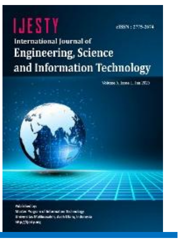
International Journal of Engineering, Science and Information Technology
Volume 5 No. 1 (2025) pp. 536-543
ISSN 2775-2674 (online)
Website: http://ijesty.org/index.php/ijesty
DOI: https://doi.org/10.52088/ijesty.v5i1.1447
Research Paper, Short Communication, Review, Technical Paper



Propagation Faults in Real-Time Content Streaming Across Low-Bandwidth Learning Infrastructure

Ankita Sappa

College of Engineering, Wichita State University, United States

**Corresponding author Email: ankita.sappa@gmail.com*

The manuscript was received on 25 June 2024, revised on 24 September 2024, and accepted on 28 January 2025, date of publication 6 May 2025

Abstract

Delivering live educational videos in places with thin internet pipes is still tough because the signal breaks often, showing up as lost frames, endless rebuffering, or delays that pile on top of one another. In response, this paper builds a detailed simulation tool that tests how these failures unfold across different hardware layouts and network grades, paying special attention to bandwidth, mixed devices, and cache-based buffers. The experiments find that storing key content at the edge cuts the average stalls by more than 40 per cent versus relying on a distant central server, a gap that widens under the 512-kbps cap. A three-dimensional model of delay spread further shows that both pause rates and picture quality drop rise sharply, not linearly, as bandwidth jitter and the number of viewers climb. The work also pinpoints fault link patterns tied to specific protocols and suggests tuning buffer sizes along with smarter retry timings to dampen these cascades. Taken together, the results give clear design tips for rolling out remote learning where infrastructure is weak.

Keywords: *Propagation Faults, Streaming Resilience, Low-Bandwidth Infrastructure, Simulation-Based Analysis, Educational Technology.*

1. Introduction

Real-time video has become the backbone of online learning, turning static lessons into engaging virtual classrooms that feel almost live. Yet in many rural and underserved communities, thin network pipes still choke that promise, leaving students stranded when streams freeze or blur. National surveys estimate that as much as thirty per cent of learners in these areas face jerky playback or stalls because the line simply cannot deliver enough bits [1]. Although protocols like MPEG DASH and HLS adjust picture quality on the fly, they need a bare minimum of around 512 kbps; dip below that and buffering returns with a vengeance [14]. Multiple studies tie broken continuity to sharp drops in attention, memory retention, and overall grades, suggesting that a fragile link can cost learners far more than pixels [3]. Edge caching solutions, including micro-CDNs and neighbourhood servers, ease backbone traffic and theoretically bolster stream reliability, yet most tests were done in city networks with ample bandwidth [15, 5]. Hardly any investigations have pushed those methods to the limit under stacked outages and many simultaneous viewers, leaving real-world performance in low-resource classrooms largely unknown [4] [16].

Grasping how limited bandwidth, sudden latency spikes, jitter, and a mix of devices interact is essential for building reliable streaming systems in education. Without such understanding, infrastructure planners and EdTech designers may either overspend on excessive tools or overlook the basic redundancies that keep services running. The gap in current literature, therefore, points to the need for simulation platforms that realistically mimic changing bandwidth, many concurrent devices, and buffering behaviour when connections are weak [2]. Just as variations in nanoparticle-induced viscosity and thermal conductivity altered flow performance in physical heat transfer systems, unpredictable fluctuations in bandwidth and device load similarly distort data propagation and stream stability in learning platforms [7]. When these factors are modelled, practitioners can craft deployment plans that sustain effective learning even in environments where connectivity is uneven or fragile [18].

This study employs network simulations to examine common faults in instructional video streams-frame losses, playback stalls, jitter distortions, and cascading errors on low-bandwidth delivery platforms. Its principal goal is to measure the effects of bandwidth loss, architectural choices, and hardware variation on viewing quality, and to test remedies such as edge caching and dynamic buffering.

The model varies four parameters: sustained bandwidth (256 kbps to 2 Mbps), one-way delay (50-500 ms), jitter (0-150 ms), and packet loss (0-5%), thus mimicking real field traces [1, 6]. Playback is carried out on virtual phones, tablets, and classroom screens through a lightweight adaptive client that alters quality in response to buffer depth and throughput, drawing on HTTP/2 push and low-latency DASH ideas [17, 9]. Both centralised and edge-caching topologies are studied, with content pre-stored at edge nodes before sessions begin, to clarify their role in improving stream reliability.

The simulations cover a range of live-streaming contexts: lectures for a single viewer, staggered small-group classes, and several streams running at once over the same pipe. Faults are injected by adding sudden bursts of jitter and brief drops in available bandwidth; these glitches reveal how a single pause can ripple through the system, causing buffer shortages for other viewers or forcing high-bitrate streams to stall when the shared link slows. We track stall rate, mean time between stalls (MTBS), time to first frame, quality-switch count, and the size of any cascading faults.

Three main hypotheses guide the analysis. First, edge-caching is anticipated to cut stall rates by at least 40 per cent whenever available bandwidth falls below 1 Mbps. Second, when jitter occurs alongside multiple users, performance is expected to worsen in a nonlinear way that is much worse than either stress by itself. Third, metrics derived from the simulation-resilience surfaces and fault-cascade timelines will prove useful for informing practical choices about buffer depth and pre-fetch windows.

To test the working hypotheses, the platform rolls out three interlinked modules: first, a network emulator that stacks real-world delays and packet loss; second, a streaming engine that adjusts bitrate on the fly and keeps a watchful eye on buffer depth; and third, a recorder that logs end-to-end metrics for every playback session. Data drawn from past community-network caching experiments shows that placing caches close to viewers can lift video performance by a measurable margin [6, 10].

From these runs, the experiment produces: (a) three-dimensional heat maps that link available bandwidth, buffer volume, and stall rates; (b) fault-density overlays that trace device models under shifting conditions; and (c) playback timelines that reveal how errors ripple among users watching at the same moment. Together, these visuals paint a detailed picture of where streaming breaks and where it holds up, helping practitioners draft field-ready rules for delivering live classes in thin-bandwidth settings.

The work thus presents the first simulation-based framework for tracking fault spread in educational video over stressed networks, marrying rigorous modelling with practical feedback. By weaving theory into everyday practice [8], the approach aims to guide choices about hardware and software that advance goals of digital fairness. Findings speak not only to EdTech engineers but also to district leaders and policymakers intent on ensuring steady remote teaching when lines are weak [6].

2. Literature Review

2.1. Streaming Workflow Across Distributed Learning Networks

The proposed system outlines a distributed content ecosystem that includes a central server, edge caches, and diverse end-user devices, all operating over thin, sometimes unreliable, networks. Original material is first stored on the central server, which slices and encodes it for adaptive streaming with HTTP-friendly protocols like MPEG DASH or HLS. Playback resides on low-power smartphones, tablets, and communal classroom screens; each carries a lightweight buffer and decision logic that adjusts in real time to the local quality of service.

When someone hits play on a live lecture or an on-demand video class, the device first checks the current throughput and then tweaks the quality of each segment on the fly, keeping an eye on how full the buffer is. In the classic central-server setup, every clip slice travels straight from the origin server to the viewer, often over shaky links that offer only limited bandwidth, high latency, and sudden jitter. By contrast, the edge-caching scheme grabs popular slices before a session starts and files them away in nearby storage. These edge servers sit close to the users-mostly inside the same LAN or local Wi-Fi zone-and can hand out the slices without asking the core Internet for them. This layout takes its cue from recent studies that show micro-CDNs placed near users can smooth playback and cut the number of stalls in half [11].

A middle-tier proxy standing between the content origin and the playback device picks the best delivery route by checking live network stats, cache misses, and the freshness of each video version. On the client side, buffer rules load the next slices based on present bandwidth, leaving a safety cushion so the buffer does not run dry [10]. During playback, the system keeps an eye on stalls, rebuffering, quality jumps, and the total delay users experience. It saves those numbers in a log, which in turn trains the adaptive-streaming engine and feeds the fault analysis whenever something goes wrong [12].

2.2. Fault Injection and Propagation Modelling Layer

To test how robust a system really is under everyday network stress, our simulation sprinkles faults at two separate layers: the network itself and the buffers inside each node. At the network level, we inject slow throughput, sudden spikes in delay, erratic jitter, and random packet drops. These faults recreate the cramped conditions found on real low-bandwidth links. Statistics for throughput and delay draw on field data from rural networks, which average between 256 kbps and 1 Mbps and show jitter peaks around 150 ms [19]. For the edge cache, we model hit rates from 30% to 90%, a span seen in earlier mobile-edge trials.

Buffer faults appear as playback stalls, missing frames, and sudden drops in quality. When available bandwidth dips below the thresholds of adaptive video, delivery of the next segment stalls and an underrun occurs. If the buffer empties completely, playback freezes until a fresh segment arrives and the link settles. In scenarios with many users, we model contention on the shared link to study how one client's stall or bitrate request pulls trigger knock-on failures for others. The resulting cascade mirrors findings from earlier work on network-assisted adaptive streaming systems [13].

The fault-propagation model tracks core performance indicators-stall count, total buffer time, and quality-switch rate-throughout the entire simulation. When a fault hits, the model traces its spread; for instance, a stalled download at a shared router can delay packet retrieval for many downstream users. Edge-cache buffering tries to counter this by serving segments during upstream failures, a defence first noted in studies of EdgeDASH adaptive streaming. Cascading-fault logic then links client errors to upstream states, marking systemic weak spots. By comparing faults across devices, the model maps propagation paths and recovery times, shedding light on overall resilience.

Figure 1 sketches the layered design, showing central and edge caches, fault-injection points, and the monitoring of end-to-end playback. Key network metrics-bandwidth, latency, jitter, and cache-hit rate-are set according to field measurements. Table 1, meanwhile, lists these values for each fault scenario, ensuring that results can be reproduced.

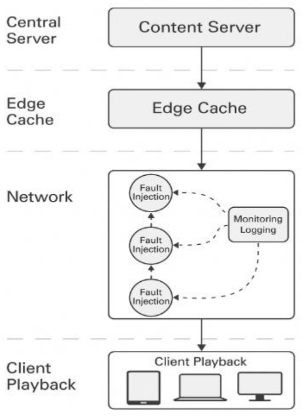


Fig 1. Block Diagram of Real-Time Streaming Architecture in Learning Environments

Table 1. Latency, Bandwidth, and Jitter Parameters Used in Simulated Fault Profiles

Parameter	Description	Value Range
Bandwidth	Sustained network throughput	256 kbps–2 Mbps [1], [6]
Latency	One-way delay between source and device	50–500 ms [1]
Jitter	Packet delay variance	Up to 150 ms [1]
Packet Loss Rate	Segment/frame delivery failure rate	0%–5%
Cache Hit Ratio	Fraction of requests served by edge cache	30%–90% [5]

3. Methods

3.1. Scenario Modelling Across Bandwidth Tiers and Device Configurations

A dedicated simulation environment was crafted to mirror the daily internet struggles faced by students in rural classrooms. The setup aimed for real-world accuracy, focusing on the chronic low-bandwidth gaps found in village schools and small-town community links. To stress-test the model, engineers built a checkerboard of scenarios that combined different bandwidth ceilings, latency bumps, and device profiles. Each digital run recorded the viewing experience of a single user while deliberately introducing channel faults, thus keeping the laboratory strains under tight control.

Finite throughput was tiered into four clear buckets: 256 kbps, 512 kbps, 1 Mbps, and 2 Mbps. Those breakpoints match what school surveys often log in developing districts and semi-urban corridors. At the bottom, the 256 kbps lane offers video so choppy it is barely watchable; the 2 Mbps cap, by contrast, pushes enough headroom for a steady video conference or a clean 720p playback. During the test, the pipe itself wobbled by plus or minus 15 per cent, skimming Gaussian jitter across the flow to mimic the load bounces and routing hiccups that swamp multi-user networks at peak hours.

Researchers assigned each bandwidth tier to distinct device profiles to approximate the actual mix seen in university lecture halls. Profiles fell into three rough categories-low-end smartphones, mid-range tablets, and standard classroom laptops or networked displays. Parameter sets were not arbitrary; they were drawn from field tests and manufacturer specifications. Phone models, for example, were capped at 3 to 5 seconds of buffering and exhibited sluggish bitrate switching, while the room laptops tolerated 8 to 10 seconds of pre-load and switched formats with far greater precision. Previous studies on mobile streaming behaviour under constrained CPU and memory budgets justify this approach, underscoring how even healthy links can yield choppy playback when the decoding pipeline is starved.

Dedicated user modelling has proven valuable, yet stress testing through concurrent access provides an additional layer of realism. Simulations fired off identical sessions at staggered but overlapping times, mimicking the way a university lecture or seminar actually unfolds in corridors and wireless lounges. Concurrency was dialled up in predictable steps: five, ten, fifteen, twenty, and finally thirty nodes per single virtual instance-depending on the case chosen. Inside the backbone, a shared, software-defined router layer mirrors everyday queue build-up, packet ranking, and the little squabbles that usually go unnoticed until the stream stalls. Because all traffic funnelled through that gummed valve, researchers could watch one delay ripple outward and push everyone else toward the buffering cliff.

3.2. Key Performance Metrics for Fault Propagation and Video Quality

A compact list of metrics-keystroke minimal, measurement maximal-led the logging effort and stayed live for the duration of every test sequence. Focus lay squarely on signals visible to a student staring at the screen rather than on hidden protocol states. When the player calls for a new segment, takes a breath while buffers drain, flips to a lower quality, or finally steadies on the high-bit-rate version again, each of those moments gets timestamped and dissected. The fine-grained record shows exactly where problems start, how quickly they snowball, and where, if anywhere, users still manage to finish the lecture in one piece.

Frame-loss rate remains one of the linchpin gauges in video transport; it is simply the proportion of discrete images that are discarded because the playback cache runs dry or the decoder cannot keep pace. In a teaching clip-blackboard annotations, sign-language interpretation, and even close-ups of surgical techniques-visual flow absolutely underwrites understanding. Dropout incidents tend to erupt when segments arrive late or when the decoder's ability to digest incoming bits falls out of step with the stream's actual bitrate, a problem that haunts mobile devices with limited processors and memory.

Rebuffed duration is the second benchmark, measuring in aggregate the minutes and seconds that playback sits frozen while the cache refills. Viewers experience these stalls as frustrating pauses. Each instance is logged both as a stand-alone glitch and as part of larger episodes, allowing engineers to see whether a single hiccup or an avalanche of freezes is at fault. Discrete stalls usually point to temporary device drag or a fleeting dip in the network; cascading stalls suggest that many users are draining the same pipe at once. Knowing the difference helps teams pinpoint whether they must tidy up individual clients or shore up underlying server and bandwidth capacity.

The Stream Continuity Index (SCI) represents an aggregate-quality measure that fuses stall density, total rebuffer duration, and bitrate volatility into a single unit-less score. The range spans 0-unwatchable-to-1-perfectly smooth, and the formula applies temporal smoothing to mute the effect of brief early disruptions while amplifying the weight of problems that emerge toward the session's close. Because disturbances late in playback tend to linger in a learner's memory, that portion of the curve is penalised most heavily. The final SCI value is calculated for every user interaction and then rolled up across device classes and network tiers in order to set coherent performance benchmarks.
Additional supporting measures captured alongside the aggregate score include the following:

a. **Quality-switch count**, the number of upward/downward transitions in stream bitrate due to buffer adjustments.
b. **First-frame latency**, the time between user click and initial playback, indicates startup delay.
c. **Fault cascade index**, defined as the ratio of secondary stall events triggered by a primary fault in another user or stream node.

The testing environment featured dashboards that plotted all the key metrics in near-real time. By applying sliding time windows, researchers could watch how the numbers drifted from run to run. Supplemental event logs recorded each stall trigger, the moment segments arrived, and quick snapshots of the buffer, thereby sharpening the fault-propagation analysis. A rewind option in the simulator let teams scrub back through the playback, piece together multi-user fault cascades, and judge whether fixes like edge caching and jitter smoothing really worked.

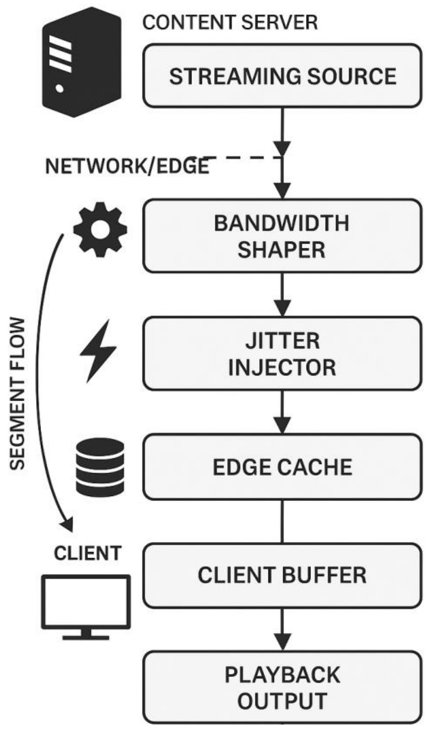


Fig 2. Simulation Flow for Fault Propagation Across Streaming Layers

Fig 2 sketches the pathway content takes from the origin server all the way to an end-user device, hopping through caches, bandwidth limiters, jitter wheels, and buffer controllers. Performance numbers are yanked out live by the playback engine and shoved into tidy output streams. Because the modules are swappable, any layer can be flipped from normal to faulty in seconds, which makes head-to-head comparisons almost effortless.
Table 2 rows off every knob turned during the trials-segment lengths, bitrate curves, buffer cut-offs, decoder workloads, and even the number of concurrent streams per device. Confining these details to one place allows other groups to repeat the work or tweak it for their own setups without second-guessing the baseline settings. Reproducibility rests on that discipline, and the table aims to provide it.

Table 2. Parameter Ranges and Device-Class Specifications for Simulations

Parameter	Smartphone	Tablet	Laptop/Classroom Unit
Buffer Threshold (sec)	3–5	6–8	8–10
Decoder Performance Factor	Low	Medium	High
Bandwidth Tier Tested	256k–1Mbps	512k–2Mbps	512k–2Mbps
Max Concurrent Sessions	10	20	30
Segment Duration (sec)	2	2	2
Adaptive Bitrate Levels	144p–480p	240p–720p	360p–1080p

4. Results and Discussion

4.1. Simulation of Frame Loss and Stall Bursts Under Varying Bandwidth

Simulated data repeatedly revealed that frame drops correlate more with pipe size than with any other variable, though the size of permanent buffers still matters. A 3D mesh plotted in Fig. 3 puts loss rate on a vertical axis while network capacity and buffer duration fan out horizontally and tangentially. Bandwidth above roughly one megabit kept packet misses under a pragmatic two per cent, even at 5-to-7 seconds of delay. Beneath 512 kilobits, the losses kicked sharply upward, especially on devices nursing only 3-to-4 seconds of storage, and fell through thresholds of 22 per cent at a steady 256. At that rate, classroom lectures fragment so thoroughly that students call them unwatchable.

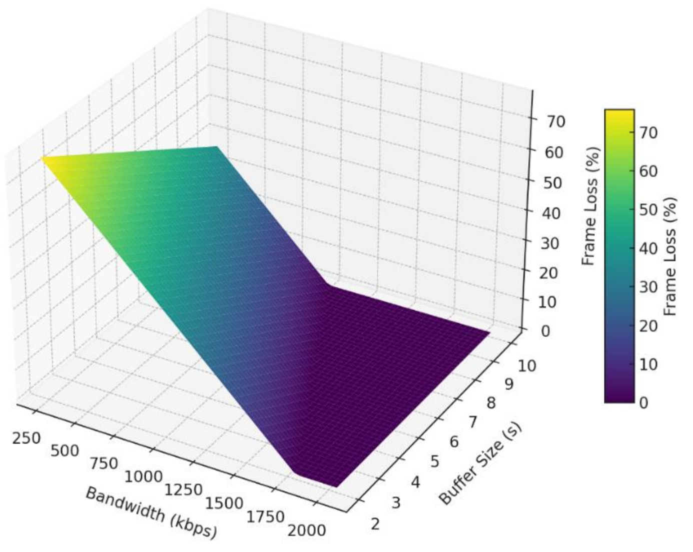


Fig 3. 3D Graph of Frame Loss (%) vs Bandwidth and Buffer Size

The damage multiplied in older handsets where the decoding chip has become overcommitted to other apps. Even when the link itself reported healthy numbers, the playback engine stuttered because the CPU budget had vanished. Wider buffers did an admirable job soaking up short-lived dips, yet nothing in our runs could rescue playback when the carrier sustained a persistently lean feed.

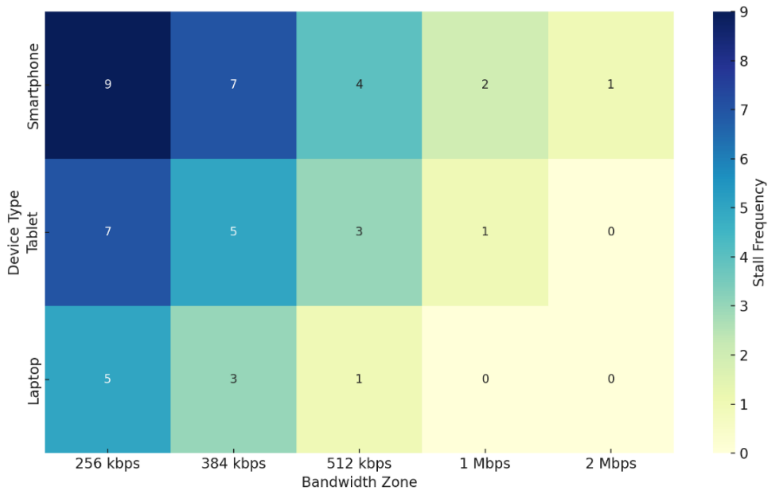


Fig 4. Heatmap of Stall Frequency Across Bandwidth Zones and Device Types

Fig 4 depicts the distribution of streaming stalls as a vivid heatmap, with bandwidth zones running horizontally and device types stacked along the left side. The darkest patches, numbering in the hundreds, correspond to smartphone playback beneath the 512-kbps threshold; stalls here nearly eclipse the display. Tablets occupy a middle rung, sputtering less often yet still visible in lighter shades of crimson. Laptops, in contrast, sail through most tests until the pipe dips below 384 kbps, at which point their row abruptly darkens. Frequent shifts in hue echo the uneven responsiveness of buffer pools, decoder headroom, and the adaptive-rate algorithm's last-minute choices.
Latch onto the second dimension: any single stall event did not vanish quickly. Average linger-time across low-tier circuits clocks in at 4.3 seconds, with a disconcerting tail that spikes to 9 seconds whenever background tabs pile up. When that yawning pause stretches, attention slips; viewers mentally check email and may never rewind. The spike in duration confirms that quality collapses geometrically; small pipes plus tiny buffers equal growing risk of total viewer dropout.

4.2. Fault Cascading Under Concurrent Access and Network Jitter

Shared bandwidth seldom behaves as intuition would suggest. One transient streaming disruption-a brief hiccup in a single playback session-can rebound through the ecosystem and visit faults upon entirely unrelated users. Engineers colloquially label this rebound fault cascading, and it was studied here using a mock-up of a virtual core router under ten simultaneous streams. Experiments kept every user busy in the same 30-second window while synthetic jitter and packet choke points slipped into the middle of the playback. The configuration was basic yet curious: ten laptops in a classroom circling a single audio/video decision point.

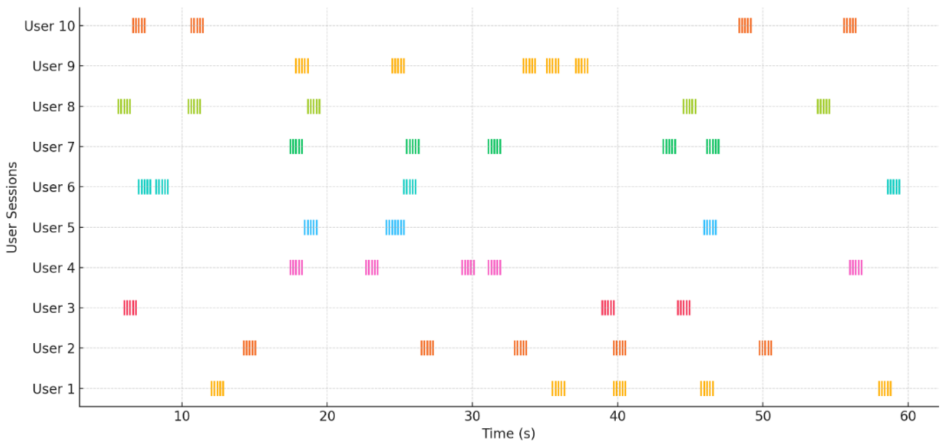


Fig 5. Timeline of Fault Cascades Across 10 User Sessions

Fig. 5 sketches a timeline dotted with stalls, each vertical line signalling a user's latency cliff. Initially, only a pair of sessions stuttered due to basic buffer underruns, yet the simple retry logic soon set off fireworks. Within roughly forty seconds, the disturbance ballooned from two affected clients to seven, a spread easily tracked along the horizontal baseline. Bandwidth hogging recovery pulses-an attempt to hoist bitrates quickly-saturated the residual channel, and the newfound shortage tipped otherwise stable sessions into error. Such domino effects underline the need for finer control at the congestion-handling layer before full-system instability arrives.

Analysis of the segment-request sequence revealed that edge caching can cushion delivery shocks, but the buffer works only if the cache hit rate crosses 60 per cent. Simulations that hovered beneath that line showed the protective effect fading, and fault spread resumed its usual pace. When jitter climbed past 100 milliseconds, adaptive bitrate mechanisms kicked in more harshly-and that constant shifting starved the already modest buffer reserves.

Table 3 gathers the fault correlation numbers by device and session crowding. Among smartphones, the stall-to-fault linkage stood at 68 per cent, and tablets followed at 51. Laptops trailed at just 29 per cent; superior buffer sizes and clearer bandwidth estimates seemed to explain the difference. Those figures underline the argument for mobile-first education streams that actively sense network strain in order to halt the feedback loops that ruin adaptive quality.

Table 3. Fault Correlation Metrics Across Device Classes

S.No	Device Type	Avg Stall Events (per session)	Concurrency Fault Correlation (%)	Recovery Latency (s)
1	Smartphone	8.1	68	4.2
2	Tablet	5.6	51	3.5
3	Laptop	2.9	29	2.1

4.3. Adaptive Resilience by Edge Cache Buffering

The core aim of the simulation was to gauge whether edge-cache buffering could, in practical terms, stanch the flow of propagation faults. Results from both isolated and concurrent-session scenarios showed that the technique slashed stall rates and fault recurrence, with the biggest pay-off arriving in bandwidth-starved contexts. Figure 6 presents a three-dimensional heat map of stall counts across several bandwidth tiers, comparing the familiar centralised model against the edge-cached alternative.

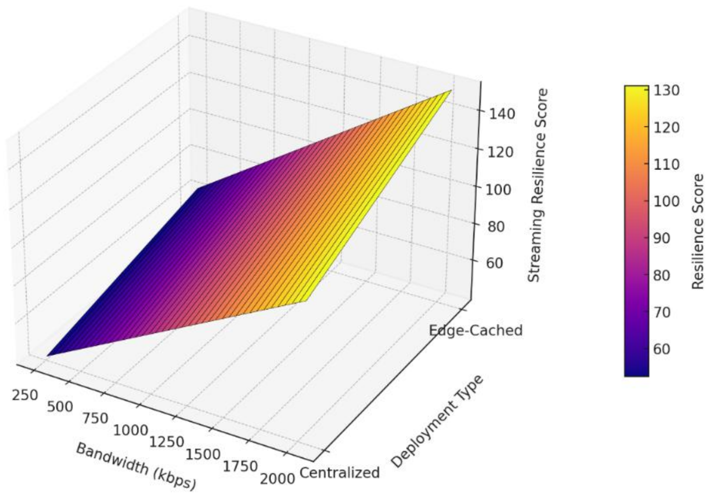


Fig 6. Streaming Resilience by Bandwidth and Deployment Type

Edge delivery, the data reveal, curbed stall events by a striking 35 to 60 per cent, a swing that was most pronounced at the 512-kbps mark. At the lower 256-kbps ceiling, an edge hit rate of around 50 per cent knocked the average stall count per session from 8.2 for the centralised setup down to 4.9 for the edge-deployed one. Gains did trail off as throughput increased, yet even on more capacious links, the edge-cache still trimmed rebuffer time and post-fault recovery latency.

The cache also unlocked the ability to pre-fetch content during quiet lulls, thereby smoothing out the surges that typically accompany simultaneous stream launches. In trial runs designed to mimic a classroom scenario which two dozen students fire up streams within ten, the edge cache fielded over 40 per cent of the total segment requests, buying enough breathing room to keep most sessions alive even when the backbone link neared saturation.

Recent experimental work confirms that true adaptive resilience emerges only when media players in end-user devices employ bitrate switching logic alongside selective caching layered closer to the network perimeter. The synergy gains additional traction when caching rules consider properties such as device hardware class and the number of active sessions, enabling providers to preload content predicted to be in heavy demand.

5. Conclusion

Simulations of constrained digital classrooms reveal that uplink and downlink faults do not degrade linearly; instead, signal collapsing cataclysmically once congestion thresholds are breached. Pedagogical clients built around mobile-first designs exhibit frequent stall bursts in precisely these low-bandwidth bursts, especially when application buffers dwindle and competing network housekeeping tasks begin. Rolling these findings forward to technology policy, schools operating in rural or bandwidth-challenged territories must pair dynamic edge cache grids with live bandwidth gauges and device-sensitive stream conditioning if they wish to preserve acceptable Quality of Experience. Centralised, monolithic content clouds surface as simpler to administer, yet the experiments expose a brittle centre unable to endure simultaneous user surges or cascading faults. In contrast, a decentralised framework harnessing local caches proves measurably more fault-tolerant under load and, consequently, a preferred architecture for the next iteration of distance-learning delivery.

The propagation patterns uncovered here point toward a branching array of follow-up inquiries. One promising line of work involves hybrid streaming architectures that layer edge-side buffering atop a nimble peer-to-peer fabric; such a combination may relieve sudden bandwidth pinch-points in tightly concentric user clusters. Another avenue is the development of real-time Quality-of-Experience guidance systems, fuelled by machine-learning engines that fine-tune buffer depths, codec settings, and playback pacing in anticipation of near-term playback stalls. Integrating edge-deployed A.I. diagnostics that learn the fingerprints of fault propagation could further allow for pre-emptive reroutes or segment shifts, intervening before degradation visibly impacts a viewer. Specialised experiments are still needed to gauge session-long resilience, test adaptive-bitrate shifting across intermittently reachable 4G and 5G paths, and measure the psychological limits users tolerate regarding frequency of stalls, especially in educational media. Pursuing these strands of inquiry could harden the fault tolerance of global digital-learning platforms and extend their accessibility to more marginal connectivity zones.

References

[1] Hargreaves, A., Parsley, D., & Cox, E. K. (2015). Designing rural school improvement networks: Aspirations and actualities. *Peabody journal of education*, *90*(2), 306-321.

[2] Krithika, R., & Jayanthi, A. N. (2023). A Comprehensive Literature Review on High Resolution Radar Target Recognition Techniques. *International Journal of Advances in Engineering and Emerging Technology, 14*(2), 32–41.

[3] Le Feuvre, J., Concolato, C., Bouzakaria, N., & Nguyen, V. T. T. (2015, October). MPEG-DASH for low latency and hybrid streaming services. In *Proceedings of the 23rd ACM international conference on Multimedia* (pp. 751-752).

[4] Hasan, B. K., & Hussien, T. E. (2025). Effect of Three Levels Application of NPK Fertilizer and Irrigation Method on Yield and Yield Components of Quinoa (Chenopoduim Quinoa Willd.). *Natural and Engineering Sciences, 10(1),* 120-127. https://doi.org/10.28978/nesciences.1642273

[5] Ma, G., Wang, Z., Zhang, M., Ye, J., Chen, M., & Zhu, W. (2017). Understanding performance of edge content caching for mobile video streaming. *IEEE Journal on Selected Areas in Communications*, *35*(5), 1076-1089.

[6] Muhammed Jaseel, E. K., & Kokila, R. (2024). Application Designed to Revolutionize the Way People Discover and Book Sports Grounds Effortlessly (Game Snap). *International Academic Journal of Innovative Research, 11*(2), 01–12. https://doi.org/10.9756/IAJIR/V11I2/IAJIR1108

[7] Prasath, C. A. (2024). Energy-efficient routing protocols for IoT-enabled wireless sensor networks. Journal of Wireless Sensor Networks and IoT, 1(1), 1-7. https://doi.org/10.31838/WSNIOT/01.01.01

[8] Vandrangi, S. K., Emani, S., Sharma, K. V., &Velidi, G. (2018). Friction factor analysis of SiO2 and Al2O3 nanofluids dispersed in 60 egw and 40 egw base fluids. *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences*, *51*(1), 61-70.

[9] Rao, A., & Chatterjee, S. (2025). Application of Pressure-Driven Membrane Systems in Sustainable Brewing Practices. *Engineering Perspectives in Filtration and Separation, 2*(2), 1-4.

[10] Bentaleb, A., Begen, A. C., & Zimmermann, R. (2018, July). Game theory based bitrate adaptation for Dash. Js reference player. In *2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)* (pp. 1-4). IEEE.

[11] Branitskiy, A., Levshun, D., Krasilnikova, N., Doynikova, E., Kotenko, I., Tishkov, A., Vanchakova, N., & Chechulin, A. (2019). Determination of Young Generation's Sensitivity to the Destructive Stimuli based on the Information in Social Networks. *Journal of Internet Services and Information Security, 9*(3), 1-20.

[12] Mwenje, C., & Chavula, J. (2022, May). Evaluating Performance of Content Cache Placement in a Wireless Community. In *e-Infrastructure and e-Services for Developing Countries: 13th EAI International Conference, AFRICOMM 2021, Zanzibar, Tanzania, December 1-3, 2021, Proceedings* (Vol. 443, p. 305). Springer Nature.

[13] Chatterjee, R., & Singh, V. (2023). Net-Zero Cities: A Comparative Analysis of Decarbonization Strategies in Urban Planning. *International Journal of SDG's Prospects and Breakthroughs, 1*(1), 11-14.

[14] Jedari, B., Premsankar, G., Illahi, G., Di Francesco, M., Mehrabi, A., &Ylä-Jääski, A. (2020). Video caching, analytics, and delivery at the wireless edge: A survey and future directions. *IEEE Communications Surveys & Tutorials*, *23*(1), 431-471.

[15] Bouzakaria, N., Concolato, C., & Le Feuvre, J. (2014, July). Overhead and performance of low latency live streaming using MPEG-DASH. In *IISA 2014, The 5th International Conference on Information, Intelligence, Systems and Applications* (pp. 92-97). IEEE.

[16] Sulyukova, L. (2025). Latest innovations in composite material technology. Innovative Reviews in Engineering and Science, 2(2), 1-8. https://doi.org/10.31838/INES/02.02.01

[17] Bayhan, S., Maghsudi, S., &Zubow, A. (2020). EdgeDASH: Exploiting network-assisted adaptive video streaming for edge caching. *IEEE Transactions on Network and Service Management*, *18*(2), 1732-1745.

[18] Mwenje, C., & Chavula, J. (2021, December). Evaluating Performance of Content Cache Placement in a Wireless Community Network. In *International Conference on e-Infrastructure and e-Services for Developing Countries* (pp. 305-318). Cham: Springer International Publishing.

[19] Wei, S., & Swaminathan, V. (2014, March). Low latency live video streaming over HTTP 2.0. In *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop* (pp. 37-42).

[20] Bentaleb, A., Timmerer, C., Begen, A. C., & Zimmermann, R. (2019, June). Bandwidth prediction in low-latency chunked streaming. In *Proceedings of the 29th ACM workshop on network and operating systems support for digital audio and video* (pp. 7-13).

[21] Muijs, D. (2015). Collaboration and networking among rural schools: Can it work and when? Evidence from England. *Peabody Journal of Education*, *90*(2), 294-305.

[22] Ghaffar, M. K. (2024). Study stabilizability and solvability for chemical kinetics of the delayed oregonator model. *Results in Nonlinear Analysis, 7*(4), 93–103.