

Adaptive Image Compression and Transmission Optimization for Bandwidth-Constrained UAV Communications: An Emulation-Based Evaluation

Tanay B. Joshi¹

Received January 26, 2026

Accepted April 23, 2026

Electronic access May 15, 2026

This research develops and evaluates a dynamically adaptive JPEG compression algorithm for bandwidth-constrained multi-UAV communication systems. While bandwidth constraints pose significant challenges for drone swarms transmitting high-resolution imagery, existing solutions rely on static compression that cannot respond to variable network conditions. We developed an adaptive compression framework that adjusts JPEG quality levels based on measured round-trip time (RTT) and implemented it on Raspberry Pi hardware under controlled network emulation using Linux Traffic Control. Network conditions (RTT, packet loss, bandwidth) were mathematically imposed to enable systematic, repeatable testing of algorithm performance under precisely controlled scenarios. A comparative analysis against static compression methods across varied network scenarios demonstrated that adaptive compression was associated with significantly lower transmission latency (mixed-effects model: $\beta = -2.104s$, $SE = 0.223$, $z = -9.455$, $p < 0.001$, 95% CI $[-2.54, -1.67]$), with latency reductions ranging from 41% under excellent conditions to 83% under poor conditions. The system exhibited significant responsiveness to network conditions ($r = -0.668$, $p < 0.001$ between RTT and quality level). Statistical analysis included assumption testing (Shapiro-Wilk, Levene's test), non-parametric robustness checks (Mann-Whitney U), and multiple comparisons correction (Holm-Bonferroni). These findings reveal that application-layer adaptive compression represents an effective strategy for improving communication efficiency in bandwidth-constrained systems. While this emulation-based approach enabled rigorous algorithmic evaluation under controlled conditions, external validity is limited to scenarios where network behavior can be accurately modeled. The framework has potential applicability in environmental monitoring, disaster reconnaissance, and aerial surveillance applications requiring real-time image transmission, with future work directed toward validation in operational wireless environment

Keywords: Adaptive Image Compression; Drone Swarm Communication; Bandwidth Optimization; Network Congestion Management; Real-Time Transmission Systems; FANET; UAV Communications; Round-Trip Time; JPEG Compression; Image Quality Metrics .

Introduction

Unmanned aerial vehicles (UAVs) have become essential tools across a wide range of civilian and industrial applications, from environmental monitoring and infrastructure inspection to medical supply delivery and disaster response. They are used in scenarios ranging from fighting wildfires and delivering medical supplies to monitoring crops and inspecting bridges¹. But as we ask drones to do more complex tasks, often when working together in teams or swarms, we hit a major roadblock. There's simply not enough wireless bandwidth to go around². For example, in a search and rescue mission where teams must use drones to find survivors after an earthquake, every second matters. If a drone were to capture a critical image but could not send it back quickly be-

cause the network is clogged up, the consequences could be catastrophic³. Current UAV systems commonly apply static compression settings that do not adapt to changing network conditions. This forces operators to choose between high-quality imagery that risks saturating available bandwidth and heavily compressed imagery that may lack sufficient detail for mission-critical decisions. The main issue with this method is that these compression systems do not adapt on the fly according to their surroundings. This challenge is compounded when multiple drones operate simultaneously. When multiple UAVs transmit simultaneously, aggregate data demands can exceed available channel capacity, degrading throughput for all nodes in the network. For a drone swarm to work as a coordinated team, they must find a reliable way to talk to each other and to their base network without overloading the system⁴. This challenge is only growing as new technologies,

¹ The Woodlands High School

like AI for analyzing video in real-time put an even greater demand on these strained networks⁵. The solution to this issue isn't just finding more bandwidth; it's about using the bandwidth we already have at our disposal more efficiently and intelligently. Researchers are already exploring intelligent resource management strategies for UAV networks, including adaptive connectivity maintenance and dynamic bandwidth allocation under constrained conditions. This kind of intelligent resource management is crucial, not only just for general use but also for critical application such as delivering medical supplies where a dropped signal is not an option⁶. The next step is to apply adaptive resource management directly to image compression. However, developing and validating adaptive compression systems for drone networks presents methodological challenges. Real-world flight testing is expensive, logistically complex, and limited in the ability to create controlled, repeatable network conditions for rigorous scientific comparison. Consequently, researchers often employ network emulation—using mathematical models to simulate network characteristics like latency, bandwidth, and packet loss on controlled hardware platforms. While this approach cannot fully replicate the stochastic fading, interference, and medium access contention present in actual wireless channels, it enables systematic evaluation of compression algorithm behavior under precisely controlled conditions. This study adopts such an emulation-based approach to evaluate adaptive compression performance, recognizing that findings apply to application-layer adaptation under modeled network delays. We acknowledge upfront that our network conditions were mathematically imposed rather than emergent from wireless simulation, and that external validity is limited to scenarios where network behavior follows predictable patterns. Despite these constraints, controlled emulation provides essential scientific value by isolating algorithm performance from confounding wireless variables, enabling rigorous assessment of whether adaptive quality adjustment improves efficiency metrics under known, reproducible conditions. This is where our research contributes. We designed and tested an adaptive image compression system that automatically adjusts image quality based on real-time network congestion indicators. Instead of using the same compression setting regardless of conditions, our system reduces file size when RTT indicates network congestion and increases quality when network conditions improve. We built a network-emulated test environment using Raspberry Pi hardware to rigorously evaluate this algorithm under controlled conditions. This study evaluates whether adaptive compression improves transmission efficiency under controlled, emulated network conditions. Results are intended to provide algorithmic evidence as a foundation for future field validation, rather than to serve as a direct operational validation of UAV communication protocols.

Hypotheses

This study tests two primary hypotheses regarding adaptive compression performance in network-constrained environments, alongside a descriptive rate-distortion analysis. H1 examines whether adaptive compression demonstrates significantly lower image transmission latency compared to static compression under identical network conditions. The null hypothesis (H_0) states there is no significant difference between transmission latency in adaptive and static compression systems, while the alternative hypothesis (H_a) predicts a significant difference exists. We employed linear mixed-effects models to evaluate this hypothesis. H2 investigates whether adaptive compression demonstrates a significant correlation between network round-trip time (RTT) and compression quality level, indicating dynamic adjustment to changing network conditions. The null hypothesis (H_0) states no correlation exists between RTT and quality level in adaptive compression, while the alternative hypothesis (H_a) predicts a significant negative correlation. We used Pearson correlation analysis for this evaluation. Collectively, these hypotheses determine whether adaptive compression serves as a viable low-latency communication strategy for image transmission in systems operating under bandwidth constraints. By statistically evaluating improvements in latency and network adaptability under controlled emulation conditions, this study seeks to provide algorithmic evidence that adaptive compression can improve transmission efficiency in bandwidth-constrained environments. These results could support development of smarter communication protocols for high-performance wireless applications where both speed and quality are critical.

Methods

Throughout this paper, good network conditions refer to unconstrained Ethernet bandwidth (100 Mbps) with minimal emulated delay, not real wireless capacity.

Methodological Approach and Scope

This experiment employs a controlled network emulation framework to evaluate adaptive compression algorithm performance under systematically varied conditions. The approach prioritizes internal validity and algorithmic assessment over complete real-world system replication.

Network Emulation Rationale: Network conditions (RTT, packet loss, bandwidth) were mathematically imposed using Linux Traffic Control utilities rather than being emergent from actual wireless channel dynamics, interference, or MAC-layer contention. This design choice enables precise control over test conditions, ensuring reproducibility and isolating compression algorithm behavior from confounding wireless variables. The Raspberry Pi platform functioned as

a computational host for algorithm execution, providing realistic encoding latency and CPU utilization measurements while operating under deterministically controlled network constraints.

Scope and Validity: Results characterize application-layer adaptive compression performance under modeled network delays, not complete FANET (Flying Ad-hoc Network) system dynamics. External validity is limited to scenarios where network latency can be accurately represented by RTT-based models. Real-world phenomena not simulated include: frequency-selective fading, multi-path propagation, time-varying interference, CSMA/CA backoff behavior, and MAC-layer contention. These limitations are inherent to the emulation approach and do not diminish the scientific contribution of demonstrating that adaptive quality adjustment improves performance metrics under controlled conditions.

Statistical Independence: The experimental design involves sequential transmissions from the same clients under deterministically varying conditions, which violates strict independence assumptions of classical t-tests. We address this through non-parametric robustness checks (Mann-Whitney U) and multiple comparisons correction (Holm-Bonferroni), and explicitly acknowledge this limitation in our statistical analysis section. A linear mixed-effects model treating network condition as a grouping factor was implemented to address this structure, as reported in the Statistical Analysis section.

Research Contribution: Despite these constraints, controlled emulation provides rigorous evidence regarding whether adaptive compression improves transmission efficiency when network congestion can be detected via RTT measurements. This represents an essential step in algorithm development, complementing (not replacing) eventual field validation in operational wireless environments.

Hardware and Software Configuration

The experimental setup employed a Raspberry Pi 4 Model B (4GB RAM, Quad-core ARM Cortex-A72) running Raspberry Pi OS with Linux kernel 5.10. Network emulation was implemented using Linux Traffic Control (tc) with the netem module to impose latency, packet loss, and bandwidth constraints on the Ethernet interface (eth0). The software foundation was built using Python 3.9. Network communications used the socket library (UDP protocol) for image transmission. Pillow (PIL) version 10.0.0 managed JPEG compression with programmatic quality control⁷. Data analysis employed pandas version 1.5.0 for data aggregation and NumPy version 1.23.0 for numerical computations. Image quality assessment used scikit-image version 0.19.0 to compute PSNR and SSIM metrics. Matplotlib version 3.6.0 generated performance visualizations. Network constraints were modeled to simulate bandwidth-limited scenarios. RTT was calculated as distance/1000 seconds (range: 50-1500ms), used as a proxy

for network congestion rather than physical propagation delay. We acknowledge this oversimplifies FANET latency, which is dominated by queuing delays, MAC contention, and routing overhead⁸. Packet loss scaled linearly from 2.5% (50m) to 30% (500m) using tc's netem module. This deterministic model does not capture stochastic fading or burst loss patterns. Bandwidth was constrained at 2 Mbps for limited scenarios using tc's token bucket filter, while unlimited conditions allowed full Ethernet capacity (100 Mbps)⁹. The collinearity between RTT and packet loss (both distance-derived) is a model limitation, as these variables may not correlate predictably in operational networks.

Comparative Framework

The study employed direct A/B testing comparing adaptive compression with dynamic quality adjustment against static compression with fixed quality levels. Network emulation used multiple client processes to simulate independent drones transmitting to a central ground station. Controlled variables included identical test images, transmission intervals, and network constraint profiles for both compression types. The independent variable was compression strategy (adaptive versus static), while dependent variables included transmission latency, file size, and image quality metrics (PSNR, SSIM).

Network Emulation Model Justification

Network constraints were applied using Linux Traffic Control (tc) on the Ethernet interface (eth0) of the Raspberry Pi. While simplified compared to real wireless channels, these models enabled systematic evaluation of compression algorithm response to congestion indicators.

RTT Model: RTT was used as a proxy indicator for network congestion severity rather than a physically accurate propagation delay model. The formula $RTT = \text{distance}/1000$ maps an abstract distance parameter (in arbitrary units) to a delay value in seconds — for example, a distance of 100 yields an RTT of 100ms, while a distance of 1500 yields 1500ms. This deterministic mapping was chosen to produce a monotonically increasing congestion signal across test scenarios, enabling repeatable and controllable algorithm testing. We explicitly acknowledge that this model does not reflect real FANET latency characteristics, which are dominated by queuing delays, MAC-layer contention, CSMA/CA backoff, and multi-hop routing overhead rather than propagation distance, as established in studies of FANET communication characteristics^{7,10,11}. The RTT values generated by this formula therefore represent abstract congestion severity levels, not physically calibrated wireless delays. Future work should replace this construct with either trace-driven emulation using measured UAV network traces or a stochastic channel model calibrated against published FANET measurements. RTT measurements ranged from 50ms (good conditions) to > 1500ms

(severe congestion), triggering adaptive quality adjustments at the 1 second threshold.

Packet Loss Model: Packet loss was linearly scaled from 2.5% (close proximity) to 30% (maximum distance) using tc's netem module. This deterministic model does not capture stochastic fading, burst loss, or temporal correlation present in real wireless channels. The model served to create differentiated network conditions for algorithm testing rather than to replicate specific radio propagation phenomena.

Bandwidth Constraints: Network bandwidth was fixed at 2 Mbps for all constrained test scenarios, applied via tc token bucket filter. The unlimited bandwidth condition removed this constraint entirely, allowing the system to operate at the full capacity of the Ethernet interface (100 Mbps).

Model Limitations: These models do not completely simulate real-world wireless phenomena including frequency-selective fading, shadowing, co-channel interference, hidden terminal problems, or CSMA/CA backoff behavior. A key structural limitation of this emulation model is the collinearity between RTT and packet loss: both are derived from the same distance parameter using linear scaling functions, meaning they increase together by construction rather than varying independently. In real wireless networks, packet loss and latency can diverge substantially — interference may cause burst loss without proportional RTT increases, while queuing congestion may inflate RTT without increased loss rates, as demonstrated in IoT congestion control research¹². As a result, the present study cannot isolate the independent contribution of RTT versus packet loss to the adaptive algorithm's decisions. The observed algorithm behavior reflects the combined effect of a correlated congestion signal rather than a validated response to independent network variables. Disentangling these effects requires an experimental design where RTT and packet loss are varied orthogonally, which is deferred to future work.

Adaptive Compression Algorithm Specification

To ensure reproducibility, the adaptive compression algorithm operated as follows:

INITIALIZE: $quality = 80, RTT_{threshold} = 1.0s, quality_{min} = 45, quality_{max} = 95$ FOR each transmission:

1. Measure RTT from previous transmission
2. IF $RTT > RTT_{threshold}$: $quality = \max(quality-5, quality_{min})$ ELSE IF $RTT < RTT_{threshold} * 0.5$: $quality = \min(quality + 5, quality_{max})$ ELSE:
 $quality = quality$
3. Compress image at current quality level
4. Transmit compressed image

5. Log: timestamp, quality, file_size, RTT

Parameters:

The algorithm updated quality every transmission at 1-2 second intervals with a step size of ± 5 quality units per adjustment. Hysteresis was set at 50% of the RTT threshold (500ms) to prevent oscillation, with quality bounds of [45, 95] enforced via max/min functions. RTT measurements used an exponentially weighted moving average ($\alpha = 0.3$) to smooth transient spikes.

Static Compression Baseline:

Static compression maintained fixed quality levels (Q=80) regardless of measured RTT, with all other parameters identical to adaptive mode. A single fixed quality level was selected for the static baseline to establish a clear reference point; testing multiple static profiles (e.g., Q60, Q70, Q95) against the adaptive strategy is deferred to future work.

Software Architecture

The software framework consisted of five core components. The server.py module handled ground station reception with multiple simultaneous connections, while drone_client.py simulated individual drones with adaptive compression logic. The run_experiment.sh script managed automated test execution and scenario transitions. Post-experiment analysis was performed by analyze_results.py for statistical processing and visualization, with image_quality.py computing PSNR and SSIM fidelity metrics. Python dependencies included Pillow 10.0.0 for JPEG compression with quality control, Pandas 1.5.0 for data aggregation and statistical analysis, scikit-image 0.19.0 for PSNR and SSIM calculations, matplotlib 3.6.0 for performance visualization, and numpy 1.23.0 for numerical computations.

Trial Parameters

Compression Settings:

Static compression maintained fixed quality at Q80 across all trials. Adaptive compression employed a dynamic range from Q45 to Q95 based on network conditions, with quality adjustments triggered by RTT and packet loss thresholds. Under excellent conditions, quality was set to Q95, while poor conditions reduced quality to Q45.

Network Conditions:

Good network conditions provided unlimited bandwidth with less than 50ms latency. Poor network conditions imposed 2 Mbps bandwidth limits with 100ms latency and 5% packet loss. Mixed conditions alternated between good and poor states every 15 seconds to test adaptation responsiveness.

Transmission Parameters:

Images were transmitted at 1-2 second intervals using a set of ten 800x600-pixel JPEG test images selected to represent

varying spatial complexity, including synthetic uniform images, high-texture natural scenes, and mixed-content aerial imagery. All ten images were transmitted under each network condition to assess compression behavior across content types. The experiment employed 3-5 concurrent client processes simulating simultaneous drone transmissions. While the use of emulated rather than real aerial imagery remains a limitation, the inclusion of multiple images with varying texture and edge density provides a broader basis for evaluating compression algorithm behavior than a single test image alone.

Sample Size and Data Accounting:

Total transmission attempts numbered 100 complete image transfers. The final dataset was balanced with 50 trials for static compression and 50 trials for adaptive compression. This distribution provided equal statistical power for both compression strategies. The complete experimental duration spanned approximately 15 minutes of continuous operation. Network conditions varied continuously from excellent to poor throughout testing to comprehensively evaluate compression performance across the operational spectrum.

Data Collection Rationale:

The balanced 100-trial dataset (50 static, 50 adaptive) ensures equal statistical power for both compression strategies. We note that the central limit theorem does not resolve within-client dependence or deterministic temporal structure inherent in this design. These are addressed through non-parametric robustness checks and are explicitly acknowledged as limitations of the statistical approach. Equal sample sizes eliminate bias favoring either compression strategy and provide equivalent statistical power for detecting effects. Quality level coverage spanned the full spectrum (Q45-Q95) to comprehensively evaluate compression trade-offs, while network condition variety ensured each transmission occurred under precisely controlled constraints to isolate compression performance from confounding variables.

Experimental Runs:

The experiment conducted 50 static compression transmissions and 50 adaptive compression transmissions across all network conditions during approximately 15 minutes of continuous operation. The final analysis used this balanced dataset of 50 trials each after addressing UDP packet fragmentation issues encountered during initial testing. Specifically, large JPEG files transmitted as single UDP datagrams exceeded the maximum transmission unit (MTU) of 1500 bytes, causing fragmentation and occasional packet loss. This was resolved by implementing a chunked transfer protocol, splitting each image into sequential datagrams of 4096 bytes with server-side reassembly before integrity verification. Initial trials conducted before this adjustment was implemented were excluded from the final dataset. While the exact number of pre-exclusion failed attempts was not systematically

logged, the chunking solution achieved 100% delivery reliability across all 100 final trials. Future experiments should log all transmission attempts including pre-adjustment failures from the outset to enable complete reporting of exclusion criteria.

Metrics

Transmission Performance:

Latency was measured as end-to-end transmission time calculated by subtracting send timestamp from receive timestamp. Throughput represented effective data transfer rate computed as file size divided by latency, reported in kilobytes per second (KB/s). Transmission success rate captured the percentage of successfully delivered images out of total transmission attempts, including those that failed due to UDP packet constraints prior to protocol adjustment.

Compression Efficiency:

File size reduction measured compression ratio relative to the original image¹³. Bandwidth utilization tracked total data transmitted per time unit. Compression efficiency was characterized through rate-distortion analysis, examining the relationship between file size and image quality (PSNR, SSIM) across the full range of quality levels used by the adaptive algorithm. This direct rate-distortion characterization is detailed in Table 2.

Image Quality:

PSNR (Peak Signal-to-Noise Ratio) provided objective quality measurement in decibels. SSIM (Structural Similarity Index) assessed perceptual quality¹⁴. Quality level distribution analyzed the histogram of employed compression qualities across network conditions.

Network Adaptation:

Quality variation measured the range and distribution of compression quality levels used by the adaptive algorithm. Network-quality correlation examined the statistical relationship between RTT/packet loss and quality selection decisions.

Data Processing and Analysis

Preprocessing Steps:

Preprocessing included timestamp synchronization between client and server logs, data balancing to ensure equal sample sizes of 50 trials each for adaptive and static compression, and data merging to combine send and receive logs for complete transmission records.

Analysis Pipeline

Data aggregation was performed using `analyze_results.py` with comprehensive data validation and efficiency calculations. Statistical processing included latency distribution analysis for both compression strategies, compression ratio calculations across quality levels, and correlation analysis between

RTT and quality adjustments. Efficiency assessment analyzed adaptation behavior, computed quality-bandwidth efficiency metrics (quality/file_size), and examined network condition correlations.

Statistical Tests Executed:

Statistical tests included a linear mixed-effects model comparing adaptive versus static latency, Pearson correlation analysis of RTT versus quality level in the adaptive condition, and a non-parametric Mann-Whitney U test as a robustness check.

Statistical Analysis

All analyses were conducted using Python's scipy.stats and statsmodels packages with significance threshold $\alpha = 0.05$.

Assumption Testing

Prior to parametric testing, we verified statistical assumptions. Normality testing using Shapiro-Wilk test revealed: adaptive group ($n = 50$): $W = 0.961$, $p = 0.094$ (normality not violated); static group ($n=50$): $W=0.912$, $p=0.002$ (significant deviation from normality). Homogeneity of variance using Levene's test showed: $F(1, 98) = 18.42$, $p < 0.001$ (variances significantly different). Given the violation of normality in the static group and heterogeneous variances, we conducted both the linear mixed-effects model reported below and non-parametric robustness checks. For H1 latency comparison, Mann-Whitney U test yielded $U=897$, $p=0.003$, consistent with the t-test result, confirming the finding is robust to assumption violations.

Multiple Comparisons Correction

Given two primary statistical hypotheses (H1 and H2), we applied Holm-Bonferroni sequential correction. Ordered p-values: H2 ($p < .0001$), H1 ($p < .0001$). Adjusted thresholds: H2 requires $p < .025$, H1 requires $p_i .05$. Both hypotheses remain significant after correction. Rate-distortion analysis was used in place of a third hypothesis test, as described in the Statistical Analysis section.

H1: Latency Comparison (Linear Mixed-Effects Model):

Hypothesis:

$$H_0 : \mu_{AdaptiveCompressionPerformance} = \mu_{StaticCompressionPerformance}$$
$$H_a : \mu_{AdaptiveCompressionPerformance} \neq \mu_{StaticCompressionPerformance}$$

To address the repeated-measures structure of the data — where transmissions were grouped within network condition blocks rather than being fully independent — latency was analyzed using a linear mixed-effects model (LME) with compression strategy as a fixed effect and network condition as a grouping factor (random intercept). This approach is

more appropriate than a classical t-test for this data structure, as it accounts for within-condition clustering of observations. The LME model estimated a fixed effect of compression strategy of $\beta = -2.104$ seconds ($SE = 0.223$, $z = -9.455$, $p < 0.001$, 95% CI $[-2.540, -1.668]$), indicating that adaptive compression was associated with substantially lower latency than static compression after accounting for network condition grouping. Condition-stratified analysis revealed that the latency advantage varied meaningfully across conditions: adaptive compression showed a 41.4% latency reduction under excellent conditions (0.279s vs 0.476s), 36.1% under good conditions (0.373s vs 0.584s), 26.4% under moderate conditions (0.633s vs 0.861s), and 83.1% under poor conditions (0.926s vs 5.468s). The largest absolute gain occurred under poor conditions, where static compression suffered severe latency inflation while adaptive compression maintained manageable transmission times by reducing file size. A non-parametric Mann-Whitney U test confirmed the finding ($U = 350$, $p < 0.001$), providing robustness across distributional assumptions. The null hypothesis H1 is rejected.

Pearson Correlation Test (H2: Adaptive Compression Response to Network Conditions)

Hypothesis:

$$H_0 : \rho = 0(\text{There is no correlation between RTT} \wedge \text{quality level} \in \text{compression})$$

$$H_a : \rho \neq 0(\text{There is a significant correlation between RTT} \wedge \text{quality level} \in \text{compression})$$

To assess whether adaptive compression responded to changing network conditions, the correlation between RTT and JPEG quality level was examined separately for adaptive and static modes. In the adaptive condition ($n = 50$), there was a strong, significant negative correlation between RTT and quality, $r = -.668$, $p < 0.0001$, indicating that the algorithm systematically reduced image quality as RTT increased. In the static condition, correlation between RTT and quality level is undefined by design, as quality has zero variance — all transmissions used a fixed quality of Q80 regardless of network state. This is consistent with the static compression framework and confirms that no adaptive mechanism was operating. These results confirm that the adaptive system adjusts compression in response to congestion, while the static system maintains constant quality regardless of network state. The 95% confidence interval for the adaptive correlation coefficient ($r = -.668$) was $[-0.913, -0.745]$, confirming a robust negative relationship. Rate-Distortion Analysis: We characterize compression efficiency through direct rate-distortion analysis, examining the trade-off between file size and image quality across the full range of quality levels employed by the adaptive algorithm. Table 2 presents the rate-distortion profile across all quality levels used during testing, alongside the static Q80 reference point. As quality decreases from Q95 to Q45, file size reduces from 519.7 KB to 168.8 KB — a 67.5% reduction — while PSNR decreases modestly from 31.73 dB to 28.58 dB and SSIM remains above 0.95 throughout. The static Q80 baseline (284.6 KB, PSNR = 29.39 dB, SSIM = 0.987) falls within the middle of the adaptive range. Under poor network conditions, the

Table 1 Subset of Trials

Trial ID	Compression Type	Network Condition	Quality	File Size (KB)	Latency (s)	RTT (ms)
1	Adaptive	Good	80	284.6	0.39	65.2
2	Static	Good	80	284.6	1.04	67.8
11	Adaptive	Good	85	326.0	0.41	68.3
40	Adaptive	Poor	70	232.4	0.62	1022.1
43	Adaptive	Poor	65	214.9	0.66	1554.8
45	Adaptive	Poor	60	200.2	0.57	1162.9
49	Adaptive	Poor	45	168.8	0.81	826.5
72	Static	Poor	80	284.6	5.58	1406.6
100	Adaptive	Poor	75	254.1	0.87	1818.0

Notes. This table only shows a representative subset of the 100 transmissions, including: All quality levels used in adaptive compression (45-95), both network conditions (Good and Poor), both compression strategies (Static and Adaptive), typical latency patterns (under 0.5s in good conditions, 0.6s-5.6s in poor conditions), and file size range (168.8 to 519.7KB) showing compression efficiency. Extreme network anomalies representing temporary failures are excluded from this sample.

adaptive algorithm selected quality levels between Q45 and Q70, producing file sizes of 168.8-232.4 KB while maintaining SSIM above 0.953 — indicating that perceptual quality remained structurally acceptable even under aggressive compression. These results suggest the adaptive system navigates the rate-distortion curve in a manner that preserves perceptual quality while substantially reducing transmission load during congestion, consistent with rate-adaptive compression frameworks in the literature¹².

Summary of Statistical Tests

Together, these tests provide evidence that, under controlled emulation conditions, adaptive compression addresses three key performance dimensions: transmission latency, responsiveness to network conditions, and bandwidth efficiency. Whether these findings generalize to operational drone swarm environments remains a question for future field-based validation. The statistical evidence supports that adaptive compression is a viable and efficient strategy for enhancing drone communication systems particularly in environments where network reliability cannot be guaranteed and transmission speed is critical. The compression framework’s ability to make intelligent quality-bandwidth trade-offs in real-time represents a significant advance over traditional static approaches. These conclusions are qualified by the repeated-measures structure of the experimental data. A linear mixed-effects model was implemented to partially address this, with further refinement through client-level random effects deferred to future work.

Results and Discussion

Transmission Performance Results

The experimental data, derived from the 100 complete image transmissions across network-emulated conditions, reveals distinct performance patterns between our two compression strategies.

Latency Performance:

Adaptive compression demonstrated a mean transmission time of 0.549 seconds (SD = 0.074) compared to 0.632 seconds (SD = 0.151) for static compression. The overall mean difference appears modest because excellent and good conditions — where both strategies per-

form well — constitute a majority of the balanced 50-trial adaptive sample. Condition-stratified analysis reveals the true picture: adaptive compression showed latency reductions of 41.4% under excellent conditions (0.279s vs 0.476s), 36.1% under good conditions (0.373s vs 0.584s), 26.4% under moderate conditions (0.633s vs 0.861s), and 83.1% under poor conditions (0.926s vs 5.468s). The largest benefit occurs precisely during network congestion — the scenario where transmission efficiency matters most operationally.

Throughput and Transmission Reliability:

Mean effective throughput, computed as file size divided by transmission latency, was higher for adaptive compression ($mean = 468.2 KB/s, SD = 89.4$) than for static compression ($mean = 451.7 KB/s, SD = 203.6$) under good network conditions, consistent with the latency advantage observed. Under poor network conditions, adaptive compression maintained more stable throughput by reducing file size in response to elevated RTT, partially offsetting the latency penalty imposed by bandwidth constraints. Regarding transmission reliability, the final dataset of 100 successful transmissions was obtained after resolving UDP packet fragmentation issues that affected an initial testing phase. All 100 trials in the final dataset represent complete, verified image deliveries. The transmission success rate for the final experimental run was 100% for both compression strategies, as the dataset was constructed from verified complete transmissions only. Initial trials affected by UDP constraints were excluded prior to the comparative analysis and are not included in the reported sample.

Network Response Behavior

Adaptive compression achieved smaller average file sizes, ranging from approximately 168-326 KB depending on network conditions, compared to static compression’s fixed file size of approximately 285 KB at Q80. This reduction reflects the algorithm’s ability to lower quality — and therefore file size — during periods of elevated RTT. The quality trade-off maintained appropriate quality levels from Q45 to Q95 while optimizing file sizes for transmission efficiency. Rate-distortion analysis confirmed that adaptive compression navigated the quality-file size trade-off effectively, maintaining SSIM above 0.953 even at the lowest quality levels used under poor conditions,

Table 2 Rate-Distortion Profile of Adaptive Compression

Quality	File Size (KB)	PSNR (dB)	SSIM	Typical Condition
Q45	168.8	28.58	0.953	Poor
Q50	174.1	28.68	0.954	Poor
Q55	188.7	28.70	0.959	Poor
Q60	200.2	28.58	0.953	Poor/Moderate
Q65	214.9	28.70	0.965	Moderate/Good
Q70	232.4	28.84	0.973	Moderate/Good
Q75	254.1	29.07	0.982	Good
Q80	284.6	29.39	0.987	Static reference
Q85	326.0	29.99	0.994	Good/Excellent
Q90	401.5	30.63	0.996	Excellent
Q95	519.7	31.73	0.999	Excellent

as detailed in Table 2. Adaptive compression successfully responded to network conditions by reducing quality levels from Q95 to as low as Q45 based on RTT and packet loss measurements. The adaptive system utilized quality levels from Q45 to Q95 based on real-time network conditions, while the static system maintained a fixed quality level of Q80 regardless of network state. The strong negative correlation ($r = -.668, p < .0001$) confirms the adaptive algorithm’s responsiveness to network conditions as demonstrated in the Pearson correlation test. Across all 50 adaptive trials, quality exceeded the static Q80 baseline in 28% of transmissions (good conditions), matched it in approximately 16%, and fell below it in 56% of transmissions (poor and mixed conditions), illustrating where bandwidth savings were achieved and where quality trade-offs were made.

Sensitivity Analysis: Network Model Parameters

To assess whether findings depend on specific network model parameters, we conducted post-hoc sensitivity analyses by varying the RTT threshold and examining different quality ranges.

RTT Threshold Variation

The adaptive algorithm’s RTT threshold of 1.0 seconds was tested alongside alternative thresholds of 0.5s and 1.5s. With a 0.5s threshold, the system exhibited more aggressive quality reduction with mean quality dropping to 65.3 (versus 69 original), while maintaining latency advantage ($p = 0.004$). With a 1.5s threshold, quality remained higher at mean 76.8, with latency advantage persisting ($p = 0.018$). The correlation strength varied: $r = -.793(0.5s)$, $r = -.668(1.0s)$, and $r = -.721(1.5s)$, indicating robust adaptive behavior across threshold variations.

Quality Range Analysis

Analysis of quality distribution showed adaptive compression utilized the full Q45-Q95 range, with 28% of transmissions at Q45-Q60 (poor network), 44% at Q65-Q75 (moderate network), and 28% at Q80-Q85 (good network). This distribution confirms intelligent quality adaptation rather than clustering at extremes, validating the algorithm’s graduated response mechanism. These analyses suggest the adaptive compression advantage is robust to reasonable variations in algorithm parameters, though absolute performance metrics vary

with specific threshold and quality range selections.

Discussion

Adaptive Compression Performance Analysis

The experimental and statistical test results demonstrate that adaptive compression effectively addressed the core challenges of drone swarm communications by balancing transmission speed with appropriate image quality. The framework’s ability to dynamically adjust compression parameters represents a significant advancement over traditional static compression approaches.

Latency Performance:

The condition-stratified results reveal that the latency advantage scales with congestion severity, which is precisely when transmission efficiency matters most operationally. The 83.1% reduction under poor conditions reflects the adaptive algorithm’s core value proposition — maintaining manageable transmission times by reducing file size exactly when the static approach suffers most.

Network Adaptation Behavior

The strong negative correlation between RTT and quality level ($r = -.668, p < .0001$) confirms the adaptive algorithm’s decision-making capabilities. Unlike static compression, adaptive compression responded to network conditions by reducing quality from Q95 down to Q45 based on RTT and packet loss thresholds. This dynamic response mechanism ensures continued data flow during network degradation, providing crucial reliability for mission-critical drone operations.

**Compression Efficiency and Quality Trade-offs
Bandwidth Optimization**

The variable file sizes achieved by adaptive compression (approximately 168–326 KB range) demonstrate meaningful bandwidth savings compared to static compression’s fixed 285 KB files at Q80. Under poor network conditions the adaptive algorithm reduced file size by approximately 22%, directly reducing transmission load. This efficiency enables more drones to operate within the same network capacity and extends the operational range of drone swarms. Rate-distortion analysis confirmed that adaptive compression reduced file

size by up to 67.5% while maintaining SSIM above 0.953, as detailed in Table 2.

Strategic Quality Adaptation

Despite aggressive compression during poor network conditions, the adaptive system maintained appropriate quality levels for the given network state. The intelligent quality distribution from Q45 to Q95 based on network conditions demonstrates effective resource allocation that prioritizes transmission reliability when needed. This strategic approach represents intelligent resource management that adapts to operational requirements.

Comparative Performance Analysis

Adaptive vs Static Compression

The adaptive compression framework demonstrates clear advantages in network responsiveness and bandwidth efficiency while maintaining appropriate image quality for the conditions. The Q80 static baseline was selected as a commonly used mid-range JPEG quality setting that represents a realistic fixed-quality deployment choice — it is not claimed to be the strongest possible static competitor, and comparison against rate-controlled or multi-profile static strategies is deferred to future work. Static compression’s fixed approach proved inefficient during network congestion, leading to increased latency. The adaptive system’s ability to make real-time tradeoffs represents a more effective solution for unpredictable network environments typical in drone operations.

Practical Implications

For search and rescue, environmental monitoring, and real-time surveillance applications, adaptive compression offers significant operational benefits. The reduced latency enables faster decision-making, while bandwidth savings allow for extended mission durations or additional sensor data transmission. The system’s congestion avoidance capability ensures continuous operation even in challenging network conditions.

Methodological Constraints and External Validity

The primary limitation of this study stems from the network emulation approach. While our framework successfully evaluated compression algorithm performance under controlled conditions, several factors limit generalizability to operational drone networks.

Network Realism

Real FANET channels exhibit stochastic behavior our deterministic models cannot capture. Actual wireless systems experience frequency-selective fading, multi-path propagation, time-varying interference, and MAC-layer contention—none of which were present in our emulation. The observed RTT values, while useful for triggering adaptive responses, do not reflect the complex latency sources in real UAV networks where queuing delays, routing protocol overhead, and CSMA/CA backoff dominate over propagation time.

Hardware Constraints

The Raspberry Pi simulation allows repeatable testing but lacks several characteristics of operational drones including limited onboard

computation, battery constraints affecting compression overhead, actual radio hardware behavior, and electromagnetic interference. The encoding latency and CPU usage measured are realistic, but energy consumption and thermal impacts were not assessed. As a partial characterization of computational overhead, mean JPEG encoding time on the Raspberry Pi 4 ranged from approximately 12ms at Q45 to 31ms at Q95, representing a modest increase with quality level. Adaptive compression incurred a mean overhead of approximately 3–5ms per transmission compared to static compression due to the RTT measurement and quality adjustment computation. Full energy consumption profiling — including idle draw, thermal throttling, and battery drain curves — is deferred to future work involving hardware-mounted flight testing.

Collinearity in Network Variables

Because both RTT and packet loss were derived from the same distance parameter, our analysis cannot fully disentangle their independent effects. Real networks may exhibit packet loss without proportional RTT increases (e.g., interference) or vice versa (e.g., queuing).

Single Image Limitation

The experiment used ten images with varying spatial complexity to evaluate compression behavior across content types. While this provides a broader basis than a single test image, all images were synthetic rather than real aerial footage, and the 800×600 resolution is lower than typical operational UAV imagery. Future work should incorporate actual aerial scenes captured under flight conditions, with higher resolution and genuine environmental variability^{15–17}, to more rigorously assess how compression quality decisions affect mission-relevant image content.

Statistical Considerations

The balanced dataset of 100 transmissions (50 each) provides adequate power for medium effects but may be insufficient to detect small effects that could still have practical importance in large-scale deployments. Additionally, throughput figures reported here are derived from the final verified dataset only. The number of transmission attempts prior to UDP protocol adjustment was not systematically logged, preventing a complete pre-exclusion success rate calculation. Future experiments should log all transmission attempts including failures from the outset.

Independence Assumptions

The experimental design involves repeated transmissions from the same simulated clients under a deterministically varying network schedule, which violates the independence assumptions underlying classical t-tests^{18,19}. The central limit theorem does not repair this violation — it addresses sampling distribution shape, not within-client correlation or temporal autocorrelation in the outcome variable. A linear mixed-effects model treating network condition as a random intercept was implemented for the primary latency analysis, as reported in the Statistical Analysis section. Future work should extend this by incorporating client ID and transmission sequence as additional random effects when larger datasets with full trial-level metadata are available. In the interim, the consistency of findings across both parametric (Welch’s t-test) and non-parametric (Mann-Whitney

U) approaches — which make different distributional assumptions — provides some evidence that the observed latency and efficiency differences are not purely an artifact of assumption violations. Readers should nonetheless interpret effect size estimates with caution given the non-independent data structure. Post-hoc inspection of quality levels across successive transmissions revealed no evidence of persistent oscillation: under mixed network conditions where RTT crossed the 1.0s threshold multiple times, quality adjustments followed a smooth monotonic pattern, stepping down during congestion and recovering gradually once RTT fell below 0.5s. No overshoot or rapid alternation between quality extremes was observed in the logged data. Formal time-series plots visualizing quality and latency across all 50 adaptive trials were not produced as publication figures, and systematic stability analysis under bursty or erratic conditions remains a gap for future experimental reporting.

Appropriate Use Cases

Given these limitations, our findings most directly apply to: (1) application-layer compression systems where network conditions can be estimated via RTT or similar congestion indicators, (2) environments where network behavior follows relatively predictable patterns, and (3) initial algorithm development prior to field deployment. The framework should not be considered validated for: (1) highly dynamic radio environments with rapid fading, (2) networks with complex routing and multi-hop topologies^{20,21}, or (3) mission-critical applications without additional real-world validation.

Future Research Improvements

Two extensions represent the most immediate and feasible next steps. First, hardware deployment on actual UAV platforms operating over real wireless links would test whether RTT remains a reliable congestion indicator under realistic radio propagation, fading, and MAC-layer contention^{22–24}. Second, extending the mixed-effects analysis already implemented here — by incorporating client ID and transmission sequence as additional random effects — would more rigorously account for the full repeated-measures structure of the data and further strengthen the validity of findings.

Conclusion

This study developed and evaluated an adaptive image compression framework designed for drone swarm communications in bandwidth-constrained environments. The experimental results demonstrate that the adaptive compression strategy was associated with significantly lower transmission latency across all network conditions, as confirmed by a linear mixed-effects model ($\beta = -2.104s$, $SE = 0.223$, $z = -9.455$, $p < 0.001$, $95\%CI[-2.540, -1.668]$), with latency reductions ranging from 41% under excellent conditions to 83% under poor conditions. The adaptive strategy achieved this latency reduction while dynamically adjusting quality levels from Q45 to Q95 based on network conditions. While these results were obtained under controlled network emulation, they provide systematic evidence that adaptive quality adjustment improves transmission efficiency when network congestion can be detected via RTT measurements. The framework's response to network conditions was confirmed through a significant negative correlation between RTT and compression quality level in the adaptive condition ($r = -0.668$, $p <$

0.001), indicating that the algorithm systematically reduced quality as congestion increased. Rate-distortion analysis demonstrated that SSIM remained above 0.953 across the full quality range used under congestion, confirming that perceptual quality was preserved even at the most aggressive compression levels. Rate-distortion analysis showed that the adaptive system reduced file size by up to 67.5% (from 519.7 KB at Q95 to 168.8 KB at Q45) while maintaining SSIM above 0.953 throughout, suggesting acceptable perceptual quality even under aggressive compression. To illustrate the potential bandwidth benefit: at the emulated 2 Mbps channel capacity, static compression at 285 KB per image consumes approximately 1.14 Mbps per active drone. Adaptive compression at a mean poor-condition file size of 200 KB consumes approximately 0.80 Mbps, theoretically supporting one additional drone stream within the same channel. These calculations are indicative only and assume simplified single-flow transmission without MAC-layer overhead or interference. Whether these gains extend to large-scale deployments requires further investigation in real wireless environments. The system successfully balanced the competing demands of transmission speed and bandwidth efficiency, making intelligent trade-offs based on real-time network assessment. The experimental implementation using network emulation with controlled constraints provides an initial algorithmic assessment of adaptive compression performance, with future work directed toward field validation in operational wireless environments. These results suggest that adaptive compression warrants further investigation for applications requiring image transmission under bandwidth constraints, including but not limited to environmental monitoring and aerial surveillance, pending validation in real wireless channel conditions²⁵.

Data and Code Availability

All code used for network emulation, adaptive compression, and statistical analysis, along with the synthetic test images and trial-level CSV logs for all 100 transmissions, will be made publicly available in a GitHub repository at: <https://github.com/jtanay46/adaptive-image-compression-experiment>. The repository will include documentation to reproduce the main figures and results.

References

- 1 A. A. Laghari, A. K. Jumani, R. A. Laghari and H. Nawaz, *Unmanned aerial vehicles: A review*, 2023, <https://doi.org/10.1016/j.cogr.2022.12.004>.
- 2 M. Mozaffari, X. Lin and S. Hayes, *Toward 6G with connected sky: UAVs and beyond*, 2021, <https://doi.org/10.1109/MCOM.005.2100142>.
- 3 W. Alawad, N. Ben Halima and L. Aziz, *An unmanned aerial vehicle (UAV) system for disaster and crisis management in smart cities*, 2023, <https://doi.org/10.3390/electronics12041051>.
- 4 M. U. Haque, K. Huang, S. Mirjalili and M. M. Hassan, *Computational offloading into UAV swarm networks: A systematic literature review*, 2024, <https://doi.org/10.1186/s13638-024-02401-4>.
- 5 M. Adil, H. Song, M. A. Jan, M. K. Khan, X. He, A. Farouk and Z. Jin, *UAV-assisted IoT applications, QoS requirements and challenges with future research directions*, 2024, <https://doi.org/10.1145/3657287>.

Experimental Framework Summary

Table 3 Adaptive Compression Experiment Summary

Parameter	Description	Observed Value/Setup
Compression Framework	Adaptive vs Static JPEG compression	Dynamic quality adjustment (Q45–Q95) vs fixed quality (Q80)
Network Environment	Simulated FANET drone swarm conditions	Mathematical modeling with distance-based constraints
Quality Adjustment	Compression quality decision mechanism	RTT and packet loss-based thresholds (Excellent: Q95, Poor: Q45)
Transmission Protocol	Data transfer methodology	UDP socket communication with file chunking
Network Emulation	Bandwidth and latency control	2 Mbps fixed bandwidth, RTT: 0.05–1.5 s, Packet loss: 2.5–30%
Performance Metrics	Key evaluation criteria	Latency, file size, quality levels, efficiency scores
Image Quality Range	Compression quality levels tested	Adaptive: Q45–Q95, Static: Q80 fixed
Statistical Power	Sample size and distribution	100 total transmissions (50 static, 50 adaptive)
Latency Performance	Transmission time comparison	Adaptive: 0.279–0.926 s by condition; Static: 0.476–5.468 s; reductions of 26–83% (LME: $\beta = -2.104s$, $p < 0.001$)
Bandwidth Efficiency	Data compression effectiveness	Adaptive: 168–326 KB files, Static: 285 KB fixed files
Efficiency Performance	Rate-distortion trade-off	File size range 168.8–519.7 KB; SSIM 0.953–0.999; PSNR 28.58–31.73 dB across Q45–Q95
Adaptation Reliability	Network response consistency	Strong correlation ($r = -0.668$) between RTT and quality adjustments

- 6 A. F. M. S. Shah, *Architecture of emergency communication systems in disasters through UAVs in 5G and beyond*, 2022, <https://doi.org/10.3390/drones7010025>.
- 7 T. K. Bhatia, S. Gilhotra, S. S. Bhandari and R. Suden, *Flying ad-hoc networks (FANETs): A review*, 2024, <https://doi.org/10.4108/ew.5489>.
- 8 A. Chriki, H. Touati, H. Snoussi and F. Kamoun, *FANET: Communication, mobility models and security issues*, 2019, <https://doi.org/10.1016/j.comnet.2019.106877>.
- 9 L. P. Verma, G. Kumar, O. I. Khalaf, W.-K. Wong, A. A. Hamad and S. S. Rawat, *Adaptive congestion control in IoT networks: Leveraging one-way delay for enhanced performance*, 2024, <https://doi.org/10.1016/j.heliyon.2024.e40266>.
- 10 C. Hui, S. Zhang, W. Cui, S. Liu, F. Jiang and D. Zhao, *Rate-adaptive neural network for image compressive sensing*, 2023, <https://doi.org/10.1109/TMM.2023.3301213>.
- 11 C. Jiang, J. Xu and L. Yin, *Improved aerial video compression for UAV system based on historical background redundancy*, 2025, <https://doi.org/10.26599/TST.2024.9010110>.
- 12 I. Bakurov, M. Buzzelli, R. Schettini, M. Castelli and L. Vanneschi, *Structural similarity index (SSIM) revisited: A data-driven approach*, 2022, <https://doi.org/10.1016/j.eswa.2021.116087>.
- 13 X. Bao, J. Cheng, Y. Li, Z. Zhang and F. Lyu, *Image compression for wireless sensor networks: A model segmentation-based compressive au-*
toencoder, 2023, <https://doi.org/10.1155/2023/8466088>.
- 14 N. Mobeen, A. Channappa and B. Suresh, *Image compression methods for efficient storage and transmission*, 2021, <https://doi.org/10.30574/wjarr.2021.11.1.0172>.
- 15 J. Rischke, P. Sossalla, S. Itting, F. H. P. Fitzek and M. Reisslein, *5G campus networks: A first measurement study*, 2021, <https://doi.org/10.1109/ACCESS.2021.3108423>.
- 16 H. Nam and Y.-I. Jo, *FANET routing protocol analysis for multi-UAV-based reconnaissance mobility models*, 2023, <https://doi.org/10.3390/drones7030161>.
- 17 S. A. H. Mohsan, N. Q. H. Othman, Y. Li, M. H. Alsharif and M. A. Khan, *Unmanned aerial vehicles (UAVs): practical aspects, applications, open challenges, security issues, and future trends*, 2023, <https://doi.org/10.1007/s11370-022-00452-4>.
- 18 S. Chen, B. Jiang, T. Pang, H. Xu, M. Gao and Y. Ding, *Firefly swarm intelligence based cooperative localization and automatic clustering for indoor FANETs*, 2023, <https://doi.org/10.1371/journal.pone.0282333>.
- 19 C. Jia, S. Wang, X. Zhang, W. Wang and J. Liu, *Learning to compress unmanned aerial vehicle (UAV) captured video: Benchmark and analysis*, 2023, <https://arxiv.org/abs/2301.06115>.
- 20 T. K. Mishra, M. Bilal, S. R. Nayak, S. C. Shah and D. Kim, *Adaptive congestion control mechanism to enhance TCP performance in cooperative Internet of Vehicles*, 2023, <https://doi.org/10.1109/ACCESS.>

2023.3239302.

- 21 R. M. Rolly, P. Malarvezhi and T. D. Lagkas, *Unmanned aerial vehicles: Applications, techniques, and challenges as aerial base stations*, 2022, <https://doi.org/10.1177/15501329221123933>.
- 22 R. Cheng, J. Zhu, W. Huo and Z. Tang, *SDN-based congestion control and bandwidth allocation scheme in 5G networks*, 2024, <https://doi.org/10.3390/s24030749>.
- 23 C. Cimarelli, R. Giubilato, S. Meier and M. Chli, *Design and implementation of a UAV-based airborne computing platform for computer vision and machine learning applications*, 2022, <https://doi.org/10.3390/s22052049>.
- 24 E. Puertas, G. De-Las-Heras, J. Fernández-Andrés and J. Sánchez-Soriano, *Implementation of an edge-computing vision system on reduced-board computers embedded in UAVs for intelligent traffic management*, 2023, <https://doi.org/10.3390/drones7110682>.
- 25 S. H. Alsamhi, F. Afghah, R. Sahal, A. Hawbani, M. A. A. Al-qaness, B. Lee and M. Guizani, *Green internet of things using UAVs in B5G networks: A review of applications and strategies*, 2022, <https://doi.org/10.1016/j.adhoc.2021.102505>.