SYSC 4906D: Artificial Intelligence in Engineering
Assignment 4
Winter 2025

Prof. Mohamed Ibn Kahla
Mohannad Abu Issa
Dept. of Systems and Computer Engineering Date: Feb 10, 2025

Mini Project 2: IoT Task Offloading in Edge Computing

❖ Objective:

- The primary goal of this mini project is to **implement the minimax theorem** in a scenario where IoT devices must decide whether to **process tasks locally** or **offload** them to an edge server. Students will learn how to model system constraints, quantify delay and energy costs, and build an **optimization** or **multi-objective** approach that leverages minimax techniques to achieve robust performance and resource utilization.

❖ Engineering Problem Context:

➢ Importance of the Engineering Problem:

- Modern Internet of Things (IoT) applications span from smart homes and healthcare monitoring to industrial automation. Many IoT devices have **limited processing power** and **finite battery resources**, making it costly or inefficient to handle computationally intensive tasks locally. By **offloading** tasks to an edge server with greater processing power, devices can reduce local energy consumption and potentially complete tasks faster. However, **communication overhead** and **network congestion** introduce additional delays and energy expenditure.

➢ Performance and Resource Utilization Optimization

- In designing IoT systems, engineers face the challenge of **balancing delay** (quality of service) and **energy consumption** (battery longevity, operational cost). The complexity arises from:
  - ♦ **Local CPU Limitations**: Devices might be too slow or energy-constrained.
  - ♦ **Communication Delays**: Offloading to an edge server incurs transmission time and additional energy usage for sending data.
  - ♦ **Edge Server Constraints**: While typically more powerful, the server's availability or network bandwidth may fluctuate, affecting overall performance.

  A **minimax** perspective helps ensure robustness against worst-case conditions—whether due to unpredictable network delays or adversarial scenarios—by formulating the decision-making process to minimize the **worst** possible performance outcome.

❖ Governing Equations

➢ Local Processing

- Local Processing Time $T_i^{\text{local}} = \frac{\kappa_i}{f_i}$

  Where κ_i = required CPU cycles for task i, and f_i = local CPU speed (cycles/second).

- Local Energy Consumption $E_i^{\text{local}} = \eta_i \times \kappa_i$

  Where η_i = energy cost coefficient (Joules per cycle).

➢ Edge Offloading

- Communication Delay $T_i^{\text{comm}} = \frac{d_i}{R_i}$

  Where d_i = data size to be sent (bits), andR_i =uplink transmission rate (bits/second).

- Edge Processing Time $T_i^{\text{edge}} = \frac{\kappa_i}{F}$

  Where κ_i = required CPU cycles, and F = edge server CPU speed (cycles/second).

- Transmission Energy $E_i^{\text{comm}} = \alpha_i \times d_i$

  Where α_i = communication energy coefficient for (Joules/bit), and d_i = data size.

➢ Constructing the Cost Function

- When deciding whether to compute locally or offload, each task i experiences a delay D_i and energy consumption E_i. Define a binary decision variable:

$$x_i = \begin{cases} 1 & \text{if task } i \text{ is offloaded} \\ 0 & \text{otherwise (local processing)} \end{cases}$$

  - Overall Delay:

$$D_i = x_i \cdot \left(T_i^{\text{comm}} + T_i^{\text{edge}}\right) + (1 - x_i) \cdot T_i^{\text{local}}$$

  - Overall Energy:

$$E_i = x_i \cdot E_i^{\text{comm}} + (1 - x_i) \cdot E_i^{\text{local}}$$

  - Total Cost for I Tasks:

$$C = \alpha \sum_{i=1}^{I} D_i + (1 - \alpha) \sum_{i=1}^{I} E_i,$$

  Where α is the weighting factor indicating how much we prioritize **delay** over **energy**.

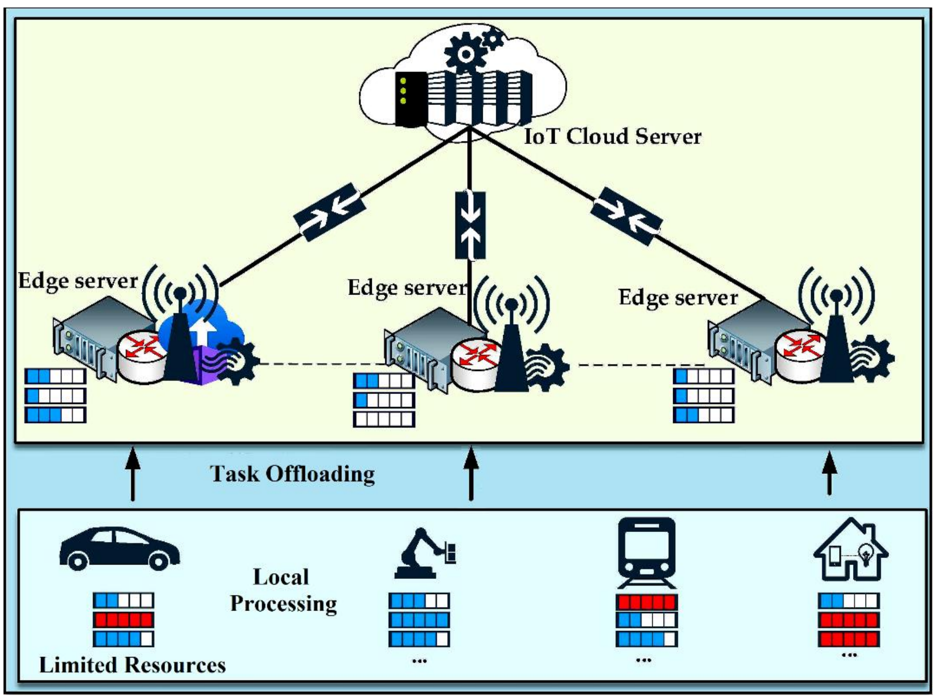
➢ In real-world IoT scenarios, the device aims to minimize both D_i and E_i subject to potential **battery constraints**, **network limitations**, or **adversarial** conditions. A **minimax formulation** ensures the chosen strategy is robust to the worst-case network or resource scenario.

❖ System Overview

➢ **IoT Devices**

- Number of Devices: N.
- Hardware Constraints: Each device has local CPU speed f_i and a maximum battery/energy budget.
- Tasks: Each device's task is characterized by κ_i (CPU cycles) and d_i (data size).

- **Edge Server**
  - Typically, more powerful CPU speed F.
  - Handles offloaded tasks at a lower per-cycle cost in terms of time and energy (from the perspective of the loT device), but with added communication overhead.



❖ Assignment Instructions

- In this assignment, we consider an IoT network over multiple time steps. An **edge manager** decides whether to process tasks on IoT devices or offload them to edge servers using **minimax** principles.
- At every time step t, a new task arrives with specific resources requirements and delays.
- At every time step t, a new task arrives with specific resource requirements and delays.
- Player 2 (Resource Manager) provides the parameters $\{T_t^{\text{local}}, E_t^{\text{local}}, T_t^{\text{comm}}, T_t^{\text{edge}}, E_t^{\text{comm}}\}$.

- Player 1 (Edge Manager) chooses local or offload for each task, aiming to minimize worst-case cost (delay + energy).

- **Time-Series Resource States example:**

Scenario	T^{local}	E^{local}	T^{comm}	T^{edge}	E^{comm}
A	3.0 s	4.0 J	1.0 s	1.0 s	1.5 J
B	1.8 s	2.0 J	1.5 s	1.0 s	2.0 J
C	2.0 s	2.5 J	2.0 s	1.0 s	1.5 J
D	2.5 s	3.0 J	1.0 s	1.0 s	2.5 J

- **Example Interpretation**:
  - **Scenario A**: Local processing is quite slow (3.0 s) and expensive (4.0 J). Offloading requires (1.0 s) of communication plus (1.0 s) edge time = (**2.0 s)** total, using (1.5 J). Likely, **offload** is better.
  - **Scenario B**: Local is faster (1.8 s) and cheaper (2.0 J), while offloading now takes (2.5 s) total and (2.0 J). Likely, **local** is better.

- For each round $t \in \{1, \dots, 120\}$, Player 2 presents a set of parameters:
  - $T_t^{\text{local}}, E_t^{\text{local:}}$ Time and energy for local processing at step t.
  - $T_t^{\text{comm}}, T_t^{\text{edge}}, E_t^{\text{comm}}$: Time and energy for offloading at step t.

- Assignment Constraint Rule:
  - **Over any 4 consecutive steps**:
    - No more than 2 local decisions (IoT devices are energy-limited and require time to harvest energy, making continuous local processing infeasible.)
    - No more than 3 offload decisions (Excessive offloading can congest the edge network and incur additional financial costs)

- Assignment Tasks:
  - **Implement** a **Minimax** approach to plan over 120 tasks.
  - **Apply** the 4-step rule to ensure feasible sequences of local/offload.
  - **Test** your code under **three weighting scenarios** for α:
    1. $\alpha = 0.5$
    2. $\alpha = 0.1$
    3. $\alpha = 0.9$

❖ Deliverables After Completing the Assignment:

- **Complete Code**: Complete the Minimax Player class.
- **Program Output**: Provide the output generated after running the program, including the plots of energy, delay, and total cost for the three scenarios.
- **Discussion:** Explain how **delay vs. energy weighting** affects the decisions and final outcomes.
- **Submission Format**: Include both the code and the output in a single PDF document. Ensure the code is presented in a format that allows it to be copied directly from the PDF. You can achieve this by embedding the code in a text box in Word before converting it to PDF.