GR JOURNALS | Journal of Information and Communications Technology: Algorithms, Systems and Applications

Volume 2 Issue 3 (September 2026)
DOI: https://doi.org/10.64189/ict.26315
J. Inf. Commun. Technol. Algorithms, Syst. Appl., 2026, 2(3), 26315

Research Article

OPEN ACCESS

Delay-Aware Reinforcement Coding for Multi-Hop Narrowband Wireless Mesh Networks: Joint RTT, Redundancy, and Forwarding Optimization

El Miloud Ar-Reyouchi[1,2,*] and Abdeljalil Alaoui Douiri[2]

[1] *Department of Informática y Automática, ETSI Informática, UNED, Madrid, 28040, Spain*
[2] *Société Nationale de Radiodiffusion et de Télévision (SNRT), Rabat, Morocco*

*Corresponding Author
El Miloud Ar-Reyouchi
e.arreyouchi@m.ieice.org

Received: 10 August 2026
Revised: 31 August 2026
Accepted: 10 September 2026
Published Online: 15 September 2026

Citation

El M. Ar-Reyouchi, A. A. Douiri, Delay-aware reinforcement coding for multi-hop narrowband wireless mesh networks: joint RTT, redundancy, and forwarding optimization, *Journal of Information and Communications Technology: Algorithms, Systems and Applications,* 2026, 2(3), 26315, https://doi.org/10.64189/ict.26315.

Abstract

Multi-hop narrowband wireless mesh networks remain useful for telemetry and supervisory control, but tail round-trip time (RTT) is strongly coupled to packet-loss probability, queue pressure, hop count, message size, retransmission, and coding overhead. In this paper Delay-Aware Reinforcement Coding (DARC), a finite Markov decision process controller that jointly selects fast-forward or store-and-forward relaying, random linear network coding (RLNC) generation size, and coded-packet redundancy, is proposed. The state includes four three-level features-end-to-end uncoded packet-loss probability, load/queue pressure, hop count, and message size-yielding 81 contexts and 12 actions (972 Q-values). DARC minimizes a fully normalized objective by combining empirical CVaR95 tail latency, residual loss, coding overhead, and goodput deficit. The load transition is action-dependent through a useful service rate, providing a long-horizon motivation for Q-learning; a contextual UCB bandit, a model-informed greedy controller, an adaptive heuristic, fixed policies, and an exact MDP oracle are included as baselines. Ten independent learning seeds are evaluated over 120,000 transitions with 95% confidence intervals. Under the exact discounted MDP benchmark, DARC achieves a normalized cost of 0.3364, 2.2% below the store-and-forward cost and within 0.29% of the exact oracle. In the representative five-hop, 12% end-to-end packet-loss condition, the learned policy selects uncoded fast-forward switching to preserve 12.25 kb/s goodput, while reliability-biased RLNC lowers p95 RTT at a substantial goodput cost. The results support DARC as a reproducible adaptive-control hypothesis; formal RF calibration remains pending the availability of raw traces.

Keywords: Wireless mesh networks; Random linear network coding; Reinforcement learning; Tail latency; Narrowband wireless communications.

1. Introduction

Practical narrowband telemetry networks exhibit a basic but difficult coupling: message size determines serialization time, hop count multiplies service and recovery exposure, packet loss drives retransmission, and queue pressure magnifies the upper RTT tail. Earlier practical and analytical studies quantified these effects in wireless mesh, narrowband, network-coding, and Internet-of-Things settings.[1–7] The operational question addressed here is therefore not whether fast-forward switching or network coding can help in isolation, but when each mechanism should be activated as conditions change.

Classical RLNC and practical wireless coding provide strong foundations for adaptive coded transmission,[8–11] while Q-learning provides the basic learning mechanism.[12] Contextual bandit methods provide a myopic adaptive comparator,[13] and CVaR and retransmission-timer concepts motivate explicit treatment of tail latency and recovery timing.[14,15] The broader routing, queueing, and recent reinforcement-learning literature is reviewed in Section 2. These foundations reinforce two requirements that are central here: the learning problem must be tied to an explicit system state transition, and the comparison set must include adaptive-not only fixed-baselines.

A contextual bandit is sufficient when an action influences only the immediate reward. DARC instead models queue/load evolution as action dependent: a coding/forwarding choice changes the useful service rate and hence the probability that the next load state rises, remains stable, or falls. This converts the controller into a finite MDP. The contextual UCB rule of Auer *et al.*[13] is nevertheless retained as a strong myopic learning baseline, allowing the data to reveal whether long-horizon state coupling materially changes the decision.

Research question. Can a compact reinforcement-learning controller jointly adapt forwarding mode, RLNC generation size, and redundancy to reduce tail latency while maintaining reliability and useful goodput under changing narrowband multi-hop network conditions?

Methodological novelty boundary: DARC does not introduce a new RLNC coding theorem or a new Q-learning update. Its contribution is the joint, interpretable, state-aware control architecture: forwarding mode, RLNC generation size, and coded-packet redundancy are selected together from an 81-state context; the action affects the subsequent load/queue transition; and the objective explicitly combines CVaR95 tail latency, residual loss, coding overhead, and useful-goodput deficit. This differs from fixed coding or forwarding, RL-based coding that does not jointly adapt forwarding in the present comparison, and a contextual-bandit controller that omits next-state value.

The main contributions are as follows:

An 81-state representation that includes packet loss probability, load/queue pressure, hop count and message size while retaining a compact 972-entry Q-table.

An action-dependent transition model in which the useful service rate affects the next queue/load state, providing an explicit P(s_{t+1}|s_t,a_t) and a defensible MDP/Q-learning formulation.

A CVaR95 tail-latency estimator using a final 256-packet window, 256-packet warm-up and 32-packet update stride is used; the choice is justified from the number of samples available in the 5% tail and is combined with residual loss, coding overhead and goodput deficit after every term is normalized to [0,1].

An explicit RLNC model over GF(2^8), including finite-field rank probability, coefficient-header overhead, source-only coding, generation-level cumulative ACK and rank-deficiency recovery.

A stronger benchmark suite: store-and-forward, fast-forward, reliability-biased RLNC, adaptive heuristic, contextual UCB, greedy one-step control and an exact value-iteration MDP oracle.

Multi-seed learning-stabilization diagnostics over 120,000 transitions, confidence intervals, a full learned-policy atlas, relative p95 gains, dual-axis-free ablation, message-size sensitivity, Pareto analysis and objective-weight sensitivity.

A reproducibility package with code, transition and reward matrices, learned policy arrays, CSV data, and an empirical-calibration template.

For traceability, Table 1 positions DARC against representative alternatives. The closed-loop MDP architecture is defined in Fig. 1, and Algorithm 1 gives the online controller. Table 2 lists the state-bin boundaries, and Table 3 summarizes the state, action, learning, normalization, and transition parameters. Fig. 2 shows the results of the multi-seed learning diagnostics; Fig. 3 shows all 81 learned decisions; and Figs. 4–10 present the distributional, scaling, robustness, ablation, message-size, Pareto, and weight-sensitivity results. Tables 4–7 summarize representative baseline performance, exact discounted-policy performance, a dynamic-workload stress test, and novelty-to-evidence traceability.

2. Related work and research gap

Earlier narrowband experiments by the authors,[1,2,5,6] together with independent studies of forwarding, network coding, capacity, and delay in multi-hop wireless networks,[3,4,7] measured or analyzed how RTT/latency and capacity vary with message size, route length, forwarding behavior, and coding/redundancy. In the present study, the authors' prior findings are used only to choose physically meaningful state variables and operating ranges; the independent literature provides

the broader comparison. Work on RLNC and wireless coding[8–11] establishes the basic trade-off between coded redundancy and recovery. ETX, ExOR, and low-power routing standards [16–21] address path quality, opportunistic forwarding, and scheduled communication. Queueing and stochastic network optimization [22,23] motivate representing load evolution and action-dependent service. The 6TiSCH minimal profile[24] and hard-real-time scheduling results[25] further motivate reproducible scheduling assumptions and examination of deadlines and tail behavior rather than relying on mean throughput alone.

Recent studies provide several points of comparison. Learning-based network coding has been examined in drone-assisted secure links and for selective RLNC in tactile-Internet scenarios.[26,27] Deep reinforcement learning has also been used for link adaptation,[28] multi-hop offloading,[29] delay-aware resource allocation in industrial IoT,[30] energy-aware forwarding,[31] and federated wireless control.[32,33] Nuwa-RL focuses on application-sensitive congestion control in dynamic wireless networks.[34] Other works have evaluated RL rate adaptation for time-critical Wi-Fi with a calibrated simulator[35] and multi-timescale DRL for delay reduction in dynamic resource allocation.[36] Belief-state RL has recently been proposed for latency-robust wireless control when observations are delayed or uncertain.[37] These contributions support adaptive decision making, but they do not combine the narrowband variables considered here-message size, forwarding mode, RLNC generation size, and redundancy-under a CVaR tail-latency objective.

The 2024–2026 literature is strengthened by recent studies on delay-aware industrial-IoT resource control, adaptive forwarding, federated and multi-timescale wireless reinforcement learning, and latency-robust digital-twin control.[30–32,36,37] In addition, recent work on offline/distributional reinforcement learning for risk-sensitive wireless optimization[38] and on digital-twin-driven reinforcement-learning reliability and resource-management transfer[39,40] highlights the importance of uncertainty, tail behavior, and model-to-deployment mismatch. These studies broaden the contemporary comparison while leaving the paper's narrower novelty claim unchanged.

The gap addressed by DARC is therefore specific. We ask whether a compact, auditable tabular controller can coordinate forwarding, generation size, and redundancy from four observable network descriptors while also representing the effect of the current action on subsequent queue conditions. The oracle, greedy controller, and contextual bandit are included as checks on that premise. If a myopic method performs as well as the exact MDP, the comparison should reveal it; no advantage from long-horizon reinforcement learning is assumed beforehand.

3. System model and MDP formulation

3.1 81-state context and 12-action controller

Consider a source sending traffic to a supervisory endpoint over H wireless hops. At control interval t, the gateway provides four quantities to DARC: the estimated uncoded end-to-end packet-loss probability p_t, the load/queue-pressure ratio ρ_t, the active route length H_t, and the pending application message size B_t. Each quantity is mapped to one of three bins. Operationally, ρ_t is a dimensionless load/queue-pressure indicator: smaller values denote lightly loaded service, whereas larger values indicate offered demand and backlog pressure approaching the available useful service capacity. It is a coarse gateway control observable rather than an exact queue-length measurement.

$$s_t = (b_p, b_\rho, b_H, b_B),\ b_i \in \{0,1,2\};\ |S| = 3^4 = 81 \quad (1)$$

Each control action specifies three choices: RLNC generation size g∈{1,2,4}, whether to send one extra coded packet r∈{0,1}, and the forwarding mode f∈{S&F,FFS}: Throughout the manuscript, S&F denotes store-and-forward switching, and FFS denotes fast-forward switching.

$$a_t = (g_t, r_t, f_t),\ g \in \{1,2,4\}, r \in \{0,1\}, f \in \{\text{S\&F}, \text{FFS}\};\ |A| = 3 \times 2 \times 2 = 12;\ |S||A| = 972 \quad (2)$$

Table 1: Positioning of DARC relative to representative approaches

Approach	Adaptive coding	Forwarding adaptation	Tail objective	Long-horizon state
Fixed narrowband routing	No	No	RTT/mean	No
Fast-forward	No	Fixed FFS	RTT/mean	No
Fixed RLNC	Fixed	Fixed	Indirect	No
RL-based network coding [26,27]	Yes	Not joint here	Application-specific	Method-specific
DRL link/resource adaptation [28–30]	Yes/indirect	Resource-level	Delay/QoS	Yes
Contextual UCB baseline [13]	Yes	Yes	Immediate normalized cost	No
DARC (proposed)	Yes	Yes	CVaR95 + reliability + efficiency	Yes, queue/load

Table 2: State-bin boundaries used in the 81-state context

State variable	Bin symbol	Low bin (0)	Medium bin (1)	High bin (2)
Raw uncoded end-to-end packet-loss probability	b_p	< 6%	6%-13%	> 13%
Load/queue-pressure indicator	b_ρ	< 0.50	0.50-0.75	> 0.75
Active route length	b_H	2-3 hops	4-5 hops	6-7 hops
Pending application message size	b_B	≤ 256 bytes	257-512 bytes	> 512 bytes

The simulation uses loss bins <6%, 6–13%, and >13%; load bins <0.50, 0.50–0.75, and >0.75; hop bins 2–3, 4–5, and 6–7; and message-size bins ≤256, 257–512, and >512 bytes. The physical payload rate is 19.2 kb/s. Unlike an instantaneous cost input, the message size is part of the state and can therefore affect the selected action. Table 2 summarizes the four quantization boundaries in one place for implementation reference.

3.2 RLNC, finite-field rank and recovery model

The simulator defines p as the raw end-to-end uncoded packet-loss probability before RLNC or ARQ. If per-hop packet-loss probabilities p_h are measured instead, the equivalent end-to-end uncoded packet-loss probability is obtained from Eq. (3).

Unless stated otherwise, packet erasures are conditionally independent Bernoulli events at the packet level for a given state value p. Equation (4) therefore uses a binomial reception model. Temporal correlation, burst errors, and channel-memory effects are not included in the present numerical model; they are treated as an empirical-validation extension rather than being hidden inside the reported loss probability.

$$p_{e2e} = 1 - \prod_{h=1}^{H}(1 - p_h);\ \text{equal links: } p_{e2e} = 1 - (1 - p_h)^H \quad (3)$$

RLNC operates over GF(2^8). For a generation of g source packets and r extra coded packets, a receiver can decode when at least g packets arrive and their coefficient matrix has full rank. The finite-field rank correction is included explicitly:

$$P_{dec}(g,r,p) = \Sigma_{k=g}^{g+r} C(g+r,k)(1-p)^k p^{g+r-k} \prod_{i=0}^{g-1}(1-256^{i-k}) \quad (4)$$

Relays do not recode in this simulator; RLNC is performed only at the source. The pair (g,r)=(1,0) is the uncoded case and bypasses the encoder; thus, it has no RLNC coefficient header. Whenever the coding is active (g>1 or r=1), each coded packet includes a 4-byte generation/rank field and g one-byte GF(2^8) coefficients. This distinction is reflected in Eq. (5), so the uncoded baseline is not charged coding overhead that it never transmits.

$$\text{If } (g,r)=(1,0): L_c = B \text{ and } n_{tx} = 1. \text{ Otherwise: } L_c = B/g + 4 + g \text{ bytes and } n_{tx} = g + r \text{ before rank-deficiency recovery} \quad (5)$$

The receiver sends one cumulative generation ACK. If rank is insufficient, the source sends additional innovative coded packets. This generation-level ACK/ARQ abstraction is deliberately stated because packet-by-packet ARQ or relay recoding would change both the overhead and the delay.

3.3 Action-dependent queue transition and Markov property

The key correction relative to a contextual-bandit formulation is that the selected action changes the useful service rate μ(s_t,a_t). Let λ_t denote the offered useful-bit demand and Q_t denote a normalized backlog indicator. A conceptual queue update is

$$Q_{t+1} = \max\{0,\ Q_t + A_t - S(s_t,a_t)\};\quad mu(s_t,a_t) = E[S(s_t,a_t)]/Delta_t \quad (6)$$

The discrete transition kernel uses the service-to-demand ratio μ/λ to update the load bin. When μ/λ>1.10, the probabilities of moving down, staying, or moving up are 0.70, 0.25, and 0.05, respectively. When μ/λ<0.82, they are 0.05, 0.25, and 0.70; the intermediate case is 0.15, 0.70, and 0.15. In the lowest and highest bins, the probability of moving outside the range is added to the boundary bin. The loss, route length, and message size follow slower exogenous nearest-neighbor kernels: 0.80 probability of remaining and 0.10 for each adjacent move, with the same boundary handling. Together, these rules define the transition law in Eq. (7) without the need for an additional hidden parameter.

$$P(s'|s,a) = P_p(b'_p|b_p)\ P_H(b'_H|b_H)\ P_B(b'_B|b_B)\ P_{rho}(b'_{rho}|s,a) \quad (7)$$

Because P_ρ depends on a through the useful service rate, the current coding/forwarding decision changes future congestion risk. This finding highlights the long-horizon coupling required for Q-learning. The contextual UCB baseline intentionally removes this next-state term and optimizes only the immediate reward.

Markov approximation and route changes: The four-feature state is a controlled Markov approximation, not a claim that the physical radio network is exactly Markovian. The transition kernel assumes that the binned loss, load, route length, and message size contain the information needed by the simulator for the next control step. Unobserved channel memory, exact queue

occupancy, interference, and protocol timers can violate this approximation and can reduce policy quality after deployment. Route length H is allowed to move through the exogenous nearest-neighbor kernel, but the routing protocol, route-repair delay, and abrupt topology churn are not explicitly modeled; rapidly changing routes therefore lie outside the validated simulation envelope.

3.4 Tail-latency estimator and CVaR95 objective

Tail latency is evaluated from a rolling RTT sample window. A 95% tail contains only about ceil(0.05W) observations: approximately 4 samples for W=64, 7 for W=128, and 13 for W=256. To reduce tail-estimator sparsity, the final experiments use W=256, a 256-packet warm-up, and a 32-packet update stride. For RTT observations $T_1,\ldots,T_W$, the empirical 95th percentile is

$$VaR_{0.95} = \inf\{x : (1/W)\ \Sigma_{i=1\ldots W}\ I(T_i <= x) >= 0.95\} \quad (8)$$

DARC optimizes the conditional value-at-risk rather than only the quantile point. The empirical CVaR95 is the mean of samples at or beyond the empirical VaR threshold:

$$CVaR_{0.95} = (1/N_{tail})\ \Sigma_{[i\ :\ T_i >= VaR_{0.95}]}\ T_i \quad (9)$$

CVaR is used because it responds to the magnitude of extreme delays, not only the 95th-percentile crossing.[14] The sliding-window RTT estimator follows established timer and loss-detection principles in explicitly separating observation windows from retransmission decisions.[15] This is especially appropriate when recovery events create a heavy upper tail.

3.5 Fully normalized multi-objective cost and Q-learning

To prevent RTT from dominating the reward numerically, every component is clipped and normalized to [0,1]. The exact definitions are d_norm=clip[(CVaR95−0.50 s)/(6.00 s−0.50 s),0,1], e_norm=clip[p_res/0.25,0,1], o_norm=clip[O_c/1.50,0,1], and g_def=clip[1−G/R,0,1], where p_res is the residual delivery-loss probability, O_c=(transmitted coded bytes−source bytes)/source bytes, G is the useful goodput and R is the physical payload rate. These constants cover the simulated operating envelope and are listed in Table 3 so that Eq. (10) is dimensionless and directly reproducible. Here, p is the raw uncoded loss presented to the channel model, whereas p_res is the residual delivery-loss probability after the selected coding/recovery action. G counts successfully delivered application payload bits per second: coded/retransmitted packets and ACK bytes are not credited as useful goodput. Coding and recovery transmissions consume simulated airtime and affect delay/overhead, whereas cumulative ACK traffic is represented by the generation-level ACK/ARQ timing abstraction rather than being added to G.

$$J(s,a) = 0.45\ d_{norm} + 0.25\ e_{norm} + 0.15\ o_{norm} + 0.15\ g_{def};\ \text{all terms are in } [0,1] \quad (10)$$

The reward is r_t=−J(s_t,a_t). DARC uses the standard tabular temporal-difference update

$$Q_{next} = Q + \alpha\,[r + \gamma \max_{a'} Q(s',a') - Q],\ \gamma = 0.82 \quad (11)$$

For comparison, contextual UCB treats each context independently and selects an action from an optimistic immediate-reward estimate.[13] In all reported UCB experiments, the exploration coefficient is β=0.10:

$$a_t = \arg\max_a\ [\hat{\mu}_a(s_t) + \beta \sqrt{2 \ln n_s / N_{(s,a)}}] \quad (12)$$

The interaction among the measured mesh conditions, the 81-state observations, tabular Q-learning, the joint action and action-dependent queue feedback is summarized in Fig. 1. The diagram is intentionally presented as a conventional engineering block schematic

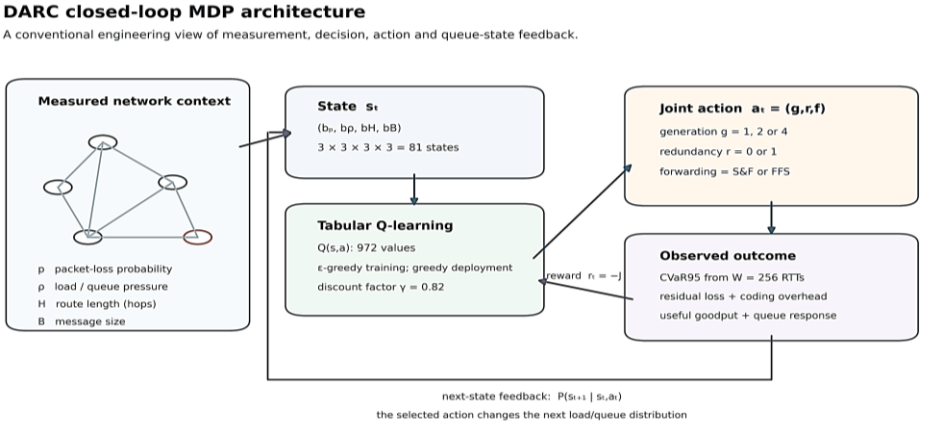

Fig. 1: DARC closed-loop MDP architecture. The measured packet loss, queue/load pressure, route length and message size form the state; the selected coding/forwarding action determines the immediate performance and influences the next load/queue state.

to make the implementation path explicit.

4. Proposed DARC algorithm

At each control interval, the gateway estimates the raw uncoded packet-loss probability from recent delivery/ACK statistics, calculates the load/queue pressure, obtains the active route length and reads the pending message size. The four values are quantized to the 81-state context. During offline training, ε-greedy exploration is used with uniform generative sampling of the 81 contexts to avoid poorly visited state bins; after training, the greedy action is applied. The RTT/CVaR reward is evaluated on the defined window, and the selected action remains active for one control interval to preserve causal attribution.

Algorithm 1: Delay-Aware Reinforcement Coding (DARC)

Step	Operation
Input	recent packet-loss probability p, load/queue ratio ρ, hops H, message size B, Q-table Q
1	Quantize (p,ρ,H,B) to s=(b_p,b_ϱ,b_H,b_B).
2	Choose a=(g,r,f) by ε-greedy exploration during training; use $\text{argmax}_a Q(s,a)$ during inference.
3	If (g,r)=(1,0), bypass RLNC; otherwise configure source RLNC over GF(2^8), redundancy r, and S&F/FFS for one control interval.
4	Collect RTT samples; after a 256-packet warm-up maintain W=256 and update tail statistics every 32 completions.
5	Compute VaR95 and CVaR95, residual loss, overhead, useful goodput, and normalized cost J using Eq. (10).
6	Observe next state s′; its load component depends on the action-modified service rate according to Eq. (7).
7	Update Q(s,a) using the Q-learning recursion in Eq. (11).
8	Repeat; after convergence optionally freeze the policy and retain a safe fixed fallback.

The algorithm therefore makes the equation reference explicit: the learning update is Eq. (11), not the RLNC delivery equation. Its online complexity is O(12) for greedy action selection, and the stored table contains only 972 values.

5. Simulation methodology and reproducibility

5.1 Experimental envelope and statistical protocol

The simulator evaluates routes from two to seven hops, application messages from 128 to 1024 bytes, raw end-to-end uncoded packet-loss probability from 1% to 22%, and offered load/queue pressure from 0.25 to 0.90. The reference condition is five hops, 512 bytes, a 12% end-to-end packet-loss probability and a load of 0.75. The environment stores an explicit 81×12 immediate-reward matrix and an 81×12×81 transition tensor. Ten independent Q-learning seeds are trained for 120,000 transitions. Offline training samples the 81 contexts uniformly from the generative simulator, while the next states are drawn from Eq. (7); this improves coverage without changing the Bellman target. Static result figures use independent repeated samples with 95% confidence intervals; the exact discounted benchmark is computed directly from the transition matrix, avoiding Monte Carlo ambiguity in the oracle comparison. The uniform distribution is also used as d_0 for the exact discounted comparison, so the oracle gap is a state-space average rather than a result conditioned on one favorable start state. Table 3 provides the parameter rationale, including the learning schedules, objective normalization constants, transition thresholds, and the exact-evaluation initial distribution.

5.2 Baselines

Seven comparison policies are used in addition to DARC: (i) uncoded store-and-forward; (ii) uncoded fast-forward; (iii) fixed reliability-biased RLNC with g=2, r=1 and FFS; (iv) a threshold heuristic that increases coding under loss/load; (v) per-context UCB, which is a true contextual bandit and ignores next-state value; (vi) a model-informed greedy one-step benchmark that selects the action minimizing the simulator's expected immediate cost matrix and therefore is not presented as a directly deployable online learner; and (vii) an exact value-iteration oracle over the same 81-state MDP. The oracle is not deployable because it assumes that the transition model is known, but it provides an upper bound on the achievable discounted performance within the stated model.

5.3 Calibration status and protocol

The present revision does not fabricate empirical calibration. The authors' historical narrowband publications [1,2,5,6] motivate the radio rate, message-size and hop-count envelope, but the raw timestamped traces needed for a defensible RMSE/MAE/R^2 calibration were not supplied with this submission draft. The supplementary package therefore includes empirical_calibration_template.csv. Once measured mean/p95 RTT points are inserted, the simulator can be fitted on serialization, processing and error parameters and compared by RMSE, MAE and R^2. Until that step is completed, all new numerical results must be interpreted as a reproducible digital-model study rather than new over-the-air evidence.

Table 3: Model, learning, normalization, and transition parameters with rationale

Parameter	Value / range	Rationale / role
Radio payload rate	19.2 kb/s	Narrowband serialization
State variables	3 loss × 3 load × 3 hops × 3 size	81 contexts
Action variables	g={1,2,4}; r={0,1}; S&F/FFS	12 actions
Q-table size	972 values	Gateway-scale interpretability
Raw loss	0.01–0.22 uncoded end-to-end packet-loss probability	Channel impairment
Route length	2–7 hops	Network scaling
Message size	128–1024 bytes	Serialization sensitivity
Load/queue pressure	0.25–0.90	Congestion
Finite field	GF(2^8)	RLNC coefficients
Tail window	256 packets; warm-up 256; stride 32	CVaR95 estimation (~13 tail samples)
Objective weights	0.45 / 0.25 / 0.15 / 0.15	Tail-emphasized generic telemetry scalarization; not an application-specific optimum; Fig. 10 tests alternatives
Learning seeds	10 independent seeds	Learning uncertainty
Discount factor	γ=0.82	Finite look-ahead for action-dependent queue effects while retaining strong immediate-cost influence
Training horizon	120,000 transitions	Extended stabilization assessment
Q initialization	$-0.36/(1-\gamma)$, uniform	Uniform prior return for reward-maximizing Q
Learning-rate schedule	$\alpha_t=0.85/(1+N(s,a))^{0.62}$	Diminishing state-action step size for stabilization across repeated visits
Exploration schedule	$\varepsilon_t=\max[0.02, \exp(-t/18{,}000)]$	Broad early exploration with a 2% floor to avoid deterministic lock-in during training
Offline context coverage	Uniform sampling over 81 contexts	Uniform state coverage during training to avoid poorly visited quantized contexts
Stabilization criterion	final Q-change <1%; policy-change <5%	Diagnostic, not a theorem
State-bin boundaries	Table 2	Explicit, reproducible quantization of p, ρ, H, and B
Normalization constants	CVaR: 0.50-6.00 s; residual loss max 0.25; overhead max 1.50	Cover the simulated envelope and keep all objective terms dimensionless in [0,1]
UCB exploration coefficient	β = 0.10	Fixed exploration strength for the myopic contextual-bandit comparator
Load transition thresholds	$\mu/\lambda > 1.10$; $\mu/\lambda < 0.82$; otherwise intermediate	Separates service-surplus, overload, and near-balance regimes
Load-bin transition probabilities	down/stay/up = 0.70/0.25/0.05, 0.05/0.25/0.70, or 0.15/0.70/0.15	Encodes directional queue response to the action-modified service-to-demand ratio
Exogenous nearest-neighbor kernel	stay 0.80; adjacent moves 0.10 each; boundary mass folded	Represents slower changes in loss, route length, and message size
Exact benchmark initial distribution	Uniform over all 81 states	Neutral state-space average for the exact discounted criterion

Note: *Parameter selection is intentionally transparent rather than empirically fitted: the normalization constants span the simulated operating envelope, the transition thresholds encode service-surplus/overload directions, and the learning schedules are used as finite-simulation design choices. The default weights (0.45 tail, 0.25 residual loss, 0.15 overhead, 0.15 goodput deficit) represent a tail-emphasized generic telemetry objective, not a universal application preference; Section 6.8 reports sensitivity to alternative weight profiles.*

6. Results and Discussion

6.1 Multi-seed learning behavior

The learning-stabilization diagnostics across ten independent seeds over an extended 120,000-transition horizon are shown in Fig. 2. The three panels show fixed-validation normalized cost, relative Q-table change and the fraction of contexts whose greedy action changes between checkpoints. At the final checkpoint, the mean

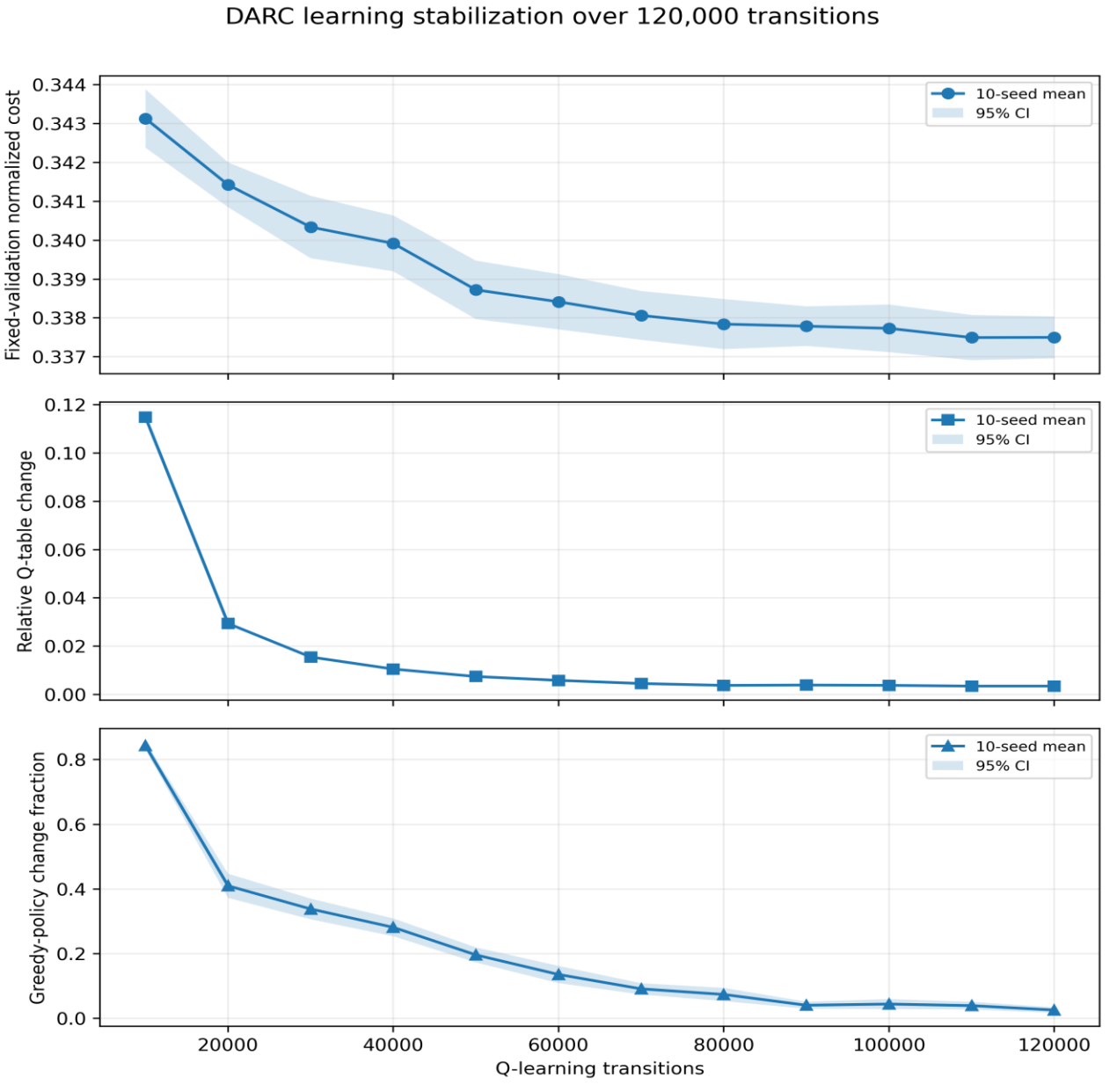

Fig. 2: Ten-seed DARC learning-stabilization diagnostics over 120,000 transitions: fixed-validation normalized cost, relative Q-table change, and greedy-policy change fraction with 95% confidence intervals.

relative Q-table change is approximately 0.0035, and the mean greedy-policy change fraction is approximately 0.026, both below the predeclared 1% and 5% stabilization thresholds. The figure is therefore described as stabilization evidence rather than a proof of asymptotic convergence; confidence bands expose seed-to-seed uncertainty.

6.2 Learned-policy interpretability

The full 81-context policy is shown in Fig. 3 using three enlarged discrete panels. The generation size, redundancy and forwarding are separated rather than compressed into a single multicolor map. Rows are encoded as packet-loss-bin/hop-bin pairs (P1/H1 through P3/H3), while columns are load-bin/message-size pairs (L1/B1 through L3/B3); the bin values are printed below the atlas. This layout makes individual decisions readable at normal manuscript zoom.

6.3 RTT distribution and hop-count scaling

The empirical RTT CDF at the representative operating point is shown in Fig. 4. The redesigned figure adds an upper-tail zoom so that the small differences among the uncoded/adaptive strategies remain visible near the 95th-percentile region. Reliability-biased RLNC shifts the distribution left because it reduces recovery exposure, but that benefit is obtained with additional coded airtime. DARC selects g=1, r=0 and FFS at this moderate-loss context, explicitly prioritizing useful throughput over heavy protection.

The data in Fig. 5 strengthen the hop-scaling analysis by adding 95% confidence intervals, and the second

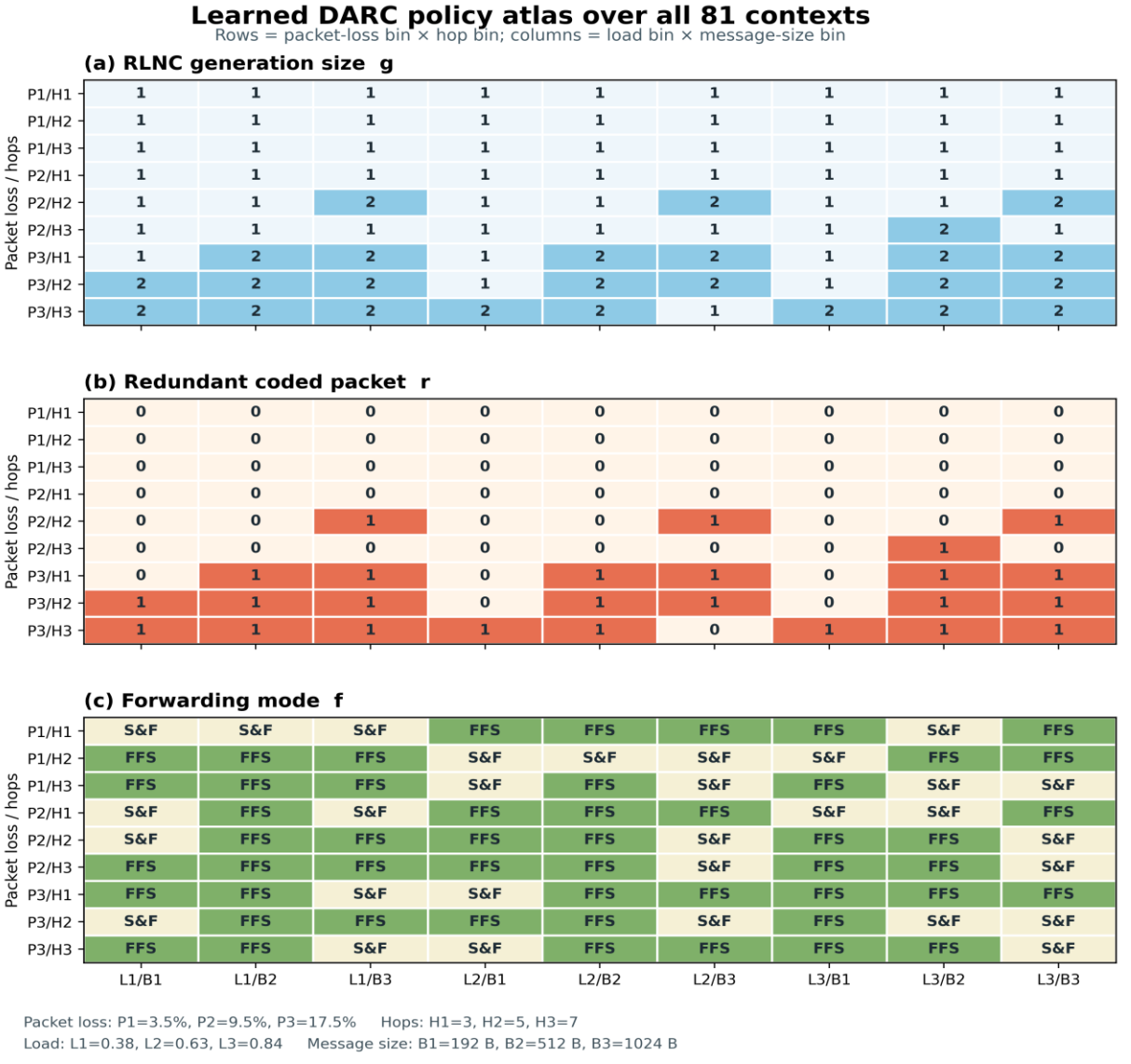

Fig. 3: Enlarged DARC policy atlas over all 81 states: (a) RLNC generation size, (b) redundant coded-packet choice, and (c) forwarding mode. The compact P/H and L/B axis keys are expanded below the panels.

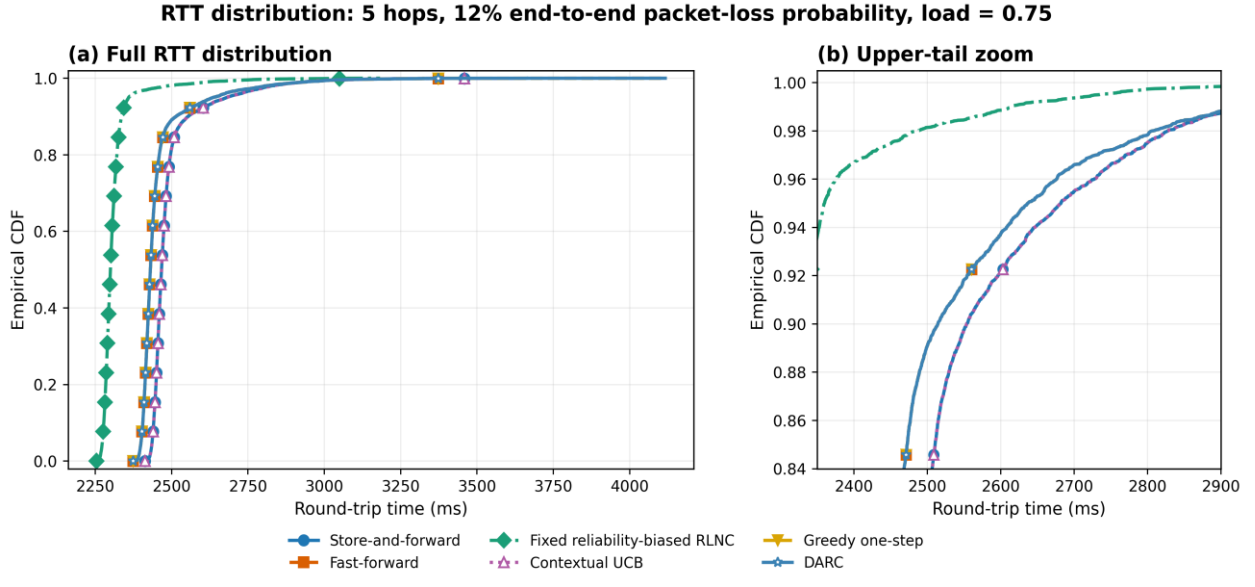

Fig. 4: Empirical RTT CDF at five hops, 12% raw end-to-end packet-loss probability and a load of 0.75: (a) full distribution and (b) enlarged upper-tail region.

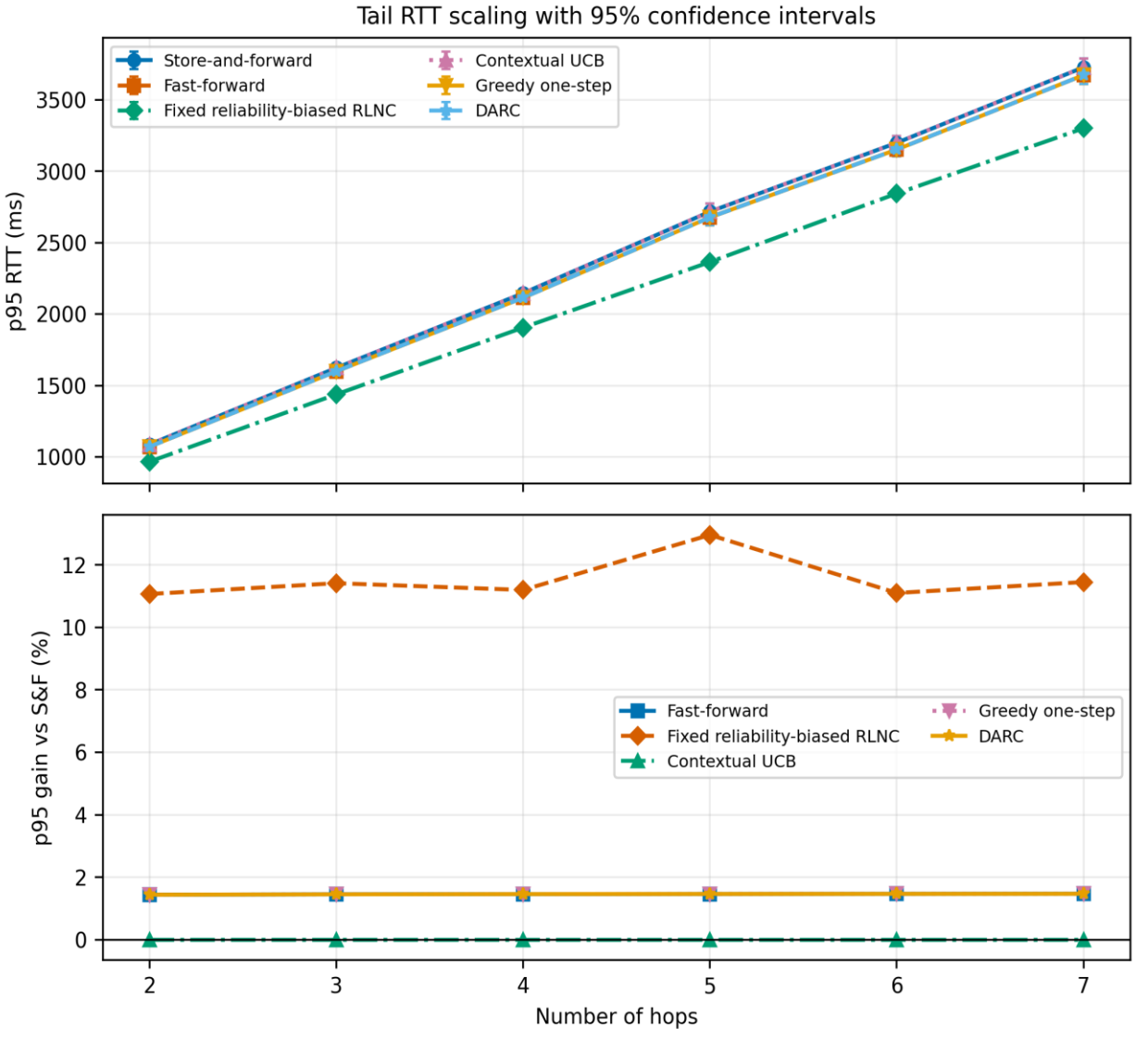

Fig. 5: p95 RTT versus hop count with 95% confidence intervals and relative p95 gain against store-and-forward.

panel shows the relative p95 gain against the store-and-forward process. The absolute curves can be visually close because serialization is dominant; the relative-gain panel reflects the smaller algorithmic contribution around that physical floor.

6.4 Channel-loss robustness and ablation

The useful goodput as the raw end-to-end packet loss probability changes is shown in Fig. 6. The upper panel reports absolute goodput with 95% confidence intervals, while the lower panel reports the relative difference against store-and-forward. Store-and-forward and fast-forward overlap in goodput by construction because the forwarding mode changes relay processing delay but not radio serialization, coding overhead or residual delivery probability in the present abstraction. The relative panel makes this zero-difference behavior explicit instead of allowing overlapping curves to be mistaken for a plotting error.

The ablation in Fig. 7 avoids a dual y-axis. p95 RTT and goodput are normalized separately to the store-and-forward baseline and displayed in two aligned panels. Fast-forward switching removes processing time without changing payload goodput, whereas RLNC trades a useful rate for a lower recovery tail. Full DARC chooses among these mechanisms rather than assuming one permanent configuration.

6.5 Message size, Pareto behavior and representative comparison

The direct serialization effect is shown in Fig. 8. Increasing B increases the time required to place a message on a 19.2-kb/s narrowband link; thus, physical airtime accounts for a larger share of RTT as messages grow. By including the message size in the state, the controller can distinguish between operating regimes with different latency floors. DARC can influence queueing, relay processing, and recovery behavior, but it cannot eliminate the serialization time itself.

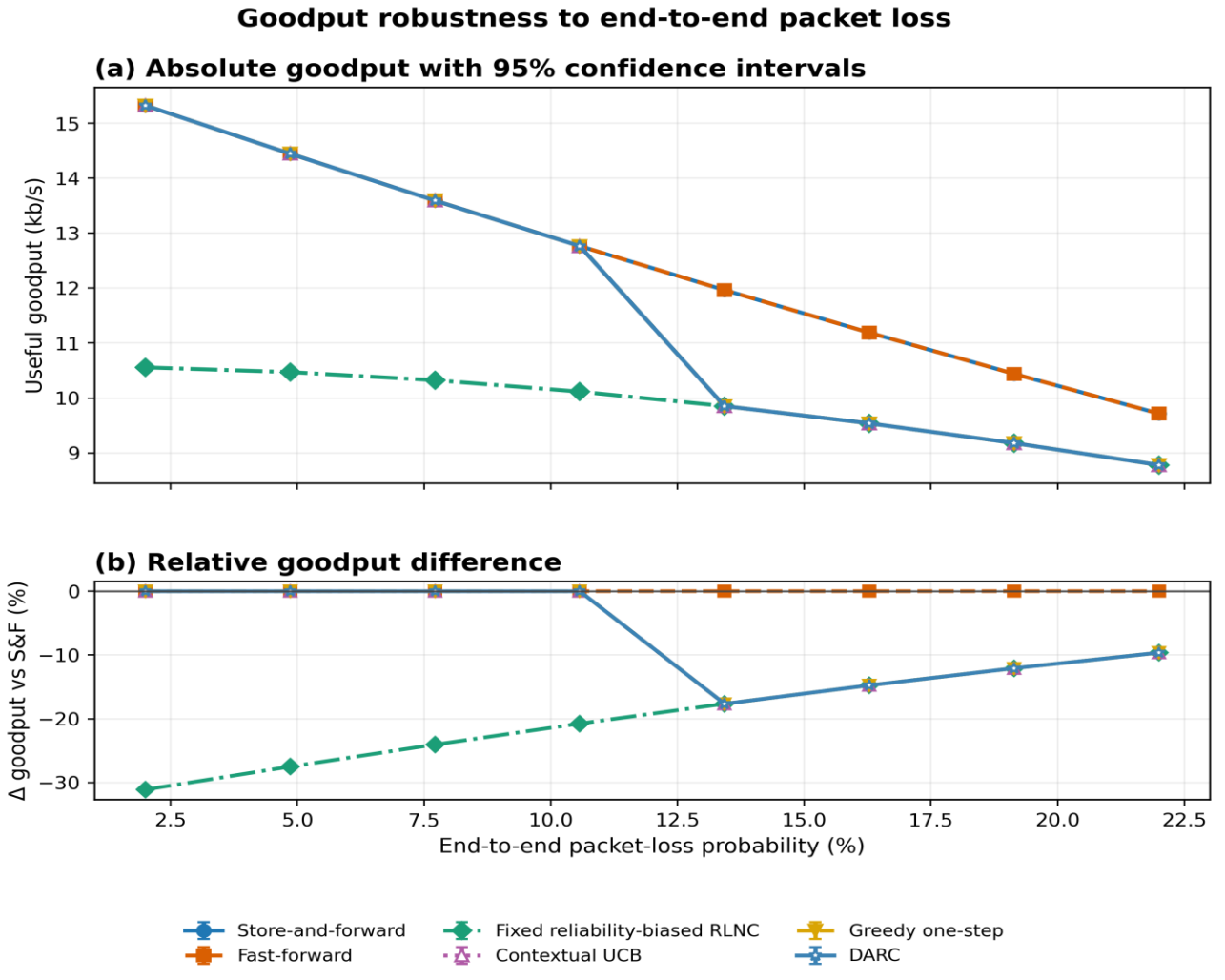

Fig. 6: Goodput robustness versus raw end-to-end packet-loss probability: (a) useful goodput with 95% confidence intervals and (b) relative goodput difference with respect to store-and-forward.

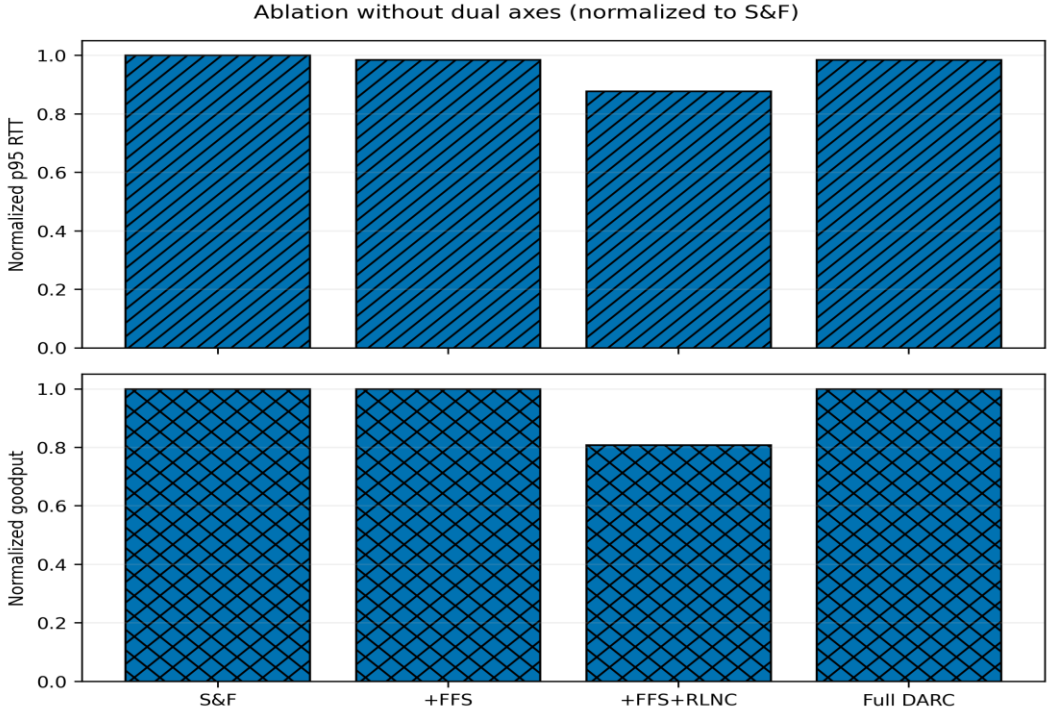

Fig. 7: Ablation of forwarding and coding components using separate normalized p95-RTT and goodput panels.

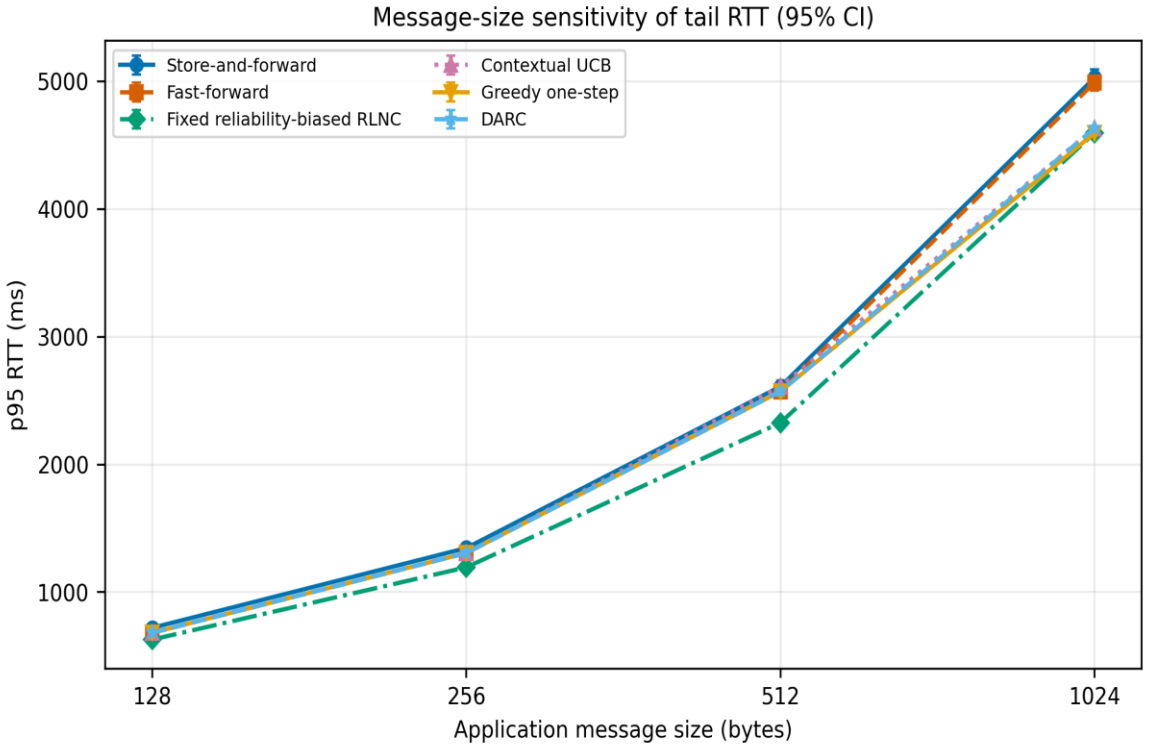

Fig. 8: Tail-RTT sensitivity to application message size with 95% confidence intervals.

Table 4 presents a compact all-policy comparison with p95 RTT, useful goodput, normalized cost, 95% confidence intervals, and the exact oracle gap. The p95 RTT, goodput, and normalized-cost columns are repeated-simulation statistics at the representative conditions, whereas the oracle-gap column is taken from the exact discounted MDP benchmark in Table 5. The same p95-goodput operating points are then visualized in Fig. 9; the numerical Pareto values and confidence intervals are also supplied in the Supporting Information as pareto_summary.csv.

At the representative operating point, the reliability-biased RLNC policy reaches approximately 2364 ms p95 RTT, but its goodput decreases to 9.90 kb/s. DARC instead chooses uncoded FFS, which is approximately 2641 ms p95 RTT while keeping 12.25 kb/s. This trade-off is easier to interpret in Fig. 9 than in a single ranking: stronger coding moves the operating point toward lower tail latency and lower goodput, whereas the uncoded adaptive choice remains in the higher-throughput region. From an application perspective, neither point is universally preferable: throughput-limited telemetry can favor DARC/uncoded FFS, whereas a reliability- or tail-latency-critical application may deliberately accept the lower goodput of stronger RLNC. The intended use is therefore policy selection under explicit application weights, not a universal ranking.

6.6 Exact dynamic MDP benchmark and contextual-bandit comparison

For the exact policy comparison, Table 5 uses the normalized discounted criterion $\bar{J}^{\pi}=(1-\gamma)\, E_{s_0\sim d_0}[\Sigma_{t=0}^{\infty}\ \gamma^t\ J(s_t,\pi(s_t))]$, where d_0 is uniform over the 81 states. The prefactor $(1-\gamma)$ keeps the reported quantity on the same normalized [0,1] cost scale as the one-step objective.

Table 4: Representative baseline comparison at five hops and 12% raw packet loss (20 independent repetitions).

Policy	p95 RTT ms (95% CI)	Goodput kb/s (95% CI)	Normalized cost (95% CI)	Exact oracle gap	Selected action
Store-and-forward	2663.1 ± 23.2	12.250 ± <0.001	0.3620 ± 0.0031	+2.59%	g=1, r=0, S&F
Fast-forward	2640.6 ± 24.6	12.250 ± <0.001	0.3582 ± 0.0025	+1.70%	g=1, r=0, FFS
Fixed reliability-biased RLNC	2364.4 ± 12.5	9.901 ± <0.001	0.3564 ± 0.0027	+8.51%	g=2, r=1, FFS
Heuristic adaptive	2364.4 ± 12.5	9.901 ± <0.001	0.3564 ± 0.0027	+11.86%	g=2, r=1, FFS
Contextual UCB	2663.1 ± 23.2	12.250 ± <0.001	0.3620 ± 0.0031	+0.32%	g=1, r=0, S&F
Greedy one-step	2640.6 ± 24.6	12.250 ± <0.001	0.3582 ± 0.0025	+0.01%	g=1, r=0, FFS
DARC	2640.6 ± 24.6	12.250 ± <0.001	0.3582 ± 0.0025	+0.29%	g=1, r=0, FFS
Oracle MDP	2640.6 ± 24.6	12.250 ± <0.001	0.3582 ± 0.0025	+0.00%	g=1, r=0, FFS

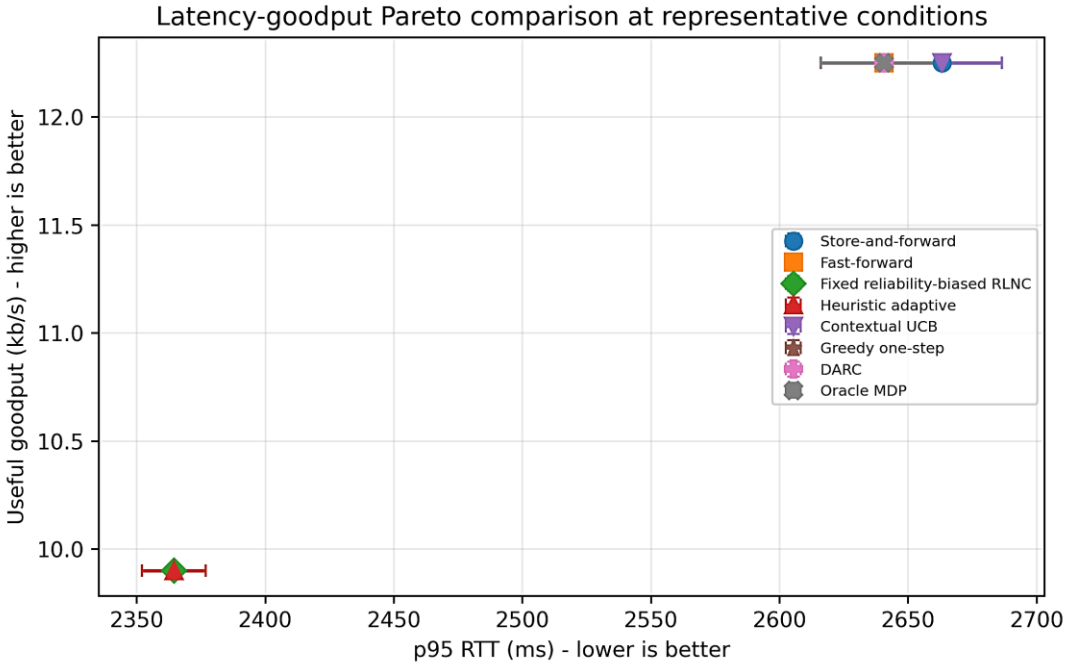

Fig. 9: Latency–goodput Pareto comparison at the representative condition; error bars show 95% confidence intervals.

Table 5: Exact normalized discounted cost over the 81-state simulated MDP

Policy	Exact discounted cost	Gap vs oracle
Oracle MDP	0.335370	+0.00%
Greedy one-step	0.335396	+0.01%
DARC	0.336350	+0.29%
Contextual UCB	0.336435	+0.32%
Fast-forward	0.341084	+1.70%
Store-and-forward	0.344052	+2.59%
Fixed reliability-biased RLNC	0.363898	+8.51%
Heuristic adaptive	0.375136	+11.86%

With respect to the exact normalized discounted objective, DARC reaches 0.336350, which is within 0.29% of the value-iteration oracle. It improves the store-and-forward and fast-forward policies by 2.24% and 1.39%, respectively. Contextual UCB is 0.336435, which is only 0.025% above DARC, while the greedy one-step controller reaches 0.335396. The 0.025% deterministic difference between DARC and UCB is too small to interpret as a practically meaningful or statistically significant superiority claim; it is smaller than the learning-seed variability visible in Fig. 2. The present evidence therefore treats DARC and contextual UCB as practically comparable, while DARC retains an explicit long-horizon queue model. More bursty measured traffic may strengthen or weaken the value of that coupling and should be tested rather than assumed. This wording deliberately avoids a universal or statistically significant superiority claim and limits the meaning of the oracle to optimality within the specified simulated MDP.

6.7 Dynamic-workload stress test

To address changing-state behavior directly, the supplied dynamic_workload_benchmark.csv was summarized over 20 independent seeds per policy. The benchmark reports the discounted normalized cost and a p95 window-cost statistic for each run. Table 6 focuses on the two adaptive myopic comparators, DARC, and the two uncoded fixed-forwarding baselines. The values are the means ± 95% t confidence interval across seeds.

Table 6: Dynamic-workload stress test over 20 independent seeds per policy

Policy	Discounted normalized cost (95% CI)	p95 window cost (95% CI)
Greedy one-step	0.3400 ± 0.0547	0.5824 ± 0.0208
Contextual UCB	0.3414 ± 0.0546	0.5760 ± 0.0184
DARC	0.3422 ± 0.0545	0.5838 ± 0.0208
Fast-forward	0.3462 ± 0.0562	0.5990 ± 0.0216
Store-and-forward	0.3493 ± 0.0561	0.6002 ± 0.0208

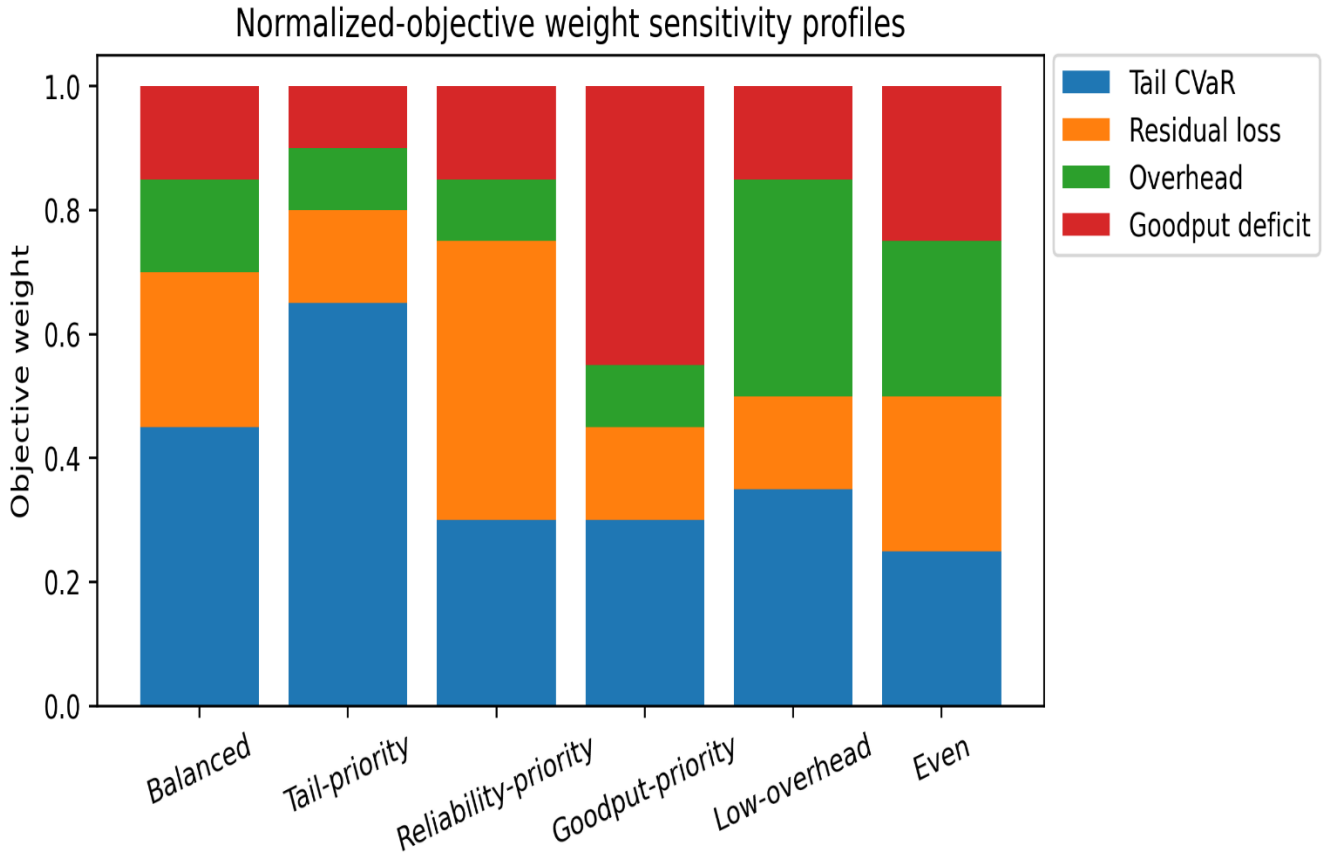

Fig. 10: Weight-sensitivity profiles for the fully normalized tail/loss/overhead/goodput-deficit objective.

The dynamic stress test does not reveal a statistically separable long-horizon advantage for DARC over greedy or contextual UCB: the confidence intervals overlap substantially, and greedy has the lowest mean discounted cost in this supplied benchmark. DARC nevertheless remains close to both adaptive comparators and below the fixed S&F/FFS means. This negative/neutral result is scientifically important: it shows that the value of the action-dependent queue model is scenario-dependent and must be demonstrated with more bursty measured traffic rather than assumed from the RL formulation alone.

6.8 Objective-weight sensitivity

The default 0.45/0.25/0.15/0.15 weights are a tail-emphasized generic telemetry preference rather than weights fitted to a particular application or to measured RF data. Six normalized weight profiles are tested in Fig. 10 instead of relying on one arbitrary scalarization. Tail- and reliability-priority profiles tend to select stronger coding, whereas goodput- or overhead-priority profiles favor light coding. This confirms that the controller's tradeoff is governed by interpretable normalized weights rather than by the unit scale of RTT. Application-specific deployment should therefore select or constrain the weights from service-level requirements rather than treat the default scalarization as universal.

7. Scalability, robustness, limitations and future work

Scalability and computational complexity: The corrected state space has 3^4=81 contexts and 12 actions; thus, only 972 Q-values are stored. The online quantization is O(1), and the greedy action selection is O(12). The exact transition tensor is required only for the offline oracle and reproducibility analysis; a deployed DARC gateway needs only the learned Q-table, observation logic, tail window and action interface.

Robustness and generalization scope: The simulation envelope covers a 1–22% raw end-to-end packet loss probability, a load/queue pressure of 0.25–0.90, two to seven hops and 128–1024-byte messages. Confidence intervals, multi-seed learning stabilization, a full policy atlas, ablation, hop scaling, message-size sensitivity and objective-weight sensitivity test internal mechanism consistency. These results support interpolation inside the stated envelope but not extrapolation to burst-dominated channels, hidden terminals, rapidly changing routes, substantially lower bit rates, many competing flows or modem-specific firmware behavior. The conclusions must not be extrapolated outside this simulated operating envelope without measurement-driven recalibration and empirical validation.

Limitations: The greatest remaining scientific limitation is the lack of newly supplied over-the-air trace data.

Table 7: Novelty-to-evidence traceability

Claimed contribution	Evidence in manuscript	Remaining validation
81-state message-aware controller	Eqs. (1–2); Fig. 3; Table 3	Gateway implementation
Genuine MDP/action-dependent transition	Eqs. (6–7); explicit transition probabilities; Table 5	Queue traces under real traffic
CVaR95 tail objective	Eqs. (8–10); Figs. 2,4,5; W=256 design rationale	Measured sliding RTT windows
Explicit RLNC model	Eqs. (3–5)	CPU/header validation on target radios
Adaptive-baseline fairness	Contextual UCB, greedy, heuristic, oracle; Tables 4–5	Independent implementation
Weight-scale robustness	Fig. 10	Application-specific preference study
Reproducibility	Code + CSV + matrices + fixed seeds	Independent rerun
Empirical calibration boundary	Section 5.3 + calibration template	Raw historical/new RF traces

Interference maps, correlated burst errors, oscillator drift, exact MAC contention, codec CPU cost and modem buffering are not modeled. The RLNC abstraction is source-only and uses generation-level cumulative ACK rather than relay recoding. The load transition is deliberately low-dimensional. In addition, the contextual UCB and the model-informed greedy baseline perform very closely to DARC in the tested MDP; thus, the paper does not claim a large generic advantage of long-horizon learning. The 120,000-transition stabilization test reduces-but does not eliminate-learning uncertainty, and formal convergence is not claimed. The binned state should therefore be interpreted as an approximate Markov state and route-length transitions as a coarse proxy rather than a routing-protocol model.

Future work should use timestamped RTT/loss traces from the radios used in earlier practical studies, fit the digital model and report the RMSE, MAE and R^2 before empirical calibration is performed. A second stage should introduce correlated loss, link asymmetry, explicit queue occupancy and multiple flows and then evaluate constrained or distributional RL for probabilistic delay guarantees. Hardware-in-the-loop testing should compare inference and coding CPU overhead with saved retransmission time. This measurement-and-calibration step is the required next stage for converting the present simulation study into an empirically calibrated system. Table 7 maps each claimed contribution to its manuscript evidence and the validation that remains open.

8. Conclusion

In conclusion, DARC provides a technically distinct continuation of practical narrowband RTT/network-coding research by turning fixed coding and forwarding choices into an auditable 81-state MDP. The corrected formulation includes the message size, an action-dependent queue transition, a CVaR95 tail estimator, normalized objectives, explicit finite-field RLNC overhead, strong adaptive baselines, an exact oracle, multi-seed diagnostics and uncertainty-aware result figures. DARC reaches near-oracle discounted cost in the stated model while revealing that contextual-bandit performance can be very close. The appropriate interpretation is therefore a reproducible, falsifiable adaptive-control design ready for measurement-driven calibration rather than a claim of universal superiority.

CRediT Author Contribution Statement

El Miloud Ar-Reyouchi: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Software, Supervision, Validation, Visualization, Writing – Original draft, Writing – Review & editing. **Abdeljalil Alaoui Douiri:** Investigation, Resources, Validation, Writing – Review & editing. Both authors have read and agreed to the published version of the manuscript.

Funding Declaration

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability Statement

The synthetic data generated and analyzed during this study, including model-configuration details, expected-reward and transition matrices, learned policy arrays, seed-level validation results, and plot-data CSV files, are included in the Supporting Information and reproducibility package accompanying the manuscript. Additional reproducibility files or clarifications are available from the corresponding author upon reasonable request. Raw over-the-air RF measurement traces are not part of the present study because they were not supplied for this analysis; therefore, the digital model is not described as empirically calibrated.

Conflict of Interest

There is no conflict of interest.

Artificial Intelligence (AI) Use Disclosure

The authors declare that Generative AI was used to assist with language editing, organization, and drafting of selected revision passages. The authors reviewed and verified all revisions and remain responsible for the scientific content and interpretation. No generative AI was used to fabricate or alter the underlying data or images; numerical values reported in this revision are derived from the accompanying reproducibility files and remain independently reproducible.

Supporting Information

The Supporting Information includes the reproducibility code, expected-reward matrix, full transition tensor, learned policy arrays, ensemble Q-table, seed-level validation outputs, plot-data CSV files, model card, and an empirical-calibration template for future measured RTT/RF traces. The complete Supporting Information is available at the link: Supporting Information.

References

[1] Y. Chatei, M. Hammouti, E. M. Ar-Reyouchi, K. Ghoumid, Downlink and uplink message size impact on round trip time metric in multi-hop wireless mesh networks, *International Journal of Advanced Computer Science and Applications*, 2017, **8**, 223–229, doi: 10.14569/IJACSA.2017.080332.

[2] E. M. Ar-Reyouchi, M. Hammouti, I. Maslouhi, K. Ghoumid, The internet of things: Network delay improvement using network coding, Proceedings of the Second International Conference on Internet of Things, Data and Cloud Computing, ACM, 2017, 8, doi: 10.1145/3018896.3018902.

[3] M. Ploumidis, N. Pappas, V. A. Siris, A. Traganitis, On the performance of network coding and forwarding schemes with different degrees of redundancy for wireless mesh networks, *Computer Communications*, 2015, **72**, 49–62, doi: 10.1016/j.comcom.2015.05.001.

[4] S. Kafaie, M. H. Ahmed, Y. Chen, O. A. Dobre, Performance Analysis of Network Coding with IEEE 802.11 DCF in Multi-Hop Wireless Networks, *IEEE Transactions on Mobile Computing*, 2018, **17(5)**, 1148–1161, doi: 10.1109/TMC.2017.2737422.

[5] S. Rattal, E. M. Ar-Reyouchi, An effective practical method for narrowband wireless mesh networks performance, *SN Applied Sciences*, 2019, **1**, 1532, doi: 10.1007/s42452-019-1595-9.

[6] E. M. Ar-Reyouchi, K. Ameziane, O. El Mrabet, K. Ghoumid, The potentials of network coding for improvement of round trip time in wireless narrowband rf communications, 2014 International Conference on Multimedia Computing and Systems, IEEE, 2014, 130–134, doi: 10.1109/ICMCS.2014.6911418.

[7] J. Laarhuis, A. Chiumento, Capacity and delay analysis of multi-hop wireless networks, *Ad Hoc Networks*, 2025, **169**, 103750, doi: 10.1016/j.adhoc.2024.103750.

[8] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, B. Leong, A random linear network coding approach to multicast, *IEEE Transactions on Information Theory*, 2006, **52**, 4413–4430, doi: 10.1109/TIT.2006.881746.

[9] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, J. Crowcroft, XORs in the Air: practical wireless network coding, *IEEE/ACM Transactions on Networking*, 2008, **16**, 497–510, doi: 10.1109/TNET.2008.923722.

[10] S. Chachulski, M. Jennings, S. Katti, D. Katabi, Trading structure for randomness in wireless opportunistic routing, *ACM SIGCOMM Computer Communication Review*, 2007, **37**, 169–180, doi: 10.1145/1282427.1282400.

[11] I. Chatzigeorgiou, A. Tassi, Decoding delay performance of random linear network coding for broadcast, *IEEE Transactions on Vehicular Technology*, 2017, **66**, 7050–7060, doi: 10.1109/TVT.2017.2670178.

[12] C. J. C. H. Watkins, P. Dayan, Q-learning, *Machine Learning*, 1992, **8**, 279–292, doi: 10.1007/BF00992698.

[13] P. Auer, N. Cesa-Bianchi, P. Fischer, Finite-time analysis of the multiarmed bandit problem, *Machine Learning*, 2002, **47**, 235–256, doi: 10.1023/A:1013689704352.

[14] R. T. Rockafellar, S. Uryasev, Optimization of conditional value-at-risk, *Journal of Risk*, 2000, **2**, 21–41, doi: 10.21314/JOR.2000.038.

[15] V. Paxson, M. Allman, J. Chu, M. Sargent, Computing TCP's retransmission timer, RFC 6298, IETF, 2011, doi: 10.17487/RFC6298.

[16] D. S. J. De Couto, D. Aguayo, J. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing, Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, ACM, 2003, 134–146, doi: 10.1145/938985.939000.

[17] S. Biswas, R. Morris, ExOR: Opportunistic multi-hop routing for wireless networks, *ACM SIGCOMM Computer Communication Review*, 2005, **35**, 133–144, doi: 10.1145/1080091.1080108.

[18] A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. P. Vasseur, R. Alexander, RPL: IPv6 routing protocol for low-power and lossy networks, RFC 6550, IETF, 2012, doi: 10.17487/RFC6550.

[19] P. Thubert, An architecture for IPv6 over the time-slotted channel hopping mode of IEEE 802.15.4, RFC 9030, IETF, 2021, doi: 10.17487/RFC9030.

[20] M. Vučinić, X. Vilajosana, S. Duquennoy, D. Dujovne, 6TiSCH Minimal Scheduling Function, RFC 9033, IETF, 2021, doi: 10.17487/RFC9033.

[21] IEEE, IEEE Standard for Low-Rate Wireless Networks, IEEE Std 802.15.4-2020, 2020, doi: 10.1109/IEEESTD.2020.9144691.

[22] J.-Y. Le Boudec, P. Thiran, Network Calculus: A theory of deterministic queuing systems for the internet, Springer, 2001, doi: 10.1007/3-540-45318-0.

[23] M. J. Neely, Stochastic network optimization with application to communication and queueing systems, Morgan & Claypool, 2010, doi: 10.2200/S00271ED1V01Y201006CNT007.

[24] K. Pister, T. Watteyne, Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e, RFC 8180, IETF, 2017, doi: 10.17487/RFC8180.

[25] C. L. Liu, J. W. Layland, Scheduling algorithms for multiprogramming in a hard-real-time environment, *Journal of the ACM*, 1973, **20**, 46–61, doi: 10.1145/321738.321743.

[26] L. Xiao, H. Li, S. Yu, Y. Zhang, L.-C. Wang, S. Ma, Reinforcement Learning Based Network Coding for Drone-Aided Secure Wireless Communications, *IEEE Transactions on Communications*, 2022, **70**, 5975–5988, doi: 10.1109/TCOMM.2022.3194074.

[27] Shahzad, R. Ali, A. Haider, H. S. Kim, RS-RLNC: A reinforcement learning-based selective random linear network coding framework for tactile internet, *IEEE Access*, 2023, **11**, 141277–141288, doi: 10.1109/ACCESS.2023.3340210.

[28] X. Ye, Y. Yu, L. Fu, Deep reinforcement learning based link adaptation technique for LTE/NR systems, *IEEE Transactions on Vehicular Technology*, 2023, **72**, 7364–7379, doi: 10.1109/TVT.2023.3236791.

[29] T. H. Nguyen, V. D. Do, L. H. Lan, N. C. Luong, D. V. Le, D. Niyato, Deep reinforcement learning for multi-hop offloading in UAV-assisted edge computing, *IEEE Transactions on Vehicular Technology*, 2023, **72**, 16917–16922, doi: 10.1109/TVT.2023.3292815.

[30] H. Wang, Y. Bai, X. Xie, Deep reinforcement learning based resource allocation in delay-tolerance-aware 5G industrial IoT systems, *IEEE Transactions on Communications*, 2024, **72**, 209–221, doi: 10.1109/TCOMM.2023.3322736.

[31] N. A. Askar, A. Habbal, RLEAFS: Reinforcement learning-based energy aware forwarding strategy for NDN-based IoT networks, *IEEE Access*, 2024, **12**, 177173–177188, doi: 10.1109/ACCESS.2024.3456669.

[32] S. K. Das, R. Mudi, M. S. Rahman, K. M. Rabie, X. Li, Federated reinforcement learning for wireless networks: fundamentals, challenges and future research trends, *IEEE Open Journal of Vehicular Technology*, 2024, **5**, 1400–1440, doi: 10.1109/OJVT.2024.3466858.

[33] R. Ali, B. Bellalta, A federated reinforcement learning framework for link activation in multi-link Wi-Fi networks, 2023 IEEE International Black Sea Conference on Communications and Networking, 2023, 360–365, doi: 10.1109/BlackSeaCom58138.2023.10299778.

[34] G. Gong, X. Jiang, G. Jin, Y. Xie, H. Chen, Nuwa-RL: A reinforcement learning based receiver-side congestion control algorithm to meet applications demands over dynamic wireless networks, 2022 IEEE International Conference on High-Performance Computing and Communications, 2022, 508–515, doi: 10.1109/HPCC-DSS-SmartCity-DependSys57074.2022.00095.

[35] G. Peserico, T. Fedullo, A. Morato, F. Tramarin, L. Rovati, S. Vitturi, SNR-based reinforcement learning rate adaptation for time critical Wi-Fi networks: assessment through a calibrated simulator, 2021 IEEE International Instrumentation and Measurement Technology Conference, 2021, 1–6, doi: 10.1109/I2MTC50364.2021.9460075.

[36] L. Liang, P. Duan, G. Cui, W. Wang, Multiagent DRL for two-timescale bandwidth allocation in multibeam satellite networks, *IEEE Internet of Things Journal*, 2025, **12**, 10613–10626, doi: 10.1109/JIOT.2024.3511672.

[37] H. Tiwari, B. Kar, P. Tiwari, Digital twin-assisted belief-state reinforcement learning for latency-robust ISAC in 6G Networks, IEEE INFOCOM 2026 - IEEE Conference on Computer Communications, 2026, doi: 10.1109/INFOCOM59046.2026.11571214.

[38] E. Eldeeb, H. Alves, Offline and distributional reinforcement learning for wireless communications, *IEEE Communications Magazine*, 2025, **63**, 71–76, doi: 10.1109/MCOM.001.2400521.

[39] Z. Tao, W. Xu, X. You, Provable Performance bounds for digital twin-driven reinforcement learning in wireless networks: A novel digital-twin bisimulation metric, *IEEE Transactions on Signal Processing*, 2025, **73**, 4430–4445, doi: 10.1109/TSP.2025.3624833.

[40] N. Cheng, X. Wang, Z. Li, Z. Yin, T. H. Luan, X. Shen, Toward enhanced reinforcement learning-based resource management via digital twin: opportunities, applications, and challenges, *IEEE Network*, 2025, **39**, 189–196, doi: 10.1109/MNET.2024.3438543.