Deep Reinforcement Learning for Latency-Sensitive Multimedia Applications Over 5G Networks

Anjali Omer†, Filippo Malandra†, Nicholas Mastronarde†, and Jacob Chakareski‡

†Department of Electrical Engineering, ‡University at Buffalo, Ying Wu College of Computing, New Jersey Institute of Technology

Introduction

- Multimedia applications running over 5G network have stringent QoS requirements such as low latency and high throughput
- It is imperative for the packet scheduler to make fast and accurate scheduling decisions every TTI, fairly allocating the available resources
- Rule based approaches are straightforward and easy to implement but not delay optimal
- **Our Contribution:** We employ various deep reinforcement learning based approaches to find an optimal scheduling policy that minimizes the user delay

System Model

In 5G every TTI, RBs are distributed among active users to satisfy their QoS requirements based on a comparison of per-RB metrics

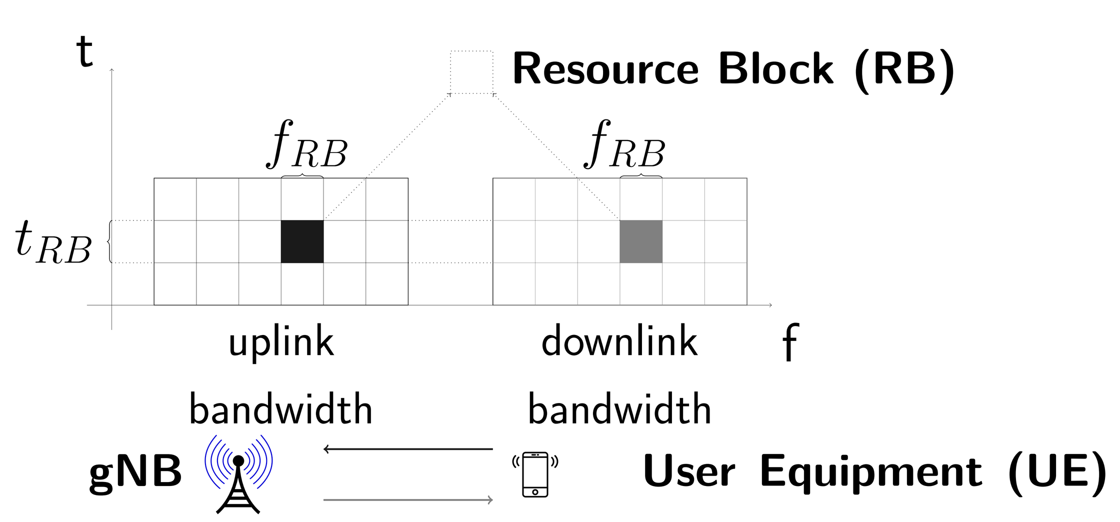


Figure: LTE Resource Grid.

We consider a downlink scheduler, where a basestation allocates M RBs to N requesting UEs

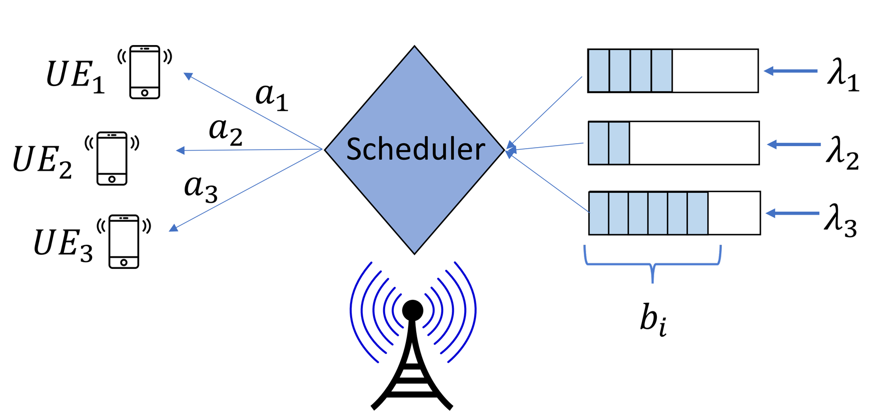


Figure: System Model. λ_i, b_i, and a_i represent the arrival rate, buffer status, and allocated bits, respectively, for UE i.

Channel Model: Based on the Urban Macro (UMa) channel model [2]. As per the model, the pathloss experienced by a UE at a distance of d from the basestation is given by:

$$PL_{dB} = 128.1 + 37.6 \log_{10}(d_{km})$$

Each channel undergoes log normal fading.

$$X_{dB} \sim \mathcal{N}(\mu, \sigma^2)$$

The channel state of each UE is characterized by its Signal to Interference plus Noise Ratio (SINR) which is subsequently mapped to Transport Block Size (TBS) index

SINR → CQI → MCS Index → TBS Index

TBS index is used to determine the transmission rate using the TBS table [2].

Buffer Model: The packet arrival process is assumed to be independent and identically distributed. The length of each packet is fixed. The buffer status of UE k at time $t+1$ is given by the Lindley's recursive equation:

$$b_k^{t+1} = \max\left(b_k^t - a_k^t, 0\right) + l_k^t$$

where b_k^t and b_k^{t+1} denotes the buffer state at time t and $t+1$ resp., a_k^t and l_k^t denote the transmitted bits and arrived bits in slot t resp.

DRL-Based Scheduler

We consider the following DRL based approaches to determine an optimal scheduling policy:

DDPG: Designed to solve problems involving continuous, high-dimensional and large action spaces but is shown to perform well in problems such as ours involving large but discrete action spaces[3][1]

DQN: A deep learning based Q-learning algorithm that represents the optimal action-value function as a neural network instead of a table

Problem Formulation

State Space : Defined as a $NM + N$ long vector containing information regarding each UE's buffer state and CQI value in all RBs

Action Space: Depends on the DRL Algorithm used:

- DDPG: A vector of size $[1 \times M]$ whose ith element represents the UE that is assigned to RB i.
- DQN: A chosen algorithm out of a number of available rule-based algorithms such as Max CQI, Max Buffer and Max Weight

Cost: To penalize large queue backlogs, we define buffer cost $c_i^t(s_i^t, a_i^t)$ for each UE

$$c_i^t(s_i^t, a_i^t) = b_i^{(t+1)} - b_i^t$$

The total cost incurred in time slot t is defined as

$$c^t(s^t, a^t) = \sum_{i=1}^{N} c_i^t(s_i^t, a_i^t)$$

Objective: To determine an optimal policy π^* that minimizes the average sum of queuing delays across the UEs.

$$\pi : \mathcal{S} \rightarrow \mathcal{A}$$

Experimental Results

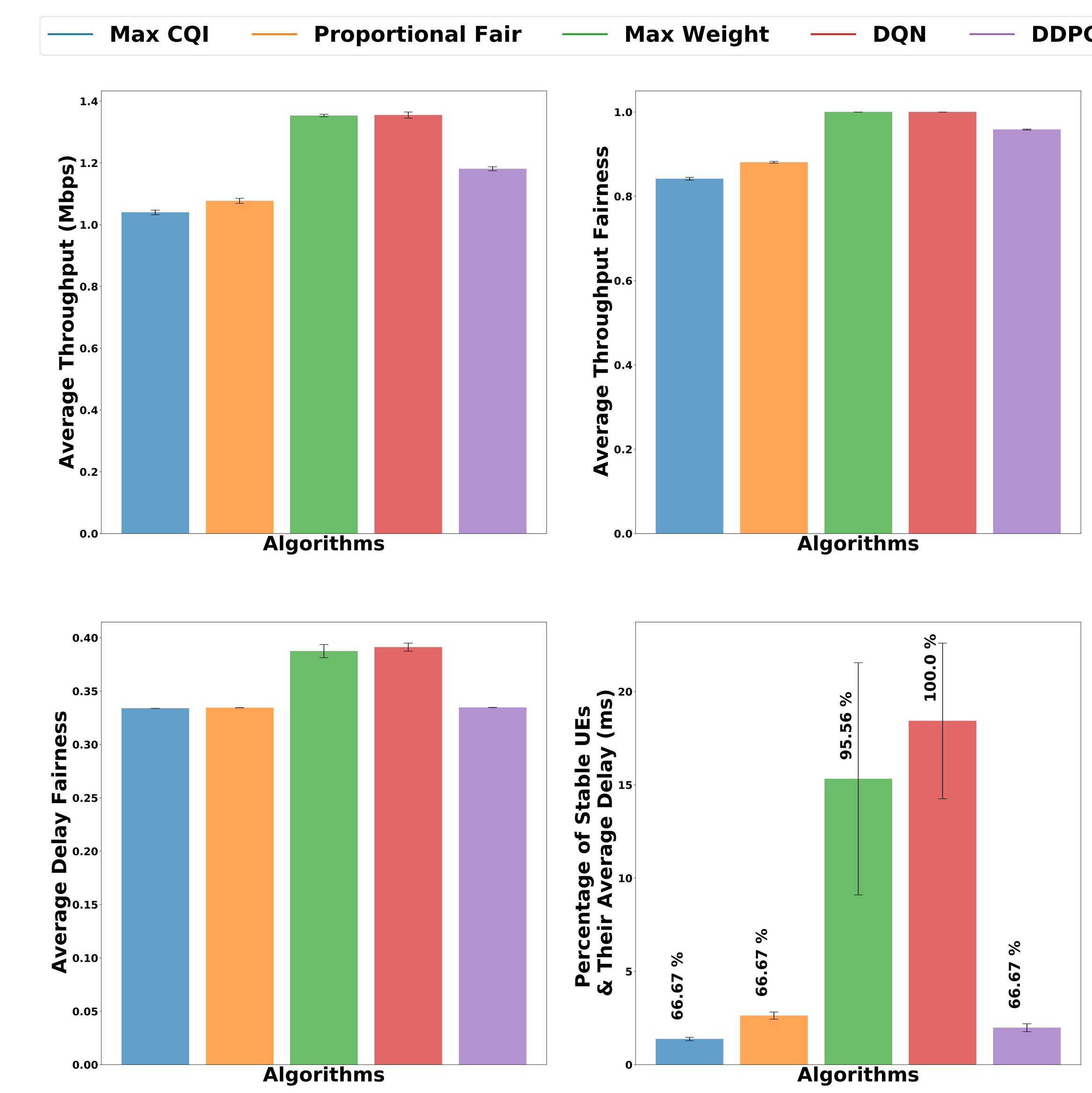
Benchmark Algorithms: Rule-based algorithms such as Max CQI, Proportional Fair (PF) and Max Weight

Performance Metrics: Include

- Average network throughput
- Average throughput fairness and average delay fairness based on Jain's fairness index
- Percentage of stable UEs and their average queuing delay. Stable UE is the one whose buffer doesn't grow infinitely

All results are averaged over 15 epochs of 10000 slots each.

Setup: We consider a hexagonal grid with 6 interfering BSs, a central BS, 3 UEs with fixed location and arrival rate of 450 Kbps, 6 RBGs with sytem bandwidth 1.4 MHz. The packet arrival process is based on Poisson Distribution



Conclusion

- In the given limited scenarios, DQN performed better than other algorithms and was able to learn policy that ensured buffer stability for all UEs
- As future work, we plan to test more scenarios with higher data rate, more RBGs and UEs, mobility and variable data rate.

References

[1] N. Sharma, *et al.*, "Deep Reinforcement Learning for Delay-Sensitive LTE Downlink Scheduling,"

[2] 3GPP TR 25.913, "Requirements for Evolved UTRA (E-UTRA) and Evolved UTRAN,"

[3] G. Dulac-Arnold, *et al.*, "Deep reinforcement learning in large discrete action spaces,"