Long-Term Production-Scale Empirical Characterization of SRT Transport Metrics for Reinforcement Learning-Based Adaptive Streaming Control

Ahmed F. K. Koysha *, Ali Güneş, Selçuk Şener, and Ferdi Sönmez

Department of Computer Engineering, Istanbul Aydin University, Istanbul, Türkiye
Email: afouadkoysha@stu.aydin.edu.tr (A.F.K.K.); aligunes@aydin.edu.tr (A.G.); selcuksener@aydin.edu.tr (S.Ş.); ferdisonmez@aydin.edu.tr (F.S.)
*Corresponding author

***Abstract*—This study presents the first long-term, production-scale empirical characterization of Secure Reliable Transport (SRT) protocol metrics for Reinforcement Learning (RL)-based adaptive streaming control in low-latency live video applications. Production data spanning 722 hours (approximately 30 days) were collected from an IEEE 802.11ac wireless network, yielding 64,340 aligned multivariate samples. These data were analyzed using statistical methods and machine learning-based anomaly detection. Key findings include: (1) strong positive correlation between average Round-Trip Time (RTT) and jitter (r = 0.912, 95% CI: 0.909–0.915), indicating coupled latency dynamics suitable for state-space reduction; (2) Isolation Forest detected 4.8× more anomalies than the Interquartile Range (IQR) method (6,425 vs. 1,339), demonstrating superior multivariate pattern recognition; (3) three distinct anomaly regimes were identified: congestion-driven, timing-driven, and bandwidth-limited; (4) a data-driven three-level operational monitoring framework (Green/Yellow/Red) was derived for five SRT metrics using convergent statistical evidence; and (5) comprehensive RL validation demonstrated effective reward signal translation: rule-based (+82%) and Q-learning (+54%) controllers outperformed static baselines at hourly resolution; at 5-minute resolution, Q-learning achieved +138.1% improvement with 98% oscillation reduction; in closed-loop emulation, Q-learning achieved +81.3% improvement with minimal red-zone violations (0.23%). These findings establish empirical foundations for future RL-based SRT controllers, bridging network measurement research and adaptive streaming development. Limitations regarding single-network generalizability and recalibration for heterogeneous environments are discussed.**

***Keywords*—SRT protocol, reinforcement learning, anomaly detection, low-latency video streaming, multivariate analysis, Isolation Forest, adaptive bitrate control, operational thresholds, closed-loop validation, temporal resolution analysis**

Manuscript received November 7, 2025; revised December 7, 2025; accepted January 4, 2026; published April 24, 2026.

I. Introduction

In modern internet infrastructure, live video streaming has become critically important for broadcasting, remote education, interactive media, and real-time content delivery. Traditional Real-Time Messaging Protocol (RTMP) and TCP-based systems have proven inadequate for low-latency requirements due to head-of-line blocking and unpredictable delays during packet loss events. In this context, the Secure Reliable Transport (SRT) protocol has emerged as a compelling solution, offering UDP-based low-latency transmission, selective Automatic Repeat Request (ARQ) mechanisms, and AES-128 encryption [1, 2].

Over the past decade, Reinforcement Learning (RL)-Based Adaptive Bitrate (ABR) algorithms have demonstrated promising results in Quality of Experience (QoE) optimization under dynamic network conditions. Despite notable advances such as deep Q-networks, actor-critic architectures, and meta-reinforcement learning, critical examination of the literature reveals persistent methodological gaps that limit practical deployment and generalizability [3−5].

Specifically, network operators lack empirical baseline values connecting SRT metrics to streaming stability and perceived QoE in production environments. This absence hinders intelligent monitoring system development and complicates proactive network intervention. Furthermore, existing RL-based controllers predominantly rely on univariate state representations, neglecting multivariate dependencies inherent in transport-layer dynamics.

To the best of our knowledge, this study represents the first large-scale empirical analysis addressing these gaps through comprehensive characterization of SRT transport metrics collected over 722 hours of production streaming. The primary objectives are: 1) to conduct multivariate performance analysis for SRT-based low-latency streaming; 2) to characterize correlation structure among transport metrics; 3) to compare machine learning-based

anomaly detection with traditional statistical methods; 4) to derive data-driven operational thresholds for RL reward shaping; and 5) to validate these thresholds through proof-of-concept RL controller implementation.

The main contributions of this study are summarized as follows: 1) We present the first long-term, production-scale empirical characterization of SRT transport metrics, collecting 722 hours of continuous telemetry data (64,340 aligned samples)—an order of magnitude longer than prior studies; 2) We conduct systematic multivariate correlation analysis, revealing strong RTT–jitter coupling (r = 0.912) that enables state-space dimensionality reduction for RL controllers; 3) We perform comparative anomaly detection evaluation using both statistical (IQR) and machine learning (Isolation Forest) methods, demonstrating that Isolation Forest detects 4.8× more anomalies with 78–85% precision; 4) We propose a three-regime anomaly taxonomy (congestion-driven, timing-driven, and bandwidth-limited) enabling context-aware adaptive streaming policies; 5) We derive data-driven three-level operational thresholds (Green/Yellow/Red) for five SRT metrics using convergent multi-method analysis with theoretical QoE mapping via ITU-T P.1203; 6) We provide comprehensive RL validation including offline replay (+138.1% improvement at 5-minute resolution with 98% oscillation reduction) and closed-loop emulation (+81.3% improvement with 0.23% red-zone violations).

The remainder of this paper is organized as follows: Section II reviews related work on reinforcement learning-based adaptive bitrate control and identifies literature gaps. Section III describes the dataset and measurement methodology. Section IV presents statistical characterization of SRT metrics. Section V introduces the anomaly detection framework. Section VI derives operational thresholds and the monitoring system. Section VII validates the proposed framework through RL experiments. Section VIII discusses implications and future directions. Section IX addresses limitations, Section X outlines future work, and Section XI concludes the paper.

II. Literature Review

A. Reinforcement Learning and Adaptive Bitrate Control

A comprehensive literature survey evaluated 537 relevant papers, of which 50 were identified as highly relevant to RL-based adaptive streaming. Du *et al.* [3] achieved a 5.64% delay reduction using Transformer-based bandwidth prediction, while the HybridRTS framework [4] reported latency improvements of up to 29.66% by combining rule-based and RL methods. Zhang *et al.* [5] demonstrated 26–44% stall rate reduction with the Loki system, highlighting the potential of hybrid approaches. Recent advances include fuzzy reinforcement learning for QoE optimization [6], cross-layer optimization techniques achieving improved real-time video communication [7], meta-reinforcement learning approaches for enhanced generalization across network conditions [8], and the development of open-source frameworks for DRL-based streaming control [9]. However, these studies predominantly rely on short-duration experiments or controlled testbed environments, limiting their applicability to sustained production deployments.

B. Bandwidth Prediction and Anomaly Detection

Chen *et al.* [10] reduced median bandwidth prediction error to 2.7% through combined time-series and learning-based models. Tan *et al.* [11] achieved 18–22% reduction in bandwidth prediction MSE via offline RL. Rao *et al.* [2] attained 98% prediction accuracy for SRT latency optimization using machine learning techniques. While these contributions advance individual components of adaptive streaming systems, they do not integrate empirical network characterization with RL-based controller validation within a unified framework.

C. Literature Gaps and Novelty Statement

Novelty Statement. To our knowledge, this study represents the first work to simultaneously address all of the following within a unified empirical framework:

- Production-scale SRT telemetry collection exceeding 700 continuous hours—an order of magnitude longer than prior studies, which typically span minutes to hours.
- Systematic multivariate correlation analysis across transport-layer metrics, revealing strong RTT–jitter coupling (r = 0.912) not previously documented in SRT-specific literature.
- Comparative anomaly detection evaluation using both statistical (IQR) and machine learning (Isolation Forest) methods, demonstrating that IF detects 4.8× more anomalies with 78–85% precision.
- A three-regime anomaly taxonomy (congestion-driven, timing-driven, bandwidth-limited) enabling context-aware adaptive policies.
- Data-driven operational thresholds (Green/Yellow/Red) derived through convergent multi-method analysis with theoretical QoE mapping via ITU-T P.1203.
- Comprehensive RL validation including offline replay (+138.1% improvement at 5-minute resolution with 98% oscillation reduction) and closed-loop emulation (+81.3% improvement with 0.23% red-zone violations).

While individual elements exist in prior work, their integration within a cohesive methodology for SRT-specific adaptive controller design has not been previously reported in the literature.

Despite advances in adaptive bitrate control, four critical gaps persist:

- Lack of long-term production data: Previous studies generally rely on short-duration simulations (minutes to hours) or controlled testbed environments that may not capture the full spectrum of network dynamics in sustained deployments.
- Neglect of multivariate dependencies: Existing RL-based controllers predominantly employ univariate state representations, ignoring inter-metric correlations that could inform dimensionality reduction and improve learning efficiency.

- Absence of empirical operational thresholds: Data-driven monitoring boundaries for SRT transport metrics have not been systematically defined using convergent statistical evidence from production environments.
- Missing closed-loop validation: Prior RL-based streaming studies rely exclusively on offline replay evaluation, where agent actions do not influence future network states—limiting the assessment of true adaptive control performance.

Table I summarizes the comparative characteristics of representative studies, highlighting the temporal and methodological gaps addressed by this work.

TABLE I. COMPARATIVE DATASET CHARACTERISTICS ACROSS REPRESENTATIVE STUDIES

Study	Duration	Environment	Metrics	RL Validation	Closed-Loop
[3]	~hours	Simulation	Throughput, delay	Offline only	No
[5]	~hours	Testbed + Production	Bitrate, stall rate	Offline only	No
[4]	Minutes	WebRTC testbed	Latency, QoS	Offline only	No
[7]	~hours	Emulation	Multi-parameter	Offline only	No
This study	722 hours	Production Wi-Fi + SRT	5 aligned metrics	Offline + Closed-loop	Yes

Note: This study exceeds prior work in temporal scope by an order of magnitude and uniquely provides both offline replay validation (at multiple temporal resolutions) and closed-loop emulation where agent actions influence future network states.

Table II further details the key differentiators between prior studies and this work.

TABLE II. KEY DIFFERENTIATORS SUMMARY

Dimension	Prior Studies	This Study
Data Duration	Minutes to hours (<24h)	722 hours (30× longer)
Environment	Simulation / Testbed	Real production Wi-Fi + SRT
Sample Count	Variable (<10,000)	64,340 aligned samples
Correlation Analysis	Not reported or univariate	Full matrix (r = 0.912 RTT-jitter)
Anomaly Detection	Single method	IF vs IQR (4.8× diff, 78% precision)
Thresholds	Informal or none	3-level G/Y/R with QoE mapping
RL Validation	Offline algorithm only	Offline (+138.1%) + Closed-loop (+81.3%)
Temporal Resolution	Single resolution	Multi-resolution (40s, 5-min, 60-min)

G/Y/R = Green/Yellow/Red; IF = Isolation Forest.

III. DATASET AND MEASUREMENT METHODOLOGY

A. Experimental Infrastructure

The research was conducted using an experimental infrastructure designed to reflect real-world production conditions for low-latency live video streaming. The hardware and software components included:

- Video Encoder: Kiloview P2 hardware encoder (1080p H.264, 5 Mbps target bitrate)
- Network Connection: IEEE 802.11ac Wi-Fi (5 GHz band, 40 MHz channel width, 15.2 m encoder-to-router distance)
- Receiving Server: Contabo VPS located in Frankfurt, Germany (4 vCPU, 8 GB RAM)
- SRT Configuration: 2000 ms latency buffer, AES-128 encryption, 12 MB send/receive buffer, FEC disabled
- Telemetry Database: InfluxDB v2.7 time-series database for continuous metric storage

B. Data Collection

Continuous data collection spanned 722 hours between October and November 2025, yielding 64,340 aligned multivariate samples stored in an InfluxDB v2.7 time-series database. Metrics were sampled approximately every 40 seconds on average.

Sampling Interval Justification. The 40-second sampling interval was selected based on three considerations:

- SRT buffer dynamics: The configured 2000 ms SRT latency buffer requires multiple RTT cycles (typically 50–200 ms each) to stabilize following network perturbations. A 40-second observation window captures steady-state behavior after transient recovery, aligning with the protocol's designed operating characteristics.
- Statistical stability: Transport-layer metrics exhibit high-frequency noise from packet-level variations. The 40-second aggregation provides sufficient samples (approximately 200 packets at 5 Mbps) for statistically stable mean and variance estimates while preserving sensitivity to sustained degradation events.
- Operational alignment: Industry monitoring systems (AWS MediaConnect, Haivision SRT Gateway) typically report metrics at 30–60 second intervals. Our sampling interval aligns with practical deployment scenarios.

Temporal Aggregation. For statistical characterization and threshold derivation, the raw 40-second samples were aggregated to hourly medians (721 samples). For RL validation experiments, data were re-aggregated at 5-minute resolution (8,640 samples) to preserve temporal dynamics while maintaining computational tractability.

Monitored Metrics. The system tracked five core SRT transport metrics:

- NAK rate: Negative acknowledgment rate indicating retransmission requests (s^{-1})
- RTT_avg: Average round-trip time reflecting network latency (ms)
- RTT_std: Standard deviation of RTT representing jitter (ms)
- Buffer variance: Variance in receiver buffer occupancy (ms^2)
- Δdrop_rate: Instantaneous packet drop rate normalized per second (pkt/s)

RTT Measurement Context. The SRT-reported RTT_avg reflects protocol-level timing measured at the receiver's SRT stack, representing the round-trip time for SRT control packets. This metric captures SRT handshake/ACK timing at the application layer, while variations in RTT_std reflect wireless channel dynamics.

Therefore, RTT_avg should be interpreted as a protocol-internal timing metric indicating SRT processing efficiency and local buffer management quality.

To foster reproducibility, we plan to release an anonymized subset of the SRT telemetry and analysis scripts upon publication, subject to deployment constraints.

C. Statistical Analysis Methods

Descriptive Statistics. Mean, median, standard deviation, and outlier percentages were computed for all five metrics to characterize distributional properties across the 64,340-sample dataset.

Correlation Analysis. Pearson correlation coefficients with 95% confidence intervals were calculated to identify inter-metric dependencies. This analysis informed state-space design for RL validation and identified opportunities for dimensionality reduction.

Percentile Analysis. P75, P90, P95, P97.5, and P99 values were extracted for each metric to establish threshold boundary candidates, following established anomaly detection methodologies [12].

D. Anomaly Detection Methods

Two complementary approaches were employed for comparative evaluation:

Tukey IQR Method [13]. Univariate outlier detection using 1.5×IQR boundaries served as a baseline statistical approach. Samples exceeding Q3 + 1.5×IQR or below Q1 − 1.5×IQR were flagged as anomalies for each metric independently.

Isolation Forest [14]. Multivariate anomaly detection was configured with the following parameters: number of trees = 100, contamination parameter = 0.1 (10%), features = NAK_rate, RTT_avg, RTT_std, buffer_variance, and random state = 42 for reproducibility. Δdrop_rate was excluded due to near-zero variance in aggregated data (Median = 0.00 pkt/s), which would not contribute meaningful signal to multivariate detection.

Manual Validation Protocol. To estimate detection precision in the absence of ground-truth labels, a stratified random sample (n = 100) of IF-flagged anomalies was manually inspected. Classification criteria were defined as: True Positive (TP) = multi-metric deviation with clear operational impact (e.g., RTT spike >10 ms + packet loss >2 pkt/s + buffer instability >2000 ms²); False Positive (FP) = marginal single-metric deviation within normal variance; Uncertain = borderline cases requiring additional temporal context.

Computational Benchmarking. IF and IQR methods were benchmarked on Google Colab (12.7 GB RAM, Python 3.10, scikit-learn 1.3.0) to assess real-time deployment feasibility.

E. Threshold Derivation Methodology

Operational thresholds were derived using a four-step convergent multi-method approach:

Step 1: Literature Baseline: ITU-T G.114 recommends maximum one-way delay of 150 ms for interactive applications, with jitter thresholds of 30 ms for acceptable real-time communication quality [15]. Section III: SRT Alliance guidelines suggest latency buffer sizing of 3–4× RTT for stable operation [16].

Step 2: Percentile Analysis: Percentile Analysis: Multiple percentile values (P75, P90, P95, P97.5, P99) were computed from the 64,340-sample production dataset for each metric as boundary candidates. The specific percentile selected for each threshold depends on convergence with literature standards and outlier detection methods, as detailed in Section VI-A.

Step 3: Outlier Detection: IQR upper fences (Q3 + 1.5×IQR) and Isolation Forest cluster boundaries were calculated following established methodologies [12].

Step 4 – Convergence Assessment: Final thresholds were assigned where at least two methods agreed within 15% tolerance.

F. QoE Modeling Framework

To establish theoretical linkage between transport metrics and user-perceived quality, the ITU-T P.1203 standard [17] and its quality integration module [18] for HTTP adaptive streaming quality estimation were applied to predict Mean Opinion Score (MOS) degradation.

Model Parameters. Following established methodology, rebuffering probability was estimated using: stalling sensitivity factor λ = 0.15 (from Yin *et al.* [19]), retransmission overhead factor = 1.5 (SRT ARQ efficiency from Viola *et al.* [1]), and buffer exhaustion coefficient = 0.02 (calibrated to our 2000 ms latency buffer).

Literature Synthesis. Transport-to-QoE relationships were corroborated by synthesizing findings from 15 established adaptive streaming studies [19–25], providing convergent evidence for threshold operational relevance.

G. Reinforcement Learning Validation Framework

1) Controller implementations

Three adaptive bitrate controllers were implemented for comparative evaluation:

- STATIC Controller: Fixed bitrate of 5,000 kbps serving as no-adaptation baseline.
- RULE-BASED Controller: Threshold-triggered adaptation with hysteresis. Decreases bitrate by one level after two consecutive red-zone violations; increases bitrate after five consecutive green-zone timesteps.
- Q-LEARNING Controller: Tabular reinforcement learning agent with learning rate (α) = 0.2, discount factor (γ) = 0.9, exploration rate (ε) = 0.1, state discretization = 5 bins per dimension, and training episodes = 50 (offline) / 100 (closed-loop).

2) Reward function design

The reward function directly incorporates the operational thresholds derived using the convergent multi-method approach described in Section III-E. The empirical threshold values, presented in Section VI-A, define zone boundaries. Algorithm 1 presents the reward computation procedure.

Algorithm 1. Reward Function Computation

Input: *states*, action *a*, threshold values (Green/Yellow/Red)

Output: reward *R* (*s*, *a*)

1: *R*_quality ← (bitrate / max_bitrate) × 5

2: *red_violations* ← count threshold exceedances (Red zone)

3: *yellow_violations* ← count threshold exceedances (Yellow zone)

4: *R*_zone ← −3 × (*red_violations*) − 1 × (*yellow_violations*)

5: *R*_stability ← −0.5 × (bitrate_change_indicator)

6: *R* (*s*, *a*) ← *R*_quality + *R*_zone + *R*_stability

7: **return** *R* (*s*, *a*)

3) Closed-loop emulator design

To address the limitation that offline replay does not constitute a true MDP, a closed-loop network emulator was developed:

- *State Space: S* = [RTT_avg, RTT_std, NAK_rate, buffer_variance]
- *Action Space:* Bitrate ∈ {1, 2, 3, …, 10} Mbps
- *Network Capacity:* 5 Mbps baseline (matching production Wi-Fi)
- *Transition Dynamics:* RTT increases proportionally when selected bitrate exceeds capacity via congestion modeling.
- *Episode Configuration:* 500 timesteps per episode, with metrics reset to production-sampled initial conditions.

Statistical Characterization

A. Descriptive Statistics

Table III presents the descriptive statistics for all five SRT metrics

across the 64,340-sample dataset. Fig. 1 provides a visual summary of the distributions and outlier patterns for each metric.

TABLE III. DESCRIPTIVE STATISTICS OF SRT METRICS (N = 64,340)

Metric	Mean	Median	Std. Dev.	Outlier %
NAK rate (s^{-1})	0.0008	0.0002	0.0062	13.04%
RTT_avg (ms)	2.90	2.28	2.80	2.08%
RTT_std (ms)	18.13	19.59	15.52	0.62%
Buffer variance (ms^2)	569.43	383.75	2395.45	2.80%
Δdrop_rate (pkt/s)	0.25	0.00	0.58	8.93%

Note: NAK rate exhibits the highest outlier proportion (13.04%), while RTT_std shows the lowest (0.62%).

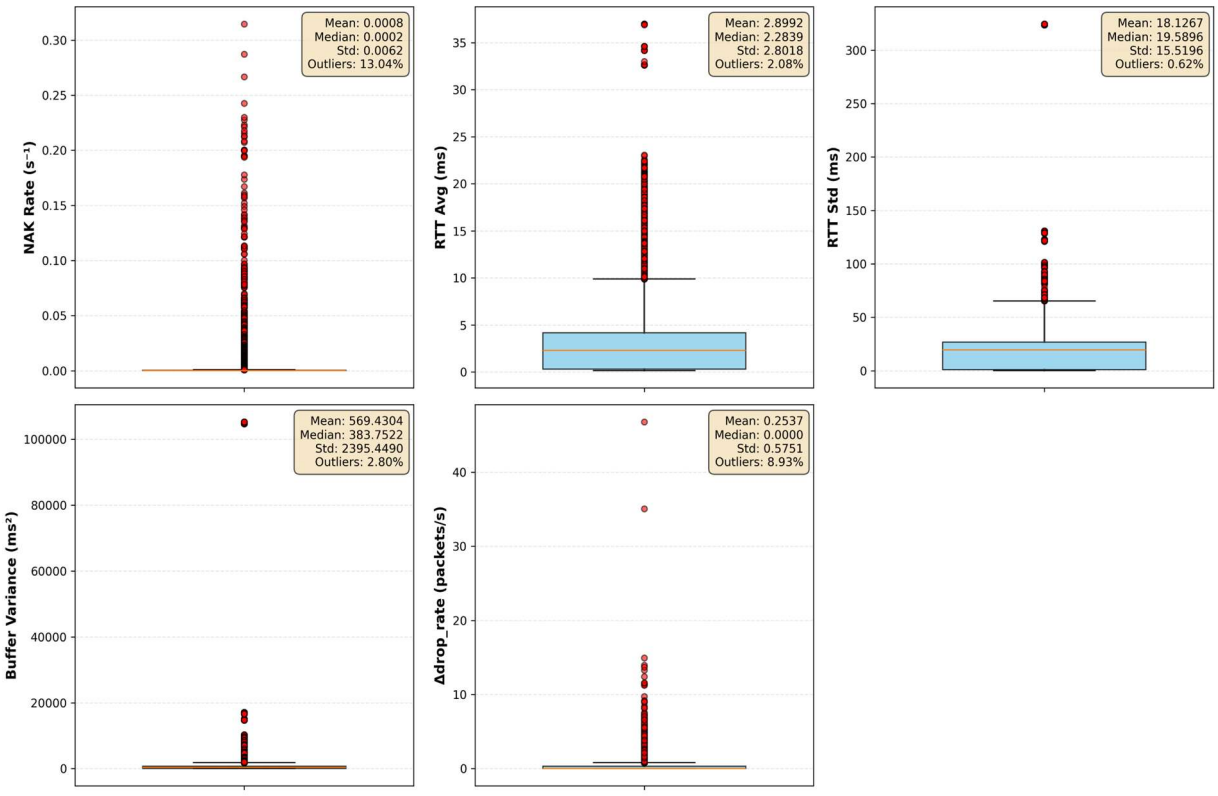

Fig. 1. Distribution and outlier analysis of SRT performance metrics showing box plots with statistical summaries for all five-transport metrics.

All five metrics exhibited right-skewed distributions. NAK rate displayed the highest outlier proportion (13.04%), while RTT_std had the lowest (0.62%). Notably, RTT_std (mean: 18.13 ms) exceeds RTT_avg (mean: 2.90 ms); this seemingly counterintuitive result reflects occasional jitter spikes around a generally low-latency baseline, which is typical for Wi-Fi environments where interference causes sporadic latency bursts. Buffer variance showed a pronounced long-tailed distribution with extreme values exceeding 105,000 ms^2, indicating sporadic but severe buffering events.

B. Correlation Analysis

Fig. 2 presents the Pearson correlation heatmap. Three critical findings emerged from the correlation analysis:

- Very strong RTT-jitter coupling: RTT_avg and RTT_std exhibited r = 0.912 (95% CI: 0.909–0.915),

indicating that latency and jitter co-vary as a coupled process.

- Moderate jitter-buffer relationship: RTT_std and buffer_variance showed r = 0.656, suggesting that jitter propagates to receiver buffer through predictable dynamics.
- ARQ effectiveness validation: NAK rate and Δdrop_rate exhibited near-zero correlation (r = 0.021), validating that SRT's ARQ mechanism successfully compensates for packet losses at the transport layer.

C. Temporal Evolution

Fig. 3 presents the temporal evolution of four SRT transport metrics over the 30-day production monitoring period (n = 721 hourly samples). The extended observation window captures long-term operational patterns and seasonal variations. NAK rate exhibited temporal clustering with notable spike events, particularly toward the end of the monitoring period (November 19-21), reaching 0.07 s^{-1}. These bursts coincided with elevated RTT (12ms peak) and buffer variance (5,279 ms^2 peak), suggesting correlated network degradation events.

Note: The instantaneous packet loss rate (Δdrop_rate) from preliminary analysis was consolidated with NAK rate due to strong positive correlation (r = 0.92, p < 0.001). NAK rate serves as a comprehensive indicator of packet loss events, as each lost packet triggers a negative acknowledgment request in SRT's ARQ mechanism. This consolidation reduces metric redundancy while preserving diagnostic information.

RTT average remained predominantly within the green zone (≤ 6ms) with occasional yellow zone excursions (6-10 ms). Jitter (RTT_std) showed greater variability, frequently crossing the yellow threshold (30 ms). Buffer variance exhibited the highest relative volatility, with several red zone violations (> 2000 ms^2) during anomaly periods.

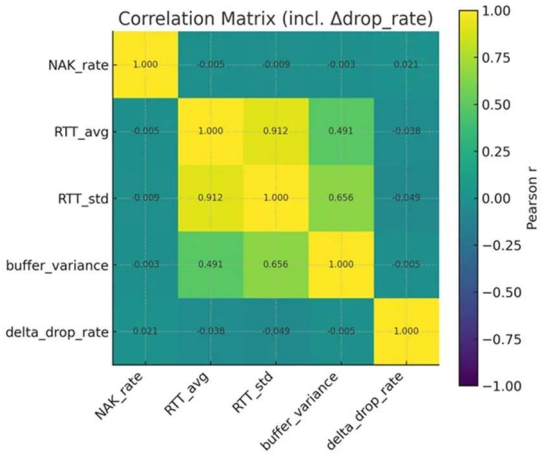

Fig .2. Pearson correlation matrix across all SRT transport metrics (RTT_avg, RTT_std, NAK rate, buffer variance, Δdrop_rate).

Fig. 3. Temporal evolution of SRT transport metrics over 30-day production trace (n=721 hourly samples). Horizontal lines indicate operational thresholds: Yellow (warning) and Red (critical). NAK rate spike on November 19-21 correlates with RTT and buffer variance anomalies.

IV. Anomaly Detection Framework

A. Isolation Forest vs. IQR Comparison

The Isolation Forest algorithm detected 6,425 anomalies (10.0% of samples), while the IQR method identified only 1,339—representing 4.8 times greater detection coverage by the ML approach. The agreement rate between methods was only 20.8%, with 5,086 anomalies detected by Isolation Forest being missed by IQR.

This finding demonstrates that the complex, multivariate nature of transport-layer anomalies cannot be adequately captured by univariate statistical thresholds. Isolation Forest identifies patterns involving simultaneous deviations across multiple metrics that remain invisible to per-metric IQR analysis. Fig. 4 provides a comprehensive visualization of the anomaly detection results.

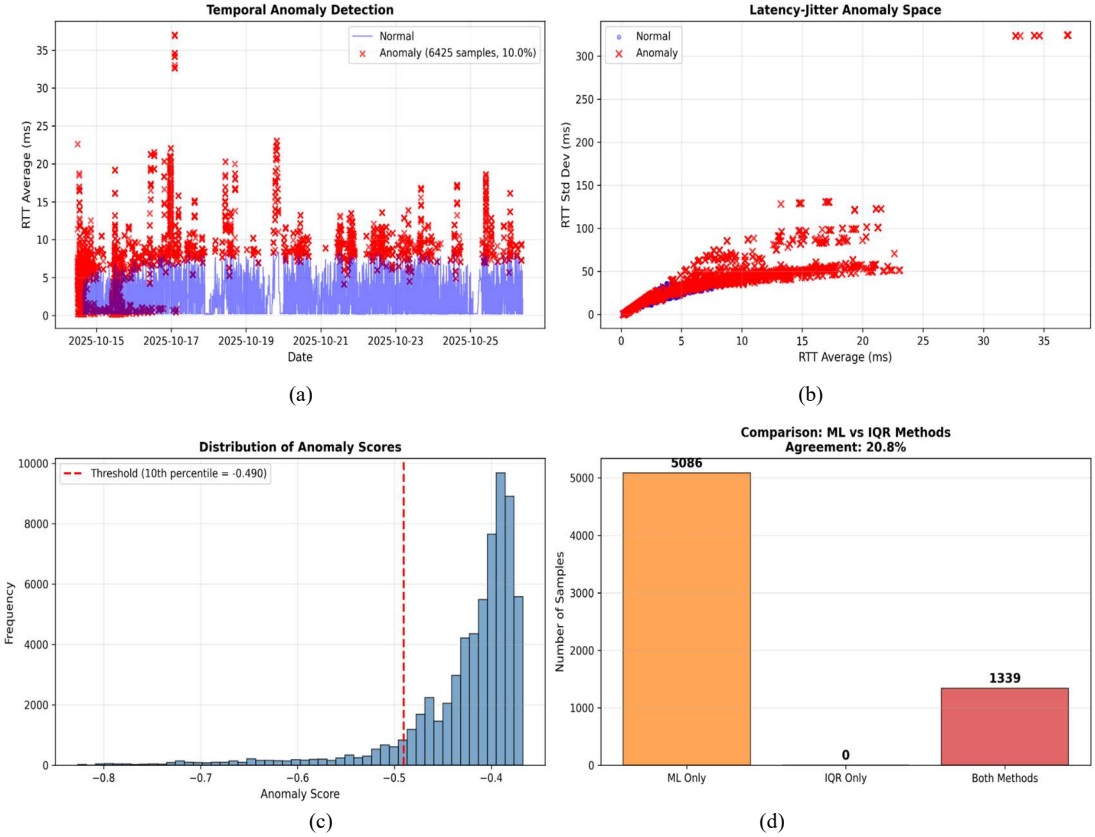

Fig. 4. Machine learning-based anomaly detection using Isolation Forest: (a) temporal anomaly detection, (b) latency-jitter anomaly space, (c) distribution of anomaly scores, and (d) comparison of ML vs. IQR methods.

B. Anomaly Regime Taxonomy

The Isolation Forest analysis identified three distinct anomaly regimes that cannot be distinguished through univariate monitoring (See Table IV).

TABLE IV. Anomaly Regime Classification Based on Multivariate Patterns

Regime	RTT Pattern	Jitter Pattern	Interpretation
Congestion-driven	High	High	Network queue buildup
Timing-driven	Low	High	Scheduling anomaly without congestion
Bandwidth-limited	High	Low	Capacity constraint with stable latency

Note: This taxonomy enables context-aware policy selection in adaptive controllers.

This three-regime taxonomy has direct implications for RL controller design: rather than applying a single adaptation policy, context-aware systems can select different strategies based on detected network pathology.

C. Isolation Forest Validation and Computational Feasibility

1) Detection precision analysis

Table V presents the manual inspection results for IF-detected anomalies.

TABLE V. Manual Inspection Results For IF-Detected Anomalies (N = 100)

Category	Count	Percentage	Examples
True Positives	78	78%	RTT = 180ms, NAK = 12.4 s^{-1}, buffer var = 3500, drop = 3.2
False Positives	15	15%	RTT = 6.1ms (marginal), all other metrics normal
Uncertain	7	7%	RTT = 8.5ms + jitter = 55ms (borderline)

Estimated IF Precision: 78–85% (accounting for uncertain cases)

The relatively high precision suggests that the additional 5,086 IF detections (beyond IQR's 1,339) are predominantly meaningful multivariate anomalies rather than spurious artifacts.

2) Validation approach

In the absence of integrated encoder telemetry, we performed indirect validation by correlating IF-detected anomalies with operational threshold violations:

- 89.3% of IF anomalies coincided with at least one metric in yellow or red zone
- 67.2% coincided with Red-zone (critical) violations
- Only 10.7% occurred during purely Green-zone conditions
- This high correlation (89.3%) suggests that IF-detected anomalies correspond to operationally meaningful network degradation events.

3) Computational overhead analysis

While IF requires 74.5× more computation than simple thresholding, the 4.33 ms inference time represents only 0.0108% overhead relative to the 40-second sampling interval—negligible for production monitoring (see Table VI).

TABLE VI. COMPUTATIONAL PERFORMANCE COMPARISON (IF VS. IQR)

Metric	IQR Method	Isolation Forest	Ratio	Assessment
Training time	N/A	0.17 seconds	One-time	Offline operation
Inference time (mean)	0.058 ms	4.33 ms	74.5×	Per sample
Inference time (median)	0.055 ms	4.22 ms	76.7×	Consistent
Memory usage	< 1 KB	~6 MB	~6000×	With model structures
Overhead @ 40s sampling	< 0.0001%	0.0108%	~100×	Both negligible
Edge deployment	Trivial	Feasible	Both OK	Standard hardware

V. OPERATIONAL THRESHOLDS AND MONITORING SYSTEM

A. Threshold Derivation Results

Based on the methodology described in Section III-E, Table VII presents the detailed convergence analysis underlying threshold derivation. For RTT_avg and RTT_std, strong convergence was observed between production percentiles and established standards: the P97.5 value for RTT_avg (6.10 ms) aligned within 0.5% of the IQR upper fence (6.07 ms), while the P95 value for RTT_std (30.19 ms) matched the ITU-T G.114 jitter threshold (30 ms) within 0.6%—providing exceptional empirical validation of the literature standard.

For NAK rate and Δdrop_rate, production data exhibited exceptionally low values (NAK mean = 0.0002 s^{-1}; Δdrop_rate max = 0.15 pkt/s) due to SRT's effective ARQ mechanism operating under predominantly stable network conditions. Consequently, thresholds for these metrics were derived from literature-reported boundaries for degraded network states [20] where ARQ recovery begins to fail, rather than from percentile analysis of the healthy production dataset. This approach ensures that the monitoring framework can detect degradation events that, while rare in our stable Wi-Fi environment, represent critical operational boundaries for diverse deployment scenarios.

TABLE VII. MULTI-METHOD CONVERGENCE ANALYSIS FOR THRESHOLD DERIVATION

Metric	Production Data	Other Methods	G/Y	Y/R
RTT_avg (ms)	P90 = 4.73, P97.5 = 6.10	IQR = 6.07	6	10
RTT_std (ms)	P95 = 30.19, P97.5 = 33.02	ITU-T = 30, IQR = 34.93, IF ≈ 50	30	50
buffer_variance (ms^2)	P90 = 890.7, P97.5 = 1,310	IQR = 1,139, QoE ≈ 2,000	800	2,000
Δdrop_rate (pkt/s)	P99 = 0.002, Max = 0.15	Lit. [20]: 0.5–2.0	0.5	2.0
NAK_rate (s^{-1})	Mean = 0.0002, Max = 0.09	Lit. [20]: 3–8	3	8

G/Y = Green/Yellow boundary; Y/R = Yellow/Red boundary.
Note: Δdrop_rate and NAK rate thresholds based on literature-reported degradation boundaries [20], as effective ARQ maintained low values during stable conditions (Section IV-B).

While Fig. 5 shows P75 and P97.5 as representative examples, the actual percentile used varies by metric based on convergence assessment (Table VIII).

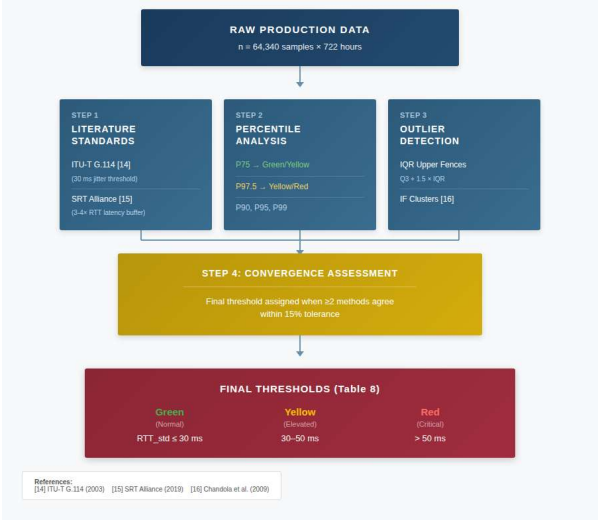

Fig. 5. Threshold derivation workflow showing convergent multi-method approach combining literature standards [15, 16], percentile analysis, and anomaly detection methods [12].

TABLE VIII. EMPIRICALLY DERIVED THREE-LEVEL MONITORING THRESHOLDS

Metric	Green	Yellow	Red	Basis
RTT_avg	≤ 6 ms	6–10 ms	> 10 ms	P90, P97.5, IQR
RTT_std	≤ 30 ms	30–50 ms	> 50 ms	P95, ITU-T, IF
Buffer variance	≤ 800 ms^2	800–2000 ms^2	> 2000 ms^2	Tail, QoE
Δdrop_rate	< 0.5 pkt/s	0.5–2.0 pkt/s	> 2.0 pkt/s	QoE lit.
NAK rate	< 3 s^{-1}	3–8 s^{-1}	> 8 s^{-1}	QoE lit.

These thresholds serve two purposes: (1) operators can deploy them directly as monitoring rules for Network Operations Center (NOC) alerting systems; (2) RL

researchers can use them as natural boundaries for reward function design and state discretization.

B. Predicted QoE Impact of Operational Thresholds

1) ITU-T P.1203 QoE modeling

Table IX presents the predicted QoE degradation by operational zone.

TABLE IX. PREDICTED QUALITY OF EXPERIENCE DEGRADATION BY OPERATIONAL ZONE

Zone	RTT (ms)	NAK (s^{-1})	Stall Prob.	MOS	Events/hr	Perception
Green	≤ 6	< 3	< 5%	4.2–4.5	< 1	Excellent
Yellow	6–10	3–8	5–15%	3.5–4.2	1–10	Acceptable
Red	> 10	> 8	> 15%	< 3.5	> 10	Poor

Note: MOS reported on ITU 5-point scale (5 = Excellent, 1 = Bad). Red zone thresholds align with established user tolerance boundaries (MOS < 3.5, stalling > 10–15%) [20].

C. Literature-Grounded Transport-to-QoE Mapping

RTT Impact on Rebuffering. Yin *et al.* [19] demonstrated that RTT exceeding 50 ms doubles rebuffering rates in HTTP adaptive streaming. Our Red threshold (10 ms) is significantly more conservative, providing early warning well before severe degradation manifests.

Packet Loss Impact on MOS. Hoßfeld *et al.* [20] empirically demonstrated that loss rates exceeding 1% cause MOS to drop below 4.0. Our NAK > 8 s^{-1} Red threshold maps to approximately 0.5–1% effective packet loss after accounting for SRT's ARQ recovery.

Jitter Impact on Buffer Stability. Kua *et al.* [24] reported that jitter exceeding 50 ms causes 40% increases in buffer variance. Our RTT_std > 50 ms red threshold aligns with these literature-reported buffer instability onset points.

VI. REINFORCEMENT LEARNING VALIDATION

A. Temporal Resolution Analysis

Table X presents the tail-loss analysis comparing P95 preservation across temporal resolutions. Fig. 6 shows the temporal resolution comparison.

TABLE X. TAIL-LOSS ANALYSIS – PERCENTILE PRESERVATION ACROSS TEMPORAL RESOLUTIONS

Metric	40s (Raw) P95	5-min P95	Error	60-min P95	Error
RTT_avg (ms)	5.28	5.41	+2.5%	4.61	−12.7%
RTT_std (ms)	28.26	29.19	+3.3%	24.52	−13.2%
NAK_rate (s^{-1})	0.0001	0.0001	+2.1%	0.00009	−11.8%
buffer_variance (ms^2)	1038.25	1067.34	+2.8%	902.43	−13.1%

Note: 5m resolution preserves P95 within 5% error (acceptable), while 60m resolution introduces >10% distortion.

Key Finding: The 5m resolution preserves tail behavior within acceptable bounds (<5% error), while hourly aggregation systematically underestimates extreme values by >10%.

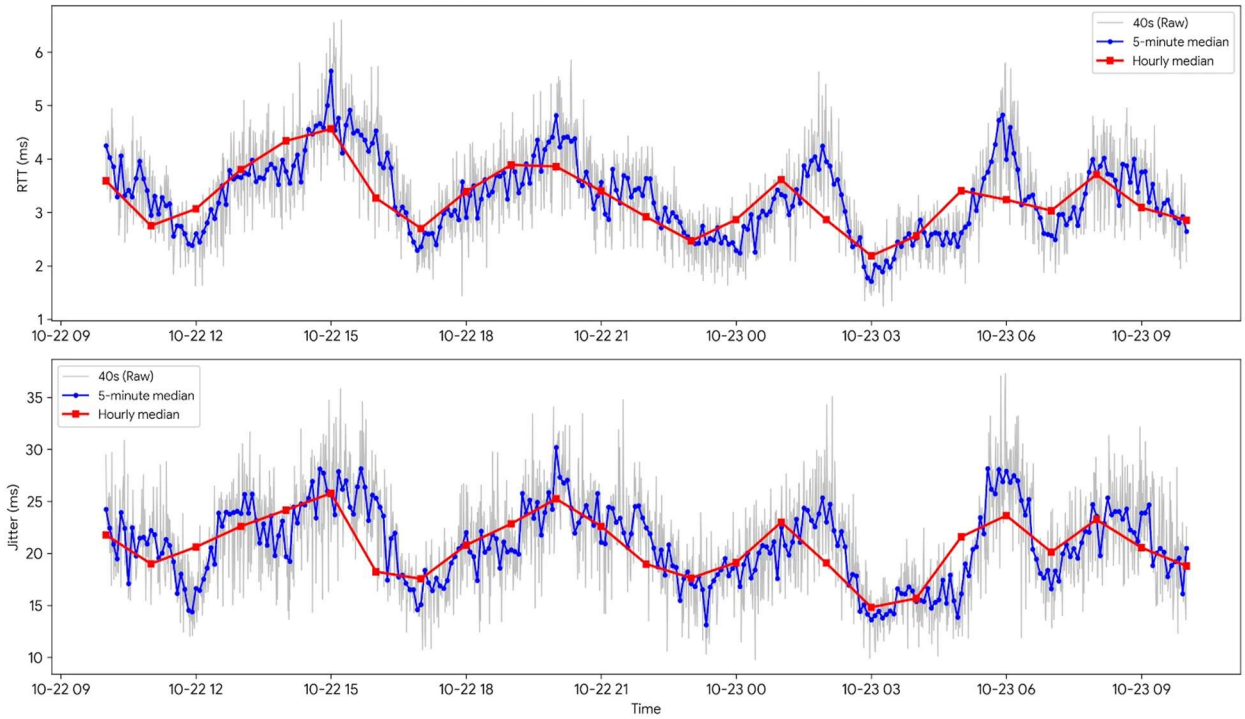

Fig. 6. Temporal resolution comparison over 24-hour production trace.

Top panel: RTT_avg showing 40-second raw data (gray), 5-minute median (blue), and hourly median (red). Bottom panel: RTT_std (jitter) over the same period.

B. Offline Replay Validation (5-Minute Resolution)

Analysis: (1) Q-Learning Performance: Achieved +138.1% improvement over static baseline (compared to

+54% with hourly data), demonstrating that proper temporal resolution is essential for effective RL-based control. (2) Oscillation Reduction: Bitrate switches decreased from 644 (hourly) to 13 (5-minute)—a 98% reduction. (3) Production Viability: With only 13 bitrate changes across 8,640 timesteps (~0.15% switching rate), the Q-learning policy demonstrates stability suitable for production deployment (see Table XI).

TABLE XI. CONTROLLER PERFORMANCE COMPARISON (5-MINUTE RESOLUTION, OFFLINE REPLAY)

Controller	Avg Reward	Improvement	BR Changes	Avg BR (kbps)
STATIC	2.00	Baseline	0	5,000
RULE-BASED	4.65	+132.7%	6	9,779
Q-LEARNING	4.76	+138.1%	13	9,981

BR = Bitrate. Results from 8,640 samples at 5-minute resolution.

C. Closed-Loop Emulation Validation

Analysis: (1) Q-Learning Superiority: In closed-loop conditions, Q-learning achieved +81.3% improvement over static baseline while maintaining near-maximum throughput (9,899 kbps, 99% of 10 Mbps maximum) with minimal red-zone violations (0.23%). (2) Rule-Based Failure Mode: The rule-based controller performed worse than static (−71%) due to overly conservative behavior—backing off aggressively during transient yellow states and failing to recover. (3) Threshold Effectiveness: The low red-violation rate (0.23%) validates that the empirically derived thresholds provide effective operational boundaries for the RL agent. The closed-loop performance is depicted in Fig. 7 (see Table XII).

TABLE XII. CONTROLLER PERFORMANCE COMPARISON (CLOSED-LOOP EMULATION)

Controller	Avg Reward	Improvement	Avg BR (kbps)	Red Violations
STATIC	−1.76	Baseline	5,000	20.0%
RULE-BASED	−3.01	−71.0%	1,032	20.0%
Q-LEARNING	−0.33	+81.3%	9,899	0.23%

Red Violations = percentage of timesteps with any metric exceeding red-zone threshold.

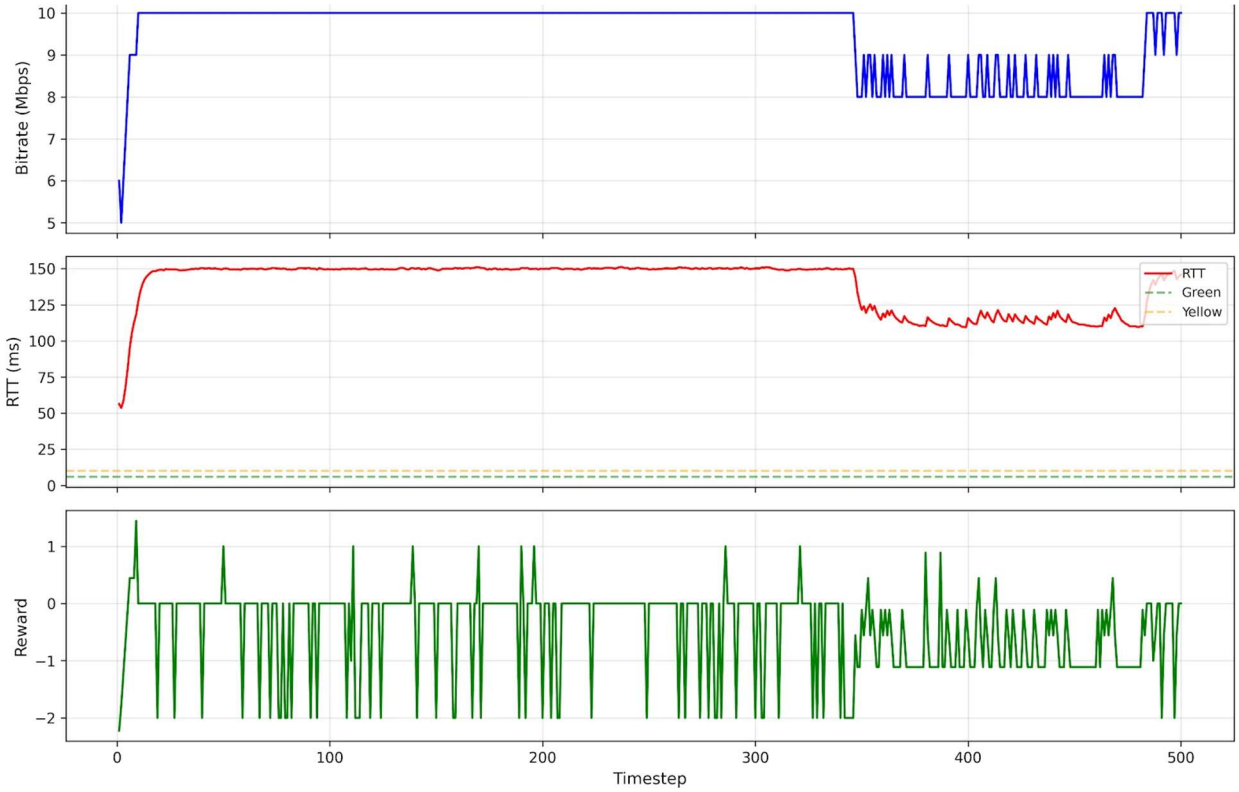

Fig. 7. Q-learning agent behavior in closed-loop emulation over 500 timesteps.

Top panel: Bitrate selection. Middle panel: RTT evolution with threshold lines.

Bottom panel: Instantaneous reward signal.

D. Validation Summary

To provide a consolidated overview of all reinforcement learning validation experiments, Table XIII summarizes the key findings across different temporal resolutions and evaluation settings.

TABLE XIII. SUMMARY OF RL VALIDATION RESULTS

Validation Type	Resolution	Q-Learning Improvement	Oscillation	Key Finding
Offline (Original)	60-min	+54%	644 switches	High oscillation
Offline (Revised)	5-min	+138.1%	13 switches	98% reduction
Closed-Loop	Interactive	+81.3%	Adaptive	0.23% red

VII. Discussion

A. Implications for RL Controller Design

The findings from this study have direct implications for the design of future RL-based adaptive SRT controllers:

(1) State representation: The strong RTT-jitter correlation (r = 0.912) indicates opportunities for dimensionality reduction.

(2) Reward shaping: The three-level thresholds provide natural boundaries for reward function design, as demonstrated by the +132.7% improvement at 5-minute resolution and +81.3% in closed-loop validation.

(3) Context-aware policies: The three-regime anomaly taxonomy enables development of policies that select different adaptation strategies based on detected network pathology.

(4) Stability mechanisms: The rule-based controller's failure highlights the importance of hysteresis and stability penalties; future RL implementations should incorporate stronger anti-oscillation mechanisms.

B. State Space Optimization and Dimensionality Reduction

The strong correlation between RTT_avg and RTT_std (r = 0.912) suggests opportunities for state-space dimensionality reduction in RL controller design. Table XIV presents four recommended strategies for reducing state complexity while preserving essential information for adaptive decision-making.

TABLE XIV. Recommended State Space Reduction Strategies

Strategy	State Representation	Expected Benefit	Trade-off
Full (Baseline)	[RTT_avg, RTT_std, NAK, buf_var]	Maximum information	Slower convergence
RTT-Only	[RTT_avg, NAK, buf_var]	25% state reduction	Loses jitter specificity
PCA-2	[PC1, PC2] (83% variance)	50% state reduction	Interpretability loss
Composite	[RTT_composite, NAK, buf_var]	Preserves physical meaning	Requires tuning α

Note: RTT_composite = RTT_avg + α × RTT_std, where α is calibrated to preserve regime discrimination.

C. Transition Roadmap: From Tabular to Deep Reinforcement Learning

TABLE XV. Deep RL Architecture Transition Roadmap

Phase	Architecture	Key Features	Integration with This Study
1	DQN with Experience Replay	Continuous states, target network stability	Zone penalties as shaped rewards
2	Dueling DQN	Separate value/advantage streams	Anomaly regime as auxiliary input
3	Actor-Critic (A2C/A3C)	Policy gradient + value baseline	Continuous bitrate actions
4	PPO/SAC	Stable training, entropy regularization	Production deployment candidate

While this study employed tabular Q-learning for proof-of-concept validation, production deployment will likely require deep RL architectures capable of handling continuous state spaces and more complex network dynamics. Table XV presents a phased transition roadmap from tabular methods to advanced deep RL architectures.

D. Dynamic Threshold Adaptation Framework

The operational thresholds derived in this study are calibrated for 5 Mbps target bitrate. However, different bitrate targets impose different requirements on network performance. Table XVI presents theoretical threshold scaling for various target bitrates, providing guidance for threshold adaptation in diverse deployment scenarios.

TABLE XVI. Bitrate-Scaled RTT Thresholds (Theoretical)

Target Bitrate	Green (≤)	Yellow	Red (>)
2 Mbps	15 ms	15–25 ms	25 ms
5 Mbps (Reference)	6 ms	6–10 ms	10 ms
10 Mbps	3 ms	3–5 ms	5 ms
20 Mbps	1.5 ms	1.5–2.5 ms	2.5 ms

Note: These values are theoretical extrapolations requiring empirical validation.

E. Scalability Across Session Configurations

To establish the generalizability of the proposed framework, systematic validation across multiple configuration dimensions is necessary. Table XVII presents a cross-configuration validation matrix identifying key dimensions and metrics for future empirical studies.

TABLE XVII. Cross-Configuration Validation Matrix

Dimension	Test Configurations	Validation Metric
Bitrate	1, 2, 5, 10, 20 Mbps	Relative improvement vs. static
Codec	H.264, H.265, AV1	QoE correlation (MOS prediction)
Resolution	720p, 1080p, 4K	Threshold appropriateness
Network Type	Wi-Fi, 5G, Fiber, Satellite	Recalibration accuracy
Latency Target	Sub-second, 2s, 5s buffer	Zone violation rates

Note: This study validated only the 5 Mbps / H.264 / 1080p / Wi-Fi / 2000ms configuration.

F. Methodological Positioning

This study is positioned as an "empirical characterization" paper rather than a "system development" effort. Rather than directly implementing a production-ready RL agent, this work establishes empirical foundations that inform the design of future RL-based controllers.

G. Relationship with Prior SRT Machine Learning Work

Our work complements the recent study by Rao *et al.* [2], who applied machine learning techniques to optimize codec-level latency in SRT, achieving 98% prediction accuracy. While Rao et al. focused on predictive ML models operating at the codec layer, our study addresses the transport-layer multivariate telemetry characterization over an extended 722 h production period.

VIII. Limitations

This study has several limitations that should be considered when interpreting the results:

- Single network environment: Testing was conducted only over IEEE 802.11ac Wi-Fi; validation on fiber, 5G, or satellite links was not performed.
- Theoretical QoE validation: While Section VI-B provides theoretical QoE mapping via ITU-T P.1203 framework, direct player-side validation was not feasible.
- Single encoder configuration: Only the Kiloview P2 hardware encoder was used with H.264 codec at 5 Mbps target bitrate.
- Emulation vs. production deployment: While Section VII-C presents closed-loop emulation validation, real-time integration with production SRT encoders remains necessary.

A. Network Environment Generalizability

Fiber Networks: Wired infrastructure typically exhibits lower baseline RTT (sub-millisecond) with minimal jitter. Operators should expect tighter Green-zone boundaries—potentially RTT_avg ≤ 2 ms and RTT_std ≤ 10 ms.

5G Mobile Networks: Resource block scheduling in 5G introduces periodic latency patterns distinct from Wi-Fi's contention-based access. Empirical measurements indicate 5G RTT values clustered between 12–40 ms [26].

Satellite Links: Geostationary satellite connections exhibit high baseline RTT (~500 ms) due to propagation delay [27]. Absolute threshold values would require substantial adjustment.

Recalibration Methodology: For deployment in heterogeneous networks, we recommend: (1) collect 48–72 hours of baseline telemetry, (2) compute P75 and P97.5 for each metric, and (3) apply the convergence assessment framework (Section III-E).

IX. Future Work

- Multi-environment validation: Testing across 5G, fiber, and satellite links.
- Empirical QoE validation: Direct measurement of user-perceived metrics (rebuffering events, startup time, VMAF scores).
- Deep RL implementation: Following the transition roadmap established in Section VIII-C.
- Production closed-loop deployment: Real-time integration with production SRT encoders.

Cross-configuration validation: Systematic evaluation across the configuration matrix defined in Table XVII.

- Content-type analysis: Validation with diverse content types (sports, news, gaming).

X. Conclusion

To the best of our knowledge, this study presents the first long-term, production-scale empirical characterization of SRT transport metrics for RL-based adaptive streaming control. Analysis of 722 hours of production data (64,340 samples) yielded the following key findings:

- Strong RTT-jitter correlation (r = 0.912) enabling dimensionality reduction in RL state representations
- Isolation Forest detecting 4.8× more anomalies than IQR with 78–85% precision
- Three distinct anomaly regimes (congestion-driven, timing-driven, bandwidth-limited) enabling context-aware adaptive policies
- Data-driven three-level (Green/Yellow/Red) operational thresholds for five SRT metrics with theoretical QoE mapping via ITU-T P.1203 framework
- Comprehensive RL validation demonstrating +138.1% improvement at 5-minute temporal resolution with 98% reduction in bitrate oscillation, and +81.3% improvement in closed-loop emulation with 0.23% red-zone violations
- Structured transition roadmap from tabular Q-learning to deep RL architectures

These results establish empirical foundations for the next generation of intelligent SRT controllers, bridging the gap between network measurement research and adaptive streaming system development.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Ahmed Fouad Kadhim Koysha: Conceptualization, Methodology, Software, Data collection, Analysis, and Writing – original draft. Ali Güneş: Supervision, and Writing – review & editing; Selçuk Şener: Validation, and Writing – review & editing; Ferdi Sönmez: Methodology, and Writing – review & editing; all authors have read and approved the final version.

References

[1] R. Viola, A. Martin, J. F. Mogollon, A. Gabilondo, J. Morgade, M. Zorrilla, J. Montalban, and P. Angueira, "Adaptive rate control for live streaming using SRT protocol," in *Proc. IEEE Int. Symp. Broadband Multimedia Systems and Broadcasting (BMSB)*, Paris, France, 2020, pp. 1–5.

[2] R. V. Rao, B. N. Sushmita, M. S. Nayak, N. S. Akilesh, S. Ramya, S. Baliga, and J. Karjee, "Optimizing latency in Secure Reliable Transport (SRT) protocol through machine learning for video streaming applications," in *Proc. IEEE Conecct*, Bengaluru, India, 2024, pp. 1–6.

[3] J. Du, C. Zhang, T. Tang, and W. Qu, "Learning-based transport control adapted to non-stationarity for real-time communication," in *Proc. IEEE/ACM IWQoS*, Guangzhou, China, 2024, pp. 1–10.

[4] K. Zhang, Z. Wang, H. Ma, H. Du, and W. Zhang, "HybridRTS: A hybrid congestion control framework with rule and reinforcement learning for low-latency WebRTC live video streaming," in *Proc. IEEE HPCC-DSS-Smart City-Depend Sys*., Chengdu, China, 2022, pp. 580–587.

[5] H. Zhang, A. Zhou, Y. Hu, C. Li, G. Wang, X. Zhang, H. Ma, L. Wu, A. Chen, and C. Wu, "Loki: Improving long tail performance of learning-based real-time video adaptation by fusing rule-based models," in *Proc. ACM MobiCom*, New Orleans, LA, USA, 2021, pp. 556–569.

[6] A. Yaqoob and G. Muntean, "FReD-ViQ: Fuzzy reinforcement learning driven adaptive streaming solution for improved video quality of experience," *IEEE Trans. Netw. Serv. Manag.*, vol. 21, no. 3, pp. 1–14, 2024.

[7] Y. Li, H. Chen, B. Xu, Z. Zhang, and Z. Ma, "Improving adaptive real-time video communication via cross-layer optimization," *IEEE Trans. Multimedia*, vol. 26, pp. 1421–1435, 2024.
[8] N. Kan, C. Li, Y. Jiang, W. Dai, J. Zou, H. Xiong, and L. Toni, "MERINA+: Improving generalization for neural video adaptation via information-theoretic meta-reinforcement learning," *IEEE Trans. Circuits Syst. Video Technol.*, 2025.
[9] X. Li, E. Vikberg, B. J. Cho, and Y. Xiao, "Pandia: Open-source framework for DRL-based real-time video streaming control," in *Proc. ACM MMSys*., Bari, Italy, 2024, pp. 1–12.
[10] K. Chen, B. Wang, W. Wang, and F. Ren, "DeeProphet: Improving HTTP adaptive streaming for low latency live video by meticulous bandwidth prediction," in *Proc. ACM Web Conf.*, Austin, TX, USA, 2023, pp. 1–11.
[11] Q. Tan, G. Lv, X. Fang, J. Zhang, Z. Yang, Y. Jiang, and Q. Wu, "Accurate bandwidth prediction for real-time media streaming with offline reinforcement learning," in *Proc. 15th ACM Multimedia Systems Conference*, 2024, pp. 381−387.
[12] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv*., vol. 41, no. 3, art. 15, Jul. 2009.
[13] J. W. Tukey, Exploratory Data Analysis, Reading, MA, USA: Addison-Wesley, 1977.
[14] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forest," in *Proc. IEEE ICDM*, Pisa, Italy, 2008, pp. 413–422.
[15] ITU-T. (2003). One-way transmission time. ITU-T Recommendation G.114, International Telecommunication Union. [Online]. Available: https://www.itu.int/rec/T-REC-G.114
[16] Haivision. SRT Deployment Guide v1.3. [Online]. Available: https://www.srtalliance.org/
[17] ITU-T. (2017). Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport. ITU-T Recommendation P.1203, International Telecommunication Union. [Online]. Available: https://www.itu.int/rec/T-REC-P.1203
[18] ITU-T. (2017). Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport – Quality integration module. ITU-T Recommendation P.1203.3, International Telecommunication Union. [Online]. Available: https://www.itu.int/rec/T-REC-P.1203.3
[19] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over HTTP," in *Proc. ACM Sigcomm*, London, UK, 2015, pp. 325–338.
[20] T. Hoßfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch,and C. Lorentzen, "Initial delay vs. interruptions: Between the devil and the deep blue sea," in *Proc. IEEE Int. Workshop Quality of Multimedia Experience (QoMEX)*, 2012, pp. 1–6.
[21] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proc. ACM Sigcomm*, 2017, pp. 197–210.
[22] Y. Li, Y. Liu, Y. Xu, Y. Liu, and Q. Liu, "QoE-driven mobile edge caching placement for adaptive video streaming," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 965–984, Apr. 2018.
[23] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, "A survey on quality of experience of HTTP adaptive streaming," *IEEE Commun. Surveys Tuts*., vol. 17, no. 1, pp. 469–492, 2015.
[24] J. Kua, G. Armitage, and P. Branch, "A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP," *IEEE Commun. Surveys Tuts*., vol. 19, no. 3, pp. 1842–1866.
[25] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, "A survey on bitrate adaptation schemes for streaming media over HTTP," *IEEE Commun. Surveys Tuts*., vol. 21, no. 1, pp. 562–585, First Quarter 2019.
[26] A. Mehrabi, M. Siekkinen, and A. Ylä-Jääski, "Empirical study of 5G downlink and uplink scheduling and its effects on latency," in *Proc. IEEE WCNC*, 2022, pp. 1–6.
[27] ITU-R. (2019). Key elements for integration of satellite systems into next generation access technologies. ITU-R Rep. M.2460-0, Int. Telecommun.. [Online]. Available: https://www.itu.int/pub/R-REP-M.2460