Reinforcement Learning for URLLC Scheduling

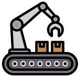
Marceau COUPECHOUX
LTCI, Telecom Paris, Institut Polytechnique de Paris
joint work with Benoît-Marie Robaglia and Dimitrios Tsilimantos

11/10/2024

Outline

Context and motivation

- **Ultra Reliable and Low Latency Communications (URLLC)** is one of the use cases of 5G/6G.



- URLLC: 99.999% reliability and latency < 1ms [3GPb].
- Uplink communications require **device coordination**.
- **Traditional MAC protocols** fail to meet the URLLC requirements:
  - May miss a lot of transmission opportunities.
  - Do not account strict latency requirements.
  - Interference and collisions degrade latency and reliability.

Uplink URLLC Access Solutions

- **Grant-Based protocols**: the scheduling of the devices is performed by the BS, see e.g. [Ca22, NGS21].
- **Grant-Free protocols**: devices access the channel without the 4 way handshake.
  - Contention-Free: the BS pre-allocates uplink resources to the devices [FNW19].
  - Contention-Based: users access the medium without coordination of the BS [M+19].
- **Advanced radio interfaces**: to further improve URLLC performance.
  - Non Orthogonal Multiple Access (NOMA) [S+13].
  - Multi-frequency channel access [LZK10].
  - Multi-connectivity, macro-diversity [MKB+19].
  - Multiple-Input Multiple-Output (MIMO) [BCC+07].

Challenges of Multiple Access for URLLC

- GB protocols: inherent latency due to access and polling
- Contention-based GF protocols: collisions
- Collision-free GF protocols: pre-allocation vs flexibility tradeoff
- Device heterogeneity: requirements, capabilities and traffic
- Dynamic environments: channels, number of devices, traffic
- Advanced radio interfaces: how to fully exploit them at the MAC layer?

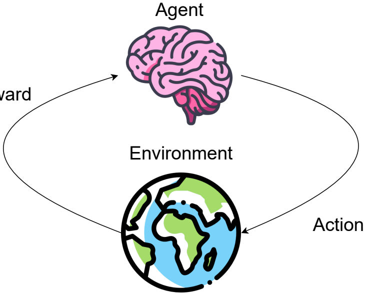
$\Rightarrow$ We have explored Reinforcement Learning solutions to address some of these challenges.

Deep RL Approaches for Uplink Access: SARL vs. MARL

- **Deep SARL Approaches**
  - Deployed at the BS to enhance GF-like protocols:
    - Transmit Power [NAM+21].
    - Number of retransmissions [LDZ+21].
    - Uplink resources [LDZ+21].
  - Challenges: partial observability, protocol overhead.
- **Deep MARL Approaches**
  - Deployed in devices for a decentralized coordination.
  - Implements Independent Learning (IL) or Centralized Training Distributed Execution (CTDE)
  - Challenges: non-stationarity, partial observability, scalability (CDTE), absence of theoretical guarantees of convergence.

Outline

Mathematical Framework



$$V^{\pi}(s_0) = \mathbb{E}_{s_{t+1}\sim\mathcal{T}(\cdot|s_t,a_t),a_t\sim\pi(\cdot|s_t)}\left[\sum_{t\geq 0}\gamma^t\mathcal{R}(s_t,a_t,s_{t+1})|a_t\sim\pi(\cdot|s_t),s_0)\right]$$

$$V^{\pi^i,\pi^{-i}}(s_0) = \mathbb{E}_{s_{t+1}\sim\mathcal{T}(\cdot|s_t,\boldsymbol{a_t}),a^{-i}\sim\pi^{-i}(\cdot|s_t)}\left[\sum_{t=0}^{T}\gamma^t\mathcal{R}^i(s_t,\boldsymbol{a_t},s_{t+1})|a_t^i\sim\pi^i(\cdot|s_t),s_0\right]$$

Policy Gradient Methods

- Policy Gradient (PG) algorithms [SMSM99] aim to maximize $V^{\pi}(s_0)$.

$$\nabla_\theta V^{\pi_\theta}(s_0) = \mathbb{E}_{\tau\sim(\pi_\theta,\mathcal{T})}\left[\sum_{t=0}^{T}\nabla_\theta \log \pi_\theta(a_t|s_t)R(\tau)\right] \tag{1}$$

- PG methods suffer from three major limitations:
  - The return creates high variance.
  - On-policy learning suffers from low sample efficiency.
  - A small change of θ can lead to a huge change of π_θ.

Proximal Policy Optimization (PPO)

- TRPO [S+15] updates the policy under a KL divergence constraint.

$$\max_{\theta} \mathbb{E}_{s,a\sim(\pi_{\text{old}},\mathcal{T})}\left[\frac{\pi_\theta(a|s)}{\pi_{\text{old}}(a|s)} A^{\pi_{\text{old}}}(s,a)\right] \tag{2}$$

$$\text{s.t. } \mathbb{E}_{s\sim\mathcal{T}}\left[KL[\pi_\theta(\cdot|s)||\pi_{\text{old}}(\cdot|s)]\right] \leq \delta \tag{3}$$

- $A^{\pi_{\text{old}}}(s_t, a_t)$ is the advantage function:

$$A^{\pi_{\text{old}}}(s_t,a_t) = Q^{\pi_{\text{old}}}(s_t,a_t) - V^{\pi_{\text{old}}}(s_t) \tag{4}$$

- PPO [Sa17] replaces the constraint by a clip:

$$\mathbb{E}_{\boldsymbol{s},\boldsymbol{a}\sim(\pi_{\text{old}},\mathcal{T})}\left[\min\left(\frac{\pi_\theta(\boldsymbol{a}|\boldsymbol{s})}{\pi_{\text{old}}(\boldsymbol{a}|\boldsymbol{s})}A^{\pi_{\text{old}}}(\boldsymbol{s},\boldsymbol{a}), g(\nu)A^{\pi_{\text{old}}}(\boldsymbol{s},\boldsymbol{a})\right)\right] \tag{5}$$

with $g(\nu) = \text{clip}\left(\frac{\pi_\theta(\boldsymbol{a}|\boldsymbol{s})}{\pi_{\text{old}}(\boldsymbol{a}|\boldsymbol{s})}, 1-\nu, 1+\nu\right)$ and $\nu \in [0,1)$

PPO Pros and Cons

Pros:

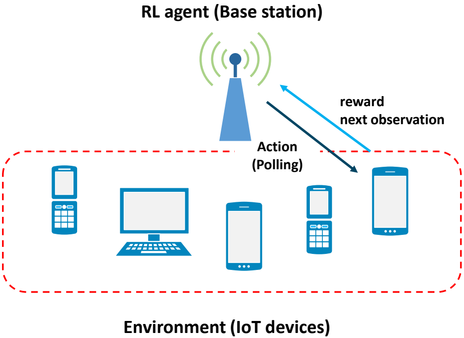
- Less computationally intensive than TRPO.
- A flexible algorithm able to work with discrete or continuous actions, in fully or partially observable environments.
- Very good performance on classical benchmarks (Atari games)
- Can be extended to multi-agent settings with good empirical performance and possibly theoretical guarantees (monotonic improvement).

Cons:

- Performance is highly dependent on implementation details.

Outline

Approach

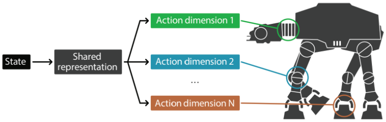


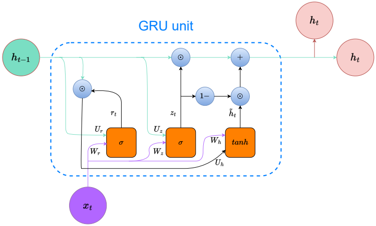
- The BS is the RL agent.
- Avoid 4-way handshake protocol.
- Allows collisions.
- NOMA is used on the uplink.

2 main limitations:

- Combinatorial action space.
- Partial observability.

Related Work





Combinatorial Action Space

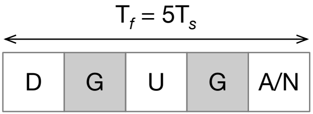
- Continuous DRL [DAa15]
- Sequential prediction [MIJD17]
- Branching architecture [TPK18]

Partial Observability

- Belief-states [KLC98]
- RNN [HS15]
- Generative model [I$^+$18].

Network model

- Time is slotted and 5 slots constitute 1 frame.



D = downlink allocation
G = guard symbol
U = uplink transmission
A/N = downlink ACK/NACK

Figure 1: Slot Structure

- The BS polls a vector of devices: $(a_1, a_2, \ldots, a_K) \in \{0,1\}^K$. *Polled* devices with at least a packet are said *active*.
- It allocates orthogonal resources for uplink pilot transmissions from the polled devices.
- A device transmits its buffer information with its packet.

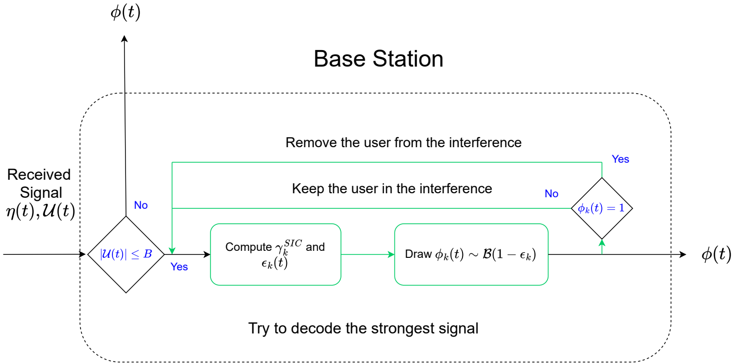
Interference Channel Model

- A user k experiences:
  - a large scale fading $g_k(t)$
  - fast fading: $\boldsymbol{h}_k(t) = [h_{k1}(t), \cdots, h_{kn_a}(t)]^T \in \mathbb{C}^{n_a \times 1}$
  - Thermal noise: $\boldsymbol{n} \in \mathbb{C}^{n_a \times 1}$
- The fast fading process $h_{ki}(t)$, for $k = 1, ..., K$ and $i = 1, ..., n_a$, follows a time-correlated Gauss-Markov model [KC07]:

$$h_{ki}(t) = \bar{a}_k h_{ki}(t-1) + z_k(t) \tag{6}$$

  where $z_k(t) \sim \mathcal{CN}(0, 1 - \bar{a}_k^2)$ and $\bar{a}_k$ the correlation coefficient [JC94].
- The coherence time T_c is controlled by $\bar{a}_k$ and plays an important role in learning the channel.

SIC Decoding Procedure

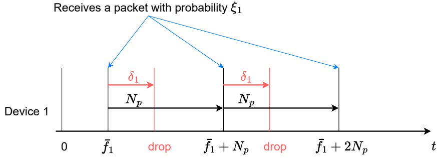


$$\gamma_k^{\text{SIC}}(t) = \frac{\eta_k(t)}{\underbrace{\sum_{j \in J_1} (1-\phi_j(t))\eta_{jk}(t)}_{\text{before } k \text{ in decoding order}} + \underbrace{\sum_{j \in J_2} \eta_{jk}(t)}_{\text{after } k} + \sigma_n^2} \tag{7}$$

Traffic Models

We study two types of traffic models described in the 3GPP standards [3GPa].

Probabilistic Periodic Traffic



$\rightarrow$ **Characteristics:** predictable traffic patterns, better use of resources.

Probabilistic Aperiodic Traffic

At every device k, packets are generated according to a Poisson process of rate λ_k.

$\rightarrow$ **Characteristics:** more complex to handle for learning algorithms because no discernible patterns to learn and exploit.

Buffer Dynamics & Deadlines

We consider packets with strict deadlines.
We have: *observed* buffer, *estimated* buffer, *real* buffer

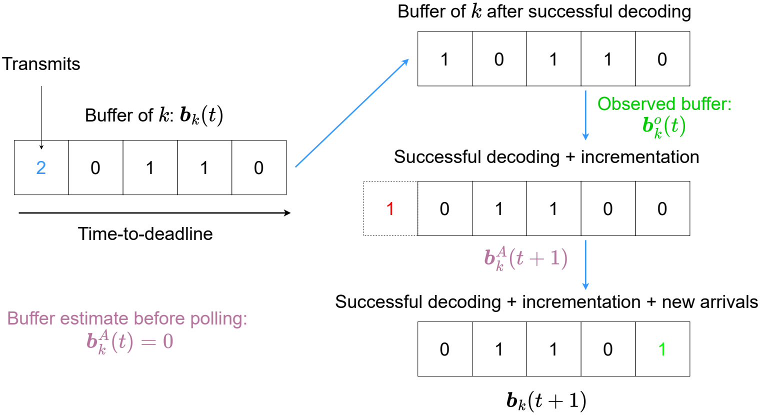


Figure 2: Buffer Dynamics $\mathcal{T}^B$

Optimization Problem

- We try to optimize the **URLLC score**: the number of *successful transmissions* over the number of *received packets*. Combines latency and reliability constraints.
- Yet, the BS doesn't have access to this information.
- We want to find the policy π maximizing:

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E}_{(\mathcal{T}^B, \mathcal{T}^H, \pi)} \left[\sum_{t=0}^{\infty} \sum_{k \in \mathcal{U}(t)} \gamma^t \phi_k(t) \right] \\ \text{s.t.} \quad & \boldsymbol{B}(t+1) \sim \mathcal{T}^B(\boldsymbol{B}(t), \phi(t)) \\ & \boldsymbol{H}(t+1) \sim \mathcal{T}^H(\boldsymbol{H}(t)) \end{aligned} \tag{P}$$

where $\gamma \in [0, 1)$ is the discount factor.

POMDP Formulation

- **State**: $\boldsymbol{s}(t) = \langle \boldsymbol{B}(t), \boldsymbol{\eta}(t), \boldsymbol{o}(t) \rangle$
- **Observation**:

$$\boldsymbol{o}(t) = \langle \boldsymbol{u}(t-1), \phi(t-1), \boldsymbol{B}^o(t-1), \boldsymbol{\eta}^o(t-1), r(t-1) \rangle.$$

- **Action**: $\boldsymbol{a} = (a_1, a_2, \dots, a_K) \in \{0,1\}^K$
- **History**: $\hbar(t) = (\boldsymbol{a}(0), \boldsymbol{o}(0), \dots, \boldsymbol{a}(t-1), \boldsymbol{o}(t-1), \boldsymbol{o}(t))$
- **Reward function**:

$$\mathcal{R}(\boldsymbol{s}(t), \boldsymbol{a}(t)) = \sum_{k \in \mathcal{U}(t)} \phi_k(t) \tag{8}$$

- **Transition function**: $\mathcal{T} = \langle \mathcal{T}^B, \mathcal{T}^H, \mathcal{O} \rangle$.

Agent State for solving the POMDP

Definition (Agent State)

At the beginning of each frame $t \geq 1$, we define the *Agent State* $\boldsymbol{A}(t)$ after the agent receives its observation $\boldsymbol{o}(t)$ as:

$$\boldsymbol{A}(t) = \langle \boldsymbol{B}^A(t), \boldsymbol{\eta}^A(t), \boldsymbol{\tau}^p(t), \boldsymbol{\tau}^a(t), \boldsymbol{\tau}^s(t), r(t-1) \rangle, \tag{9}$$

- $\boldsymbol{b}_k^A(t)$: buffer estimates: follow the same buffer dynamics.
- $\boldsymbol{\eta}^A(t)$: last known received power of the active devices.
- $\boldsymbol{\tau}^p(t), \boldsymbol{\tau}^a(t), \boldsymbol{\tau}^s(t)$: last time the devices have been polled, active and successfully decoded respectively.

Properties of the Agent State

The agent state at t, $\boldsymbol{A}(t)$ is Markovian:

$$\boldsymbol{A}(t) = f^A(\boldsymbol{A}(t-1), \boldsymbol{o}(t), \boldsymbol{a}(t-1)) \tag{10}$$

Proposition

$\boldsymbol{A}$ is a sufficient statistic for the action-observation history i.e.

$$P(s(t)|\hbar(t)) = P(s(t)|A(t)) \tag{11}$$

Proposition

The tuple $(\mathcal{S}^A, \mathcal{A}, \mathcal{T}^A, \mathcal{R}^A)$ forms an MDP where $\mathcal{T}^A : \mathcal{S}^A \times \mathcal{A} \mapsto \Delta(\mathcal{S}^A)$ is the agent state transition function and $\mathcal{R}^A : \mathcal{S}^A \times \mathcal{A} \mapsto \mathbb{R}$.

Branching Architecture

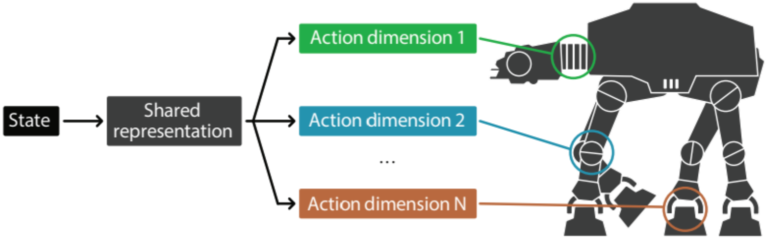


Figure 3: Branching Architecture. Image from [TPK18]

- The policy network produces K activation probabilities coordinated by hidden layers of coordination shared by all branches to capture inter-dependencies.
- Tradeoff between providing autonomy to the branches and coordinating them.

Bayesian Policies

- We use a prior f over the buffer and channel estimates.
- **EDF scheduler**: polls the users with the smallest time-to-deadline d_k^h.
- **Channel Prior**: deactivate the "bad channels".

$$f_{\text{ch}}(\boldsymbol{\eta}^{\boldsymbol{A}}(t), \boldsymbol{\tau}^{\boldsymbol{a}}) = (a_1, \ldots, a_K), \quad \text{where } a_k = \begin{cases} 0 & \text{if } \eta_k \leq \eta^* \text{ and } \tau_k^a \leq \tau^* \\ 1 & \text{otherwise} \end{cases} \tag{12}$$

Prior:

$$f(\boldsymbol{a}; \boldsymbol{A}) = EDF(\boldsymbol{B}^{\boldsymbol{A}}(t)) \odot f_{\text{ch}}(\boldsymbol{\eta}^{\boldsymbol{A}}(t), \boldsymbol{\tau}^{\boldsymbol{a}}) \tag{13}$$

Posterior policy:

$$q(\boldsymbol{a}|\boldsymbol{A}; \theta_\pi) \propto \pi(\boldsymbol{a}|\boldsymbol{A}; \theta_\pi) \odot f(\boldsymbol{a}; \boldsymbol{A}) \tag{14}$$

NOMA-PPO training algorithm

Algorithm 6: NOMA-PPO for URLLC uplink scheduling in NOMA systems.

Input: prior f, initial parameters of the policy network π_{θ_0} and the value network V_{φ_0};

for $j = 1, 2, \dots, J$ **do**

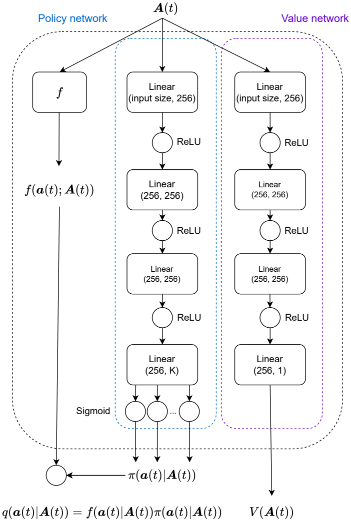
- Run the posterior policy q_{θ_j} and collect a set of β trajectories $\{(\boldsymbol{A}_b(t), \pi_{\theta_j}(\boldsymbol{a}_b(t)|\boldsymbol{A}_b(t)), r_b(t))_{t=1,\dots,T}\}_{b=1\dots\beta}$.
- Compute the rewards-to-go $\hat{R}_b(t)$ for each trajectory: $\hat{R}_b(t) = \sum_{t'=t}^{T} \gamma^{t'} r_b(t')$
- Compute the values $V_{\phi_j}(\boldsymbol{A}_b(t))$ using the value network.
- Compute the advantage estimates $\hat{A}_b^{GAE}(t)$.
- Update the policy network by maximizing (2.14) with the Adam algorithm [124]:

$$\theta_{j+1} = \arg\max_{\theta} \frac{1}{\beta T}\left[\sum_{b=1}^{\beta}\sum_{t=1}^{T} \min\left(\frac{\pi_\theta(\boldsymbol{a}_b(t)|\boldsymbol{A}_b(t))}{\pi_{\theta_j}(\boldsymbol{a}_b(t)|\boldsymbol{A}_b(t))}\hat{A}_b^{GAE}(t), g(\nu)\hat{A}_b^{GAE}(t)\right)\right]$$

Update the value network by minimizing the mean-squared error with the Adam algorithm:

$$\varphi_{j+1} = \arg\min_{\varphi} \frac{1}{\beta T}\sum_{b=1}^{\beta}\sum_{t=1}^{T}\left(V_\varphi(\boldsymbol{A}_b(t)) - \hat{R}_b(t)\right)^2 \qquad (5.23)$$

NOMA-PPO architecture



Training and convergence analysis

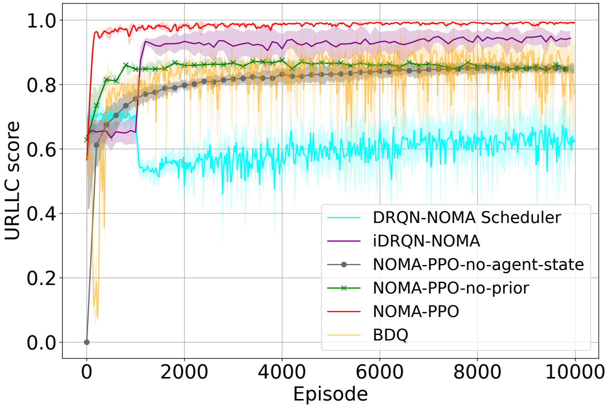


Figure 4: Evolution of the URLLC score during training for 18 users.

- The agent state can replace a RNN to handle partial observability.
- The combination of the agent state and the prior is necessary.

Performance on the 3GPP scenario

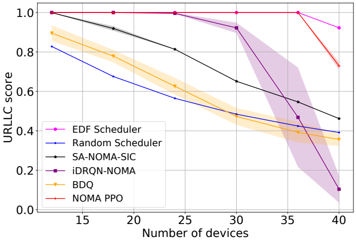


Figure 5: URLLC score in the 3GPP deterministic periodic scenario

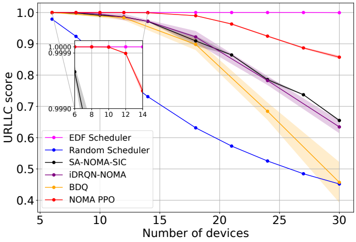


Figure 6: URLLC score in the 3GPP probabilistic aperiodic scenario

- EDF is an oracle wrt buffer info
- iDRQN does not converge for $K > 30$.
- BDQ does not manage partial observability.
- Slotted Aloha and random scheduler are not aware of the URLLC constraints.
- Aperiodic traffic is more difficult to handle.

Performance in Different Channel Conditions

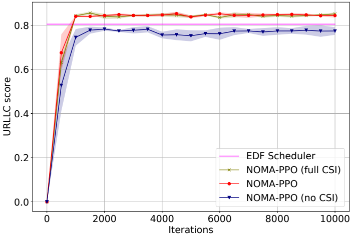


Figure 7: Long coherence time, $T_c = 1.4$ms, 10 users.

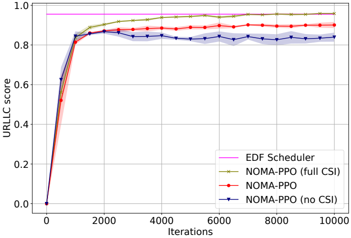


Figure 8: Short coherence time $T_c = 0.34$ms, 10 users.

- For long T_c, NOMA-PPO leverages CSI (outperforming even EDF).
- For short T_c, NOMA-PPO does not manage to exploit enough CSI.

Conclusion: Contributions

- Agent state: sufficient statistic for the past observation-action history.
  1. It expresses past actions and observations in a compact way.
  2. It converts the POMDP problem to an MDP.
- NOMA-PPO: enhances PPO with:
  1. a branching policy network architecture to linearly manage the combinatorial action space.
  2. a Bayesian policy, to use prior information about the wireless problem [TN18].
- We numerically outperform traditional MAC protocols and DRL benchmarks across several 3GPP scenarios.

Other proposed approaches

Other approaches for the uplink URLLC scheduling problem with strict deadlines:

- **FilteredPPO**, a SARL algorithm using RNN for tackling partial observability and *invalid action masking* to improve performance [RDCT21].
- **SeqDQN**, a MARL algorithm that sequentially updates Q-functions based on a Dec-POMDP formulation. It reduces non-stationarity, improves training speed and scalability vs CDTE [RCTD23].
- **MCA-PPO and MCA-iPPO** for the multi-channel access problem. MCA-PPO benefits from the monotonic improvement guarantee [RCT24b].
- **NOMA-PPO** in [RCT24a]

Thank you for your attention!

References I

3GPP, *Study on physical layer enhancements for NR ultra-reliable and low latency case (URLLC)*, TR 38.824, 3rd Generation Partnership Project (3GPP).

_______, *Study on scenarios and requirements for next generation access technologies*, TR 38.913, 3rd Generation Partnership Project (3GPP).

Ezio Biglieri, Robert Calderbank, Anthony Constantinides, Andrea Goldsmith, Arogyaswami Paulraj, and H Vincent Poor, *Mimo wireless communications*, Cambridge university press, 2007.

Giampaolo Cuozzo and authors, *Enabling urllc in 5g nr iiot networks: A full-stack end-to-end analysis*, 2022 Joint European Conference on Networks and Communications and 6G Summit (EuCNC/6G Summit), 2022, pp. 333–338.

Gabriel Dulac-Arnold and authors, *Deep reinforcement learning in large discrete action spaces*, arXiv preprint arXiv:1512.07679 (2015).

References II

- Ye Feng, Ampalavanapillai Nirmalathas, and Elaine Wong, *A predictive semi-persistent scheduling scheme for low-latency applications in lte and nr networks*, ICC 2019 - 2019 IEEE International Conference on Communications (ICC), 2019, pp. 1–6.
- Matthew Hausknecht and Peter Stone, *Deep recurrent q-learning for partially observable mdps*, 2015 aaai fall symposium series, 2015.
- Maximilian Igl et al., *Deep variational reinforcement learning for pomdps*, International Conference on Machine Learning, PMLR, 2018, pp. 2117–2126.
- William C Jakes and Donald C Cox, *Microwave mobile communications*, Wiley-IEEE press, 1994.
- Mari Kobayashi and Giuseppe Caire, *Joint beamforming and scheduling for a multi-antenna downlink with imperfect transmitter channel knowledge*, IEEE Journal on Selected Areas in Communications **25** (2007), no. 7, 1468–1477.

References III

- Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra, *Planning and acting in partially observable stochastic domains*, Artificial intelligence **101** (1998), no. 1-2, 99–134.
- Yan Liu, Yansha Deng, Hui Zhou, Maged Elkashlan, and Arumugam Nallanathan, *A general deep reinforcement learning framework for grant-free noma optimization in murllc*, arXiv preprint arXiv:2101.00515 (2021).
- Keqin Liu, Qing Zhao, and Bhaskar Krishnamachari, *Dynamic multichannel access with imperfect channel state detection*, IEEE Transactions on Signal Processing **58** (2010), no. 5, 2795–2808.
- Nurul Huda Mahmood et al., *Uplink grant-free access solutions for urllc services in 5g new radio*, 2019 16th International Symposium on Wireless Communication Systems (ISWCS), IEEE, 2019, pp. 607–612.
- Luke Metz, Julian Ibarz, Navdeep Jaitly, and James Davidson, *Discrete sequential prediction of continuous actions for deep rl*, arXiv preprint arXiv:1705.05035 (2017).

References IV

Nurul Huda Mahmood, Ali Karimi, Gilberto Berardinelli, Klaus I Pedersen, and Daniela Laselva, *On the resource utilization of multi-connectivity transmission for urllc services in 5g new radio*, 2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW), IEEE, 2019, pp. 1–6.

Francisco Hugo Costa Neto, Daniel Costa Araújo, Mateus Pontes Mota, Tarcisio F Maciel, and André LF de Almeida, *Uplink power control framework based on reinforcement learning for 5g networks*, IEEE Transactions on Vehicular Technology **70** (2021), no. 6, 5734–5748.

Mohamed W. Nomeir, Yasser Gadallah, and Karim G. Seddik, *Uplink scheduling for mixed grant-based embb and grant-free urllc traffic in 5g networks*, 2021 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2021, pp. 187–192.

References V

Benoît-Marie Robaglia, Marceau Coupechoux, and Dimitrios Tsilimantos, *Deep Reinforcement Learning for Uplink Scheduling in NOMA-URLLC Networks*, IEEE Transactions on Machine Learning in Communications and Networking **2** (2024), 1142–1158.

______, *Multi-Agent Proximal Policy Optimization for Dynamic Multi-Channel URLLC Access*, 2021 IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2024.

Benoît-Marie Robaglia, Marceau Coupechoux, Dimitrios Tsilimantos, and Apostolos Destounis, *SeqDQN: Multi-Agent Deep Reinforcement Learning for Uplink URLLC with Strict Deadlines*, 2023 Joint European Conference on Networks and Communications amp; 6G Summit (EuCNC/6G Summit), vol. 2, IEEE, June 2023, pp. 623–628.

Benoit-Marie Robaglia, Apostolos Destounis, Marceau Coupechoux, and Dimitrios Tsilimantos, *Deep reinforcement learning for scheduling uplink iot traffic with strict deadlines*, 2021 IEEE Global Communications Conference (GLOBECOM), IEEE, December 2021.

References VI

- Yuya Saito et al., *Non-orthogonal multiple access (noma) for cellular future radio access*, 2013 IEEE 77th vehicular technology conference (VTC Spring), IEEE, 2013, pp. 1–5.
- John Schulman et al., *Trust region policy optimization*, International conference on machine learning, PMLR, 2015, pp. 1889–1897.
- John Schulman and authors, *Proximal policy optimization algorithms*, arXiv preprint arXiv:1707.06347 (2017).
- Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour, *Policy gradient methods for reinforcement learning with function approximation*, Advances in neural information processing systems **12** (1999).
- Michalis K Titsias and Sotirios Nikoloutsopoulos, *Bayesian transfer reinforcement learning with prior knowledge rules*, arXiv preprint arXiv:1810.00468 (2018).

References VII

Arash Tavakoli, Fabio Pardo, and Petar Kormushev, *Action branching architectures for deep reinforcement learning*, Proceedings of the AAAI Conference on Artificial Intelligence, 2018.