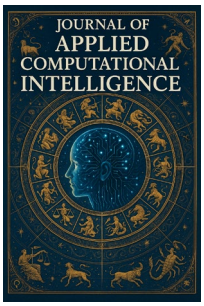
UNIVERSITY OF WEST ATTICA

eISSN: 3136-6937

Journal of Applied Computational Intelligence

Vol. 4, No. 2, April, 2026



Safe SAC Model for RDMA Congestion Control with Queue Length Barrier Constraints

Lena Kaleta[1,*], Hanna Kędziora[1], Michał Kosiński[1]

[1] *Faculty of Computer Science and Information Technology, Poznan University of Technology, Poznan, 60-965, Poland*

Received 14 December 2025; Revised 12 March 2026; Accepted 16 April 2026; Published 20 April 2026

Abstract

Safe rate adaptation for remote direct memory access requires concurrent foreground throughput, transfer completion and bounded switch-queue occupancy under delayed telemetry. This study evaluates a soft actor-critic controller whose proposed sender rates pass through a queue-length barrier projection before execution. Four sender work queues feed a shared fluid-model bottleneck. Fourteen actor descriptors summarize delayed occupancy, congestion feedback, pending work and previous rates, while a trusted counter and admission history form a conservative current-queue envelope. A 100 Gbit/s reference link and 50 microsecond control slot define 625,000 bytes per traffic unit. The admissible aggregate rate uses a 0.90-unit service lower bound, 0.10-unit background upper allowance and barrier gain 0.35. Training uses 12,000 transitions, and evaluation uses twelve independently seeded traffic episodes across four dimensionless offered-load levels. Safe SAC yields mean foreground goodput 79.55 Gbit/s and mean episode-level 99th percentile flow completion time 2.863 ms. Relative to additive-increase multiplicative-decrease control, goodput rises 11.30% and completion tail falls 34.55%; mean queue 99th percentile falls 52.11%. Fixed-proposal filtering through the same barrier also has zero observed nominal safety-boundary violations, while learned proposals improve goodput by 2.06% and completion tail by 17.53% against that comparator. Projection modifies 99.22% of learned proposals versus 25.65% of fixed proposals. When actual service falls below the assumed bound, violation fractions increase, reaching 30.73% at service multiplier 0.90. Median local actor-plus-projection time is 27.30 microseconds, with three of 1,000 calls above the 50 microsecond slot. These results separate policy efficiency, conditional barrier protection and execution cost within the declared queue simulator.

Keywords: Safe Reinforcement Learning; Soft Actor Critic; RDMA Congestion Control; Queue Length Barrier; Action Projection; Flow Completion Time

1. Introduction

RDMA congestion control increasingly combines network feedback with computational rate adaptation to coordinate concurrent transfers [1]. Queue occupancy supplies a direct indication of the pressure imposed on shared switching resources [2]. Learning based controllers represent congestion histories through state dependent transmission decisions [3]. Safe reinforcement learning introduces explicit action constraints to regulate decisions near operational boundaries [4]. These developments are relevant when synchronized senders build a queue faster than delayed feedback can reduce their rates. A useful controller must connect its transmission objective with the amount of work already admitted to the bottleneck. Effective throughput, completion latency and occupancy then become interacting properties of the same control sequence.

* Corresponding author: kaleta.le@put.poznan.pl

http://dx.doi.org/10.68406/jaci.2026.vol4iss2no3:30-43

➢ This is an open access article under the CC BY-NC-SA 4.0 license (https://creativecommons.org/licenses/by/4.0/).

© Authors retain all copyrights.

A queue threshold and a learned congestion penalty affect different parts of rate control. Threshold-based feedback research [5] links occupancy events with sender adjustments, while continuous control studies [6] choose actions within a numerical operating range. A reward penalty influences policy fitting, whereas a feasible rate set limits the command executed in the next queue slot. Network observation studies [7] motivate accounting for information age and intervening transmissions. The present filter uses trusted delayed telemetry, recorded admissions and conservative service and background bounds to determine that set. Comparing a learned proposal with a fixed proposal under the same projection distinguishes the contribution of policy adaptation from barrier protection.

A constrained comparator is essential for attributing gains to reinforcement learning. Controlled learning assessment [8] connects an architectural claim with matched information, execution conditions and component comparisons. A fixed rate proposal sent through the identical barrier isolates performance changes caused by a learned proposal. An unfiltered SAC route separately tests the effect of executing policy actions directly. Evaluation method [9] relates completion statistics to the proportion of transfers finalized within the observation period. Foreground goodput, transfer completion coverage, queue occupancy and safety-boundary violations describe complementary outcomes, so the analysis reports their denominators and operating assumptions alongside the model ranking.

This study evaluates barrier-filtered soft actor-critic rate adaptation for four senders sharing a simulated RDMA bottleneck. Delayed trusted counters and recorded admissions form a current occupancy envelope, from which a feasible aggregate rate budget constrains execution. Section 2 reviews congestion telemetry, learning-based rate control and safe-action filtering. Section 3 defines queue units, fourteen actor descriptors, envelope propagation, scalar policy updates, vector projection, replay training and runtime measurement. Section 4 compares matched traffic performance, completion coverage, barrier interventions, ablations, service sensitivity and local timing. Section 5 summarizes three quantitative findings and the bounds of the simulated safety claim.

2 Related Work

2.1 Congestion Signals in RDMA Networks

RDMA rate regulation adjusts the transmission pressure of multiple flows based on observations of congestion. Queue-based feedback research [10] is related to buffer occupancy, and delay-based methods [11] can infer the existence of path congestion from changes in round-trip time. These signals show the situation of the network at different times and aggregation levels. A queue counter is a type of switch resource, and a time observation can have accumulated effects along a path.

Telemetry-assisted congestion control [12] has several counters to distinguish between changes in admitted traffic, service and queue occupancy, etc. The resulting state can be used for more accurate control calculations when its timestamps are aligned. Record sender admissions provide information on work added after the latest queue report. A service lower bound and a background arrival allowance then define the range of change for that report. Such accounting is particularly suitable for synchronous arrivals and can use the available headroom in a small number of control slots. The queue envelope used here maintains this time relationship and provides a specific location for every admitted byte in the occupancy update.

2.2 Learning Based Transmission Rate Control

Continuous policy learning [13] is a way to change the rate according to the current situation and accumulate a reward from previous decisions. Soft Actor-Critic Methods [14] are based on stochastic action generation and value estimation. Congestion control can record pending transmissions and the state of the queue, and then decide how aggressively to change the current rate. It has a general decision scale and can limit the total contribution of the queue projection.

Off-policy control studies [15] learn from stored transitions produced by an older version of a policy. This is relevant to changes in behaviour prior to execution, and the fitted value function will then be based on the altered environment. Store the proposed action and the corresponding reward and next observation in the replay record. The current implementation is as follows: the actor samples a rate multiplier, the filter applies feasible rates, and the recorded transition is the resulting queue evolution. A separate, unfiltered policy has the same network dimensions, optimisation schedule and traffic construction. A fixed proposal with filtering can be used to determine the contribution of the learning component.

2.3 Barrier Functions and Safe Action Filtering

Barrier-based control [16] is an operating mode that uses the remaining distance from a state to its limit. Research on safety filters [17] then modifies the proposed action to keep a reasonable evolution of that distance. In a queue, headroom is consumed by newly admitted bytes, and the service process recovers headroom through the removal of work. Projection can obtain the nearest feasible rate vector for an explicit aggregated admission constraint. Thus, the

intervention can be expressed in terms of rates and its size can be recorded independently of the reward. Together, the queue boundary and the available service/observation age determine how much of the original command can be kept at a certain decision time.

Robust control methods [18] address uncertainty by setting an upper bound on disturbances, and network execution studies [19] investigate the effect of controller complexity on decision-making time. Both will be used in the barrier-filtered policy. Service estimation and background allowance determine the validity of the occupancy envelope, and inference and projection use some of the control interval. Therefore, the affected parties are capacity fluctuations, delayed monitoring and timing errors. It also records how often the projection alters the proposal. A controller that repeatedly reaches the feasible boundary may attribute much of the queue behaviour to the filter. A constrained fixed proposal can help quantify that possibility in an operationally feasible way.

3 Safe Rate Adaptation through Queue State Constraints

3.1 Telemetry Alignment and Queue State Construction

Four sender work queues feed one first-in-first-out bottleneck. A control slot lasts 50 microseconds, and one reference traffic unit equals the 625,000 bytes carried by a 100 Gbit/s link in that interval. Figure 1 traces delayed switch occupancy, sender admission counters, pending application work and previous rate settings to the current decision. The actor receives a fourteen-descriptor observation, while the barrier additionally uses a trusted counter and recorded admissions to estimate present occupancy. The workflow distinguishes proposed shares, executed shares and the next measured queue state under one common unit convention.

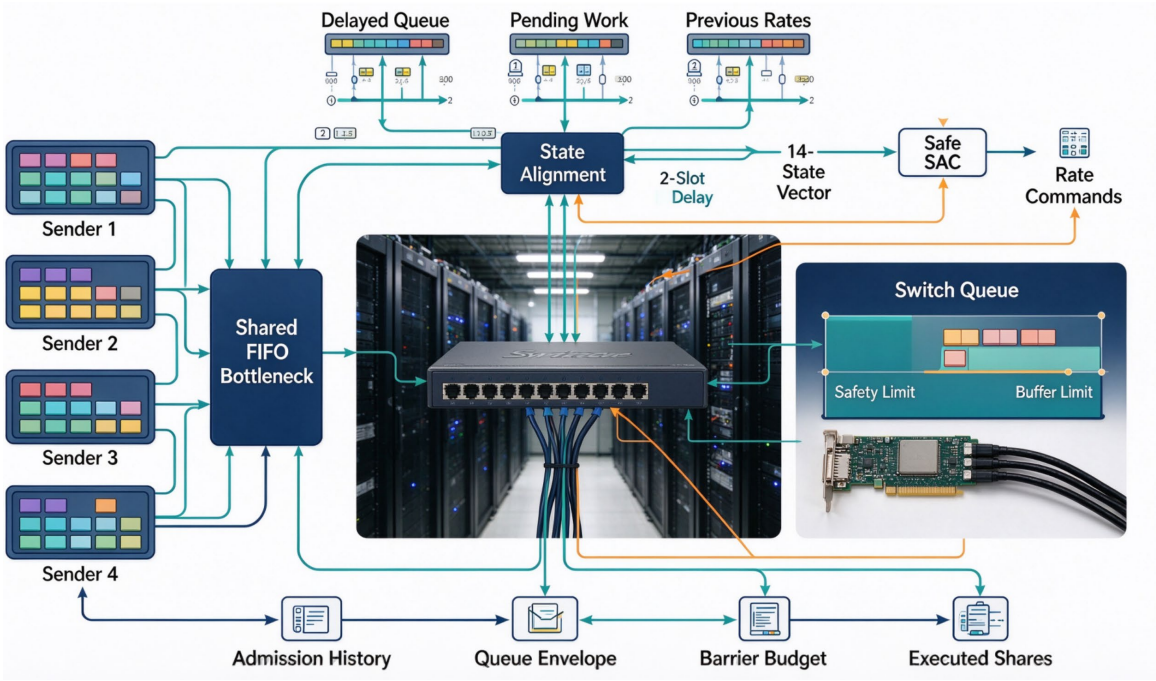


Figure 1. Telemetry alignment and RDMA rate regulation workflow.

The actor receives fourteen real-valued descriptors. Six aggregate descriptors are delayed occupancy divided by the safety limit, its change from the preceding received value, a congestion indicator, previous service, nominal offered-load tag and remaining pause duration. Four descriptors represent pending sender work divided by four traffic units and clipped at four. Four more store previous sender rate shares. The load tag is an episode-level dimensionless condition label. Concatenating aggregate state z, pending work d and previous shares rho preserves the timing and dimensional role of each input group.

$$s_t = [z_t; d_t; \rho_{t-1}] \tag{1}$$

The nominal trusted queue report is two 50-microsecond slots old. Queue differences use successive received reports, and the congestion indicator activates when reported occupancy exceeds 0.8 traffic units. The service descriptor records capacity in the preceding slot. Current pending work and rate settings are available at the sender interface. Fixed engineering scales retain the physical interpretation of each descriptor across offered-load levels. The trusted queue counter and intervening foreground-admission log additionally support barrier-envelope propagation at decision time.

Queue evolution is calculated after sender admission and background arrivals, followed by service within the current slot. The controlled admitted volume is A, background volume is b and available service is c. These quantities share the reference slot unit, and the positive part operator enforces an empty queue boundary. Application queues supply bytes up to the executed sender budgets. Admitted foreground fragments retain their sender identity and transfer identifier inside the common queue. This permits a transfer to complete when its final foreground fragment leaves the bottleneck. The physical buffer holds 6.4 units, while the operational safety boundary is 1.6 units. Their separate roles distinguish admissible occupancy from the storage available to absorb an adverse condition.

$$q_{t+1} = [q_t + A_t + b_t - c_t]_+ \tag{2}$$

The conservative occupancy envelope starts at the latest trusted queue counter and advances across each intervening slot using recorded foreground admissions. Every propagation adds at most 0.10 reference units of background traffic and subtracts at least 0.90 units of service, then applies the empty-queue boundary. The resulting barred occupancy estimate defines barrier headroom h as its distance from the 1.6-unit operational limit. Nominal service lies between 0.90 and 1.00 units and background arrivals between zero and 0.10 units per slot. The envelope therefore ties delayed telemetry and trusted sender history to the current feasible admission calculation.

$$h_t = B - \bar{q}_t \tag{3}$$

3.2 SAC Decision Learning with Barrier Filtering

A two-hidden-layer actor with 32 rectified linear units per layer outputs mean and log standard deviation for one Gaussian action. Clipping the log standard deviation from minus three to 0.5 and exponentiating gives the sampling scale. A reparameterized Gaussian draw and hyperbolic tangent produce a training action between minus one and one. Evaluation uses the deterministic mean before the same bounded mapping. The actor contains 1,602 trainable parameters. One scalar adjustment supplies four symmetric senders with a common multiplicative direction while their previous individual shares remain distinct state descriptors.

$$a_t = tanh(\mu_t + \sigma_t \varepsilon_t) \tag{4}$$

Each sender proposal multiplies its preceding share by the exponential of 0.25 times the bounded scalar action. Proposed shares are clipped from 0.005 to 0.8 reference link units before vector filtering. This rule permits bounded multiplicative acceleration or reduction while retaining rate history in the actor state. Stochastic actions provide exploration during fitting, and the mean-action evaluation path yields a repeatable proposal for each stored traffic condition. The next queue slot responds to the projected executed vector.

The barrier filter receives four proposed shares as vector v. It permits nonnegative executed shares whose sum is at most the positive part of 0.80 plus kappa times current headroom h, with kappa equal to 0.35 in the main experiment. The constant 0.80 equals the 0.90 service lower bound minus the 0.10 background allowance. Euclidean projection finds the nearest admissible vector x by sorting four proposals, identifying active entries and subtracting a shared threshold. This operation retains symmetry among equal proposals and records the correction norm for every intervention.

$$x_t = \Pi_{S_t}(v_t) \tag{5}$$

The one-slot envelope argument combines executed foreground admissions, maximum declared background arrivals and minimum declared service. If current barred occupancy is within the 1.6-unit operational boundary, the projected aggregate budget restricts the next envelope to a weighted combination of current occupancy and that boundary under gain 0.35. The argument depends on trusted admission records and service and background values remaining within their declared bounds. A reduction in actual service changes this premise and is assessed separately through capacity perturbations.

Figure 2 separates the learned proposal and executed command paths. The actor outputs one scalar adjustment, a rate adapter forms four sender proposals, and the barrier projects them before foreground admission. Replay stores the original proposed action with reward and next actor observation generated by the projected queue interaction. Twin critics estimate the consequence of this combined pathway, and target critics support bootstrap values. A fixed proposal traverses the same barrier, while a separately trained raw SAC checkpoint follows an unfiltered execution route. These matched paths identify learning and filtering contributions.

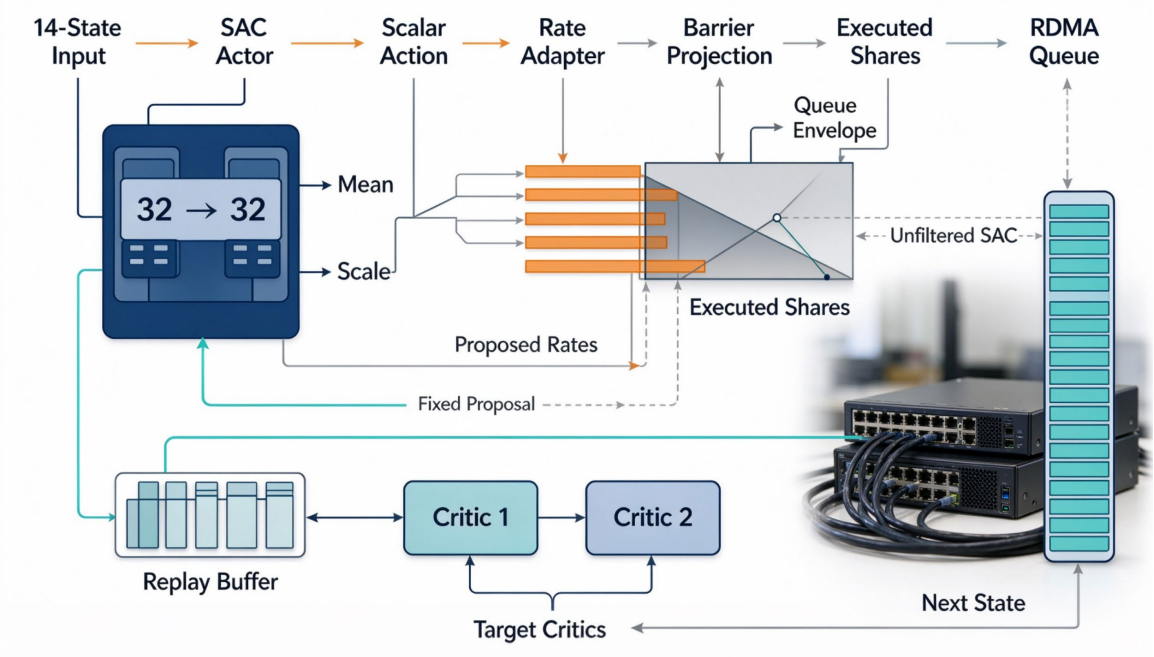


Figure 2. SAC decision learning with queue barrier filtering and matched control routes.

3.3 Training Protocol and Runtime Safeguards

Each of two critics receives fourteen state descriptors and the scalar proposed action, giving fifteen inputs. Two 32-unit rectified hidden layers precede a scalar value estimate. The bootstrap target adds immediate reward to discounted next-state soft value, using the smaller target critic and an entropy-weighted action log probability. Terminal transitions use immediate reward. The policy samples a scalar proposal, while the queue executes a projected four-share vector. Stored transitions therefore connect the proposal to its realized filtered reward and next observation.

$$y_t = r_t + \gamma V(s_{t+1}) \tag{6}$$

Slot reward begins with foreground bytes served in reference units. It subtracts queue occupancy weighted 0.08, absolute rate-share movement weighted 0.03, positive occupancy above the operational boundary weighted two and projection distance weighted 0.05. Sender-order admission and arrival-order bottleneck service retain each transfer identifier. A pause of two slots follows occupancy above 4.8 units. Completion time runs from application arrival to delivery of the final fragment and includes waiting before admission. These definitions connect queue protection with delivered work and application latency.

Each learning route uses 12,000 transitions, a replay capacity of 20,000 and 640 initial uniform-action transitions. Subsequent actor and critic updates occur every four transitions in batches of 64, producing 2,840 update rounds. Adam learning rate is 0.0003, global gradient-norm limit is five, discount is 0.98, entropy coefficient is 0.03 and target mixing coefficient is 0.01. Training draws dimensionless nominal offered-load levels uniformly from 0.6 to 1.2 and alternates steady and burst-enabled episodes up to 256 slots. Both learning routes use seed 296 and their final fixed-budget checkpoints.

$$L_\pi = E[\alpha log\pi - Q_{min}] \tag{7}$$

Evaluation freezes the actor and applies its mean-action proposal before barrier projection. Each slot records raw proposal, executed shares, occupancy envelope, measured queue and admitted bytes. A positive projection distance identifies an intervention and its norm measures correction in reference rate units. Independent traffic seeds separate evaluation episodes from training. Runtime measurement uses 50 warm-up calls followed by 1,000 actor and projection calls on the local processor. These measured computations are distinct from full queue simulation and end-to-end RDMA control latency.

4 Traffic Dependent Performance and Safety Trade Offs

4.1 Workload Design and Evaluation Settings

Figure 3 is the foreground transfer population and the burst exposure used in the evaluation. The two panels are used to provide the input for the controller. Figure 3(a) shows the transfer sizes of 16, 64, 256 and 1024 KiB and their respective probabilities of 0.4, 0.3, 0.2 and 0.1. A combination of frequently occurring small transfers and larger items

with work in the application queue. Workload analysis [20] is associated with different latency and bandwidth requirements for such-sized mixtures. Figure 3(b) is the offered load level of 0.6, 0.8, 1.0 and 1.2, along with the repeated burst intervals. Research on traffic evaluation [21] suggests keeping the arrival sequence of the controller for comparison to avoid queue excursions under synchronous admission at a moderate mean load. Repeat the above three times with different traffic seeds.

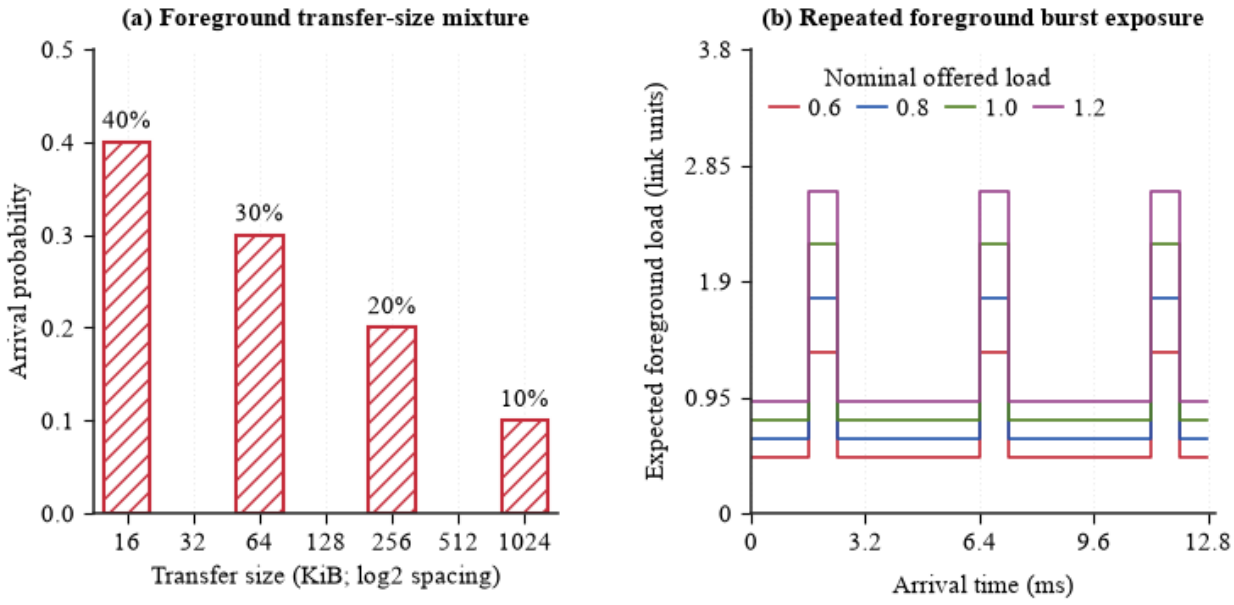


Figure 3. Foreground transfer population and traffic exposure. (a) Transfer size distribution. (b) Offered load and burst arrival pattern.

Table 1 is the numerical network and learning configuration. A reference capacity of 100 Gbit/s and a 50-microsecond slot specify a conversion rate of 625,000 bytes per traffic unit. The Safety Boundary is 1000000 bytes, the Storage Capacity is 4000000 bytes, and the Pause Threshold is 3000000 bytes. These values are used to set the scale of occupancy measurement. Reproducible configuration reporting [22] connects these settings to the units in the performance evaluation. The two slots for telemetry are 100 μs in age, and the maximum nominal background volume per slot is 62500 bytes. As shown in the table, the initial sender share was 0.20, the predicted gain is 0.35, and the final training budget is 12,000 transitions. This difference corresponds to the network conditions of the numerical decisions of the actor and filter.

Table 1. Numerical network and learning configuration.

Parameter	Value	Unit
Reference link capacity	100	Gbit/s
Control interval	50	µs
Sender interfaces	4	Count
Operational queue boundary	1000	kB
Physical buffer capacity	4000	kB
Pause trigger occupancy	3000	kB
Main telemetry delay	2	Slots
Foreground arrival interval	256	Slots
Completion drain interval	128	Slots

Service lower bound	0.90	Reference units
Background upper allowance	0.10	Reference units
Barrier gain	0.35	Dimensionless
Training transitions	12000	Count
Uniform exploration transitions	640	Count
Replay batch size	64	Transitions
Actor hidden width	32	Units
Initial sender share	0.20	Reference fraction

Each evaluation episode contains 256 arrival slots and 128 drain slots, corresponding to 12.8 ms of offered traffic and 6.4 ms for observing completion of pending work. Four sender arrivals follow independent Poisson processes, with intensities determined by dimensionless nominal offered load and expected transfer size. Burst episodes multiply arrival intensity by 2.2 in slots 32-47 of each 96-slot cycle and use factor 0.76 elsewhere. Partition methodology [23] keeps complete traffic episodes independent across fitting and evaluation. Twelve primary episodes span four load levels and three traffic seeds, with identical arrivals for every controller. Goodput and occupancy use the arrival interval; completion coverage includes the drain interval.

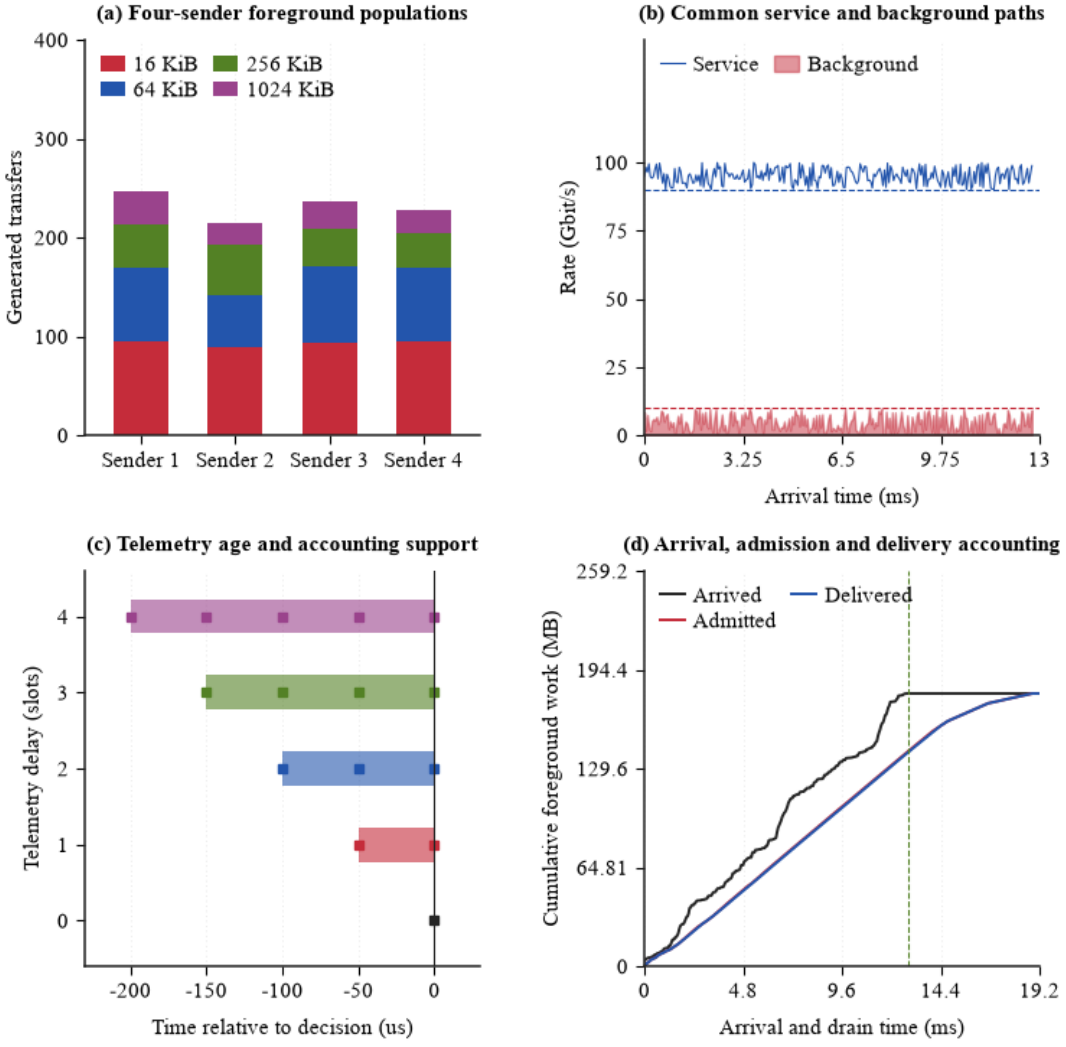


Figure 4. Shared network conditions and observation timing. (a) Sender and bottleneck structure. (b) Service and background traffic. (c) Telemetry age. (d) Pending work and completion accounting.

Figure 4 shows the other operating conditions and when they occur. The panel shows the declared deviation from the achieved control performance. Figure 4(a) shows the four simultaneous sender interfaces and their common bottleneck; these are the inputs to the aggregate admission queue. Every sender has its own pending work queue, and all admitted fragments use the same bottleneck storage. Figure 4(b) shows the service sequence of 90 and 100 Gbit/s, along with the separate background volume between zero and 10 Gbit/s. Controlled Network Assessment [24] is related to the shared conditions for interpretable rate adaptation. The actor sees the previous service value, and the filter has a fixed lower bound. Therefore, the observation and constraint paths use different calculations to obtain the relevant information. The general traffic sequence maintains the burst placement and transfer size after a change in the decision policy, so the resulting occupancy difference can be attributed to control behaviour.

Figure 4(c) is the nominal 100-microsecond telemetry age and the sensitivity range of zero to four slots. The delayed report is propagated via the admission history before the barrier is evaluated, and information age is linked to the amount of intervening work. Figure 4(d) shows the pending application load and the separation of admission and completion. A transfer can wait at the sender before joining the bottleneck and may be partially served in the common queue. The two places will have different speeds of completion. There are about 824 transfers in the 12 arrival intervals of an episode, and both small and large transfers occur frequently. Recording the number admitted, delivered and completed saves the relationship between a controller's throughput and its latency sample population during the comparison.

4.2 Transmission Efficiency and Flow Fairness

Figure 5 compares foreground service and transfer completion under six control routes. Figure 5(a) reports mean goodput of 79.55 Gbit/s for safe SAC, 77.94 for fixed proposal filtering, 78.08 for fixed rates, 71.47 for additive-increase multiplicative-decrease control, 68.86 for proportional queue feedback and 33.18 for raw SAC. Matched comparison research [25] supports common traffic inputs and information conditions. Figure 5(b) gives corresponding mean episode-level 99th percentile completion times of 2.863, 3.471, 3.369, 4.374, 4.946 and 11.329 ms. Relative to additive-increase multiplicative-decrease control, safe SAC raises goodput by 11.30% and lowers this completion tail by 34.55% in the declared episode population.

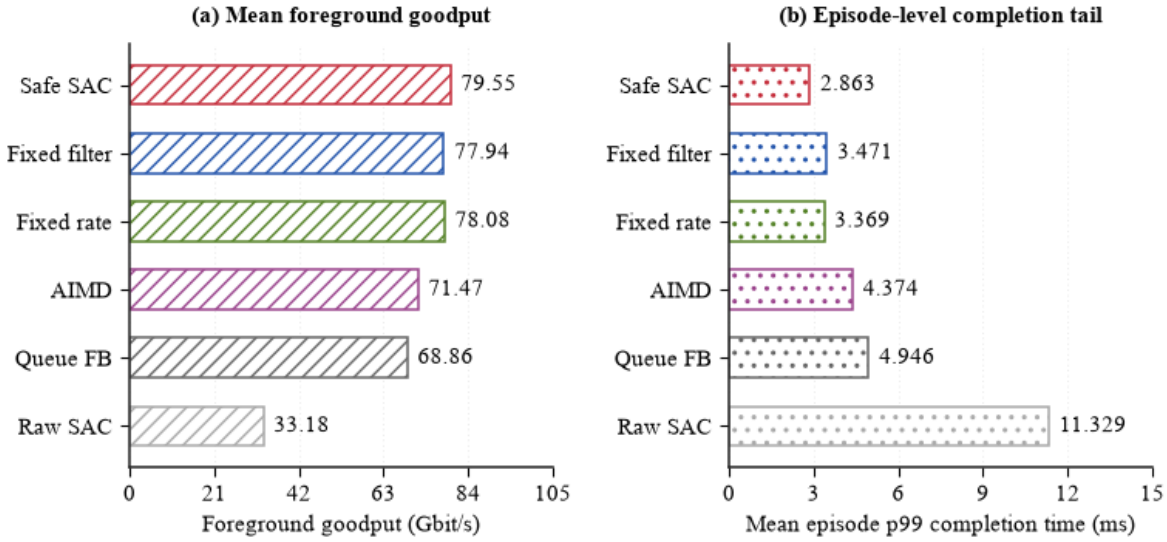


Figure 5. Transmission efficiency and completion latency. (a) Mean foreground goodput. (b) Mean episode level 99th percentile flow completion time.

Additive-increase multiplicative-decrease feedback increases a sender share by 0.008 per slot below the delayed congestion threshold and multiplies it by 0.85 above that threshold. Proportional feedback uses gain 0.025 on delayed occupancy error relative to 0.8 units. Fixed-rate control retains a prescribed sender share, and fixed proposal filtering sends four 0.24-unit proposals through the identical barrier. Baseline design studies [26] motivate this matched constrained comparator. It has zero observed nominal violations; learned safe SAC improves mean goodput by 2.06% and reduces mean episode tail completion by 17.53% relative to it. These differences describe the extra performance of a learned proposal within the same projected action set.

Figure 6 examines load and demand dependence. Figure 6(a) reports safe-controller goodput of 63.73, 78.21, 87.70 and 88.55 Gbit/s at dimensionless nominal offered-load levels 0.6, 0.8, 1.0 and 1.2. Corresponding queue 99th percentiles are 596.77, 727.46, 830.53 and 853.45 KiB. Load-dependent assessment [27] relates rising demand to bottleneck saturation. Figure 6(b) separates small transfers up to 64 KiB from larger ones; their mean episode-level tail

times are 2.826 and 2.895 ms. Both populations include application-side waiting before admission, making completion latency broader than bottleneck service time alone.

Figure 6(c) is the delivered-byte fairness of the four senders. The mean Jain index of the safe route is 0.9865, that of fixed proposal filtering is 0.9866, and that of additive increase and multiplicative decrease is 0.9868. Research on fair resource allocation [28] aims to maximise the utilisation of aggregate goodput. The same fairness value indicates that the observed improvement in completion occurs with a similar distribution of delivered services in the symmetric sender case. Figure 6(d) is the mean of the sum of absolute share changes per slot. The values are as follows: Learned Safe Route is 0.0248, Fixed Proposal Filtering is 0.0035, Additive Increase and Multiplicative Decrease is 0.0511, and Proportional Queue Feedback is 0.0843. Therefore, the learned route has more command movement than a filtered fixed proposal and is smoother in adjustment than the two numerical feedback comparators in this evaluation.

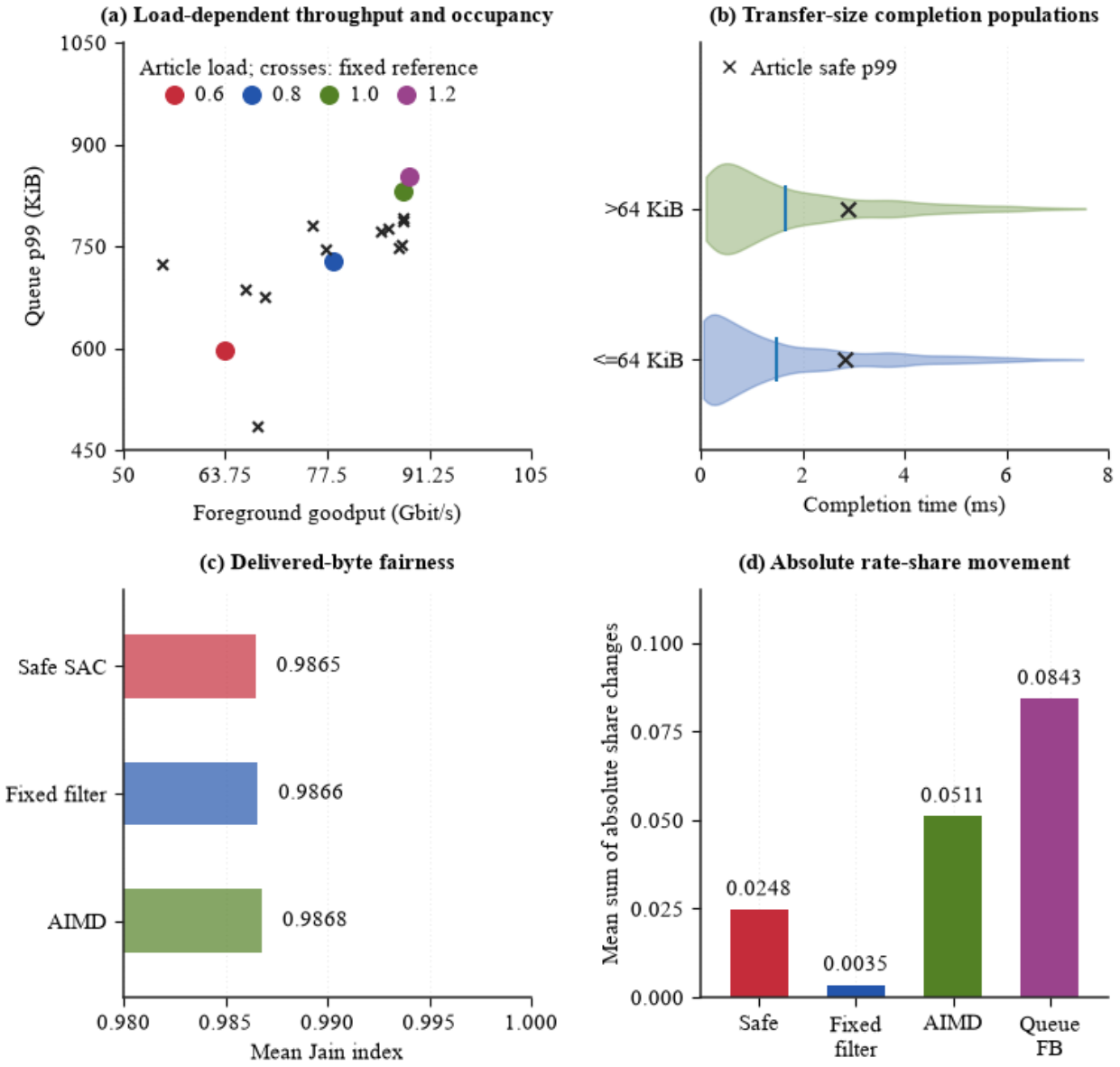


Figure 6. Performance by load and sender demand. (a) Goodput and occupancy by offered load. (b) Small and large transfer latency. (c) Delivered byte fairness. (d) Rate adjustment magnitude.

Completion coverage determines the population behind reported latency quantiles. By the end of the drain interval, safe SAC completes 99.74% of transfers, fixed proposal filtering 99.65%, additive-increase multiplicative-decrease control 97.47% and raw SAC 38.47%. Metric interpretation research [29] associates a completion quantile with the number of completed transfers. Safe-control coverage remains 98.95% at nominal load 1.2. Its mean episode tail rises from 1.050 ms at load 0.6 to 5.904 ms at load 1.2 as application backlog approaches service saturation. Reporting coverage alongside completion-time tails distinguishes fast completion from a small set of observed completions.

4.3 Barrier Effectiveness and Robustness Analysis

Figure 7 reports queue protection and projection intervention under nominal service and background bounds. Figure 7(a) has zero observed violation slots for both safe SAC and fixed proposal filtering. Their mean queue 99th percentiles are 752.05 and 623.23 KiB, respectively. Barrier evaluation studies [30] connect this result to the declared occupancy envelope. Figure 7(b) shows that projection changes 99.22% of learned proposals and 25.65% of fixed proposals during arrivals, with mean correction norms 0.1288 and 0.0087 rate units. The learned policy requests higher rates more frequently, and the barrier supplies the executed queue constraint. This intervention contrast separates policy aggressiveness from nominal boundary protection.

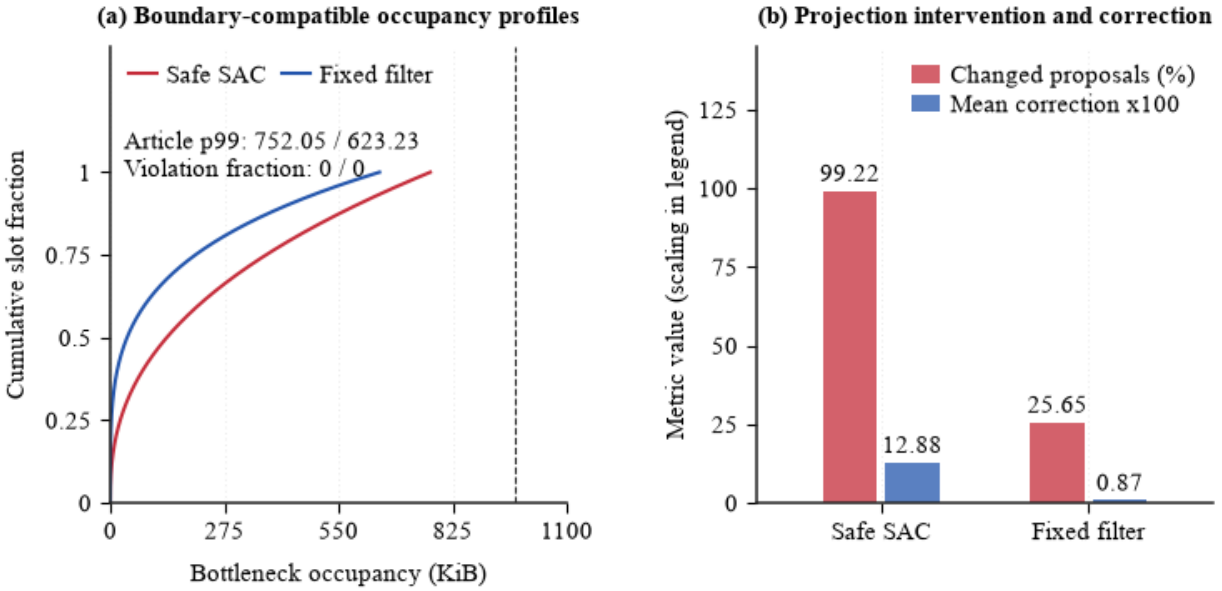


Figure 7. Queue boundary and intervention behaviour. (a) Occupancy distribution and violation fraction. (b) Projection frequency and correction magnitude.

Figure 8 groups barrier ablations, observation disturbance, telemetry age and local computation. Figure 8(a) removes either conservative service/background reserve or delayed-counter propagation while retaining the fitted actor. Component assessment research [31] identifies distinct envelope contributions: reserve removal gives queue 99th percentile 1083.99 KiB, violation fraction 21.06% and goodput 79.92 Gbit/s; direct use of the old counter gives 957.60 KiB, violation 1.20% and goodput 79.57 Gbit/s. The nominal filtered route has zero observed violations. Small goodput changes therefore coexist with substantially different queue exposure under altered protection operations.

Figure 8(b) perturbs only the actor queue descriptor with Gaussian noise up to standard deviation 0.10 traffic units, equivalent to 62,500 bytes. The trusted queue counter and admission history remain available to the barrier. Observation sensitivity research [32] distinguishes this route from perturbing the filter state. At nominal load 1.0, goodput remains 87.70 Gbit/s and queue 99th percentile 830.53 KiB. Figure 8(c) varies telemetry age from zero through four slots. At four slots, goodput is 87.59 Gbit/s and queue 99th percentile is 755.60 KiB, with zero observed violations under the declared bounds. Additional propagated uncertainty tightens the admission budget as report age rises.

Figure 8(d) reports local actor and four-entry projection timing in microseconds. Median actor time is 18.90 µs and its 99th percentile is 23.80 µs; projection median is 8.40 µs and its 99th percentile is 13.00 µs. Execution assessment research [33] relates these measurements to the 50 µs control slot. After 50 warm-up calls, 1,000 measured paired calls have combined median 27.30 µs and 99th percentile 34.61 µs. Three calls exceed 50 µs. These timings cover actor inference and projection on the local processor, while full queue calculation, telemetry handling and deployment scheduling require separate end-to-end measurement.

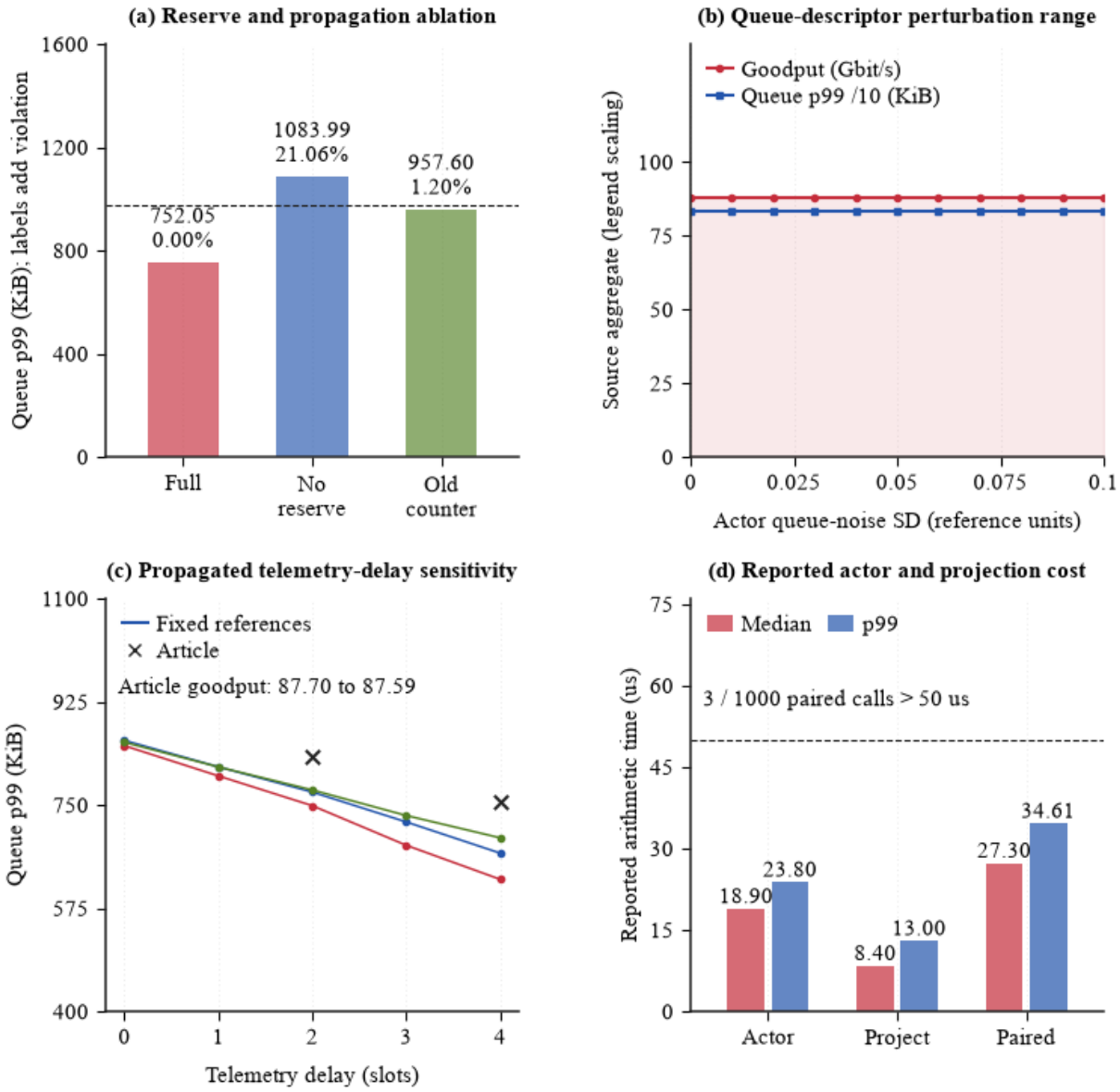


Figure 8. Mechanism attribution and execution sensitivity. (a) Reserve and propagation ablation. (b) Actor observation perturbation. (c) Telemetry delay sensitivity. (d) Actor and projection timing.

Capacity perturbation tests the service premise of the queue barrier. At nominal load 1.0, scaling actual service by 0.95 gives violation fraction 1.30%, and scaling by 0.90 gives 30.73%. Boundary-aware evaluation [34] relates these results to the fixed 0.90-unit lower service assumption used by the filter. At multiplier 0.80, queue 99th percentile is 1406.28 KiB and violation fraction is 87.24%. Barrier gain also changes the efficiency-protection balance: gains 0.15 and 0.55 give queue 99th percentiles 670.06 and 873.93 KiB and completion tails 3.513 and 2.667 ms. The reported nominal safety result is therefore conditional on the declared capacity and background envelope.

5 Conclusion

This study evaluated a queue-barrier-filtered SAC controller for rate adaptation by four senders sharing a simulated RDMA bottleneck. Fourteen actor descriptors summarize delayed congestion and sender work, while trusted counter and admission records construct a conservative current-occupancy envelope. A scalar learned proposal changes sender shares, and Euclidean projection maps the four proposed rates into a feasible aggregate admission budget. Replay and evaluation preserve separate records of proposal, executed command and resulting queue transition.

First, across twelve matched traffic episodes, safe SAC attains mean goodput 79.55 Gbit/s and mean episode-level 99th percentile completion time 2.863 ms; relative to additive-increase multiplicative-decrease control, these represent 11.30% higher goodput and 34.55% lower completion tail. Second, both safe SAC and fixed proposal filtering have zero observed nominal boundary violations; learned proposals improve goodput 2.06% and completion tail 17.53% against the fixed filtered route, while projection interventions occur in 99.22% versus 25.65% of proposals. Third, removing

conservative reserve yields 21.06% violation, and a 0.90 service multiplier yields 30.73%. These results separate efficiency, barrier protection and sensitivity to assumptions.

The service lower bound and background allowance define the operating scope of the barrier result. Local inference and projection have combined median time 27.30 μs and 99th percentile 34.61 μs, with three of 1,000 measured calls exceeding the 50 μs slot. Completion coverage, telemetry age and intervention frequency provide additional operating context for the latency and queue measures. Future validation should assess hardware RDMA queues, timestamp errors, changing service and background bounds, and full end-to-end decision timing while retaining matched traffic episodes and explicit safety denominators.

6. Declarations

6.1 Author Contributions

Lena Kaleta contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision, project administration, and funding acquisition. Hanna Kędziora contributes to validation, analysis, investigation, data collection, draft preparation. Michał Kosiński contributes to conceptualization, validation, analysis. All authors have read and agreed with the manuscript before its submission and publication.

6.2 Funding and Acknowledgments

This research received no specific financial support from any funding agency.

6.3 Institutional Review Board Statement

Not applicable.

6.4 Informed Consent Statement

Not applicable.

6.5 Date Availability

Not applicable.

6.6 Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

6.7 Declaration of generative AI and AI-assisted technologies

During the preparation of this manuscript, the authors utilized Grammarly for linguistic polishing, logical organization and draft revision. The generative AI tool was not involved in research design, data collection, data analysis, or the equation of core conclusions. All outputs generated by AI were reviewed, revised and verified manually by the authors. The authors take full responsibility for all content of this manuscript.

7. References

[1] Hu, J., Shen, H., Liu, X., & Wang, J. (2024). RDMA transports in datacenter networks: Survey. IEEE Network, 38(6), 380–387. https://doi.org/10.1109/MNET.2024.3397781

[2] Zhang, J., Wang, Y., Zhong, X., Yu, M., Pan, H., Zhang, Y., Guan, Z., Che, B., Wan, Z., Pan, T., & Huang, T. (2024). PACC: A proactive CNP generation scheme for datacenter networks. IEEE/ACM Transactions on Networking, 32(3), 2586–2599. https://doi.org/10.1109/TNET.2024.3361771

[3] Tian, H., Liao, X., Zeng, C., Sun, D., Zhang, J., & Chen, K. (2024). Efficient DRL-based congestion control with ultra-low overhead. IEEE/ACM Transactions on Networking, 32(3), 1888–1903. https://doi.org/10.1109/TNET.2023.3330737

[4] Gu, S., Yang, L., Du, Y., Chen, G., Walter, F., Wang, J., & Knoll, A. (2024). A review of safe reinforcement learning: Methods, theories, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46(12), 11216–11235. https://doi.org/10.1109/TPAMI.2024.3457538

[5] Liu, Y., Yao, X., Yang, Z., & Li, W. (2024). A multi-queue-based ECN marking strategy for multi-class QoS guarantee in programmable networks. PeerJ Computer Science, 10, e2382. https://doi.org/10.7717/peerj-cs.2382

[6] Du, Z., Zheng, J., Yu, H., Zhang, H., & Chen, G. (2025). Libra: A congestion control framework for diverse application preferences and network conditions. IEEE/ACM Transactions on Networking, 33(2), 510–525. https://doi.org/10.1109/TNET.2024.3492096

[7] Sun, H., Huang, Q., Lee, P. P. C., Bai, W., Zhu, F., & Bao, Y. (2024). Distributed network telemetry with resource efficiency and full accuracy. IEEE/ACM Transactions on Networking, 32(3), 1857–1872. https://doi.org/10.1109/TNET.2023.3327345

[8] Giacomoni, L., & Parisis, G. (2024). Reinforcement learning-based congestion control: A systematic evaluation of fairness, efficiency and responsiveness. In IEEE INFOCOM 2024—IEEE Conference on Computer Communications (pp. 1451–1460). IEEE. https://doi.org/10.1109/INFOCOM52122.2024.10621288

[9] Vielhaus, C. L., von Lengerke, C., Latzko, V., Rischke, J., Reisslein, M., & Fitzek, F. H. P. (2026). Evaluating transport layer congestion control algorithms: A comprehensive survey. IEEE Communications Surveys & Tutorials, 28, 278–340. https://doi.org/10.1109/COMST.2025.3533303

[10] Tan, Z., & Hong, P. (2024). FRCS: Congestion control in datacenter networks based on fair rate calculation by switches. In 2024 10th International Conference on Computer and Communications (ICCC) (pp. 1445–1450). IEEE. https://doi.org/10.1109/ICCC62609.2024.10941816

[11] Che, B., Wang, Y., Wan, Z., Chen, Y., Wang, Z., Tian, Y., Zhao, J., & Zhang, J. (2026). A fast-converging low-latency RDMA congestion control algorithm. IEEE Network, 40(1), 219–226. https://doi.org/10.1109/MNET.2025.3589913

[12] Zhao, Y., Wang, S., Han, S., Zhou, D., Peng, G., & Huang, T. (2024). RACC: Rapid and accurate INT-based congestion control in RDMA network. In GLOBECOM 2024—2024 IEEE Global Communications Conference (pp. 3811–3816). IEEE. https://doi.org/10.1109/GLOBECOM52923.2024.10901068

[13] Boussaoud, K., En-Nouaary, A., & Ayache, M. (2025). Adaptive congestion detection and traffic control in software-defined networks via data-driven multi-agent reinforcement learning. Computers, 14(6), 236. https://doi.org/10.3390/computers14060236

[14] Banerjee, C., Chen, Z., & Noman, N. (2024). Improved soft actor-critic: Mixing prioritized off-policy samples with on-policy experiences. IEEE Transactions on Neural Networks and Learning Systems, 35(3), 3121–3129. https://doi.org/10.1109/TNNLS.2022.3174051

[15] Jha, M. S., & Kiumarsi, B. (2025). Off-policy safe reinforcement learning for nonlinear discrete-time systems. Neurocomputing, 611, 128677. https://doi.org/10.1016/j.neucom.2024.128677

[16] Yang, Y., Zhang, Y., Zou, W., Chen, J., Yin, Y., & Li, S. E. (2024). Synthesizing control barrier functions with feasible region iteration for safe reinforcement learning. IEEE Transactions on Automatic Control, 69(4), 2713–2720. https://doi.org/10.1109/TAC.2023.3336263

[17] Kong, X., Xia, Y., Sun, Z., Zhai, D.-H., Deng, Y., & Zhang, S. (2025). Differential high order control barrier function-based safe reinforcement learning. IEEE Robotics and Automation Letters, 10(7), 7524–7531. https://doi.org/10.1109/LRA.2025.3575310

[18] Alan, A., Molnar, T. G., Ames, A. D., & Orosz, G. (2025). Generalizing robust control barrier functions from a controller design perspective. IEEE Open Journal of Control Systems, 4, 54–69. https://doi.org/10.1109/OJCSYS.2025.3529364

[19] Wang, Z., Wan, X., Li, L., Sun, Y., Xie, P., Wei, X., Ning, Q., Zhang, J., & Chen, K. (2024). Fast, scalable, and accurate rate limiter for RDMA NICs. In Proceedings of the ACM SIGCOMM 2024 Conference (pp. 568–580). ACM. https://doi.org/10.1145/3651890.3672215

[20] Pichetti, L., De Sensi, D., Sivalingam, K., Nassyr, S., Cesarini, D., Turisini, M., Pleiter, D., Artigiani, A., & Vella, F. (2024). Benchmarking Ethernet interconnect for HPC/AI workloads. In SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis (pp. 869–875). IEEE. https://doi.org/10.1109/SCW63240.2024.00124

[21] Wan, Z., Zhang, J., Yu, M., Liu, J., Yao, J., Zhao, X., & Huang, T. (2024). BiCC: Bilateral congestion control in cross-datacenter RDMA networks. In IEEE INFOCOM 2024—IEEE Conference on Computer Communications (pp. 1381–1390). IEEE. https://doi.org/10.1109/INFOCOM52122.2024.10621412

[22] Wojnar, M., Szott, S., Rusek, K., & Ciezobka, W. (2024). Reinforced-lib: Rapid prototyping of reinforcement learning solutions. SoftwareX, 26, 101706. https://doi.org/10.1016/j.softx.2024.101706

[23] Flageat, M., Lim, B., & Cully, A. (2024). Beyond expected return: Accounting for policy reproducibility when evaluating reinforcement learning algorithms. Proceedings of the AAAI Conference on Artificial Intelligence, 38(11), 12024–12032. https://doi.org/10.1609/aaai.v38i11.29090

[24] Lu, Y., Ma, X., & Cui, C. (2024). DCCS: A dual congestion control signals-based TCP for datacenter networks. Computer Networks, 247, 110457. https://doi.org/10.1016/j.comnet.2024.110457

[25] Zheng, S., Liu, J., Yan, X., Xing, Z., Di, X., & Qi, H. (2024). BBR-R: Improving BBR performance in multi-flow competition scenarios. Computer Networks, 254, 110816. https://doi.org/10.1016/j.comnet.2024.110816

[26] Arghavani, M., Zhang, H., Eyers, D., & Arghavani, A. (2024). SUSS: Improving TCP performance by speeding up slow-start. In Proceedings of the ACM SIGCOMM 2024 Conference (pp. 151–165). ACM. https://doi.org/10.1145/3651890.3672234

[27] Hu, J., Zeng, C., Wang, Z., Zhang, J., Guo, K., Xu, H., Huang, J., & Chen, K. (2024). Load balancing with multi-level signals for lossless datacenter networks. IEEE/ACM Transactions on Networking, 32(3), 2736–2748. https://doi.org/10.1109/TNET.2024.3366336

[28] Brown, L., Gran Alcoz, A., Cangialosi, F., Narayan, A., Alizadeh, M., Balakrishnan, H., Friedman, E., Katz-Bassett, E., Krishnamurthy, A., Schapira, M., & Shenker, S. (2024). Principles for Internet congestion management. In Proceedings of the ACM SIGCOMM 2024 Conference (pp. 166–180). ACM. https://doi.org/10.1145/3651890.3672247

[29] Wiegrebe, S., Kopper, P., Sonabend, R., Bischl, B., & Bender, A. (2024). Deep learning for survival analysis: A review. Artificial Intelligence Review, 57, 65. https://doi.org/10.1007/s10462-023-10681-3

[30] van Wijk, D. E. J., Coogan, S., Molnar, T. G., Majji, M., & Hobbs, K. L. (2024). Disturbance-robust backup control barrier functions: Safety under uncertain dynamics. IEEE Control Systems Letters, 8, 2817–2822. https://doi.org/10.1109/LCSYS.2024.3514998

[31] Hu, J., Zhou, Z., Rao, S., Peng, Y., Bao, B., & Ruan, C. (2024). Learning-based hierarchical adaptive congestion control with low training overhead. In 2024 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA) (pp. 262–267). IEEE. https://doi.org/10.1109/ISPA63168.2024.00041

[32] Wang, R., Tian, Y., & Kashima, K. (2024). Density estimation based soft actor-critic: Deep reinforcement learning for static output feedback control with measurement noise. Advanced Robotics, 38(6), 398–409. https://doi.org/10.1080/01691864.2024.2309621

[33] Ji, X., Han, B., Xu, Y., & Su, J. (2026). ICCP: Toward congestion control agent via controlling logic decoupling and algorithm integration. IEEE Transactions on Network and Service Management, 23, 6463–6479. https://doi.org/10.1109/TNSM.2026.3714992

[34] Lindemann, L., Robey, A., Jiang, L., Das, S., Tu, S., & Matni, N. (2024). Learning robust output control barrier functions from safe expert demonstrations. IEEE Open Journal of Control Systems, 3, 158–172. https://doi.org/10.1109/OJCSYS.2024.3385348