# MICAS912: Sequential Decision-Making Processing
# Part II - Reinforcement Learning

Mireille Sarkiss

Abdelaziz Bounhar, Ibrahim Djemai

## Lab 1

## 1 Introduction

In this Lab, we aim to tackle a practical problem for scheduling the communication of users in a network. Communication networks commonly accommodate various applications with distinct security requirements, levels of importance, and delay constraints. In a context where only a given number of devices can communicate simultaneously, scheduling becomes a challenging problem. In particular, at each time-instance it has to be decided which devices should transmit, and which communications can be deferred to a latter stage.

This lab focuses on the **centralized scheduling problem** where a central entity (here the base station) does the scheduling of the devices based on global knowledge of the system.

## 2 Problem Statement

We consider the communication setup in Figure 1. A Base Station (BS) serves two User Equipments. Both UEs wishes to communicate a number of data packets (of some number of bits). The users store their data packets in a memory buffer while waiting for communication (we shall detail in the following section the buffer model).
Furthermore, both UEs are equipped with some semantic information. More specifically, for each UE $u \in \{1, 2\}$, the following attributes are assigned

- $\Delta_u \in \mathbb{R}_0^+$: a **maximum delay constraint** which allows the device $u$ to hold its communication from a given random wake-up time $t_u$ until a limiting time $t_u + \Delta_u$, otherwise the packets are removed from the buffer and the agent is penalized.

At each time slot $t \in \mathcal{T}$, it has to be decided which user should transmit and which communication can be delayed to a latter stage. In this setup, we consider the **centralised** case, i.e. the BS acts as the centralized decision center that selects which user should communicate as well as the number of data packets he can transmit.

Due to concerns related to computational complexity, we examine this problem in its simplest form. For each user $k \in 1, 2$, we assume a **maximum delay of 1**, enabling us to represent the buffer with a variable tracking the number of packets it contains. Additionally, the **maximum allowable packets in the buffer** is capped at **1**, and the **battery level** is constrained not to exceed **2 energy units**. Finally, we assume that the BS takes actions based on a **quantized** version of the channel. Therefore, the **channel** is modeled by a **3-level** (discrete) variable that indicate the quality of the SNR;

- level 0: user 1 is the only one that can communicate,
- level 1: user 2 is the only one that can communicate,

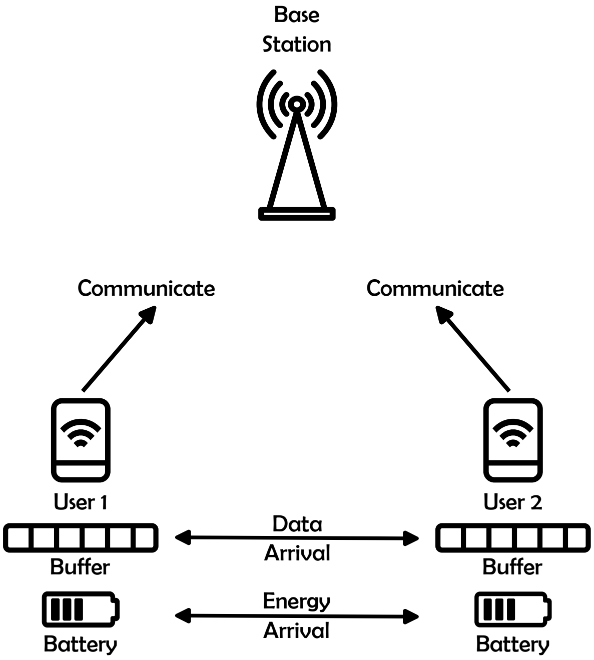

Figure 1: System model in the Uplink (UL) and Downlink (DL) for the centralized scheduling problem of one covert and one non-covert users

- level 2: both users can communicate simultaneously,

Both **data arrival** and **energy arrival** are modeled by **Bernoulli** $\mathcal{B}(p)$ random variables for $p \in [0,1]$. In contrast, the channel quality **uniformly** transitions between states $\{0,1,2\}$.

# 3 Markov Decision Processes

As the data buffer, battery and channel states dynamics satisfy Markov property, we can formulate the problem as a Markov Decision Processes (MDP) for which we will describe the state and action spaces $\mathcal{S}$ and $\mathcal{A}$ respectively, as well as the reward model $R_t$.

## 3.1 States and Actions space

*States space*: We define our state space as a vector of:

- Buffer states of the two UEs, i.e. the vector $[D_1, D_2]$,
- Battery states of the two UEs, i.e. the vector $[B_1, B_2]$,
- The SNR level of the two UEs, i.e. the vector $[SNR_1, SNR_2]$,

We can thus write

$$\mathcal{S} = \{D_1, B_1, SNR_1, D_2, B_2, SNR_2\}. \tag{1}$$

This state space is of size

$$|\mathcal{S}| = \prod_{u=1}^{2} \left[ \underbrace{(\Delta_u + 1)^1}_{\text{Buffer space size}} \cdot \underbrace{(2+1)}_{\text{Battery space size}} \cdot \underbrace{3}_{\text{SNR space size}} \right] = 324. \tag{2}$$

*Actions space*: We define our actions space as a vector of all possible decisions that can be made by the BS, i.e. how many (including the 0) data packets to be transmitted. At each time instance $t \in \mathcal{T}$, each user $u \in \{1, 2\}$ either **communicate** or remain **idle**. Therefore

$$\mathcal{A} = \{0, \dots, 3\}. \tag{3}$$

This actions space is of size

$$|\mathcal{A}| = 4 \tag{4}$$

Which makes the states-actions space of size $324 \cdot 4 = 1296$.

### 3.2 Reward Model

The agent seeks to minimize the number of lost packets. The reward function for our problem will be the penalized sum

$$R_t = -\sum_{u=1}^{2} \sum_{j=1}^{2} \mathbb{1}\{D_u[j] > \Delta_u\}, \forall t \in \mathcal{T}. \tag{5}$$

As we formalise our problem as in Infinite Discounted Horizon problem, the overall reward function following a policy $\pi$ is defined as

$$R^{\pi} = \lim_{T \to \infty} \mathbb{E}^{\pi} \left[ \sum_{t=0}^{T} \gamma^t \cdot R_t \right], \tag{6}$$

with $\gamma$ being the discount factor.

## 4 Problem Solving

In this lab, you are given the implementation of the Model and you are asked to implement the **Value Iteration** algorithm as well as the **Policy Iteration** algorithm to solve this problem. A template of these algorithms is given to you and you should complete it. Best of success!