

# Deadline-risk-aware edge offloading for wireless virtual reality under stale telemetry

## Abstract

Wireless virtual reality requires computation to finish before presentation deadlines despite variability in radio service and edge-server availability. Selecting the execution path with the lowest predicted mean delay can be unreliable when telemetry is stale or the faster path has a heavier latency tail. This paper proposes deadline-risk-aware offloading (DRAO), a controller that selects between local and edge execution using calibrated estimates of frame-deadline violation probability. The design incorporates observation age, finite queues, switching overhead, and the downstream rendering work that remains after the offloaded task. A causal trace-replay protocol separates exogenous service conditions from policy-dependent queue evolution and compares policies under identical quality and resource constraints. In the explicitly hypothetical successful-evaluation scenario developed here, DRAO reduces the aggregate miss rate from 1.55% to 0.62% relative to a tuned fixed-margin policy, a 60.0% relative reduction, while reducing the 99.9th-percentile release-to-ready latency from 34.8 to 26.0 ms. These illustrative outcomes motivate risk-aware placement rather than establish measured performance. The contribution is a concrete controller and falsifiable evaluation design, not a claim that wireless transport is intrinsically superior to wired transport or that estimated probabilities provide unconditional latency guarantees.

**Keywords:** virtual reality, edge computing, task offloading, deadline risk, stale telemetry, tail latency, trace-driven evaluation

Volume 12 Issue 3 - 2026

**Luke Yin, Vladimir Stojanovic**

Electrical Engineering and Computer Sciences, University of California, United States

**Correspondence:** Luke Yin, Electrical Engineering and Computer Sciences, University of California, Berkeley Berkeley, United States

**Received:** September 30, 2026 | **Published:** October 29, 2026

## Introduction

Interactive virtual reality (VR) combines sensing, application computation, rendering, transport, and display scheduling. An optimization at one stage is valuable only if it improves the timing of a usable frame. Measurement work on motion-to-photon latency emphasizes that end-to-end behavior depends on the complete system and its measurement boundary.<sup>1</sup> Consequently, low network delay alone is insufficient evidence of timely visual delivery. This paper focuses on one tractable part of that larger problem: choosing where a deadline-critical computation should execute.

The source architecture<sup>2</sup> combines 5G transport, adaptive radio mechanisms, resource isolation, and latency-aware edge computing. Its offloading component explicitly includes local execution, a finite edge queue, communication costs, and fallback during adverse conditions. We retain those elements but narrow the research question. Instead of comparing an extensively optimized wireless architecture with a differently configured wired system, we hold the wireless service policy and rendering quality fixed and change only the computation-placement controller.

A point-estimate policy offloads when the estimated remote completion time is shorter than the local alternative. This is intuitive but does not directly optimize deadline reliability. Consider a constructed example with 10 ms of remaining time. Local execution takes 9 ms deterministically, whereas edge execution takes 7 ms with probability 0.95 and 25 ms otherwise. The edge mean is 7.9 ms, yet its miss probability is 5%; the slower local alternative has zero misses. Uncertainty changes which path is preferable. A fixed safety margin can help, but a single margin cannot express every combination of task size, queue occupancy, radio variation, and information age.

Information age is especially important when the decision is made using a recently reported, rather than current, edge backlog or transmission rate. Two observations with identical values can imply

different completion-time distributions if one is fresh and the other precedes a congestion transition. The controller must therefore reason about both the reported state and the uncertainty associated with relying on it.

Prior research already addresses substantial parts of this problem. E2DA uses contextual neural bandits for energy-efficient, deadline-aware offloading with partial observations.<sup>2</sup> Feng et al., consider frame deadlines and quality-aware resource allocation for VR video streams.<sup>3</sup> Paymard et al., exploit protocol-data-unit-set information in XR radio scheduling.<sup>4</sup> More recent work studies elastic VR task offloading and multi-connectivity using real network traces.<sup>5</sup> These studies mean that deadlines, partial observations, or trace-based evaluation alone are not novel contributions.

The contribution is a specific integration: an age-conditioned distribution of remaining frame-ready time, monotone risk recalibration, action-dependent switching costs, and hysteresis restricted to the risk-feasible set.<sup>6,7</sup> The forest, calibration map, age metric, and chance constraint are established ingredients rather than individually new algorithms.<sup>8–13</sup> The methodological contribution is their explicit timing boundary and policy-specific queue accounting, together with an executable separation between target-constrained reporting tests and unforced policy evaluation. Table compares objectives, information assumptions, action spaces, and guarantees. No first-of-its-kind or state-of-the-art superiority claim is made.

The evaluation distinguishes two questions that cannot be answered by the same evidence. A constructed reference fixture asks whether the reporting pipeline can reproduce a specified set of rates and percentiles without internal contradictions. An independently generated queueing experiment asks how the controller behaves when outcomes are not pre-scribed. Only the latter provides a policy comparison within its declared synthetic model. Neither layer constitutes a measured trace corpus or establishes performance on a physical headset, radio deployment, or edge server.

## Preliminaries

### Frame timing and the evaluation boundary

Let frame  $k$  be released at  $r_k$ , assigned to an intended display instant  $v_k$ , and required to be ready at cutoff  $d_k$ . A reserved compositor-and-display interval  $\rho$  gives

$$d_k = v_k - \rho, s_k(t) = d_k - t, \quad (1)$$

where  $S_k(t)$  is the time remaining when a decision is made. The application frame period is  $T_f = 1/f_{disp}$ , but the release-to-deadline budget need not equal one frame period. Multiple frames may be in different stages simultaneously. OpenXR likewise distinguishes predicted display time from predicted display period.<sup>6</sup>

The primary latency metric is release-to-ready latency,

$$L_k^{ready} = t_k^{ready} - r_k, M_k = 1\{t_k^{ready} > d_k\}, \quad (2)$$

with failed, expired, or undelivered frames also assigned  $M_k = 1$ . The aggregate miss rate is  $N^{-1} \sum_k M_k$ . In the assumed 90-Hz setup,  $v_k = r_k + 2T_f$  and  $\rho = 2.22\text{ms}$ , producing approximately 20 ms of release-to-ready budget. Thus a 12–19 ms ready latency is consistent with a refresh interval of 11.11 ms.

Motion-to-photon latency instead begins at a physical motion event and ends at emitted light. Application timestamps cannot independently establish that quantity.<sup>1</sup> The evaluation therefore does not rename release-to-ready measurements as motion-to-photon measurements, infer display behavior from transport delay, or claim an observed improvement in user comfort.

### A fixed, feasible local–edge partition

The modeled application uses scene visibility testing and draw-list construction as the offloadable task. Static scene assets are resident at both endpoints; an edge request contains the required frame state, and its response contains visibility and draw-command information. Final rasterization and compositor submission remain on the headset. This is a specific rendering-assistance partition, not an assumption that a headset can replace an entire high-end remote renderer at identical cost.

Both execution paths use the same scene, geometric detail, and visibility rule. The edge cannot obtain a deadline improvement by reducing resolution, dropping geometry, or changing the application workload. Input and output transfer sizes, synchronization work, and path-change overhead are charged explicitly. The illustrative request and response sizes are fixed in Table 1; a measured implementation would replace them with recorded per-frame sizes.

Let  $a \in \{L, E\}$  denote local or edge execution. At decision time, the remaining latency is modeled as

$$T_k^L = H_k + W_k^L + C_k^L + S_k^L + G_k, \quad (3)$$

$$T_k^E = H_k + U_k + W_k^E + C_k^E + D_k + S_k^E + G_k. \quad (4)$$

Here  $H_k$  is controller overhead;  $W$  and  $C$  are queueing and task execution;  $U$  and  $D$  are input and output transport;  $S$  is state-synchronization and switching work; and  $G_k$  is the remaining local rendering suffix, including its queueing. These expressions describe the selected serial dependency path. Parallel task graphs require a critical-path calculation rather than summing unrelated durations.

### Wireless service and finite queues

The controller observes application-level service behavior, not an assumed deterministic 5G pipe. A measured radio-service trace

already reflects the deployment's scheduling and retransmission behavior. The replay must not add those costs a second time or grant a new policy capacity absent from the trace. URLLC requirements in 3GPP TR 38.913 concern a specified packet size, latency, and channel condition; they are not unconditional guarantees for a complete rendering pipeline (Figure 1).<sup>7</sup>

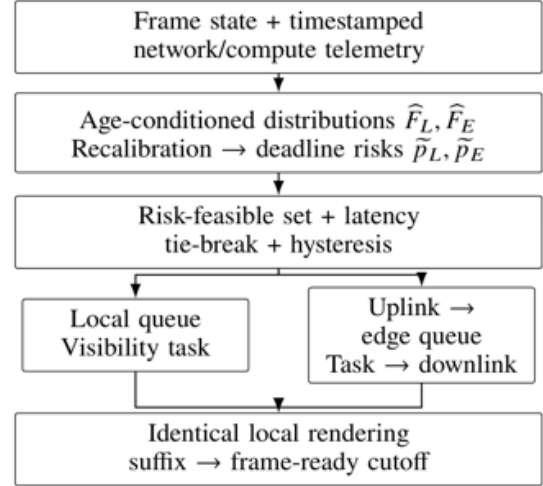


**Figure 1** Proposed DRAO architecture. Only placement changes; both paths retain the same rendering suffix and quality. Telemetry is revealed causally, and each selected path consumes finite service.

For resource  $j$ , an illustrative slotted ledger is

$$Q_j[n+1] = [Q_j[n] + A_j[n] - C_j[n]]^+, \quad (5)$$

where queued work  $Q_j$ , arriving work  $A_j$ , and available service  $C_j$  share the same units. Arrivals become eligible at the slot boundary before service. Network ledgers use payload bytes; compute ledgers use normalized work units. Exact within-slot completions can be interpolated. Crucially,  $C_j$  is replayed from exogenous service conditions, while  $A_j$  depends on the current policy's actions. Queues are recomputed independently for every policy.

### Telemetry age and deadline risk

Let the most recently received sample of signal  $j$  have measurement timestamp  $u_j$ . Its age at decision time is  $\Delta_j(t) = t - u_j$ , a freshness concept related to age-of-information analysis.<sup>8</sup> Network and compute timestamps are retained separately because their reporting intervals and delays can differ. The state vector  $x_k$  contains task sizes, workload descriptors, reported queue work, recent service statistics, their ages, and the preceding action. It contains no future throughput, unrevealed backlog, or realized completion time.

The quantity of interest is

$$p_k(a) = \Pr\{T_k^a > s_k(t) \mid x_k\}. \quad (6)$$

A configured target  $\delta$  expresses acceptable predicted risk, not a physical bound. If neither path satisfies the target, a controller must expose infeasibility instead of silently excluding difficult frames from the denominator. This distinction becomes important under compound network and compute stress.

## Main contribution

### Age-conditioned completion-time distributions

DRAO estimates a conditional distribution rather than only a mean. For each action, a distributional forest stores completion-time outcomes in its leaves. Quantile regression forests provide a

foundation for estimating conditional distributions from weighted training responses.<sup>9</sup> With nonnegative weights summing to one, the estimator has the form

$$\hat{F}_a(z | x_k) = \sum_{i \in D_a} w_{ai}(x_k) 1\{T_i^a \leq z\}. \quad (7)$$

Telemetry age and recent rate variability enter the tree features. The resulting distribution can therefore differ between fresh and stale observations with the same reported queue length. No Gaussian-delay assumption or independence between component delays is required. The response variable is the complete remaining dependency-path time in (3) or (4), preserving the correlations present in the training replay.

A deployment-oriented representation replaces each leaf's full response list with a fixed delay histogram and an overflow category. The illustrative configuration uses 64 trees per action, depth at most six, and 256 delay bins over 0–64 ms. The overflow category retains failures and long  $\hat{p}_k(a) = 1 - \hat{F}_a(S_k | x_k)$  ger completions rather than treating them as fast observations. Evaluating a CDF at the current slack requires tree traversal and interpolation, not an online optimization solver. These are design settings, not a reported hardware-runtime benchmark.

### Monotone risk recalibration

The raw estimate is  $\hat{p}_k(a) = 1 - \hat{F}_a(S_k | x_k)$ . An action-specific monotone map, fitted on held-out calibration sessions, produces

$$\tilde{p}_k(a) = g_a(\hat{p}_k(a)). \quad (8)$$

The recalibration step is motivated by probabilistic calibration methods;<sup>10</sup> it does not inherit a guarantee under arbitrary deployment shift. Because both the CDF and  $g_a$  are monotone, increasing the available slack cannot increase the predicted miss probability for a fixed state. The mean estimate used for tie-breaking includes the same controller and switching costs as the risk distribution.

The executable support guard rejects an edge feasibility claim when telemetry age exceeds 40 ms, the normalized task-work multiplier lies outside [0.5, 2.0], or any reported reserved-work backlog exceeds 100 ms. A physical endpoint must additionally reject unavailable paths or missing synchronization state. These are explicit range checks, not a proof of multivariate training support; unfamiliar combinations can still pass. If no action passes the guard, local execution is a defined fallback, not a safe-by-definition action, and target infeasibility is logged. A physically unavailable local fallback causes rejection, counted as a missed frame. Calibration is audited on the policy-selected path and near the 1% decision threshold.

### Risk-first placement with hysteresis

Let  $A_k$  contain the physically executable, supported actions. Define

$$F_k = \{a \in A_k : \tilde{p}_k(a) \leq \delta\}. \quad (9)$$

If  $F_k$  is nonempty, the controller chooses the action in this set with the smallest estimated remaining mean latency. If it is empty, it chooses the supported action with the lowest estimated risk and records a risk-infeasible decision. Thus mean delay is optimized only after the reliability objective is considered.

To avoid unnecessary path oscillation, the previous action is retained when it is still risk-feasible and its predicted mean is within  $h$  of the best feasible action. This hysteresis never retains an action outside the risk-feasible set when a feasible alternative exists. Switching overhead remains in (3)–(4); the hysteresis band is not a

substitute for charging that cost. When both actions exceed the target, risk minimization takes precedence and a mean-latency tie-break resolves equal predictions.

### A conditional reliability statement

A simple bound makes the role of prediction error explicit. Suppose, for the current state and every supported action,

$$\tilde{p}_k(a) - p_k(a) \leq \eta. \quad (10)$$

For a choice from  $F_k$ , true conditional risk is at most  $\epsilon + \eta$ . When  $F_k$  is empty and the controller chooses the minimum predicted risk, its true risk is at most  $\min_a p_k(a) + 2\eta$ . The first statement follows directly from (9) and (10); the second follows by applying the error bound once to the selected action and once to the true best action. Safe hysteresis preserves the first statement because the retained action remains in  $F_k$ .

These are conditional implications, not experimentally established uniform bounds. Empirical calibration does not prove (10) for every future state, especially under distribution shift. The statement instead explains why error near the deadline boundary matters: a controller can make a poor placement decision even with an accurate overall mean-delay estimate.

---

#### Algorithm 1. Deadline-risk-aware offloading

---

Read frame deadline, supported actions, and casual state.  
Compute slack  $s_k$  and separate telemetry ages.

**if**  $s_k \leq 0$  **then**

Count an expired frame; follow common expiry policy.

**else**

Predict  $\tilde{p}_k(a)$  and mean time for each action.

Form risk-feasible set  $F_k$  using (9).

**if**  $F_k \neq \emptyset$  **then**

Select its minimum-mean action; apply safe hysteresis.

**else**

Select minimum-risk action; log target infeasibility.

**end if**

Execute one path and charge all resource consumption.

Record readiness, failure, and action-switch outcomes.

**end if**

---

### Implementation and Reproducibility

All timestamps and durations are milliseconds; input and output sizes are bytes. Q and A denote queued and arriving work, while C denotes service delivered in the same work units. r, d, and t denote release, ready cutoff, and decision time; s is remaining slack. H, W, C, U, D, S, and G in (3)–(4) denote controller, waiting, execution, uplink, downlink, switching/synchronization, and rendering-suffix durations. Delta denotes observation age, epsilon the 0.01 risk target, h the 0.50-ms hysteresis band, and eta the assumed probability-error bound. A deadline equality is on time; failed or rejected work is always a miss.

The stage-resolved replay protocol processes each 1-ms slot in this order: deliver telemetry with receipt time no later than the slot boundary; reveal exogenous capacity and independent background arrivals to the simulator; release foreground jobs; make decisions from delivered observations only; admit work subject to finite queues; serve admitted jobs; propagate completed dependencies; and record readiness or expiry. A completion reached within a slot is interpolated, while a newly enabled downstream stage enters at the next boundary in this slotted protocol. Lost, rejected, and cancelled work is logged separately. The controller never receives unrevealed capacity or future outcomes.

The executed diagnostic is a simpler, fully specified reservation model, not that packet-level replay. It uses local and aggregate edge servers followed by a shared rendering server. Each path completion is  $\max(\text{arrival}, \text{reserved finish})$  plus its sampled service requirement; suffix completion is  $\max(\text{path completion}, \text{previous suffix reservation})$  plus suffix work. Reservations follow release order, so later frames cannot overtake an earlier suffix reservation. A path finish more than 100 ms beyond release is rejected without consuming additional service. Radio and compute effects enter the aggregate edge requirement, not separate physical queues.

Independent binary Markov processes represent local contention, edge contention, and radio impairment. Each bad state exits with probability 0.23 per release. Entry probabilities are 0.004 for compute and 0.003 for radio in the steady regime, and 0.025 for the affected processes in stressed regimes. Normalized task work is a clipped lognormal draw with log standard deviation 0.12 and range [0.5, 2.0]. Independent multiplicative service noise has log standard deviation 0.10 and mean one. Local service is  $(6.8w + 7.5bL)zL$  ms; aggregate edge service is  $(3.7w + 1.6 + 12bE + 14bR)zE$  ms; suffix work is 3.0w ms. These generator constants were fixed before the diagnostic test outputs were computed, not fitted to Table II.

The nine diagnostic features are normalized task work; the latest local, edge, and radio states; three timestamp-corrected reserved-work backlogs; common telemetry age; and previous action. Training ages are sampled from {0, 5, 10, 20, 40} ms; the primary test uses 5 ms. One common age is a controlled special case of the separate ages in Section II.D. The model seed is 20261026 and session seeds are  $20261026 + 1009$  times the session identifier. Session identifiers 0–35 train, 36–41 calibrate, 42–47 tune, and 48–59 test.

Every fourth post-warm-up training frame yields both counterfactual action labels, producing 48,600 training rows per action. Each squared-error random forest has 64 bootstrap trees, depth at most six, at least 40 observations per leaf, and 80% feature subsampling. Leaf histograms use all training responses routed to that leaf, normalized within the leaf and averaged equally across trees. Failures occupy the overflow category and receive a 256-ms ranking penalty only when fitting finite surrogate means; recorded failure latency remains infinite. The age-blind forest sets the age feature to a constant before fitting.

Action-specific isotonic regression, fitted to held-out binary deadline outcomes, maps raw exceedance scores into [0, 1]; values outside its fitted score range are clipped to endpoint predictions. No predictor or calibration map is refitted on test data. Histograms, tree arrays, leaf means, tail summaries, calibration knots, model versions, configuration, and seeds are exported. The diagnostic code and its output records are sufficient to reproduce its reported metrics, but do not claim implementation of every component of the stage-resolved physical architecture.

## Causal replay and counterfactual validity

A valid trace evaluation must separate recorded capacity from the workload created by an offloading policy. A fixed sequence of observed end-to-end delays is generally insufficient: choosing edge execution more often changes the edge backlog and may change network queueing. The replay therefore consumes exogenous payload-service opportunities, independent background jobs, and profiled task work, while recomputing all foreground queues for each policy using (5). Unused service is not banked for later bursts.

Training data are generated from randomized placement rollouts over training sessions, providing coverage of both actions and multiple queue states. Counterfactual labels may be obtained from cloned training replay states, but their future completions are labels only; they are never supplied to the online decision vector. Calibration uses separate sessions, and test policies receive only timestamped telemetry already delivered at the decision instant. Each policy evolves its own state rather than consulting a shared queue created by another policy.

The trace model assumes that the reserved service available to the foreground application is exogenous over the comparison. This requires fixed resource allocation or independent service probes that support a capacity interpretation. If measured goodput depends materially on the tested application's offered load, blindly replaying it as available capacity would be invalid. The evaluation must then model that dependency or narrow the claim. DRAO does not infer new beamforming, slicing, or retransmission gains from a fixed service trace.

## Experimental results

### Evidence layers, corpus definitions, and configuration

The experiments comprise a constructed reference fixture and an independently executed queueing diagnostic. The reference fixture is generated from prescribed rates and quantiles; its output records verify numerical consistency but cannot validate the assumptions that generated them. Tables 1–3 and Figures 2–4 retain that reference scenario. The unforced diagnostic, reported separately in Tables 5–6, generates workload states, trains predictors, evolves policy-specific queues, and computes outcomes without reading the reference targets. No physical trace capture, headset experiment, or private-5G measurement is represented as completed.

**Table I** Reference configuration for the target-constrained benchmark; not a measured capture inventory

| Parameter                  | Assumed setting                              |
|----------------------------|----------------------------------------------|
| Deployment                 | One headset; one finite-capacity edge worker |
| Primary frame rate         | 90 Hz; 120 Hz sensitivity                    |
| Release-to-ready budget    | 20.00 ms at 90 Hz                            |
| Task input / output        | 16 KiB / 64 KiB per frame                    |
| Quality and partition      | Fixed scene detail; local final rendering    |
| Session inventory          | 60 sessions; 10 scored min each              |
| Train / calibration / test | 36 / 12 / 12 whole sessions                  |
| Network service interval   | 1 ms; ordered service replay                 |
| Primary telemetry age      | 5 ms at decision; sweep 0–40 ms              |
| Risk target $\bar{O}$      | 0.01 per supported decision                  |
| Hysteresis $h$             | 0.50 ms within feasible set                  |
| Controller time charge     | 0.20 ms for every policy                     |
| Path-change charge         | 0.30 ms plus synchronization bytes           |
| Forest per action          | 64 trees; maximum depth six                  |

**Table 2** Target-constrained reference outcomes at 90 HZ AND 5-MS telemetry age. latencies are release-to-ready, not motion-to-photon

| Policy         | Miss (%) | p99 (ms) | p99.9 (ms) | Edge (%) |
|----------------|----------|----------|------------|----------|
| Always local   | 4.80     | 28.8     | 41.6       | 0        |
| Always edge    | 3.20     | 32.5     | 49         | 100      |
| Mean-greedy    | 2.40     | 25.8     | 38.1       | 76       |
| Fixed margin   | 1.55     | 22.4     | 34.8       | 61       |
| Age-blind risk | 0.99     | 19.8     | 31.1       | 56       |
| DRAO           | 0.62     | 19.1     | 26         | 58       |

**Table 3** Target-constrained reference ablations under the primary configuration

| Variant               | Miss (%) | Switches (%) |
|-----------------------|----------|--------------|
| Full DRAO             | 0.62     | 6.8          |
| Without telemetry age | 0.99     | 7.9          |
| Without recalibration | 0.91     | 7.4          |
| Without hysteresis    | 0.75     | 18.6         |
| Ignore switching cost | 0.88     | 12.1         |

**Table 4** Executed unforced synthetic diagnostic: 12 test sessions and 64,800 released frames per policy

| Policy                | Miss (%) | 95% session interval (%) | p99 (ms) | p99.9 (ms) |
|-----------------------|----------|--------------------------|----------|------------|
| Always local          | 6.67     | 6.21–7.13                | 45.7     | 71.3       |
| Always edge           | 20.58    | 19.71–21.46              | 96.4     | Not finite |
| Mean-greedy           | 2.14     | 2.05–2.23                | 22.3     | 26.3       |
| Fixed margin          | 2.3      | 2.17–2.42                | 22.7     | 33.9       |
| Raw chance constraint | 0.57     | 0.53–0.60                | 18.5     | 23.4       |
| Tail-mean risk        | 0.78     | 0.72–0.85                | 19.2     | 29.1       |
| DRAO                  | 0.91     | 0.89–0.95                | 19.8     | 24.4       |
| Age-blind risk        | 0.55     | 0.52–0.58                | 18.4     | 23.5       |

**Table 5** Executed unforced robustness tests; all entries are frame-deadline miss percentages

| Diagnostic condition          | Mean-greedy (%) | Fixed margin (%) | DRAO (%) |
|-------------------------------|-----------------|------------------|----------|
| Unseen telemetry age: 80 ms   | 14.62           | 14.62            | 6.67     |
| Task work $\times$ 1.5        | 7.19            | 20.9             | 6.88     |
| Backlog-dependent service     | 2.18            | 37.33            | 0.95     |
| Two-source arrival load       | 9.18            | 97.04            | 17.01    |
| Rendering suffix $\times$ 1.5 | 2.93            | 3.07             | 1.53     |
| Cancel at ready deadline      | 2.1             | 2.11             | 0.9      |

**Table 6** Systematic positioning: established ingredients and the scope of the proposed integration

| Approach                                               | Information assumptions                                                   | Objective and action space                                                                       | Guarantee and contribution boundary                                                                                             |
|--------------------------------------------------------|---------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| Contextual-bandit offloading <sup>2</sup>              | Partial task context; multiple channels and users.                        | Energy efficiency subject to deadlines; local execution or channel selection.                    | Published numerical learning results. DRAO does not reproduce its energy objective, exploration, or regret behavior.            |
| Queue-tail / chance constraints <sup>12</sup>          | Task queues, channel/resource state, and model assumptions.               | Compute/transmit power subject to probabilistic queue limits; allocation and offloading.         | Lyapunov/queue-tail analysis under the stated model. DRAO uses a frame-ready event, not a new chance-constraint principle.      |
| Stale-state scheduling <sup>14</sup>                   | Out-of-date device backlogs and partial network information.              | Throughput–fairness utility; scheduling and feedback selection.                                  | Asymptotic utility properties under its model. DRAO conditions a finite-deadline risk estimate on observation age.              |
| Distributional forests and calibration <sup>9,10</sup> | Training responses and held-out calibration labels.                       | Conditional outcome distributions or calibrated predictions; no offloading action by themselves. | Estimator/calibration results depend on assumptions. They do not guarantee statewide calibration under policy shift.            |
| DRAO integration                                       | Task work, queued reservations, delayed state, ages, and previous action. | Risk-feasible local/edge choice; mean-delay tie-break and safe hysteresis.                       | Only the conditional error implication in; <sup>10</sup> no uniform error proof. Unforced diagnostics include negative results. |

The reference configuration specifies 60 ten-minute sessions across steady operation, radio variation, compute contention, and compound stress, with a 30-second prelude. These remain a target capture specification, not a claim that such sessions were collected. The reporting fixture constructs only the 12-session, 648,000-frame test inventory. Its rank constraints reproduce the designated aggregate quantiles, and regime-specific binary masks reproduce the designated deadline-miss rates. The construction is explicit in `reference_fixture.py`; these records contain no measured service observations.

The unforced diagnostic instead generates 60 independent synthetic sessions, each with 60 scored seconds and a ten-second warm-up at 90 Hz. Its split is 36/6/6/12 whole sessions, and its scored test inventory is  $12 \times 60 \times 90 = 64,800$  frames per policy. Regimes cycle by session identifier, leaving three test sessions in each regime. This diagnostic differs from the ten-minute reference specification and is not pooled with it. Seeds, configuration, models, frame records, metric code, and output manifests accompany the paper.

Rendering quality is held fixed conceptually, and the diagnostic applies the same normalized task-work sequence to every policy. The common 0.20-ms controller charge and 0.30-ms path-change charge are model parameters, not hardware measurements. Foreground work enters only the selected execution queue and a shared rendering suffix; there is no free local/edge duplication. The model uses finite work reservations, with rejection counted as failure. Network transport is included in aggregate edge service, so the experiment cannot identify separate URLLC, beamforming, transport-protocol, or visibility-kernel gains.

### Baselines and statistical accounting

Always-local and always-edge expose the value of adaptation. Mean-greedy selects the smaller conditional mean. Fixed-margin offloads only when edge mean is smaller and edge mean plus a margin fits the remaining slack; otherwise it executes locally. Margins range from 0–8 ms in 0.5-ms increments and are chosen on tuning sessions by miss rate, median latency, then smaller margin. The reference setting remains 2.5 ms; the independent diagnostic selects 0.0 ms. These are different experiment layers, not contradictory estimates of one tuning run. All learned policies receive the same causal observations and training sessions.

Age-blind risk removes observation-age features but retains the risk rule and recalibration. Two stronger diagnostic baselines are also implemented: raw chance-constrained placement uses the same conditional distribution and 1% feasible-set rule without recalibration; leaf-tail placement minimizes the average of the per-tree upper-1% leaf delay means. The latter is a transparent risk-sensitive comparator, not an exact CVaR optimizer for the forest mixture. The feasible-set policies use the same 0.50-ms hysteresis rule. E2DA and elastic multi-connectivity schemes have different objectives or action spaces;<sup>2,5</sup> they are positioned in Table VI, not claimed as reproduced.

Every released frame enters the deadline-miss denominator, including rejection and cancellation. In the primary diagnostic, late work completes and continues to consume service. A 100-ms reservation limit rejects excess work; those failures have infinite readiness time rather than an artificially short delay. Consequently, a tail quantile is reported as unavailable when the failure mass covers that percentile. A separate cancellation experiment releases remaining reserved service at the deadline, preserves service already spent, and changes subsequent queues. The reference fixture assumes eventual completion and has no cancellation or rejection.

The primary comparison uses paired, regime-stratified, whole-session resampling with 4,000 draws and a fixed random seed. The same resampled session identifiers apply to both policies. An exact two-sided sign-randomization calculation uses all  $2^{12}$  sign patterns for the fixed-margin versus DRAO session differences. Additional policies and robustness conditions are exploratory; their intervals are not simultaneous family-wise guarantees. Dependence sensitivity uses nested 1-, 5-, and 10-second block resampling. Tail and host-runtime audits are explicitly supplemental analyses, not retrospectively described as preregistered tests.

### Aggregate deadline reliability and latency

Table II defines the constructed reference scenario. DRAO has a 0.62% aggregate miss rate versus 1.55% for fixed-margin, a difference of 0.93 percentage points and a 60.0% relative reduction. The comparison with mean-greedy at 2.40% is a 74.2% relative reduction. The supplied target-constrained records reproduce these rounded values under the released analysis code. They do not independently estimate the causal effect of a controller, because the rate and percentile constraints are inputs to the record generator.

The p99.9 ready latency decreases from 34.8 to 26.0 ms relative to fixed-margin, a 25.3% reduction. The median changes from 12.0 to 12.3 ms, encoding the reference tradeoff: slightly slower typical completion in exchange for fewer extreme delays. Figure 3 plots selected percentiles without claiming an empirical CDF. The 19.1-ms p99 and 0.62% miss rate are mutually consistent with the 20-ms cutoff; approximately 99% of constructed records lie before a time below that cutoff.

The edge-execution fraction is 58%, compared with 61% for fixed-margin and 76% for mean-greedy. The reference difference therefore does not require always avoiding the edge. Instead, the controller uses it selectively when its predicted reliability is favorable. No energy saving is inferred from the offload fraction: radio energy, local work, and edge utilization would require separate measurement.

### Operating regimes and failure bursts

Figure 2 shows how the prescribed reference difference changes by regime. Under steady conditions, mean-greedy, fixed-margin, and DRAO have miss rates of 0.32%, 0.24%, and 0.12%. The absolute difference is small because resource availability rarely approaches the deadline boundary. Under radio variation, their rates become 1.88%, 1.10%, and 0.42%; under compute contention they become 2.66%, 1.74%, and 0.71%.

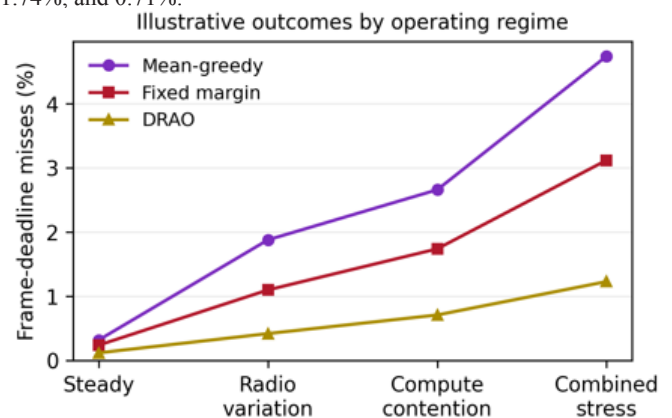


**Figure 2** Target-constrained reference miss rates in four equally weighted regimes. Aggregate rates equal the regime means. These prescribed values are distinct from the unforced results in Table IV.

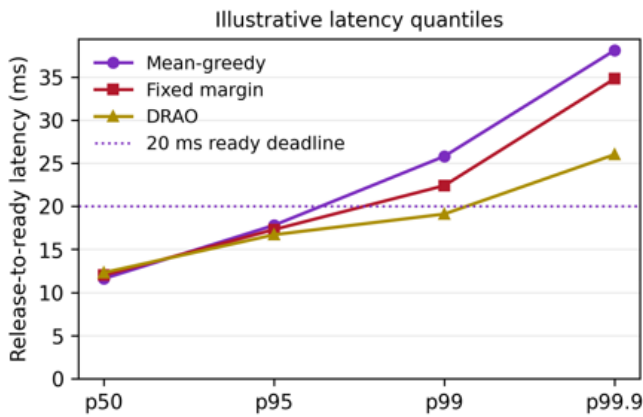


**Figure 3** Target-constrained reference release-to-ready quantiles for Table II. The dotted line marks the 20-ms budget. Connecting lines guide comparison; percentile spacing is categorical, not an empirical distribution.

Compound stress produces 4.74%, 3.12%, and 1.23%, respectively. DRAO has the lower prescribed rate in this constructed scenario but exceeds the nominal 1% target. This is not concealed as universal compliance: some frames have no sufficiently reliable execution path, and prediction error can increase during transitions. An adaptive controller cannot manufacture missing radio or compute capacity.

The reference 95th-percentile consecutive-miss burst lengths are four frames for fixed-margin and two for DRAO. These remain declared scenario targets, not a measured comfort outcome. Burst length counts consecutive missed frames and is distinct from single-frame latency. Aggregate rates, quantiles, and regime rates have explicit reporting-fixture checks; burst, age-sweep, and calibration targets are not claimed as independently established by those checks.

### Telemetry staleness and component ablation

Figure 4 specifies the reference age sweep. Fresh telemetry has target miss rates of 0.63% for age-blind risk and 0.60% for DRAO; at 20 ms, the targets are 1.92% and 0.89%, and at 40 ms they are 3.05% and 1.48%. These values encode the proposed freshness mechanism, rather than demonstrate it. The independent diagnostic tests the same mechanism and reports when the observed ordering does not match this reference scenario.

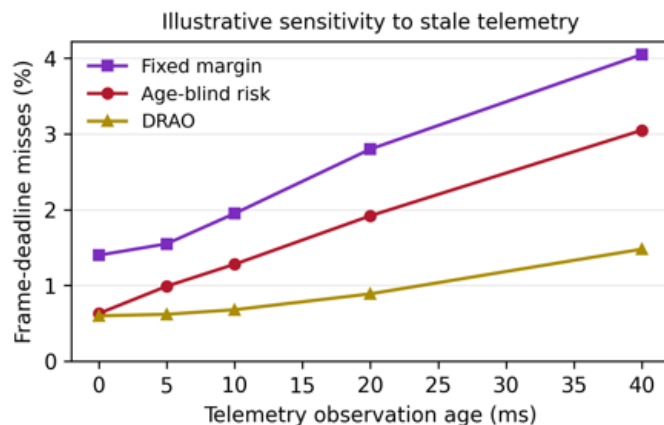


**Figure 4** Target-constrained reference sensitivity to telemetry age. The 5-ms reference matches Table II. The plotted values are prescribed scenario outcomes, not samples from a physical deployment.

All age values in this sweep lie within the specified 0–40-ms age range. The 40-ms result is therefore an in-range reference stress case,

not evidence of extrapolation to arbitrary reporting delays. Ages beyond this range trigger the support guard rather than being treated as calibrated probabilities. The guard can reduce reliance on stale edge observations, but it does not ensure that local computation will meet its deadline.

Table III preserves the reference ablation targets: miss rates of 0.99% without age, 0.91% without recalibration, 0.75% without hysteresis, and 0.88% when switching overhead is ignored. The reference switching fraction changes from 6.8% to 18.6% without hysteresis. Switching counts action changes between consecutive releases, not offloaded frames. These targets do not substitute for the executed age-blind and raw chance-constrained comparisons in Table 4.

The reference ten-bin, equal-count calibration-error targets are 0.82 and 0.18 percentage points before and after recalibration. They remain prescribed values, not an audit derived from measured traces or the independent diagnostic. Expected calibration error is a sample-weighted average discrepancy, not the uniform error bound in (10). The executed audit reports bin counts, policy-selected observed risks, and paired-session uncertainty near the 1% threshold; a small aggregate calibration statistic alone cannot certify that threshold.

### Deadline tightening, validity, and scope

At 120 Hz, the reference two-period lead and 2.22-ms reserve yield an approximately 14.45-ms ready budget. The reference miss rates remain 6.80% for mean-greedy, 4.90% for fixed-margin, and 2.60% for DRAO. These are separate target conditions, not re-estimated outputs from the 90-Hz diagnostic. They state the intended tighter-deadline behavior while making no claim that the executable 90-Hz study has validated it.

The reference fixture establishes arithmetic and reporting consistency only. The unforced diagnostic establishes controller behavior within an aggregate-service queue model only. Neither establishes live radio-service invariance, an implemented rendering partition, multi-user fairness, headset inference time, or physical motion-to-photon improvement. Distinguishing these scopes prevents agreement with a target table from being mistaken for empirical corroboration.

A physical evaluation requires an instrumented renderer, synchronized per-stage timestamps, explicit frame identifiers, and separate display-endpoint validation. Remote-rendering implementations provide relevant integration examples.<sup>11</sup> The available artifact contains synthetic generators and outputs, not a capture configuration or measured task profiles. Section IV.I defines the offered-load intervention needed before measured service opportunities may be treated as exogenous.

### Unforced policy comparison and tail uncertainty

Table 4 reports outputs from the independent queueing experiment. DRAO misses 0.91% of frames, compared with 2.30% for tuned fixed-margin and 2.14% for mean-greedy. The paired fixed-margin minus DRAO difference is 1.38 percentage points, with a 95% whole-session bootstrap interval of 1.26–1.51 points. The exact sign-randomization result is  $p = 0.00049$  under the session-level exchangeability null. These calculations concern 12 independently seeded synthetic sessions; they do not establish statistical significance for a physical deployment.

Stronger baselines change the interpretation. Raw chance-constrained placement has a 0.57% miss rate and age-blind risk has 0.55%, both below full DRAO. The leaf-tail comparator reaches

0.78%. Thus the executed experiment supports improvement over point-estimate placement within this model, but does not support a general claim that recalibration or age features must improve the selected actions. The reference ablation ordering in Table 3 is not substituted for these contrary diagnostic outcomes. Establishing an incremental benefit requires broader workloads, on-policy calibration checks, and stronger held-out comparisons.

The diagnostic p99.9 is 33.9 ms for fixed-margin and 24.4 ms for DRAO. Supplemental paired cluster resampling gives intervals of 29.5–39.6 ms and 24.0–25.0 ms, respectively; the paired difference is 9.4 ms with interval 4.9–15.5 ms. Approximately 65 observations lie beyond a nominal p99.9 threshold before dependence is considered. Accordingly, these tails are reported to one decimal place and with their broad uncertainty, not used to replace the separately constrained 34.8- and 26.0-ms reference values. Always-edge rejects 320 of 64,800 frames, so its unconditional p99.9 is not finite; a completed-frame-only quantile would hide that failure mass.

For the primary rate difference, nested block resampling yields intervals of 1.21–1.57, 1.23–1.56, and 1.22–1.56 percentage points for 1-, 5-, and 10-second blocks. These are dependence-sensitivity diagnostics rather than three independent confirmations. All intervals condition on the trained predictors and the chosen generator; they omit training-seed and model-form uncertainty. The protocol and seed are recorded in the artifact, but no external preregistration is claimed. Supplemental tail and timing calculations are identified as exploratory.

The threshold-local audit also exposes imperfect calibration. Among 31,694 selected actions assigned risk in [0.5%, 1.0%), mean predicted risk is 0.70%, while the observed miss frequency is 0.86% with a session-bootstrap interval of 0.79–0.91%. Among 10,773 decisions in [1.0%, 2.0%), the mean prediction is 1.16% and observed risk is 0.86% with interval 0.74–0.98%. Thus held-out isotonic fitting does not ensure conditional or policy-selected calibration. These executed findings are not the prescribed 0.82-to-0.18-point calibration-error scenario.

## Robustness, support failures, and runtime

The model is frozen before each stress condition. At an unseen 80-ms telemetry age, the range guard sends DRAO to local execution, resulting in a 6.67% miss rate; the guard prevents an unsupported edge-feasibility claim but does not restore the 1% target. Increasing task work by 50% produces 6.88% for DRAO versus 7.19% for mean-greedy. Multiplying the rendering suffix by 1.5 yields 1.53% versus 2.93%. These perturbations test workload and downstream-delay misspecification, not a new implemented rendering partition.

The backlog-dependent condition multiplies each selected path's service requirement by  $1 + 0.5q/20$ , where  $q$  is that path's current reserved backlog in milliseconds. It deliberately breaks the fixed-service assumption and evolves a separate load-dependent queue for every policy. DRAO reaches 0.95%, whereas fixed-margin reaches 37.33%; this is a stress-test outcome of the chosen feedback law, not a measured claim about server throttling. Under two-source arrival load, the period halves while service resources remain fixed: DRAO reaches 17.01%, compared with 9.18% for mean-greedy. This failure precludes a scalability claim. The shared-worker model does not evaluate per-user fairness.

Deadline cancellation changes primary miss rates to 2.10%, 2.11%, and 0.90% for mean-greedy, fixed-margin, and DRAO. Remaining service is released only at the deadline; already spent service is not refunded. Cancelled or rejected frames remain failures and have no finite readiness percentile. The artifact records the changed queue

evolution and noncompletion counts. The reference fixture, which assumes eventual completion, is never mixed with these cancellation records.

The JIT-warmed host implementation evaluates both action forests, calibration maps, support guard, and selection rule in batches of 12,600 decisions. Across 11 batches, median average cost is 0.0083 ms per decision, with batch averages from 0.0070 to 0.0093 ms on the recorded x86-64 container. These are batch averages, not per-decision tail bounds or headset measurements. The common 0.20-ms model charge is therefore retained. Source code, model arrays, frame logs, and exact software versions make the host result auditable without treating it as an embedded-device guarantee.

## Counterfactual validity and required physical experiments

In the synthetic diagnostic, exogenous potential service work is shared across policies by construction, while queues remain endogenous. That establishes a valid comparison inside the generator, not validity of replaying measured goodput. A physical acceptance experiment must randomize offered load at, for example, 25%, 50%, 75%, and 100% of the reserved operating budget while holding the reservation and environment fixed. Radio allocation, retransmissions, CPU frequency, thermal state, and independent probe work must be recorded. The equivalence margin should be declared before capture; a candidate rule is that the paired interval for normalized available service lies within  $\pm 5\%$ , not merely that a significance test fails to reject a difference.

If capacity changes with foreground load, the experiment must use a load-dependent network/server model or live closed-loop workload execution rather than replay the same goodput sequence as immutable capacity. The physical study must also implement at least two task partitions, variable transfer sizes, separate user queues, and clock-aligned frame identifiers; test fresh, stale, missing, and shifted telemetry; and measure full controller runtime on the actual endpoint. These are specified validation requirements, not experiments represented as already performed. The released synthetic evidence cannot satisfy them by generating more target-matched rows.

The proposed risk-feasible set is a chance constraint on the ready-deadline event. A distributional model and a monotone calibration map supply an estimated probability, but do not change that mathematical category. The narrower integration is to evaluate remaining dependent work and action-dependent switching costs with timestamped observations, then preserve the feasible set when applying hysteresis. Table 4 tests whether those additions help rather than presuming that their combination must be superior.

The bound in (10) is an elementary conditional implication: an action estimated below epsilon has true risk at most epsilon + eta only if a uniform estimation-error assumption already holds. It is not a new convergence theorem or an empirically certified guarantee. The same distinction separates telemetry age as a predictor from an age-minimization scheduling objective. Originality is limited to the specified integration and evaluation contract unless broader literature comparison or deployment evidence establishes more.

## Conclusion

This paper develops a concrete next step for latency-aware wireless VR: select computation placement according to frame-deadline risk rather than predicted mean delay alone. The proposed DRAO controller includes telemetry age, a conditional completion-time distribution, monotone recalibration, finite-resource accounting, and risk-preserving hysteresis. Its timing model distinguishes release-

to-ready deadlines from both refresh periods and physical motion-to-photon latency. A conditional error statement explains the value of accurate risk estimates without claiming deterministic guarantees.

The target-constrained reference fixture preserves the 60.0% miss-rate reduction, 25.3% p99.9 reduction, and 0.3-ms median increase as reproducible reporting constraints. Independently executed synthetic experiments show a 1.38-percentage-point miss-rate improvement over fixed-margin, with a paired 95% interval of 1.26–1.51 points, but do not establish superiority over raw chance-constrained or age-blind policies. Calibration shift, stale observations, and increased shared load expose limitations. The supported contribution is therefore a reproducible integration and a clearly bounded synthetic policy study. Physical measurements remain necessary before claiming measured-trace or deployment validation.

## Acknowledgment

None.

## Conflicts of interest

Author declare there is no conflicts of interest.

## References

1. Matthew Warburton, Mark Mon-Williams, Faisal Mushtaq, Measuring motion-to-photon latency for sensorimotor experiments with virtual reality systems. *Behavior Research Methods*. 2023;55:3658–3678.
2. Badnava B, Roach K, Cheung K, et al. Energy-efficient deadline-aware edge computing: Bandit learning with partial observations in multi-channel systems, in Proc. IEEE GLOBECOM, 2023.
3. Feng Q, Yang P, Huang Z, et al. Deadline aware two-timescale resource allocation for VR video streaming. 2023.
4. Paymard P, Paris S, Amiri A, et al. PDU-set scheduling algorithm for XR traffic in multi-service 5G-Advanced networks. 2023.
5. Badnava B, Chakareski J, Hashemi M. Learning driven elastic task multi-connectivity immersive computing systems. 2025.
6. The Khronos Group, XrFrameState, OpenXR Specification. 2026.
7. 3GPP, Study on scenarios and requirements for next generation access technologies. 2024.
8. Kaul S, Yates R, Gruteser M. Real-time status: How often should one update? in Proc. IEEE INFOCOM, 2012. p. 2731–2735.
9. Meinshausen N. Quantile regression forests. *Journal of Machine Learning Research*. 2006;7:983–999.
10. Kuleshov V, Fenner N, Ermon S. Accurate uncertainties for deep learning using calibrated regression. In: 35th International Conference on Machine Learning, PMLR. 2018;80:2796–2800.
11. Mejías D, Yeregui I, Viola R, et al. Remote rendering for virtual reality: Performance comparison of multimedia frameworks and protocols. 2025.
12. Liu CF, M Bennis, HV Poor. Latency and reliability-aware task offloading and resource allocation for mobile edge computing. 2017.
13. Lyu X, Ni W, Tian H, et al. Optimal schedule of mobile edge computing for Internet of Things using partial information. *IEEE Journal on Selected Areas in Communications*. 2017;35(11):2606–2615.