Can Bandits Learn to Share a Channel? Distributed Multi-Armed Bandits for Slotted ALOHA

Kamil Szczech
AGH University of Krakow
Krakow, Poland
kamil.szczech@agh.edu.pl

Maksymilian Wojnar
AGH University of Krakow
Krakow, Poland
maksymilian.wojnar@agh.edu.pl

Krzysztof Rusek
AGH University of Krakow
Krakow, Poland
krzysztof.rusek@agh.edu.pl

Katarzyna Kosek-Szott
AGH University of Krakow
Krakow, Poland
katarzyna.kosek-szott@agh.edu.pl

Szymon Szott
AGH University of Krakow
Krakow, Poland
szymon.szott@agh.edu.pl

***Abstract*—Slotted ALOHA networks constitute a playground for research on machine learning (ML)-based channel access schemes. We consider ML agents deployed in such networks and operating under severe constraints: fully distributed (independent operation), fully online (no pre-training), aperiodic (no fixed schedule), and with no explicit coordination (no signaling). We ask whether a multi-armed bandit (MAB), a minimal reinforcement learning agent, can learn the optimal policy for a capture-free slotted ALOHA system. We evaluate several MAB algorithms on aggregate network throughput and per-node fairness. Our best-performing agent is based on a Softmax bandit, which we call SALSA (Softmax ALOHA Learning for Shared Access). SALSA learns a per-node transmission policy close to optimal, with throughput near the theoretical maximum for fair random access. We achieve performance comparable to a competing state-of-the-art solution, but without requiring a priori knowledge of the network size.**

***Keywords*—Machine learning; multiaccess communication; reinforcement learning; wireless networks.**

I. Introduction

Wireless channel access networks have evolved considerably since the original ALOHA protocol [1]. Despite the undeniable success of IEEE 802.11 (Wi-Fi) and cellular networks, slotted ALOHA access remains in use in the Internet of Things (IoT) [2] and satellite communications [3]. With the emergence of machine learning (ML), slotted ALOHA networks have become an attractive testbed for evaluating ML-based channel access schemes [4]. While the operating limits of capture-free slotted ALOHA are known analytically,[1] we ask whether ML agents can rediscover efficient behavior under severe constraints: in a fully distributed environment without explicit communication and without relying on scheduled (cyclic) operation. In particular, we evaluate minimal reinforcement learning (RL) agents: multi-armed bandits (MABs), which operate fully online (without pre-training).

So far, MABs have been mostly used in research on wireless networks to optimize channel access parameters or to perform resource allocation [5]–[8]. Most relevant to our work, Zhao et al. [4] design MAB-based throughput-optimal access with local rewards (MTOA-L) for both capture-free and capture-based systems. The latter (capture-based MTOA-L) achieves high throughput but suffers from short-term unfairness. In contrast, capture-free MTOA-L requires *a priori* knowledge of the network size. Our goal is to maintain high short-term fairness without knowing the network size.

Our contributions are the following. First, we show that under our operating assumptions (no mobility, a fixed set of nodes, and full buffers), distributed channel access reduces to a multi-agent, non-contextual bandit problem (Section II-A). Second, we implement several MAB agents along with state-of-the-art baselines (Section II-D) in an open-source channel access simulator[2] (Section II-B). Third, we evaluate our pre-selected and hyperparameter-tuned group of MAB algorithms (Section III-A) and determine that Softmax-based agents perform the best. We call this approach SALSA (Softmax ALOHA Learning for Shared Access). Fourth, we evaluate SALSA against selected baselines (Section III-B) and study the impact of PHY layer errors on convergence and network performance (Section III-C). We find that SALSA operates close to the theoretically optimal throughput of fair channel access, i.e., $\frac{1}{e}$, while ensuring high per-node fairness, without requiring *a priori* knowledge of the network size. Finally, in Section IV, we conclude our study and outline future work.

II. System Model

A. Problem Formulation

The network consists of N independent agents that share a common wireless channel and operate in synchronous time slots indexed by $t = 0, 1, \ldots, T-1$. Each agent is fully

This research was funded by the National Science Centre, Poland (2023/05/Y/ST7/00004). We gratefully acknowledge the support of the National Research Institute, grant number POIR.04.02.00-00-D008/20-01 on "National Laboratory for Advanced 5G Research" (acronym PL-5G) for providing computing facilities. *(Corresponding author: Kamil Szczech.)*

[1]Under full buffer and perfect node fairness, slotted ALOHA approaches maximum throughput at $\frac{1}{e}$ as the network size increases [1].

[2]https://github.com/mldr-devs/ltc/tree/mab

autonomous: it runs its own learning and decision making, the network has no central controller, and no control messages. We further assume full-buffer traffic, so that every node always has data to send, no mobility, and a fixed number of nodes throughout the experiment. Thus, the problem involves no state transitions and takes the form of a bandit problem. The agents are trained separately and optimize the same reward function, which makes the interaction a non-cooperative game in which one agent's payoff is coupled to the choices of the others. Therefore, from the perspective of an individual agent, the environment is not stationary, since the channel state is shaped by the evolving policies of other agents. We investigate whether the agents find an equilibrium point, maximizing the aggregate network throughput while achieving high fairness.

We formulate the problem as a multi-agent, non-contextual bandit problem with N agents. No agent observes the actions or rewards of the others, and the goal of each agent is to maximize its own cumulative reward over the time horizon of $T = 30\,000$ slots. At every slot t, each agent i selects one action $a_i^t \in \mathcal{A} = \{\mathrm{TX},\ \mathrm{ID}\}$, corresponding to transmit or idle. When the agent selects TX, let $\tau_i^t \in \{-1, 1\}$ denote the transmission outcome[3], where $\tau_i^t = 1$ marks a successful transmission and $\tau_i^t = -1$ a collision. The reward is then

$$R_i^t = \begin{cases} 1, & a_i^t = \mathrm{TX} \wedge \tau_i^t = 1, \\ 0, & a_i^t = \mathrm{ID}, \\ -1, & a_i^t = \mathrm{TX} \wedge \tau_i^t = -1. \end{cases} \tag{1}$$

Thus, the agent is rewarded for a successful transmission but penalized for colliding.

B. Network Simulator

We implement the system in a custom Python-based network simulator, using JAX to accelerate execution. Our simulator follows a slotted (Monte Carlo) model with abstract, unitless time intervals. Within each slot, exactly one of three channel states occurs: a successful packet transmission, a collision, or an idle channel. In addition, random channel errors can be configured following a Bernoulli process with a specified error probability. Such errors are indistinguishable from a collision. We assume all nodes lie within each other's transmission range (no hidden nodes) so no channel attenuation is required.

Our main metric is throughput defined as the ratio of the number of slots with successful transmissions to all slots. We use aggregate throughput (of all nodes) to determine network performance and per-node throughput x_i as input for Jain's fairness index, $J = (\sum_i x_i)^2/(N \sum_i x_i^2) \in [1/N, 1]$, to assess throughput distribution, where $J = 1$ denotes a perfectly equal share.

C. MAB Agents

We apply well-known MAB algorithms to the problem: Thompson sampling (TS), EXP3 [9], and Softmax (i.e., the gradient bandit) [10]. Two variants of TS are used: continuous and discrete. The former models the reward with a Gaussian likelihood and a normal-inverse-gamma conjugate prior [11]. The latter uses a Dirichlet prior, which is the conjugate prior of the categorical reward distribution (since the reward takes only one of three values: -1, 0, and 1) [12]. Softmax is implemented with a temperature parameter that controls the exploration and exploitation trade-off, together with an exponential recency-weighted average of the rewards. The continuous TS additionally discounts its posterior parameters over time, preventing it from collapsing into an $\arg\max$ policy.

[3]Following other state-of-the-art papers, we assume that acknowledgments are sent back immediately and that their transmission is part of the slot duration.

TABLE I
HYPERPARAMETERS OF THE MAB AGENTS.

Algorithm	Parameter	Value	Description
EXP3	γ	0.001	Exploration factor
Softmax	η	0.02	Step size
	α	0.93	Recency weight
	τ	9.9	Temperature
Continuous TS	α	0.75	Inverse-gamma shape
	β	0.06	Inverse-gamma scale
	λ	0.004	Pseudo-observation count
	μ	-0.2	Prior mean
	γ	0.96	Discount factor
Discrete TS	α	94.0	Dirichlet concentration

Algorithm hyperparameters are tuned to maximize both throughput and fairness and are uniform across all agents in the network. We perform the tuning with Optuna [13] on networks of 5 and 20 agents, and the resulting values are reused in all experiments throughout the paper (Table I).

D. Baselines

We evaluate our approach against three baselines:

- **Exponential Backoff ALOHA (EB-ALOHA)** [14]: A standard slotted ALOHA baseline with a contention window range of $[4, 16]$.
- **ALOHA-Q** [15]: A Q-learning MAC protocol where autonomous agents learn to occupy reserved slots within synchronized transmission cycles. As the active node count N is unknown *a priori*, we use a fixed cycle length of $C = 20$ slots across all network sizes.
- **MTOA-L** [4]: The distributed variant of MTOA, where nodes select between a transmission action and L null actions. We evaluate both its capture-free and capture-based variants. For capture-free operation, optimal throughput requires $L = N - 1$. Since N is unknown *a priori*, we set $L = 19$ (matching $N = 20$) as a fixed reference.

The first baseline is a heuristic and approximates channel access in IEEE 802.11 networks. The remaining baselines are state-of-the-art competing solutions.

III. PERFORMANCE EVALUATION

We evaluate the performance of the selected MAB agents, present the policies they learn, compare our best-performing agent against the baselines, and assess its robustness under a

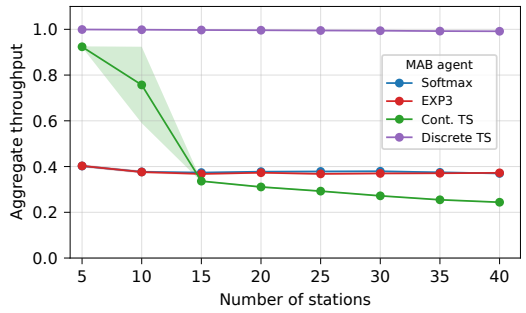


Fig. 1. Aggregate throughput of MAB agents. Results averaged over the last 10% of slots.

noisy channel. We present the means of five independent runs as well as 95% confidence intervals.

A. MAB Evaluation

Fig. 1 shows the aggregate throughput achieved by each MAB agent as a function of the number of stations N in the network. For large N, Softmax and EXP3 remain close to $1/e \approx 0.368$, i.e., the maximum theoretical throughput of fair random channel access [1]. Continuous TS achieves high throughput for a small number of stations (up to $\sim$0.92 at $N = 5$), but this drops sharply between $N = 5$ and $N = 15$ with substantial variance across seeds, and continues to decline gradually thereafter, falling below Softmax and EXP3. Discrete TS maintains throughput close to 1 regardless of N.

Since aggregate throughput alone does not reveal how evenly this throughput is shared among stations, we examine the per-station policies learned by each agent. Fig. 2 shows that Softmax and EXP3 agents converge to nearly identical transmission probabilities p_{TX} across all stations, so the resulting throughput is distributed almost equally. However, the variance of the Softmax policy is slightly lower, implying a higher Jain's fairness index than EXP3. Conversely, agents using continuous and discrete TS learn a highly bimodal policy; most stations converge to $p_{\mathrm{TX}} \approx 0$, while a few stations (or even just one for discrete TS) reach $p_{\mathrm{TX}} \approx 1$. Thus, a small subset of stations captures almost all the channel access opportunities, while the remaining stations almost never transmit, which translates into low fairness.

In general, among the MAB agents evaluated, Softmax achieves the best trade-off between throughput and fairness. Henceforth, we refer to our solution with Softmax agents as SALSA and evaluate it against the literature baselines.

B. Comparing SALSA with Baselines

Fig. 3 presents the Pareto front, comparing the combined throughput and fairness of SALSA with baseline channel access schemes. The performance of EB-ALOHA and ALOHA-Q is highly dependent on the number of nodes in the network. EB-ALOHA's performance declines as the number of nodes increases, and the algorithm operates close to the optimum only when the number of nodes is 5 or 10, due to the fixed contention window range used by the algorithm. Similarly,

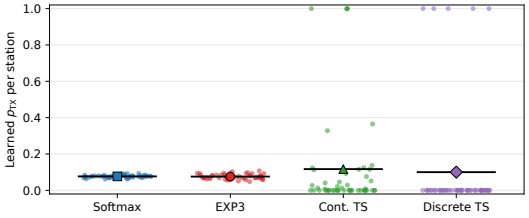


Fig. 2. Learned per-node transmission policy p_{TX} for $N = 10$ nodes across all seeds. Each point is one node in one run; the highlighted marker denotes the mean value for each agent type. The optimal policy is $p_{\mathrm{TX}} = 1/N$.

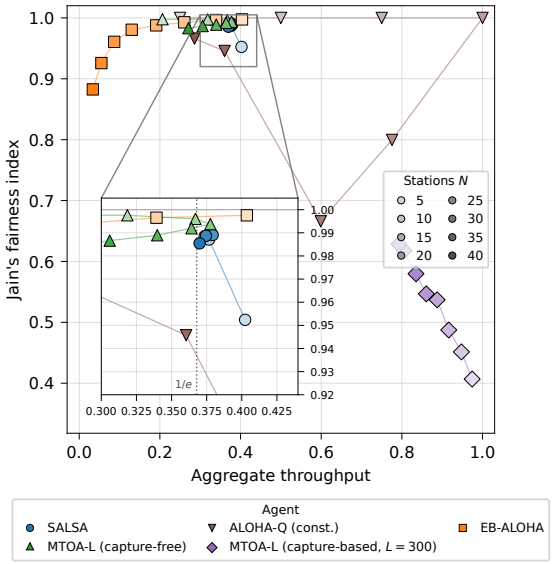


Fig. 3. Pareto front of aggregate throughput vs. Jain's fairness index for the proposed SALSA and baseline channel-access algorithms as a function of the number of nodes. Partially overlapping points around $1/e$ have been enlarged to improve readability. Results averaged over the last 10% of slots.

ALOHA-Q operates at full efficiency only when $N = 20$, matching its fixed cycle length; otherwise throughput drops.

Another technique that requires *a priori* knowledge of the number of nodes is capture-free MTOA-L. With $L = N - 1$, it achieves performance similar to our proposed approach [4]. In our tests, we use a fixed $L = 19$ to illustrate the impact of this setting on MTOA-L operation under a varying number of nodes. We observe that, for $N = 20$, fairness is high and throughput is similar to that of SALSA. As N deviates from this optimal value in either direction, MTOA-L throughput likewise degrades.

MTOA-L also offers a capture-based mode that does not require knowledge of the exact number of nodes, using $L \gg N$ (e.g., $L = 300$). In this mode, network throughput is close to 1 regardless of the number of nodes in the network. However, this comes at the cost of a drastic drop in the fairness coefficient, since MTOA-L achieves high aggregate throughput at the expense of short-term fairness. In contrast, SALSA provides throughput close to the theoretical maximum for fair channel access (for large N), i.e., near $\frac{1}{e}$, while maintaining

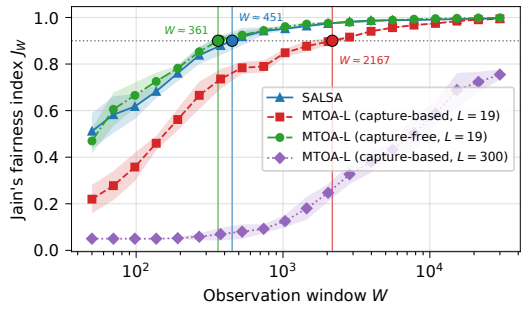


Fig. 4. Short-term Jain's fairness index J_W as a function of the observation window W (number of trailing slots) for $N = 20$ nodes and no channel errors, for SALSA and three MTOA-L variants. Vertical lines mark the window size at which each curve reaches a fairness of 0.9 (horizontal dotted line).

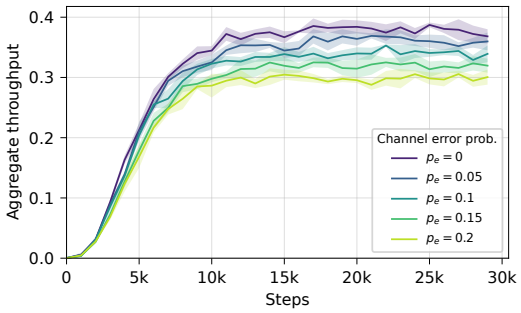


Fig. 5. Aggregate network throughput over successive steps for SALSA at $N = 20$ nodes, under different channel error probabilities p_e.

high fairness (over 0.93).

A further study of fairness is given in Fig. 4, which shows Jain's fairness index in the short-term as a function of the observation window W for the network composed of $N = 20$ nodes. SALSA and capture-free MTOA-L (with $L = N - 1$) reach a fairness of 0.9 the fastest, after $W \approx 450$ and $W \approx 350$ slots, respectively. Capture-based MTOA-L with $L = N - 1$ reaches the same fairness level much slower ($W \approx 2200$), indicating that the capture-based resolution mechanism itself decreases short-term fairness even when L is set to its optimal value. Capture-based MTOA-L with $L = 300$ does not reach a fairness of 0.9 within the observation range considered, which is consistent with its low fairness observed in Fig. 3.

C. Robustness to Channel Errors

Fig. 5 shows the aggregate network throughput over successive steps for SALSA tested in five independent trials on channels with varying probabilities of channel errors p_e. As p_e increases, the maximum achievable network throughput decreases in proportion to $1-p_e$. Furthermore, we do not observe a prolonged convergence time as p_e increases, meaning that SALSA is able to find the optimal solution even in a non-ideal radio channel, which increases the algorithm's potential for practical application.

IV. Conclusions

We investigated whether minimal RL agents can learn to equally share a channel. Our answer is affirmative: of the MAB algorithms evaluated, Softmax-based agents (SALSA) converge to near-identical per-node transmission probabilities and operate close to $\frac{1}{e}$, the throughput limit of fair random access, while maintaining a high Jain's fairness index, all without *a priori* knowledge of N, without signaling, and without pre-training. Comparing to the state of the art, SALSA matches capture-free MTOA-L configured with its optimal $L = N - 1$ but does not require that configuration, and reaches short-term fairness several times faster than the capture-based variant. SALSA also tolerates imperfect channels, with throughput scaling as $1 - p_e$ while having convergence time unaffected. The main limitation of MABs is that they are stateless so as future work we foresee replacing bandits with deep RL agents, while retaining the same constraints. Relaxing the fixed, saturated, homogeneous population assumption towards dynamic arrivals, non-saturated traffic, and legacy coexistence remains open.

References

[1] N. Abramson, "The throughput of packet broadcasting channels," *IEEE Transactions on Communications*, vol. 25, no. 1, pp. 117–128, 1977.

[2] J.-B. Seo, Y. Hu, H. Jin, and S. De, "ALOHA With SIC-Aided Collision Resolution," *IEEE Internet of Things Journal*, vol. 12, no. 8, 2025.

[3] V. Fidele Adanvo *et al.*, "Analytical Modeling of Slotted Aloha-Based Direct-to-Satellite-IoT Sensor Networks Over Nakagami-m Fading Channels," *IEEE Sensors Journal*, vol. 26, no. 2, 2026.

[4] X. Zhao and L. Dai, "Throughput-optimal random access: A queueing-theoretical analysis for learning-based access design," *arXiv preprint arXiv:2504.03178*, 2025.

[5] Z. Xu, Z. Zhang, S. Wang, X. Hu, Y. Jia, and B. Ren, "Energy-Constrained Distributed MAC in CR-IoT Networks: A Budgeted Multi-Player Multi-Armed Bandit Approach," *IEEE Transactions on Cognitive Communications and Networking*, vol. 11, no. 4, pp. 2512–2526, 2025.

[6] M. Casasnovas, F. Wilhelmi, R. Combes, M. Wojnar, K. Kosek-Szott, S. Szott, A. Jonsson, L. Esteve, and B. Bellalta, "Learning-Based Channel Access in Wi-Fi: A Multi-Armed Bandit Approach," *arXiv preprint arXiv:2511.10143*, 2025.

[7] M. A. Raza *et al.*, "Multi-agent multi-armed bandit learning for grant-free access in ultra-dense IoT networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 10, no. 4, 2024.

[8] N. Peng and L. Dai, "Multi-armed-bandit-based framed slotted ALOHA for throughput optimization," *IEEE Communications Letters*, vol. 28, no. 4, pp. 847–851, 2024.

[9] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The non-stochastic multiarmed bandit problem," *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2003.

[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[11] K. P. Murphy, "Conjugate Bayesian analysis of the Gaussian distribution," University of British Columbia, Tech. Rep., 2007, https://www.cs.ubc.ca/~murphyk/Papers/bayesGauss.pdf, accessed 2026-06-25.

[12] S. Tu, "The Dirichlet-Multinomial and Dirichlet-Categorical models for Bayesian inference," https://stephentu.github.io/writeups/dirichlet-conjugate-prior.pdf, accessed 2026-06-25.

[13] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. of the 25th ACM SIGKDD*. ACM, 2019, pp. 2623–2631.

[14] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1277–1290, 2019.

[15] Y. Chu, P. D. Mitchell, and D. Grace, "ALOHA and Q-Learning based medium access control for Wireless Sensor Networks," in *Proc. of ISWCS*, 2012.