Institut Mines-Télécom

JOINT RESOURCE SCHEDULING AND COMPUTATION OFFLOADING OF ENERGY HARVESTING DEVICES

Mireille Sarkiss
Telecom SudParis

Joint work with Ibrahim Fawaz and Philippe Ciblat

Colloque IMT : Réseaux du futur
October 14, 2021

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

IMT Nord Europe
École Mines-Télécom
IMT-Université de Lille

IMT Mines Albi-Carmaux
École Mines-Télécom

IMT Mines Alès
École Mines-Télécom

Institut Mines-Télécom
Business School

MINES
Saint-Étienne
Une école de l'IMT

TELECOM Paris
IP PARIS

TELECOM SudParis
IP PARIS

EURECOM
Sophia Antipolis

InSIC

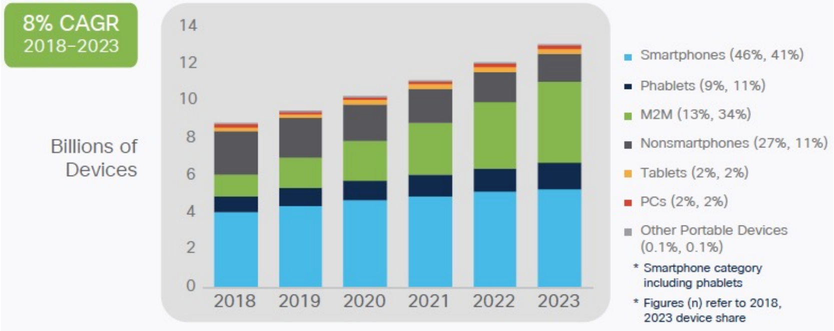
OUTLINE

1. INTRODUCTION
2. PROBLEM STATEMENT
3. MODEL - BASED APPROACH
4. MODEL - FREE APPROACH
5. CONCLUSION

Institut Mines-Télécom

- 5G and Future networks
  - Enormous number of connected devices
  - Resource-hungry applications
  - Exponential growth of mobile traffic

- Global ICT ecosystem: more than 2000 TWh of electricity annually
  - predicted to grow to 20% of global electricity demand by 2030
  - greatly increased emitted carbon footprint



Cisco report 2018-2023

Challenges and Promising Solutions

- Mobile terminals limitations:
  - Processing capacity
  - Storage
  - Energy

- Promising solutions:
  - Energy Harvesting (EH)
  - Computation offloading

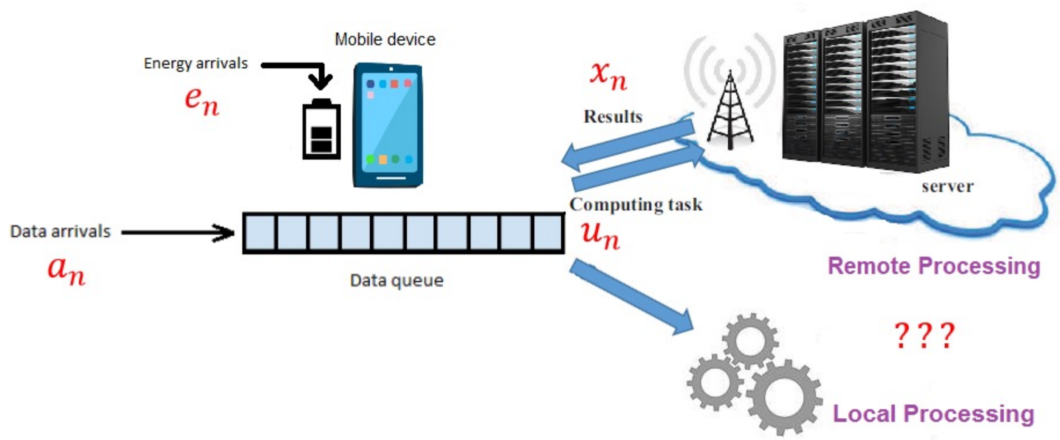
Work Objective

- Design efficient policies for resource scheduling and computation offloading under
  - Energy harvesting constraint
  - Strict delay constraint

- Optimize transmission policies taking into account:
  - Random data arrivals statistics
  - Sporadic energy arrivals statistics
  - Channel conditions
  - Packet queue status
  - Battery energy level

- Work achieved during:
  - Thesis with CEA LIST & Telecom Paris within MSAC-ITN project SCAVENGE
  - Post-doc with Telecom SudParis

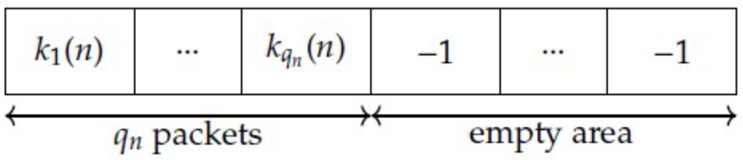
Joint Resource Scheduling and Computation Offloading



- Data arrival ~ Poisson distribution with mean λ_d
- Energy arrival ~ Poisson distribution with mean λ_e
- Constant channel during a time slot with perfect CSIT
- At the beginning of each time slot, mobile device decides:
  - Type of processing: locally or remotely
  - Number of packets to be processed

- Previous works: Average delay constraint
  - Drawback: packets can stay in the buffer for long time

- Proposed scheme: Strict delay constraint



- $k_i(n)$ is the age of the *i*-th packet at the beginning of time slot n
- A packet can be discarded due to
  - Delay violation: The *i*-th packet is discarded if $k_i(n) > K_0$
  - Buffer overflow: New arrivals are discarded if $q_n = B_d$

- 3 possible processing decisions at the beginning of each time slot:
  - Local processing: Mobile device executes u packets

$$E_\ell(u) = \left\lceil u.P_\ell.\frac{T_s}{\mathcal{E}_U} \right\rceil$$

  - Remote processing: Mobile device transmits u packets to be executed at BS

$$E_o(x,u) = \left\lceil \frac{u}{\mathcal{E}_U} \left(\frac{L.P_t}{W_{UL}.\log_2\left(1+\frac{P_t.x}{W_{UL}.N_0}\right)} + T_w.P_w + \frac{L_{DL}.P_r}{W_{DL}.\log_2\left(1+\frac{P_s.x}{W_{DL}.N_0}\right)} \right) \right\rceil$$

  - Idle: Mobile device waits for the next time slot

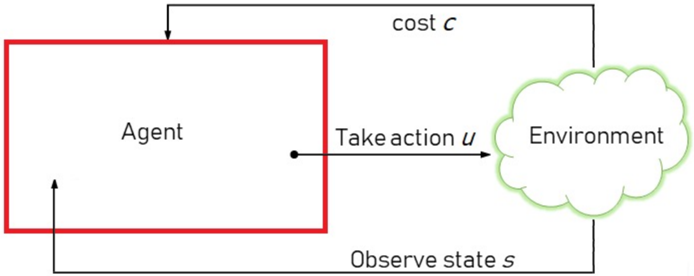
$$E_I = 0$$

- State space : $\mathcal{S} = (\mathbf{k}, b, x)$
  - $\mathbf{k} = [k_1, \cdots, k_{B_d}]$: age of each packet in the data buffer
  - b : battery level
  - x : Flat fading channel gain (quantized value)
- Action space $\mathcal{U}$: Type of processing and number of packets u
- Transition probabilities $p(s'|s,u)$
- Cost $c(s,u)$: Average number of discarded packets due to
  - Delay

$$\varepsilon_d(\mathbf{s}_n, \nu_n) = \begin{cases} 0 & \text{if } m_n = 0 \text{ or } m_n \leqslant u_{\nu_n} \\ m_n - u_{\nu_n} & \text{otherwise.} \end{cases}$$

  - Overflow

$$\varepsilon_o(\mathbf{s}_n, \nu_n) = \sum_{a=B_d-q_n+w_n+1}^{+\infty} (q_n - w_n + a - B_d).e^{-\lambda_d}.\frac{(\lambda_d)^a}{a!}$$



- Objective : Minimize the expected long-term average cost considering an infinite horizon

$$C = \lim_{T \to +\infty} \frac{1}{T} \mathbb{E}\left[\sum_{t=0}^{T-1} c(s_t, u_t)\right]$$

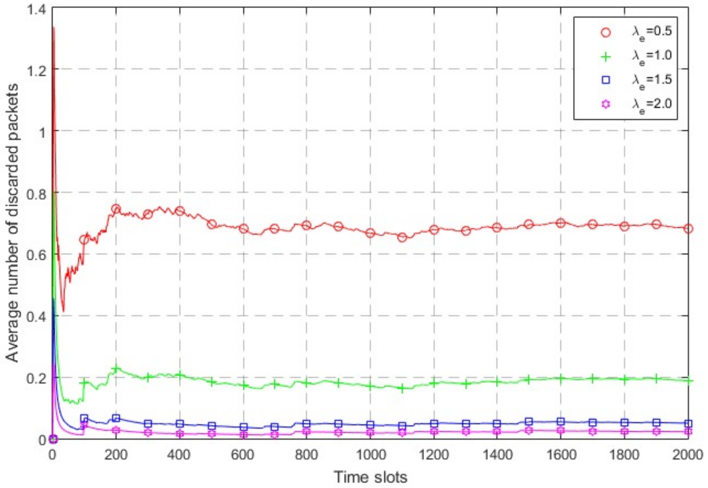
- DP : Fully-known system states and transitions
- Optimal Deterministic Offline policy $\mu^\star : \mathcal{S} \to \mathcal{U}$ using Policy Iteration algorithm
- Policy Iteration
  - Policy evaluation :

$$\beta^{n-1}\mathbf{1} + (\mathbf{Id} - \mathbf{P})\mathbf{v}^{n-1} = \mathbf{c}^{n-1}$$
$$\sum_{\mathbf{s} \in \mathcal{S}} v^{n-1}(s) = 0$$

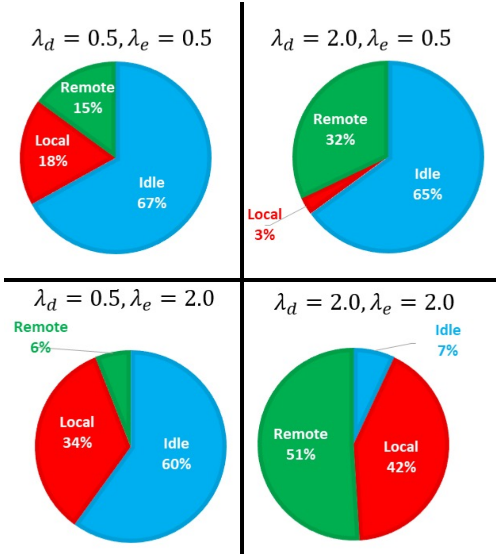
  - Policy improvement :

$$\mu^n(s) = \operatorname*{argmin}_{u \in \mathcal{U}} \left[c(s,u) + \sum_{s' \in \mathcal{S}} p(s'|s,u) v^{n-1}(s')\right]$$

Numerical Results - Convergence and Processing Decisions



Convergence of average number of discarded packets for different energy arrival rates



Percentage of processing decisions

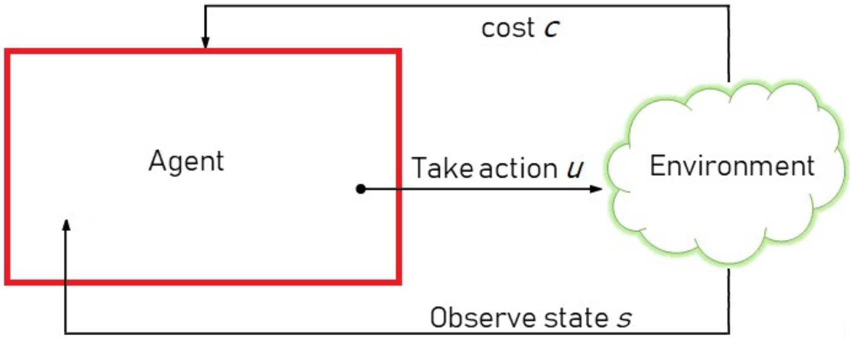
- Only few hundreds of slots are needed for the system to achieve the long-term cost
  - $\lambda_e \nearrow \;\Rightarrow$ Average number of discarded packets $\searrow$
  - $\lambda_d \nearrow \;\Rightarrow$ Idle mode $\searrow$

- DP Solution:
  - Advantage: Optimal Solution
  - Drawback: Only applicable when the environment model is known

- Alternative Solution: Reinforcement Learning (RL)
  - Learn the state-action function : $Q(s,u)$ while interacting with the environment
  - Q-Learning :
    - Updates of $Q(s,u)$ function :

$$Q(s,u) \longleftarrow$$

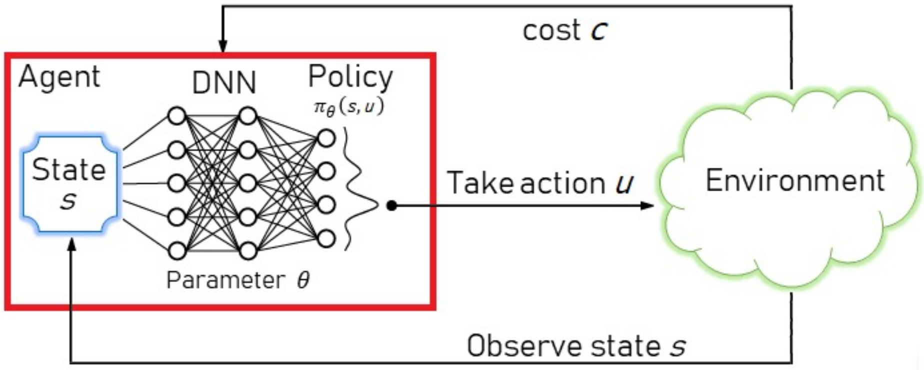
$$(1-\alpha)Q(s,u) + \alpha(c(s_n,u_n) + \min_u Q(s_{n+1},u))$$



Deep Reinforcement Learning

- DP and RL Solutions:
  - Drawback: Impractical and very complex with large system states

- Alternative Solution: Function Approximation
  - Estimate of the state-action function : $Q(s, u, \theta) \approx Q^{\star}(s, u)$
  - Non-linear function : Neural-Network (NN) $\rightarrow$ Deep Q-Network (DQN)

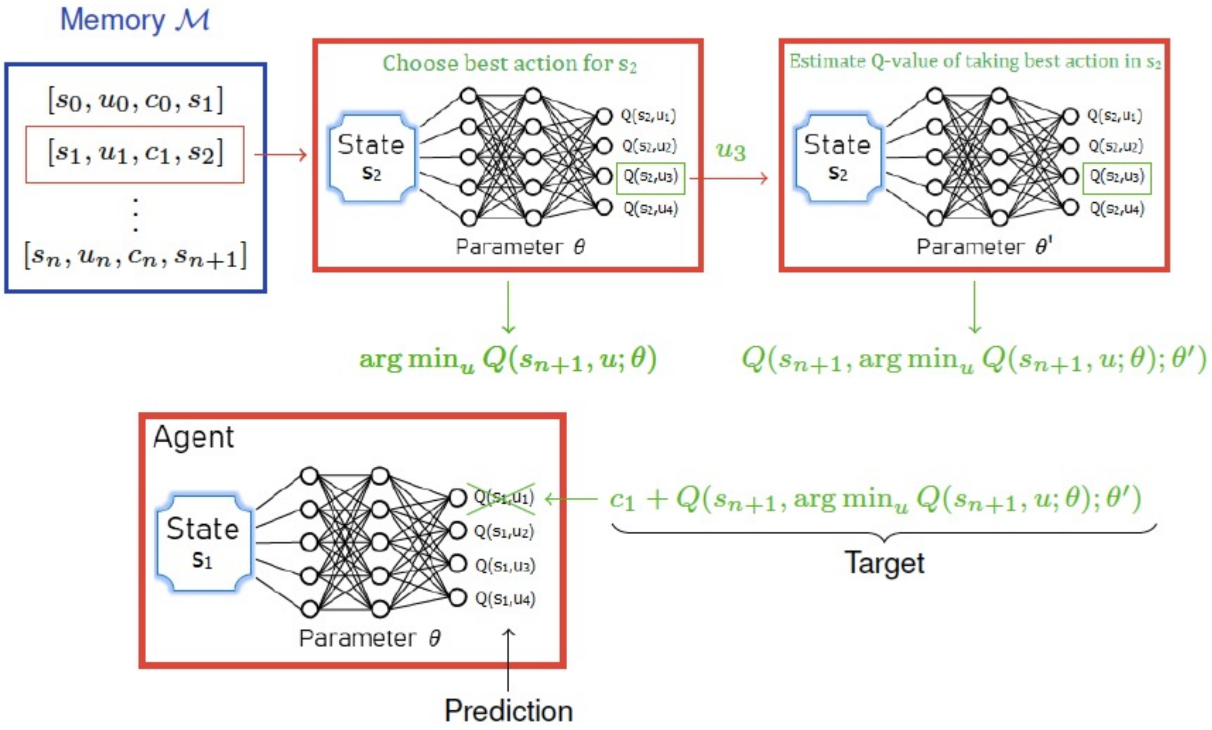


Training the Neural Network

- Learn θ by minimizing the Mean Square Error between:
  - Target : $c(s_n, u_n) + \min_u Q(s_{n+1}, u; \theta)$
  - Prediction : $Q(s_n, u_n; \theta)$
- Ensure stable learning by applying:
  - Experience Replay : Store the experience $[s_n, u_n, c(s_n, u_n), s_{n+1}]$ in replay memory $\mathcal{M}$ and train using random mini-batches from $\mathcal{M}$
  - Fixed target Network : Use a second network where its weights θ' are fixed, and only periodically or slowly updated to the primary network values for $Q(s_{n+1}, u; \theta')$
  - Double DQN : Use a second network to decouple the action selection from the target Q value generation, i.e. $Q(s_{n+1}, \mathrm{argmin}_u Q(s_{n+1}, u; \theta); \theta')$
- Ensure adequate exploration of the state space by using ϵ - greedy strategy:
  - Choose best action $u_n = \min_u Q(s_n, u; \theta)$ with probability $1 - \epsilon$
  - Select random action with probability ϵ

Step Training Example

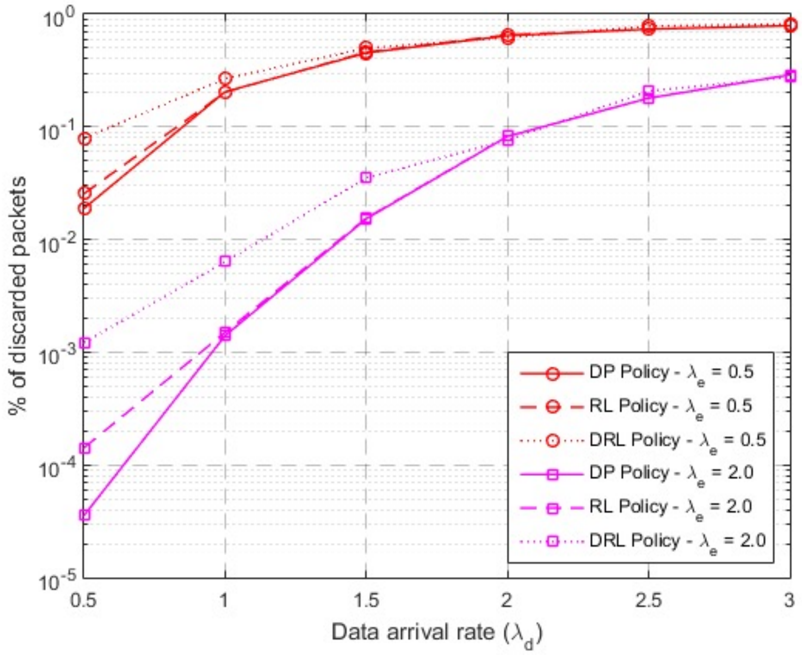
1. Interact/Explore → Replay memory $\mathcal{M}$
2. Prepare data/Train the network



Numerical Results - Discarded Packets

- RL policy is almost optimal in most of the cases

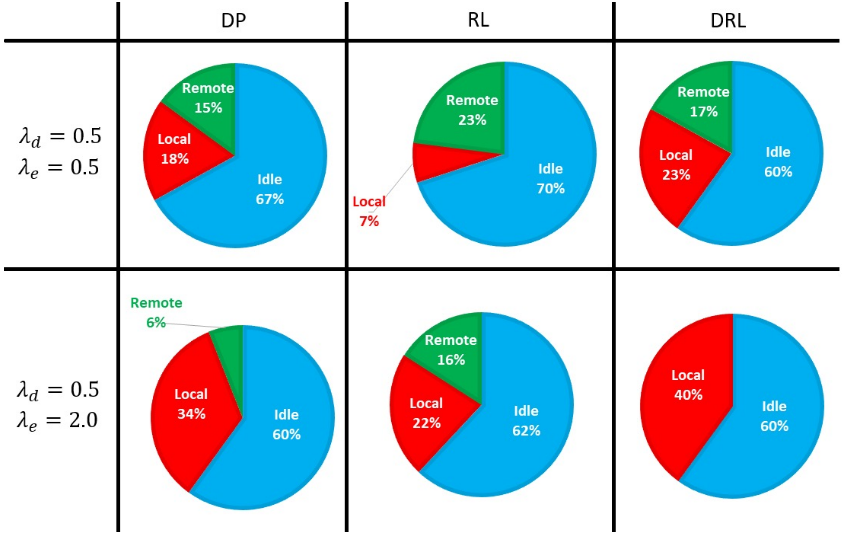
- DRL policy achieves optimal performance for high λ_e



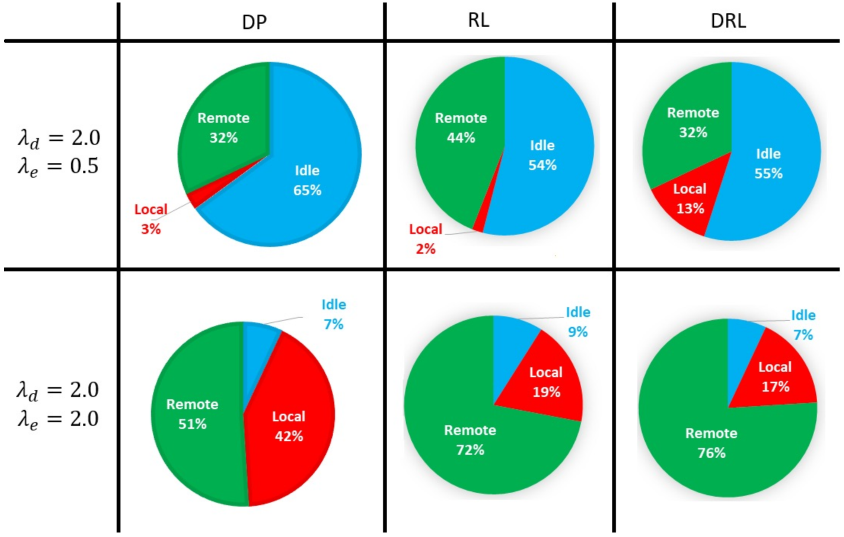
Percentage of discarded packets versus data arrival rate for different energy arrival rates

MODEL- FREE APPROACH

Numerical Results - Processing Decisions



Percentage of processing decisions



Percentage of processing decisions

- Small $\lambda_d, \lambda_e \nearrow \Rightarrow$ local mode $\nearrow$
- High $\lambda_d, \lambda_e \nearrow \Rightarrow$ remote and local modes $\nearrow$

- Investigate resource scheduling and computation offloading for EH mobile device
  - Optimal policy outperforms other policies by adapting the number of executed packets to the system states
  - DRL-based policy can be improved by improving training
    - with larger training set
    - using multistep learning algorithms

Ongoing and future work

- Consider multiple users
- Investigate Non-orthogonal multiple access NOMA
  - « Bourse d'excellence » Telecom SudParis for PhD thesis