Scheduling for Minimizing Uplink Delay in 5G Systems

AVD643 : Innovative Design Project

Sudarsanan A K
SC21M063

Guide : Dr. Vineeth B S

Department of Avionics
Indian Institute of Space science and Technology

May 24, 2022

Overview

Introduction

- The main objective of the Radio Resource Scheduling (RRS) task is to dynamically allocate spectrum resources to UEs.
- For 5G network management, it is difficult to model the network state and traffic due to the diversity of the applications and traffic it supports. [1]

- Network dynamics are difficult to anticipate and exact mathematical models are not scalable. [1]
- Large state space composed of different properties of the UEs in the network.
- Large action space as the number of UEs in the network grow.
- Task to fill the resource grid by deciding which UE will win the current RBG in the current slot.
- Our focus is on the delay minimization in uplink.

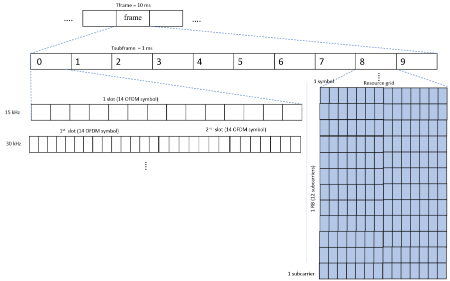

Figure: 5G NR frame structure, resource grid and basic terminologies

[2]

Literature Survey

Paper	Major Findings
F. Al-Tam, N. Correia and J. Rodriguez, **"Learn to Schedule (LEASCH): A Deep Reinforcement Learning Approach for Radio Resource Scheduling in the 5G MAC Layer,"** in IEEE Access, vol. 8, pp. 108088-108101, 2020, doi: 10.1109/ACCESS.2020.3000893.	▪ Fine-grained centralized scheduling strategy ▪ State -Space is a combination of eligibility, data rate, and fairness ▪ Action space is selection of a UE ▪ Reward is a variant of a discounted bestCQI function
A. Anand, G. de Veciana and S. Shakkottai, **"Joint Scheduling of URLLC and eMBB Traffic in 5G Wireless Networks,"** in *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 477-490, April 2020, doi: 10.1109/TNET.2020.2968373.	▪ Uses the notion of minislots ▪ "superposition/puncturing" framework for multiplexing eMBB and URLLC in 5G cellular system
D. Zavyalova and V. Drozdova, **"5G Scheduling using Reinforcement Learning,"** *2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, 2020, pp. 1-5, doi: 10.1109/FarEastCon50210.2020.9271421.	▪ Agent: DNN with 1D convolution layer (conv1D), softmax activation in output layer ▪ Reward is cell throughput
I. Comşa et al., **"Towards 5G: A Reinforcement Learning-Based Scheduling Solution for Data Traffic Management,"** in IEEE Transactions on Network and Service Management, vol. 15, no. 4, pp. 1661-1675, Dec. 2018, doi: 10.1109/TNSM.2018.2863563.	▪ Flexible RRM packet scheduler ▪ RL-based framework ▪ Neural Networks (NNs) based rule selection ▪ Scheduler state space compression technique

Figure: Literature review summary

- The problem is to fill the resource grid by deciding which UE will win the current RBG in the current slot while achieving an optimal trade-off between some key performance indicators (KPIs), like spectrum efficiency (SE), fairness, delay, and so on. We focus on the buffer status report (BSR) as an indication of the delay in the uplink.

Reinforcement Learning Setup

- The environment is the 5G stack modelled by a 5G scheduling simulator.
- We need a 5G scheduling simulator.
- Two main challenges
  1. Environment - we have to build the simulation environment.
  2. Agent - design of RL algorithms for scheduling.

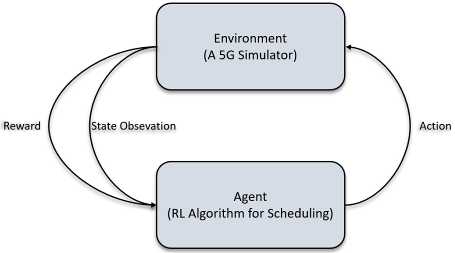

Figure: RL setup

5G Scheduling Simulators

5G-LENA Simulator[3]

- 5G-LENA is a GPLv2 New Radio (NR) network simulator, designed as a pluggable module to ns-3.
- It incorporates fundamental PHY-MAC NR features aligned with NR Release 15 TS 38.300.

MATLAB[4]

- 5G Toolbox™ provides standard-compliant functions and reference examples for the modeling, simulation, and verification of 5G New Radio (NR) communication systems
- The simulations operate across a protocol stack that includes physical (PHY), medium access control (MAC), radio link control (RLC), and application layers.

Tetcos Netsim[5]

- NetSim is an end-to-end, full stack, packet level network simulator and emulator.

We have chosen MATLAB for the simulation because of our background in MATLAB and limited time. Evaluation of above mentioned platforms is part of future work.

Some Details of MATLAB 5G Scheduler

This simulator models the following

- Slot-based and symbol-based DL and UL scheduling.
- Non-contiguous allocation of frequency-domain resources in terms of resource block groups (RBGs).
- Configurable subcarrier spacing (SCS) resulting in different slot durations.
- Asynchronous adaptive hybrid automatic repeat request (HARQ) mechanism in UL and DL.
- Multiple logical channels (LCHs) to support different kind of applications.

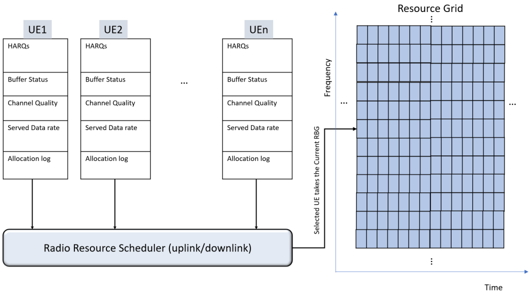

Figure: Radio resource scheduler

Scheduler Implementations

Existing Implementations

Round-Robin

```
scheduledUE  = lastSelectedUE
while i < length(UEs)
  scheduledUE = mod(scheduledUE,
length(UE))+1
  index = find(eligibleUE == selectedUE, 1)
  if index is not empty
    bufferStatus = bufferStatus(index)
    if bufferStatus > 0
      selectedUE = eligibleUE(index)
    break
    end
   end
   i = i+1
end
```

Best CQI

```
selectedUE = -1
bestAvgCQI = 0
while i < length(eligibleUEs)
  bufferStatus = bufferStatus(i)
  if  bufferStatus > 0
    cqiRBG = cqiRBG(i, :)
    cqiAvg = floor(mean(cqiRBG))
    if(cqiAvg > bestAvgCQI)
       bestAvgCQI = cqiAvg
       selectedUE = eligibleUEs(i)
    end
   end
  i = i+1
end
```

Implementation 1: Backpressure based Policy

- Weight the eligible UEs with product of their buffer status and the achievable data rate
- Select the UE with maximum weight in the current slot
- The intuition behind this policy is that we should try to serve the UE with maximum queue length and which also has a good rate so that the queue length would come down from a large value.

Backpressure based Policy

```
Initialize selectedUE, maxWeightage, mcsIndex
while i < length(UEs)
 index = find(eligibleUEs == i, 1)
 if index not empty
    bufferStatus = bufferStatus(index);
    bitsPerSym = mcsRBG(index)
    numLayers = selectedRank(index);
    achievableDataRate = ((numLayers *RBGSize * bitsPerSym * 14 * 12)*1000)/ttiDur
    Weightage = achievableDataRate * bufferStatus
    if Weightage > maxWeightage
      maxWeightage = Weightage;
      selectedUE = eligibleUEs(index);
      mcsIndex = mcsRBG(index);
    end
  end
  i = i+1
end
```

Implementation 2: SARSA based Policy

- Slotwise RL based scheduler.
- Sequential decisions in both frequency and time.
- Compress the state space by quantization.
- Find the current state.
- Apply ϵ - greedy policy.
- Update the Q function according to SARSA algorithm. [6]

SARSA based Policy

```
#Initialize the parameters
# finding out the current state
State space  -  bufferStatus , achievable data rate of all the eligible UEs
Actions -  which UE to assign in a slot
Reward = −∑_{k=0}^{eligibleUEs} bufferStatus_k
Quantize the state space to reduce complexity
while i < length(UEs)
   index = find(eligibleUE == i, 1)
   if index is not empty
        current state ← bufferStatus and achievable data rate of UE indexed by index
   end
   i = i+1
end

 #Implementation of epsilon greedy policy
if rand() < epsilon
  select  UE randomly from the eligible ones
else
   select the UE which maximizes the Q function corresponding  to the current state
end

# Q function updation - SARSA
Q function(last state) = Q function(last state)  + α*(last reward + ɣ* Q function(current state) - Q function(last state) )
laststate = current_state
last action = selectedUE
lastreward = -sum(obj.weight.*current_state
```

$$Q(lastState) = Q(lastState) + \alpha(lastReward + \gamma Q(currentState) - Q(lastState))$$

Results

UE-x(n) : Transmission
UE-x(n) : Retransmission
x : UE RNTI
n : HARQ Process ID
Frame Number : 99
Select Link: Downlink
Select RB range: RB 0-19

Resource Grid Allocation for Cell ID - 1

Resource Blocks	Slot-0	Slot-1	Slot-2	Slot-3	Slot-4	Slot-5	Slot-6	Slot-7	Slot-8	Slot-9
RB-19	UE-3(3)	UE-4(0)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-2(3)
RB-18	UE-3(3)	UE-4(0)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-2(3)
RB-17	UE-3(3)	UE-4(0)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-2(3)
RB-16	UE-3(3)	UE-4(0)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-2(3)
RB-15	UE-1(4)	UE-1(5)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-3(4)
RB-14	UE-1(4)	UE-1(5)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-3(4)
RB-13	UE-1(4)	UE-1(5)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-3(4)
RB-12	UE-1(4)	UE-1(5)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-3(4)
RB-11	UE-1(4)	UE-1(5)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-3(4)
RB-10	UE-1(4)	UE-1(5)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-3(4)
RB-9	UE-1(4)	UE-1(5)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-3(4)
RB-8	UE-1(4)	UE-1(5)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-3(4)
RB-7	UE-1(4)	UE-1(5)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-3(4)
RB-6	UE-1(4)	UE-1(5)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-3(4)
RB-5	UE-1(4)	UE-1(5)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-3(4)
RB-4	UE-1(4)	UE-1(5)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-3(4)
RB-3	UE-1(4)	UE-1(5)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-3(4)
RB-2	UE-1(4)	UE-1(5)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-3(4)
RB-1	UE-1(4)	UE-1(5)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-3(4)
RB-0	UE-1(4)	UE-1(5)	UE-1(6)	UE-1(2)	UE-1(1)	UE-4(3)	UE-4(4)	UE-3(2)	UE-2(0)	UE-3(4)

Slots in 10 ms Frame

Figure: Resource allocation in SARSA based policy

Results

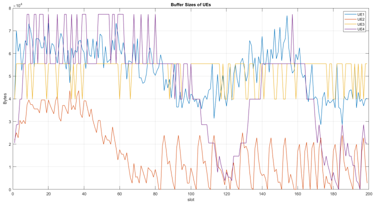

Figure: Buffer status of UEs under Round-Robin policy

Figure: Buffer status of UEs under Best CQI policy

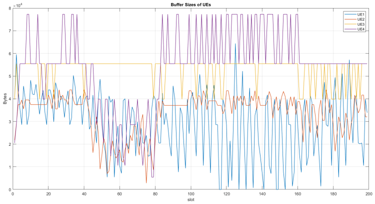

Figure: Buffer status of UEs under backpressure based policy

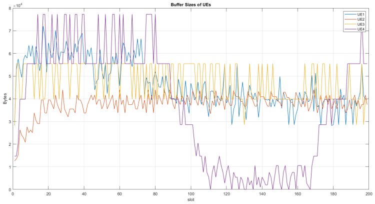

Figure: Buffer status of UEs under SARSA based policy

Results

Performance Metric		RR	BestCQI	Back Pressure	RL SARSA (α = 0.1)	RL SARSA (α = 0.05)
Time Average BSR of UEs (in Bytes)	UE 1	52651.43719	52621.09548	29302.37186	59273.07538	47041.88945
	UE2	16598.8191	21744.9196	33440.91457	12142.44724	37584.14573
	UE3	52563.08543	48165.62312	52956.45226	52663.96482	47695.48744
	UE4	45444.86935	49845.55276	54557.80402	60127.23618	37179.00503
Average BSR across UEs (in Bytes)		**4.18E+04**	4.31E+04	**4.26E+04**	4.61E+04	**4.24E+04**

Figure: Performance metric comparison

Future Work

- Extending the simulation with more number of users
- Including 5G Toolbox™ PHY layer for more realistic modelling
- Simulations with large simulation times
- Backpressure's Q can be used as the initial value for the RL algorithm.
- Parameter tuning and detailed simulations are required.

Appendix

Simulation parameter values

- Number of UEs = 4
- UE positions = [100, 0, 0; 250, 0, 0; 700, 0, 0; 750, 0, 0]
- Number of Frames = 100
- DL Bandwidth = 30 MHz
- UL Bandwidth = 30 MHz
- Number of RBs = 160
- SCS = 15 kHz
- DL carrier frequency = 2.635 GHz
- UL carrier frequency = 2.515 GHz
- Channel update periodicity = 0.2 sec
- CQI Delta = 2
- CQIvsDistance = [200, 15; 500, 12; 800, 10; 1000, 8; 1200, 7]
- BSR periodicity = 5 ms
- Number of HARQ = 16
- Scheduler periodicity = 4 slots

References

- F. Al-Tam, N. Correia, and J. Rodriguez, "Learn to schedule (leasch): A deep reinforcement learning approach for radio resource scheduling in the 5g mac layer," *IEEE Access*, vol. 8, pp. 108088–108101, 2020.
- L. Marijanović, S. Schwarz, and M. Rupp, "Optimal resource allocation with flexible numerology," in *2018 IEEE International Conference on Communication Systems (ICCS)*, pp. 136–141, 2018.
- A. M. group.
- Mathworks, "Nr fdd scheduling performance evaluation." Available at `https://in.mathworks.com/help/5g/ug/nr-fdd-scheduling-performance-evaluation.html`.
- T. develops NetSim.
- R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, second ed., 2018.