JOINT SCHEDULING-OFFLOADING POLICIES IN NOMA-BASED MOBILE EDGE COMPUTING SYSTEMS

Ibrahim Djemai, Mireille Sarkiss And Philippe Ciblat

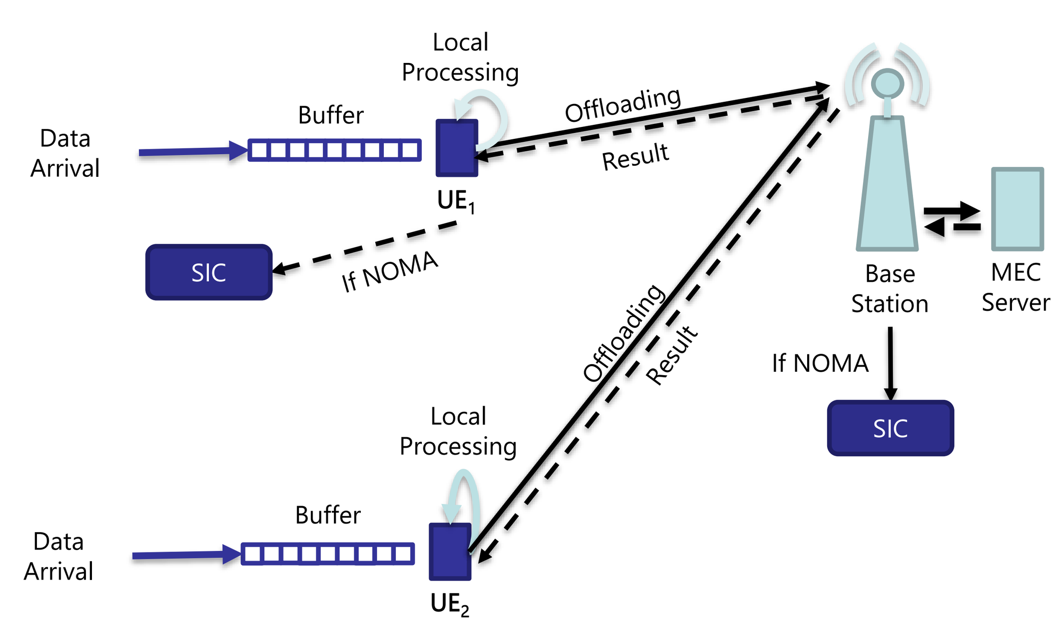
1. CONTEXT AND MOTIVATION

CONTEXT

- A wireless communications scenario with multiple NOMA users connected to Base Station (BS) that has Mobile Edge Computing Capabilities (MEC).
- Users can execute the buffered packets with strict delay either **locally** or by **offloading** them to the MEC server.

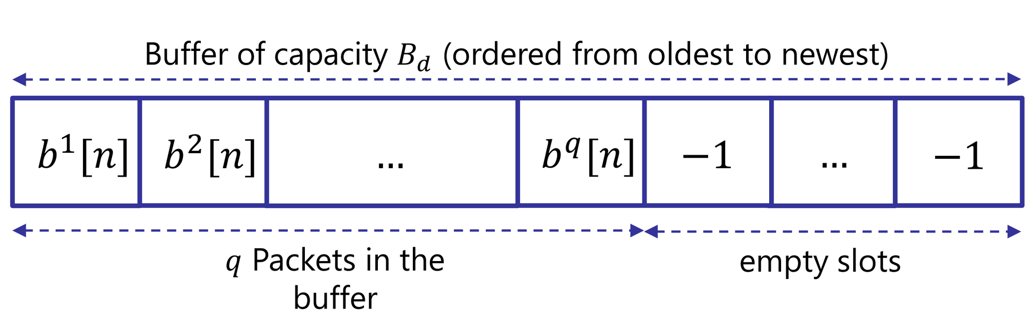
GOAL

- Design efficient policies for joint resource scheduling and computation offloading, to **minimize the overall number of dropped packets.**



2. SYSTEM MODEL

- 2 Users communicating with the BS, in a NOMA mode.
- 3 Decisions can be made at the beginning of each time slot:
  - Stay **Idle**
  - Execute packets **Locally**
  - **Offload** the packets to the BS (one or both users).
- With the number of packets to be processed.

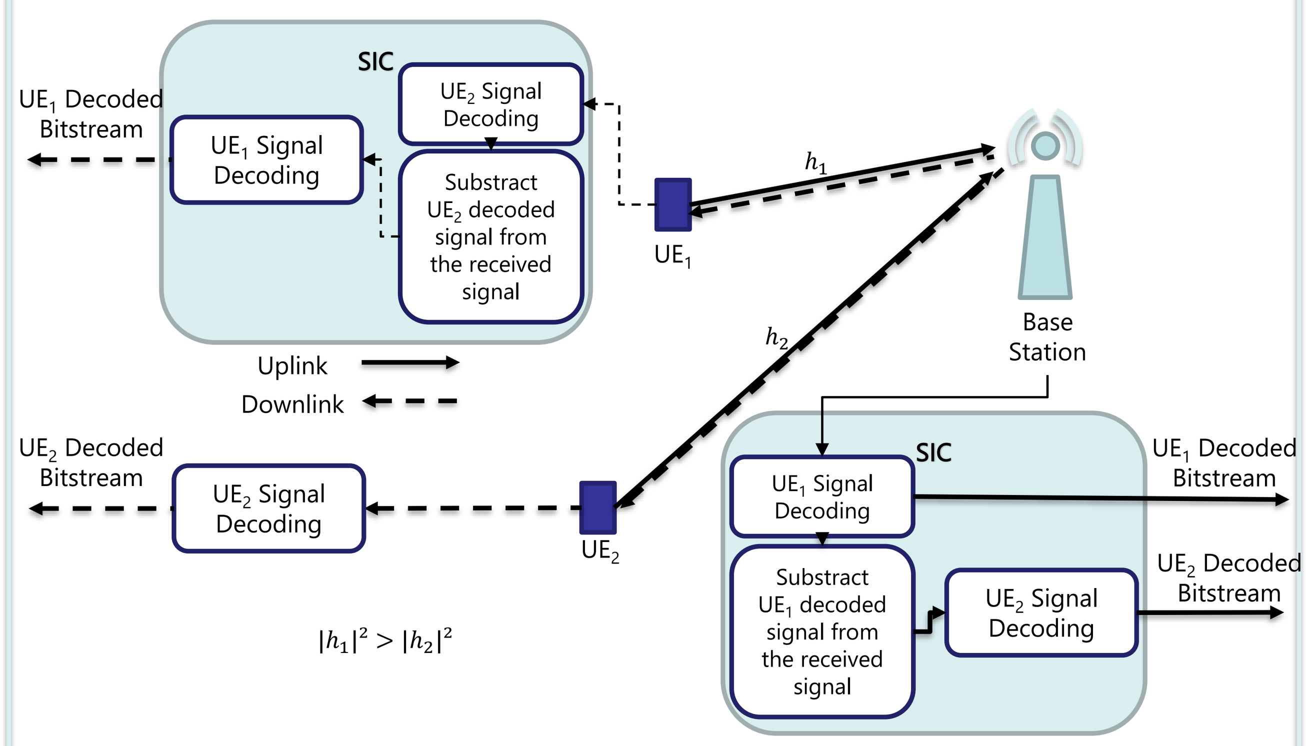


3. BUFFER AND DATA MODELS

- Random Arrival of packets following the Poisson distribution with mean λ_d
- Strict delay constraint for the buffer with size B_d
- A packet can be discarded if :
  - It reaches the maximum packet age K_0 : **Delay Violation.**
  - The buffer reaches its maximum capacity B_d : **Buffer Overflow.**

4. CHANNEL MODEL

- Rayleigh Flat-fading channel.
- $g_i = |h_i|^2$
- Exponential distribution for the channel variations.
- The channel is quantized $\tilde{g}_i = Q(g_i)$ into finite states.



5. NON-ORTHOGONAL MULTIPLE ACCESS (NOMA)

- Uplink:

$$\begin{cases} C_1^{NOMA,UL} = W^{UL} \log_2\left(1 + \dfrac{P_1^o \tilde{g}_1}{P_2^o \tilde{g}_2^+ + W^{UL} N_0}\right) \\ C_2^{NOMA,UL} = W^{UL} \log_2\left(1 + \dfrac{P_2^o \tilde{g}_2}{W^{UL} N_0}\right) \end{cases}$$

- Downlink:

$$\begin{cases} C_1^{NOMA,DL} = W^{UL} \log_2\left(1 + \dfrac{\alpha P^s \tilde{g}_1}{W^{UL} N_0}\right) \\ C_2^{NOMA,DL} = W^{UL} \log_2\left(1 + \dfrac{(1-\alpha) P^s \tilde{g}_2}{\alpha P^s \tilde{g}_2 + W^{UL} N_0}\right) \end{cases}$$

6. PROBLEM FORMULATION

- The problem is modeled as a Markov Decision Process (MDP).
- State Space : $\boldsymbol{s} = (b_1, b_2, \tilde{g}_1, \tilde{g}_2)$
- Action space : $\boldsymbol{a}$
  - Decision type (**idle**, **local** or **offload**).
  - Number of packets to transmit.
- Transition Function : $\boldsymbol{T}$

$$p(\boldsymbol{s}'|\boldsymbol{s}) = \prod_{i \in \{1,2\}} p(\boldsymbol{b}_i'|\boldsymbol{b}_i, \boldsymbol{a})\, p(\tilde{g}_i)$$

- Cost Function: $\boldsymbol{J}$

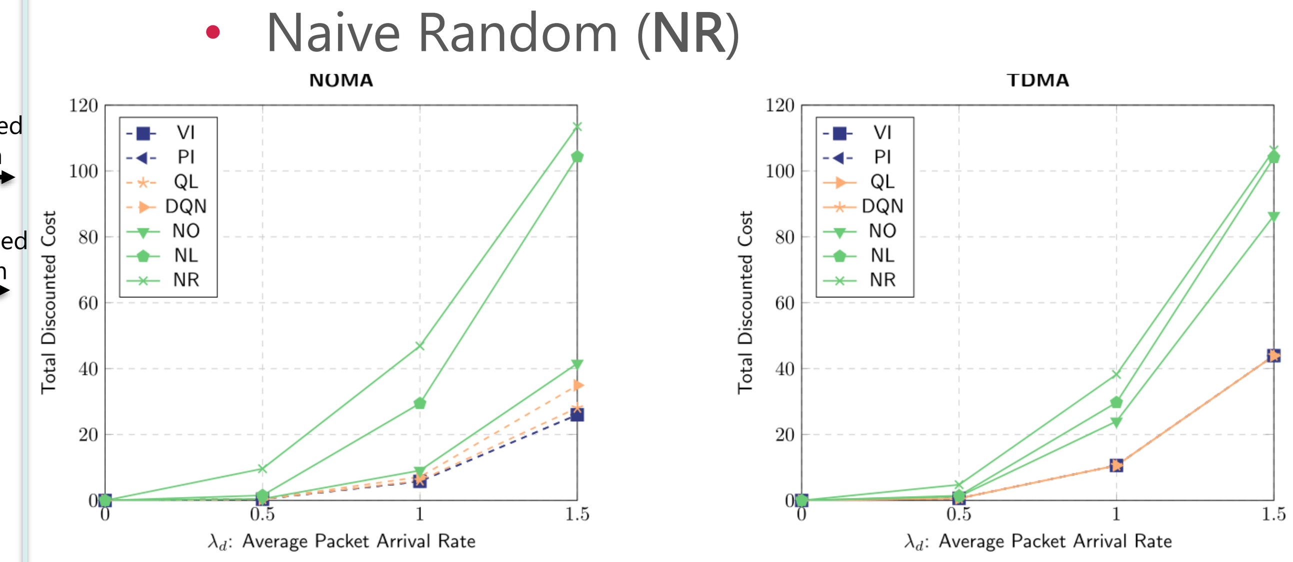
$$J^\pi = \lim_{N \to \infty} \mathbb{E}^\pi \left[\sum_{n=0}^{N} \gamma^n (c^o[n] + c^v[n]) \right]$$

- c^o: Cost due to buffer overflow
- c^v: Cost due to delay violation

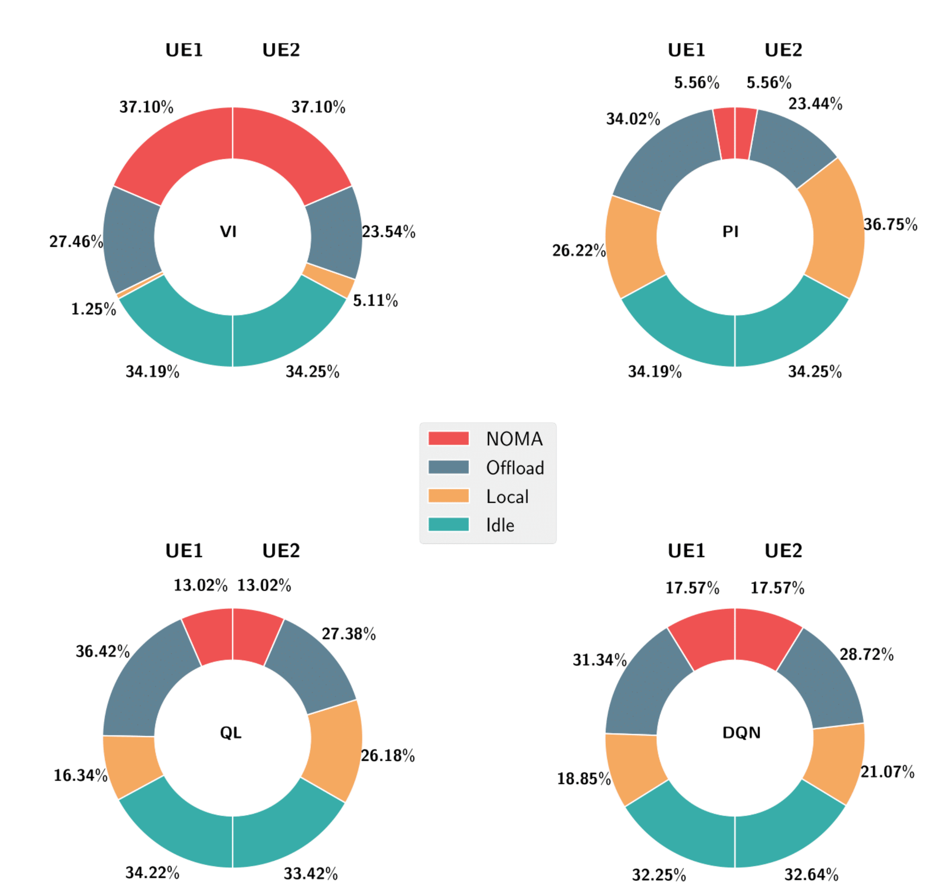
7. PROBLEM RESOLUTION

$$\pi^* = \operatorname{argmin}(J^\pi)$$

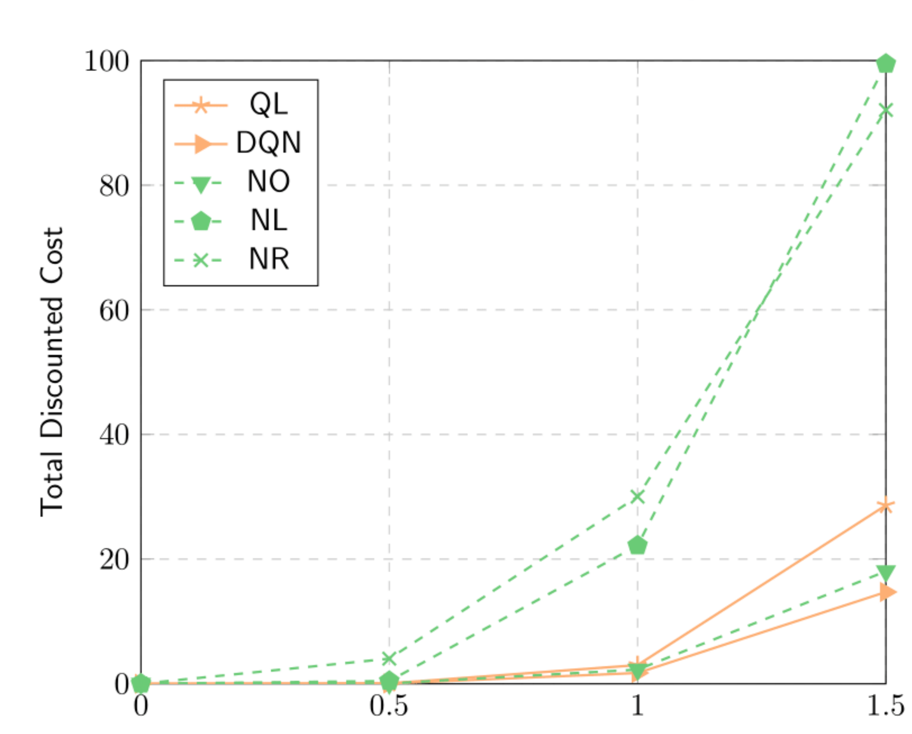
- Solve it using Sequential Decision Making and Reinforcement Learning approaches:
  - Policy Iteration (**PI**)
  - Value Iteration (**VI**)
  - Q-Learning (**QL**)
  - Deep Q-Learning Network (**DQN**)
- Compare the results against naive methods :
  - Naive Offload (**NO**)
  - Naive Local (**NL**)
  - Naive Random (**NR**)



Performance Evaluation



Action Distribution Analysis



Scalability Experiments

8. CONCLUSIONS

- PI and VI are optimal but not scalable.
- QL and DQN perform better than naive methods, and DQN scales well.
- NOMA and MEC advantages are shown

Samovar PhD day

Mars 2023

Contact ibrahim.djemai@telecom-sudparis.eu