

DEEP REINFORCEMENT LEARNING FOR URLLC DATA MANAGEMENT ON TOP OF SCHEDULED EMBB TRAFFIC

Fabio Saggese¹, Luca Pasqualini², Marco Moretti¹, Andrea Abrardo²

¹Dip. di Ingegneria dell'Informazione, Università di Pisa

²Dip. di Ingegneria dell'Informazione e Scienze Matematiche, Università di Siena

GTTI workshop “Wireless Intelligence: From Reconfigurable Surfaces to Edge/Cloud Communication”



UNIVERSITÀ DI PISA



UNIVERSITÀ DI SIENA 1240

OUTLINE

1 INTRODUCTION

2 MODEL

3 DEEP REINFORCEMENT LEARNING

4 RESULTS

5 CONCLUSIONS

- Finding a solution for the RAN slicing of eMBB and URLLC traffics;
- The solution should exploit the dynamic nature of URLLC traffic.

WHY REINFORCEMENT LEARNING?

- Performing slicing of different type of traffic is a hard task;
- RL is able to find very good policies for systems that dynamically change through time;
- differently from other machine learning techniques, RL does not require collected data for training.

OUTLINE

1 INTRODUCTION

2 MODEL

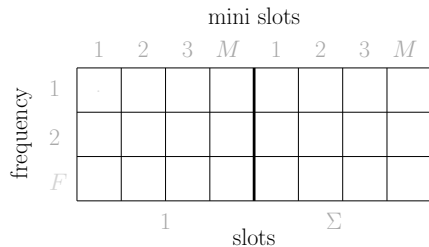
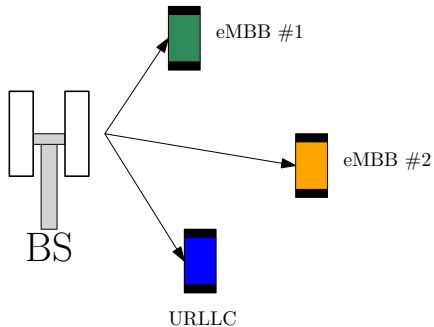
3 DEEP REINFORCEMENT LEARNING

4 RESULTS

5 CONCLUSIONS

SCENARIO & RESOURCES

We consider a downlink transmission for single cell scenario with two set of users: *eMBB* and *URLLC* users.

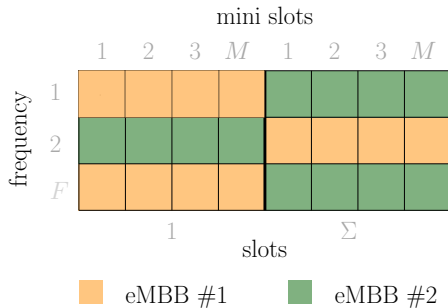


- F frequency resources;
- time resources are divided in Σ time slots;
- each time slot is divided in M mini slots.

ENHANCED MOBILE BROADBAND (eMBB)

SPECIFICATIONS:

- high throughput;
- no latency requirement;
- resource allocation is made on a *time slot* basis;
- perfect knowledge of channel state information (CSI) at the BS.



eMBB resource allocation can be performed following conventional methods (e.g. water-filling, wMMSE, etc.)

ULTRA RELIABLE LOW-LATENCY COMMUNICATION (URLLC)

SPECIFICATIONS:

- low outage probability ($\leq 10^{-5}$);
- stringent latency requirement (≤ 1 ms);
- resource allocation is made on a *mini slot* basis

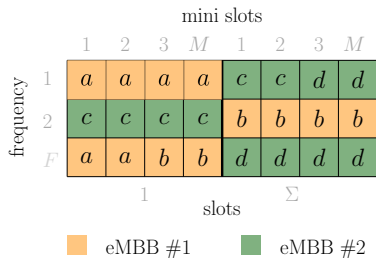
IN OUR SETTING, EACH PACKET:

- has length equal to a mini slot;
- is generated in each mini slot following a Bernoulli with probability p_u ;
- has a strict latency requirement l_u^{\max} ;
- is stored in a in FIFO queue \mathcal{Q} of infinite length;

URLLC AND EMBB COEXISTENCE

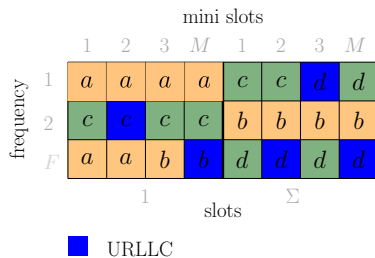
EMBB CODEWORDS:

- the BS distributes different codeword for each user;
- codeword length is multiple of a mini slot;
- each codeword is protected by a code able to recover D erased mini slots.



URLLC PUNCTURING:

- each URLLC packet is transmitted through puncturing selecting a time-freq resource;
- a transmitted URLLC packet is always *successfully decoded*.



OUTLINE

1 INTRODUCTION

2 MODEL

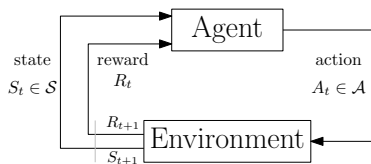
3 DEEP REINFORCEMENT LEARNING

4 RESULTS

5 CONCLUSIONS

REINFORCEMENT LEARNING

Reinforcement Learning is a field of Machine Learning studying the behaviour (policy) of a certain agent (model) acting in an environment (in which Markov property holds).



- The agent accumulates a *discounted return* $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$ with $\gamma \in [0, 1)$;
- The probability distribution of G_t depends on the policy $\pi(a|s)$ that chooses the action a for each possible state s ;

The objective of RL is to find a policy that maximizes the expected discounted return.

MODEL AS A MARKOV DECISION PROCESS

Each step t of the simulation is a mini slot of the allocation grid.

- *Actions:* $\mathcal{A}_t = \{0, 1, \dots, F\}$, where 0 means no transmission, while otherwise the action indicates the FR index.
- *State:* $\mathcal{S}_t = \{\mathcal{S}_t^{(u)}, \mathcal{S}_t^{(e)}\}$ where
 - $\mathcal{S}_t^{(u)} = \{Q_t, \Delta_t\}$
 - $\mathcal{S}_t^{(e)} = \{s_t(f)\}_{f=1}^F$ tracking how much the eMBB codeword on (t, f) is protected.
- *Reward:* $R_t = \sum_{w \in \mathcal{W}_t} e_t(w) + L_t$

$$e_t(w) = \begin{cases} -1, & \text{if } \mathcal{A}_t \text{ causes the outage of } w, \\ 0, & \text{otherwise,} \end{cases} \quad L_t = \begin{cases} 0, & \Delta_t \geq 0, \\ -\frac{3T}{F+1}, & \Delta_t < 0. \end{cases}$$

PPO AND NN ARCHITECTURE

PROXIMAL POLICY OPTIMIZATION (PPO):

- aims to the biggest possible improvement step on a policy without ending too far from the starting point one, thus avoiding the risk of performance collapse;
- is an actor-critic algorithm \rightarrow two different neural networks are required.

NN ARCHITECTURE:

Two completely separated subnetworks:

- *value function*: three ReLU dense layers with 128, 64, and 32 neurons + fourth with single neuron to estimate the value with no activation;
- *policy function*: three ReLU dense layers with 128, 64, and 32 neurons + dense fourth layer with $F + 1$ neurons to choose the actions with *softmax* activation.

OUTLINE

- 1 INTRODUCTION
- 2 MODEL
- 3 DEEP REINFORCEMENT LEARNING
- 4 RESULTS**
- 5 CONCLUSIONS

SIMULATION PARAMETERS

- Resource grid: $F = 12$, 10 time slot, $M = 14$ mini slot each $\rightarrow T = 140$;
- Users: 1 URLLC user, 10 eMBB users;
- Maximum delay constraint to $l_u^{\max} = 7$;
- Degree of protection of eMBB codewords: $D \in \{0, 1\}$.

LEARNING PHASE

- the parameters related to eMMB resource allocation and URLLC traffic generation are randomized on an episode basis: random allocation, codeword placement, protection of each codeword, and probability of URLLC packet.
- We initialize each episode with a random number of URLLC packets in the queue (always smaller than l_u^{\max}).

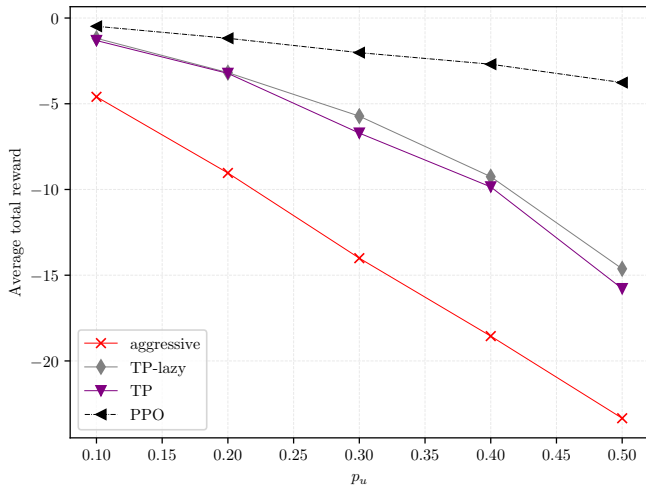
COMPARISON SCHEMES

- *Aggressive*. The URLLC packet is transmitted immediately on a randomly chosen frequency.
- *Threshold Proportional (TP)*. The URLLC packet is transmitted immediately on the frequency resource occupied by the codeword with the highest puncturing threshold¹.
- *TP-lazy*. As long as $\Delta_t > 0$, the packet is transmitted only if the present state is somehow better (or equal) than the next one. If $\Delta_t = 0$, the transmission is forced in the present mini slot. In any case, the choice of the frequency is made according to the TP scheme.

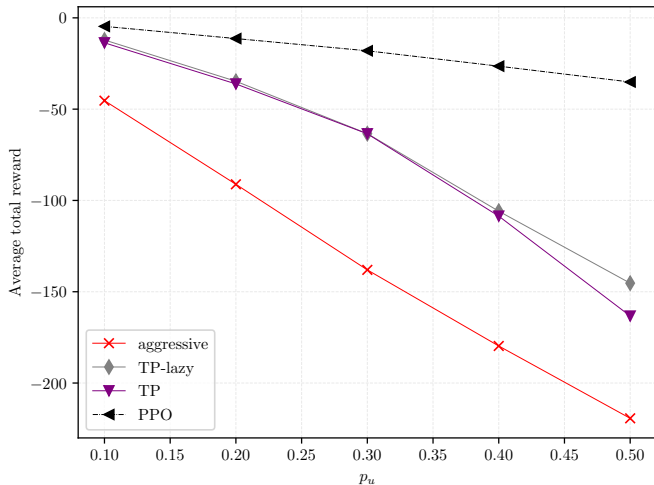
¹A. Anand, G. de Veciana, and S. Shakkottai. “Joint Scheduling of URLLC and eMBB Traffic in 5G Wireless



AVERAGE TOTAL REWARD, $T = 140$

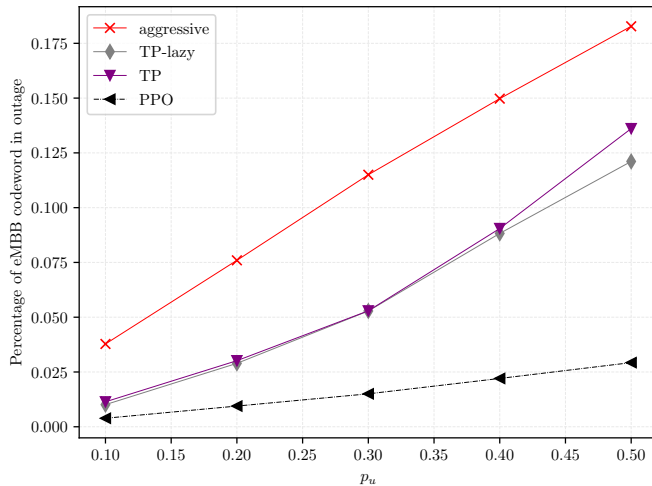


AVERAGE TOTAL REWARD, $T = 1400$

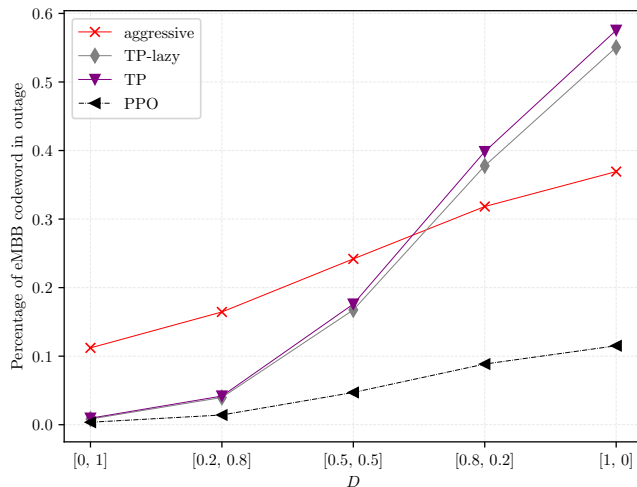


The same NN trained for $T = 140$ is used here, proving the generalization capacity of the agent.

EMBB CODEWORD IN OUTAGE VS p_u , $T = 1400$



EMBB CODEWORD IN OUTAGE VS DEGREE OF PROTECTION



OUTLINE

- 1 INTRODUCTION
- 2 MODEL
- 3 DEEP REINFORCEMENT LEARNING
- 4 RESULTS
- 5 CONCLUSIONS

CONCLUSIONS

- We proposed a Deep Reinforcement Learning approach to the slicing task with respect to eMBB and URLLC traffic;
- Our agent learns from scratch a policy outperforming all the man-made schemes over multiple performance metrics;
- The agent's learned policy is agnostic to the length of each episode allowing for fast self-training while still being applicable to the real world task.

FUTURE WORKS

- Taking into account the reliability of the URLLC user;
- Adopting a Poissonian distribution to better simulate the generation of URLLC packets;
- the agent should be able to transmit more than one packet over multiple frequencies;
- Enabling the Power Domain Non-Orthogonal Multiple Access (NOMA) communication.

BIBLIOGRAPHY & RESOURCES

- **Paper:** Fabio Saggese, Luca Pasqualini, Marco Moretti, and Andrea Abrardo. *Deep Reinforcement Learning for URLLC data management on top of scheduled eMBB traffic*. 2021. arXiv: 2103.01801 [eess.SP]
- **Github:** <https://github.com/InsaneMonster/teler12021>
- **Framework:** <https://github.com/InsaneMonster/USienaRL>

Thank you for the attention.



ABOUT ME

Fabio Saggese, fabio.saggese@phd.unipi.it
PhD. Student in Information Engineering,
University of Pisa