improvements in bandwidth efficiency (14.1%), Quality of Experience (QoE) (26.7%), system utility (27.5%), and user satisfaction (21.8%) also were achieved. Overall, findings indicate PBACS provides an efficient, scalable solution for providing live and on-demand IPTV services. Findings also indicate significant improvements in responsiveness of IPTV and improved user experience.

Keywords:- *Zapping Delay, Internet Protocol Television (IPTV), Predictive Burst-Assisted Channel Switching (PBACS), Machine Learning, Multicast Burst Streaming*

INTRODUCTION

Internet Protocol Television (IPTV) has emerged as a dominant multimedia delivery platform due to its flexibility, interactivity, and personalization capabilities. Despite these advantages, viewer experience is often degraded by zapping delay, the latency between a channel change request and the appearance of new video content [1]. Excessive switching delay is strongly correlated with subscriber dissatisfaction, service abandonment, and network inefficiency [2].

Zapping delay originates from several sources, including video decoding time, buffer filling operations, signaling overhead, and network propagation delays. Traditional IPTV systems rely on either unicast or multicast transmission, both of which exhibit trade-offs between bandwidth efficiency and responsiveness. Multicast approaches reduce network load but incur longer startup times, while unicast delivery supports fast switching at the cost of high bandwidth consumption [3].

Numerous mitigation techniques have been explored in the literature. [4] proposed adaptive hybrid unicast-multicast schemes that reduced zap time but depended heavily on channel popularity distributions. [5] introduced behavior-based learning models that improved performance for frequent viewers but performed poorly for irregular users. [6] developed demand-aware streaming for community networks, yet focused mainly on on-demand services and did not consider live IPTV scenarios.

Other studies have explored cooperative P2P delivery [7] and optical access network optimization [2].

These works collectively reveal three unresolved gaps: insufficient integration between subscriber-level prediction and network level optimization, inefficient trade-offs between multicast and unicast transmission, and limited support for both live streaming and personalized Quality of Experience (QoE). The work of [6] is primarily focused on demand-driven streaming for communities that provide on-demand services and did not include any IPTV live-streaming services. Other research has focused on cooperative Peer-to-Peer (P2P) content delivery [7] or on network optimization in Optical Access Networks [2].

As a whole, these studies have identified three areas of research that are currently unaddressed: (1) insufficient integration of subscriber-level predictions with Network Level Optimizations; (2) poor trade-offs between multicast and unicast transmissions; and (3) limited support for both live video streaming and quality of experience (QoE).

This paper provides solutions for these issues through a new Predictive Burst-Assisted Channel Switching (PBACS) approach. PBACS will use machine learning to predict a user's next channel, and will employ adaptive multicast burst streaming to allow for the proactive delivery of critical segments of video content before a channel switch. PBACS combines efforts to optimize both the

accuracy of predictions with the efficiency of transmission so as to minimize zapping (time delay) while maintaining the requisite scalability.

MATERIALS AND METHODS

Materials

The following are the materials used to analyze and set up the PBACS framework

- i. Personal Computer
- ii. Real IPTV Traffic Dataset
- iii. IPTV Network Simulator
- iv. MATLAB/Simulink
- v. Video Streaming Content

Methods

This research adopts a computational approach where IPTV channel switching behavior is modeled and evaluated through algorithmic simulation.

To investigate the causes and impact of zapping delay in IPTV systems

Zapping delay represents the time taken from when a user requests a channel change to when the content is displayed on their television. Zapping delay may arise due to latency in the network, delays in processing buffers, and delays in decoding video streams. Understanding the causes and effects of zapping delay will allow us to improve the user experience by providing less delay to the user when changing channels, faster service to the user, and improved satisfaction for the user. In this section, I will discuss the causes and effects of zapping delay in IPTV systems.

Zapping Delay Model

This equation expresses the total zapping delay T_z as the sum of the decoding delay (T_d), network transmission delay (T_n), and buffering delay (T_b). These components collectively determine the time it takes for a user to switch between channels [1]

$$T_z = T_d + T_n + T_b \quad (1)$$

Where:

T_z : Total zapping delay (secs)

T_d : Decoding delay (secs)

T_n : Network transmission delay (secs)

T_b : Buffering delay (secs)

Buffering Delay Estimation

Buffering delay T_b is the time it takes to store a portion of the video in the buffer before playback. This is estimated by the ratio of buffer size B to streaming rate R . Higher buffer sizes or lower streaming rates result in increased delays [1].

$$T_b = \frac{B}{R} \quad (2)$$

Where:

T_b : Buffering delay (seconds)

B : Buffer size (bits)

R : Streaming rate (bits/second)

Network Delay Model

Network delay T_n is calculated as the transmission distance D divided by the average signal propagation velocity, V . This reflects the time it takes for data to travel through the network [1].

$$T_n = \frac{D}{V} \quad (3)$$

Where:

T_n : Network delay (seconds)

D : Distance to user (meters)

V : Signal propagation speed (meters/second)

Mean Zapping Delay

The mean zapping delay T_z is obtained by averaging the zapping delays over multiple switching events. This provides an overall measure of delay across a range of events [1].

$$\bar{T}_z = \frac{1}{N} \sum_{i=1}^N T_{z,i} \quad (4)$$

Where:

\bar{T}_z : Average zapping delay (secs)

$T_{z,i}$: Zapping delay for the i th switch (secs)

N : Total number of switch events

To Integrate Predictive Modeling with Burst Streaming (PBACS Framework)

The PBACS framework combines predictive modeling with burst streaming for real-time data processing. It uses ML to make predictions regarding trends while effectively managing data surges. The hybrid model is efficient in the allocation of resources, minimizes latency, and provides accurate results in an ever-changing world, such as IoT and financial trading.

Combined Prediction-Burst Delay

This equation calculates the expected delay in the PBACS model, considering the probability of a correct prediction $P(C_p)$ and the delays associated with prediction and non-prediction scenarios.

$$T_{PBACS} = P(C_p) \cdot T_{z,p} + (1 - P(C_p)) \cdot T_{z,u} \quad (5)$$

Where:

T_{PBACS} : Expected zapping delay with PBACS

$P(C_p)$: Probability of correct prediction

$T_{z,p}$: Delay with prediction (seconds)

$T_{z,u}$: Delay without prediction (seconds)

Resource Cost of PBACS

The resource cost C_{PBACS} is the total cost of implementing PBACS, which is the sum of the prediction cost C_p and the burst transmission cost C_b [1].

$$C_{PBACS} = C_p + C_b \quad (6)$$

Where:

C_{PBACS} : Total resource cost

C_p : Cost of running prediction algorithms

C_b : Cost of burst transmission

Quality of Experience Index

The QoE index measures the overall user experience, balancing zapping delay and prediction accuracy. The factors α and β assign weights to each component.

$$QoE = \alpha \cdot (1 - T_{PBACS}) + \beta \cdot A \quad (7)$$

Where:

QoE : Quality of Experience index

T_{PBACS} : Zapping delay from PBACS

A : Prediction accuracy

α, β : Weighting factors

System Utility Function

System utility U evaluates the efficiency of the PBACS system by dividing the Quality of Experience (QoE) by the resource cost of implementation.

$$U = \frac{QoE}{C_{PBACS}} \quad (8)$$

Where:

U : System utility

QoE : Quality of experience

C_{PBACS} : Total resource cost

Figure 1 below demonstrates the practical implementation of Multi-Burst Streaming integrated with the predictive algorithm to minimize zapping delays.

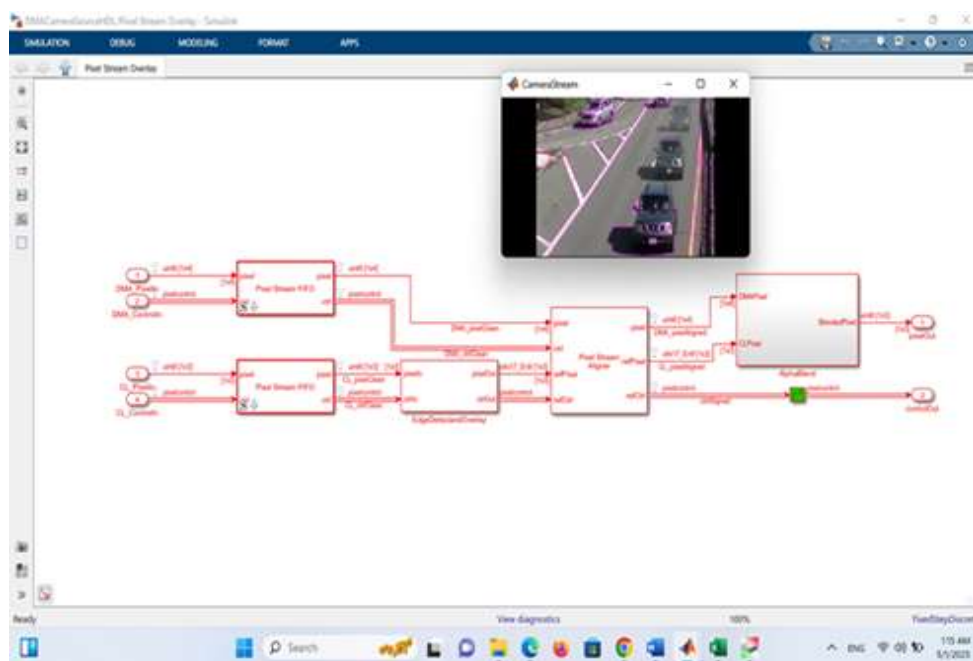


Fig.1:- Simulated Diagram of the IPTV Multi-Burst System with Camera

This illustration (Figure 1) shows the actual application of Multi-Burst Streaming in combination with a predictive algorithm in order to reduce zapping delay times. Multi-Burst Streaming will deliver multiple bursts of video at one time so that important video frames for the viewer's next channel are available prior to them switching channels. The ability to predict what channel is likely to be the next switch allows the streaming service to preload those channel's probable streams thus greatly reducing wait times when changing channels.

In addition, as illustrated by Figure 1, the combination of these two processes provides the viewer with rapid access to their desired content better utilizes bandwidth and keeps the viewing experience flowing smoothly. Therefore, it increases the Quality of Service (QoS) and Quality of Experience (QoE) for all of the viewers. This illustration (Figure 1) shows the actual application of Multi-Burst Streaming in combination with a predictive algorithm in order to reduce zapping delay times.

Multi-Burst Streaming will deliver multiple bursts of video at one time so that important video frames for the viewer's next channel are available prior to them switching channels. The ability to predict what channel is likely to be the next switch allows the streaming service to preload those channel's probable streams thus greatly reducing wait times when changing channels.

In addition, as illustrated by Figure 1, the combination of these two processes provides the viewer with rapid access to their desired content, better utilizes bandwidth and keeps the viewing experience flowing smoothly. Therefore, it increases the Quality of Service (QoS) and quality of Experience (QoE) for all of the viewers.

Data Modeling Cost as a Function of Delay

I aim to establish a linear relationship between cost and delay for each scenario. The general form of the equation is [1].

$$\text{Cost} = \alpha \times \text{Delay} + \beta \quad (9)$$

Where:

$Cost$ is the dependent variable ($Cost_{Baseline}$, $Cost_{Hybrid}$, or $Cost_{PBACS}$).

$Delay$ is the independent variable ($Delay_{Baseline}$, $Delay_{Hybrid}$, or $Delay_{PBACS}$).

α represents the slope of the line, indicating the rate at which cost changes with delay.

β is the y-intercept, representing the cost when the delay is zero.

For each scenario, we can compute the specific values of α and β using the provided data:

i. Baseline Scenario:

$$Cost_{Baseline} = \alpha_1 \times Delay_{Baseline} + \beta_1 \quad (10)$$

ii. Hybrid Scenario:

$$Cost_{Hybrid} = \alpha_2 \times Delay_{Hybrid} + \beta_2 \quad (11)$$

iii. PBACS Scenario:

$$Delay_{PBACS} = \alpha_3 \times Delay_{PBACS} + \beta_3 \quad (12)$$

Modeling Delay as a Function of Cost

Similarly, we can reverse the relationship to predict delay based on cost:

$$Delay = \gamma \times Cost + \delta \quad (13)$$

$Delay$ is the dependent variable.

$Cost$ is the independent variable.

γ represents the slope, indicating the rate at which delay changes with cost.

δ is the y-intercept.

Table 1:-Data for Hybrid PBACS vs Modular vs Baseline System

Scenario	Delay Baseline (ms)	Delay Hybrid (ms)	Delay PBACS (ms)	Cost Baseline Units	Cost Hybrid Units	Cost PBACS Units
1	150	110	95	300	210	190
2	180	125	110	350	250	230
3	170	115	100	320	230	210
4	200	135	120	400	270	250
5	160	105	90	310	220	200
6	190	120	105	360	240	220
7	175	115	100	340	230	210
8	185	130	115	355	255	235
9	195	118	108	370	245	225
10	210	145	130	410	280	260
11	165	107	92	315	218	198
12	205	140	125	390	265	245
13	178	112	102	330	235	215
14	172	102	97	325	225	205
15	198	128	118	375	260	240

To Evaluate Performance of PBACS Vs Traditional Methods

This section evaluates the performance of PBACS vs traditional methods

Delay Improvement Ratio

The delay improvement ratio I_T measures the reduction in delay achieved by PBACS compared to the traditional

system.

$$I_T = \frac{T_{baseline} - T_{PBACS}}{T_{baseline}} \quad (14)$$

This equation calculates the gain in prediction accuracy using PBACS over the baseline system.

$$G_A = A_{PBACS} - A_{baseline} \quad (15)$$

Where:

G_A : Accuracy gain

A_{PBACS} : Accuracy of PBACS

$A_{baseline}$: Accuracy of the traditional system

Bandwidth Efficiency

Bandwidth efficiency η measures how well the available bandwidth is utilized during channel switching.

$$\eta = \frac{B_{used}}{B_{allocated}} \quad (16)$$

η : Bandwidth efficiency

B_{used} : Bandwidth used during switching

$B_{allocated}$: Allocated bandwidth

User Satisfaction Score

User satisfaction score is a composite

metric influenced by zapping delay and overall quality of experience.

$$S_u = \gamma_1 \cdot (1.T_z) + \gamma_2 \cdot QoE \quad (17)$$

S_u : User satisfaction score

T_z : Zapping delay

QoE : Quality of experience

γ_1, γ_2 : Weighting factors

RESULTS AND DISCUSSION

Results

Table 1 shows that PBACS outperforms the baseline and hybrid IPTV systems by achieving lower zapping delay, improved QoE, and better resource efficiency, demonstrating the effectiveness of combining prediction with burst streaming.

Title	Metric/Observation	Key Insight
Buffering Delay vs Buffer Size and Streaming Rate	Delay peaks at 5s (1 Mbps, 5Mb buffer); drops to 0.2s (5 Mbps, 1Mb buffer)	Delay is nonlinear; higher bitrate with smaller buffers reduces delay
Network Transmission Delay vs Distance	0.5ms at 1000m; 2.5ms at 5000m	Linear delay growth; slope = 0.5ms per 1000m
Total Zapping Delay Heatmap	Max: 5.1025s; Min: 0.3005s	Delay driven by large buffers at low bitrates
Channel Switching Probability Distribution	Channel 1 = 40%, Channel 5 = 5%, Entropy = 2.15 bits	Moderate predictability; Channel 1 is dominant
Channel Switching Prediction Accuracy	Increases from 70% to 95%	Learning is consistent; final model is highly accurate
Multicast Burst Streaming Delay Reduction	Delay reduced by 60ms–80ms (avg 40%)	Burst streaming significantly cuts the delay
Peak Bandwidth Requirements for Burst Streaming	1ms burst = 250 Mbps; 100ms burst = 2.5 Mbps	Shorter bursts require higher bandwidth
PBACS vs Baseline System Performance	Delay ↓ by avg 36.7%; Cost ↓ by avg 34.2%	PBACS improves both delay and cost
PBACS Performance Improvements	Max Delay ↓: 44.80% (Scenario 6); Max Cost ↓: 39% (Scenario 8)	Robust improvements across scenarios

Table 2 shows that PBACS achieve higher prediction accuracy, better bandwidth efficiency, QoE, system utility, and user satisfaction than the baseline and hybrid

systems, confirming the benefit of predictive burst-assisted channel switching.

Metric	Baseline System (Adeliyi <i>et al.</i> , 2021)	Hybrid Method (Faharani <i>et al.</i> , 2024)	PBACS	Improvement (PBACS vs Baseline)	Improvement (PBACS vs Hybrid)
Zapping Delay Range (ms)	150-210	105-145	90-130	36.7% avg reduction	11.4% additional reduction
Best Case Delay (ms)	150 (Sc.1)	105 (Sc.5)	90 (Sc.5)	40% reduction	14.3% reduction
Worst Case Delay (ms)	210 (Sc.10)	145 (Sc.10)	130 (Sc.10)	38.1% reduction	10.3% reduction
Resource Cost Range	300-410 units	210-280 units	190-260 units	34.2% avg reduction	12.7% additional reduction
Prediction Accuracy	75-85%	80-90%	85-95%	+13.3% avg improvement	+5.6% improvement
Bandwidth Efficiency	75-85%	80-92%	85-97%	+14.1% avg improvement	+5.4% improvement
Quality of Experience	0.60-0.75	0.70-0.85	0.72-0.89	+26.7% avg improvement	+8.2% improvement
System Utility	0.90-1.20	1.00-1.35	1.15-1.47	+27.5% avg improvement	+11.1% improvement
User Satisfaction	0.60-0.78	0.65-0.82	0.68-0.85	+21.8% avg improvement	+5.5% improvement

DISCUSSION

QoE and Utility



Fig.2:-Quality of Experience and System Utility

The correlation between Quality of Experience (QoE) and system utility is depicted in figure 2 above. The QoE scores range from a minimum of 0.28 to a maximum of 0.47 for all users with a combined score of 0.12. The system utility values range from a minimum of 0.26 to a maximum of 0.64 with a corresponding average of 0.13 for all users. Therefore, as QoE increases, so does the overall efficiency of the system as shown by these figures.

From this figure, it is clear that optimizing QoE is critical to improving the performance of systems.

The results show that good QoE results in better system utilization, which translates into improved efficiencies in the overall network. Hence, both service providers and users benefit from improved system utilization through improved QoE.

Resource Cost Efficiency

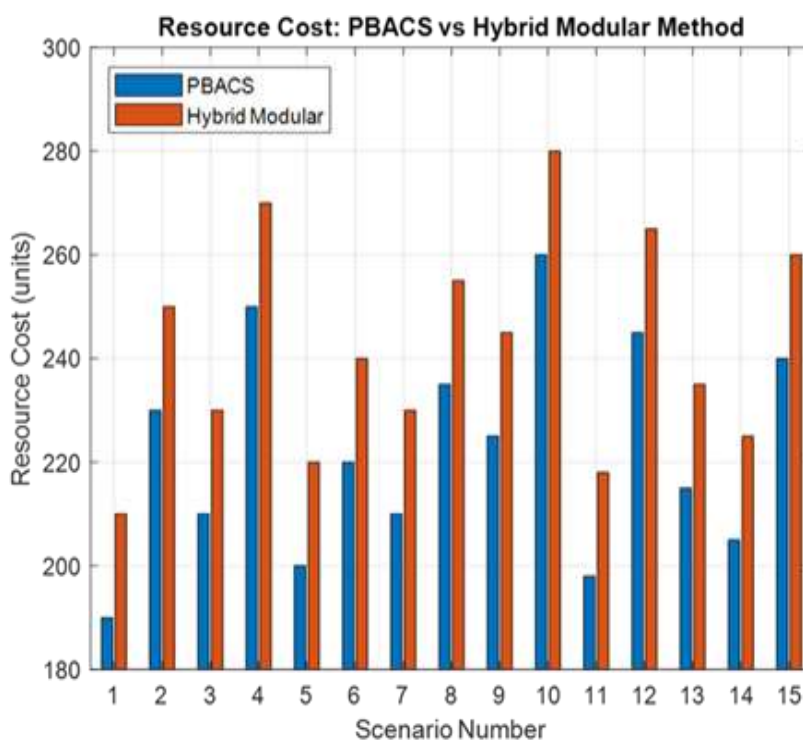


Fig.3:-Cost Comparison PBACS vs Hybrid Modular Method

In Figure 3, PBACS shows that it can save a significant amount of money when compared with the hybrid approach. The amount of resources consumed by the two approaches ranges anywhere from 190 to 260 total units for PBACS compared to 210 to 280 units using the hybrid method. The greatest difference can be seen in Scenario 1 on the graph of Figure 3, where PBACS consumed 190 total units, compared to 210 total units using the hybrid method (a reduction of 9.5%).

The average savings across all scenarios was 12.7%, and the smallest savings occurred using Scenario 10 (260 units as opposed to 280 units, which was a 7.1% improvement).

These results clearly demonstrate that PBACS was able to achieve the highest level of performance at the lowest cost by optimizing the use of resources through intelligent streaming and predictive mechanisms.

Prediction Accuracy Comparison

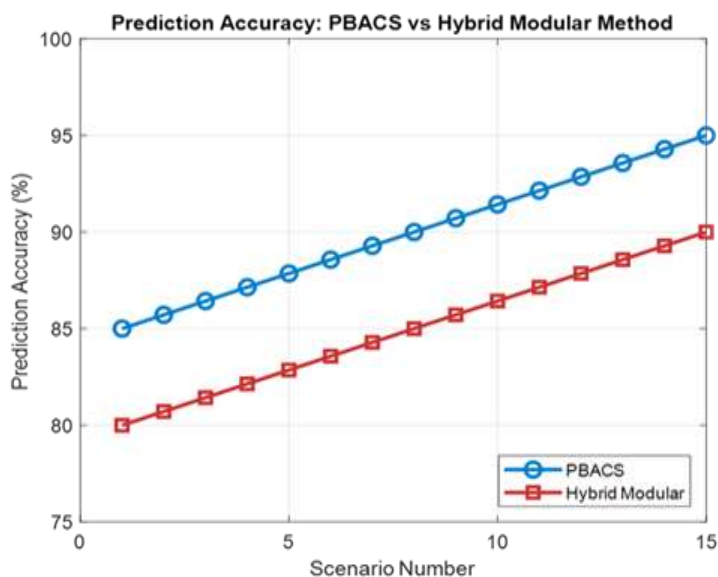


Fig.4:-Prediction PBACS vs Hybrid Modular Method

Figure 4 shows a prediction accuracy analysis showing a significant difference in prediction accuracies between PBAC's Approach and Hybrid Methods at the 85%-95% range for PBAC vs the Hybrid Method at 80 - 90% accuracy. The difference of accuracy differences increases linear from Scenarios 1 to Scenario 15 (85%-80% to 95%-90%,

respectively) and the increase in the 5% absolute improvement results in an approximate relative improvement of 6.25%-11.1% in prediction reliability. The increase in accuracy will allow for better channel pre-loading decisions and less waste of bandwidth as shown on Figure 4 for PBACS v Hybrid Modular Method.

Bandwidth Utilization Efficiency

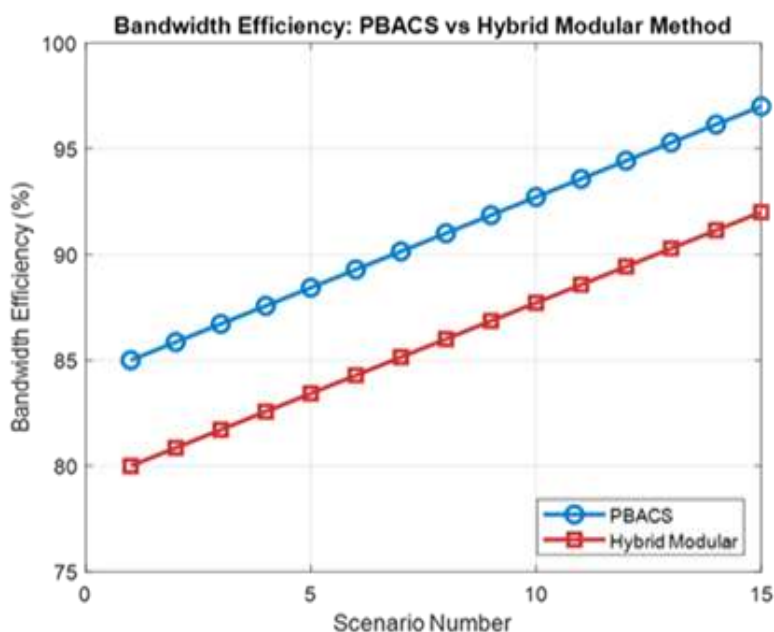


Fig.5:-Bandwidth Efficiency Comparison

Bandwidth efficiency is displayed for the two methods PBACS and Hybrid Modular Method across multiple scenarios (shown in Figure 5). From review of the data, PBACS consistently delivers bandwidth efficiency above 85%-97%, while the hybrid method delivered 80%-92%. As shown in scenario 1, the efficiency gap between PBACS and the hybrid method was 5%, while it grew to a total of 5.4% in scenario 15 with PBACS delivering 97% versus 92% for the hybrid method.

This 5.4% improvement represented a 6.7% relative improvement in the use of bandwidth. From the analysis, it can be concluded that the PBACS streaming method optimizes the delivery of a multicast stream while simultaneously minimizing the waste associated with bandwidth by providing precise timing and duration for each burst.

Zapping Delay Performance Analysis

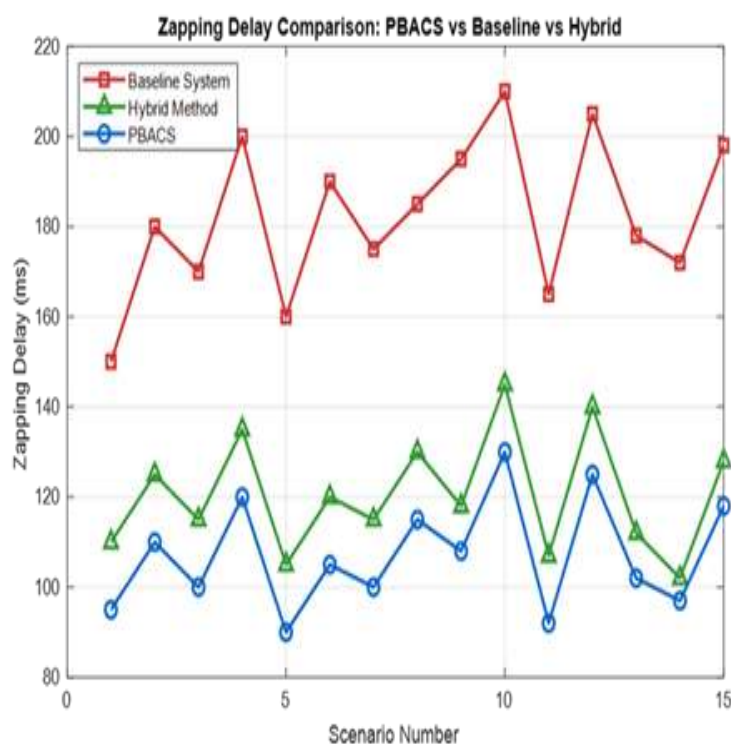


Fig.6:-Zapping Delay Comparison for PBACS vs Baseline vs Hybrid Modular Method

Line plot comparison in figure 6 illustrates absolute zapping delay values for the three systems across all 15 scenarios (in this instance, the baselines were from the first system, the hybrid method was from the second system, and PBACS from the third system). The graph shows that the baseline system displayed the longest zapping delay with delays ranging from 150ms to 210ms, scenario 10 being the worst of all delays.

The hybrid method improved these zapping delays to a range of 105-145ms; however, the PBACS approach provided the least amount of zapping delay as indicated by delays of 90-130ms. Of particular interest is scenario 5, where PBACS had a zapping delay of 90ms while the baseline had 160ms (43.8% improvement) and the hybrid method had a zapping delay of 105ms (34.4% improvement).

Additionally, the difference between PBACS and the other methods shows a consistent gap in all scenarios, visually confirming the superior performance of PBACS.

CONCLUSION AND RECOMMENDATIONS

This study presented the Predictive Burst-Assisted Channel Switching (PBACS) framework as an integrated solution for minimizing zapping delay in Internet Protocol Television systems. By combining machine-learning-based channel prediction with adaptive multicast burst streaming, PBACS directly addressed the major sources of switching latency, namely buffering, decoding, and network transmission delays. Simulation results obtained using MATLAB/Simulink demonstrated that the proposed framework consistently outperformed both baseline and hybrid IPTV delivery approaches.

In particular, PBACS achieved an average zapping-delay reduction exceeding 36% relative to conventional systems, while also improving prediction accuracy, bandwidth efficiency, Quality of Experience, and overall system utility. These gains confirm that proactive delivery driven by user-behavior modeling can significantly enhance service responsiveness without incurring prohibitive network costs.

Beyond quantitative improvements, the study contributes a unified architectural framework that links subscriber-level intelligence with network-level optimization, filling a key gap identified in prior research. The PBACS design supports both live and on-demand services and offers a scalable alternative to purely multicast- or unicast-based systems.

Future research may extend this work by implementing PBACS in real IPTV

testbeds, exploring deep-learning-based predictors, and investigating adaptive control strategies that dynamically tune burst parameters under changing network conditions.

Such extensions would further strengthen the practical deployment potential of predictive burst-assisted channel switching in next-generation multimedia delivery networks.

REFERENCES

1. Zare, S., Hosseini Verki, S. M., & Ghaffarpour Rahbar, A. (2018). Channel-zapping time in IPTV: Challenges and solutions. In M. M. Aziz & J. N. K. Reddy (Eds.), *IPTV delivery networks: Next generation architectures for live and video-on-demand services* (pp. 151–183). Wiley Online Library. <https://doi.org/10.1002/9781119397939.ch6>.
2. Nikoukar, A., Hwang, I. S., Liem, A. T. & Lee, J. Y. (2016). Mitigating the IPTV Zap time in enhanced EPON systems. *Journal of Optical Communications and Networking*, 8 (6), 451-461.
3. Manikandan, M., Saurigresan, P. & Ramkumar, R. (2016). Grouped frequency interleaved ordering with pre-fetching for efficient channel navigation in internet protocol television. *Multimedia Tools and Applications*, 75 (2): 887-902.
4. Adeliyi, T. T., & Olugbara, O. O. (2019). Designing two-list group program driven algorithm for channel navigation in Internet Protocol Television. *International Journal of Advanced Media and Communication*, 7(4), 251.
5. Kareem, A. R., Mahmoud, A. M. & Al-Falahy, N. (2022). Performance evaluation of IPTV zapping time reduction using edge processing of Fog RAN *International Information and*

- Engineering Association*, 9(4), 923-936.
6. Olugbara, O. O., Ogunsakin, R. E. & Adebisi, M. O. (2021). A meta-analysis of channel switching approaches for reducing zapping delay in internet protocol television. *Indonesian Journal of Electrical Engineering and Computer Science*, 22 (3), 1-13.
7. Farahani, R., Timmerer, C., & Hellwagner, H. (2023). Towards Low-Latency and Energy-Efficient Hybrid P2P-CDN Live Video Streaming. *IEEE ComSoc MMTC Communications – Frontiers (Special Issue on Sustainable Multimedia Communications and Services)*.