PhD Open Days

Adaptive Network Edge Orchestrator for Dynamic Environments

PHD PROGRAMME IN ELECTRICAL AND COMPUTER ENGINEERING (PDEEC)

Eliana Neuza Silva (elianaenuza@tecnico.ulisboa.pt)

1. Introduction & Motivation

Edge Computing (EC)
– Moves the Cloud's resources and processing closer to devices.
– Decreases Latency, Delay, and supports high Mobility.
→ Propels the future of Vehicular Networks, Internet of Things (IoT), Smart Cities, Advanced Healthcare, and many more.

Vehicular Edge Computing (VEC) Offloading Management
Smart and self-driving vehicles run delay sensitive applications (APPs), stemming from operational and safety demands, that should, be computed at the network Edge, closer to the vehicles, to reduce latency.
→ Because of the high demand of requests/replies at the Edge, and vehicle mobility it is essential to have a solution to manage vehicular APPs' offloads to the network, a mechanism called orchestration.

Edge Offloading Orchestration attempts to solve:
- Best Offloading Location
- Routing Optimization
- Edge Resource Allocation
- Load Balancing of Network Nodes

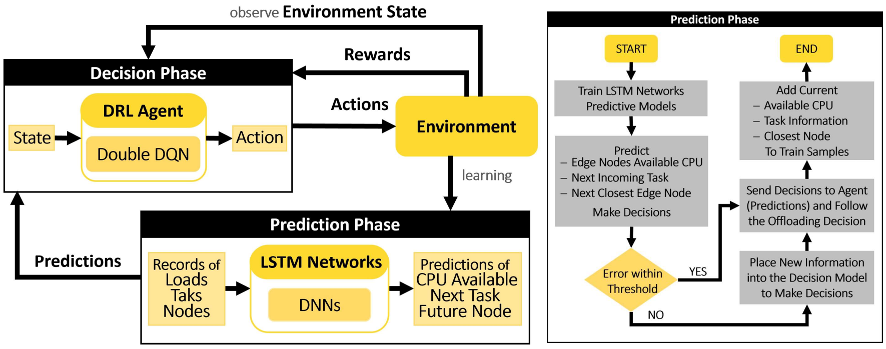
Our DRL algorithm outperforms other solutions since it:
- Removes from the offloading decision nodes that won't have enough CPU and that don't match the predicted movement.
- Predicts next possible task using task dependency.
- Decreases task failure and processing latency.

2. Deep Reinforcement Learning (DRL) Offloading Orchestrator

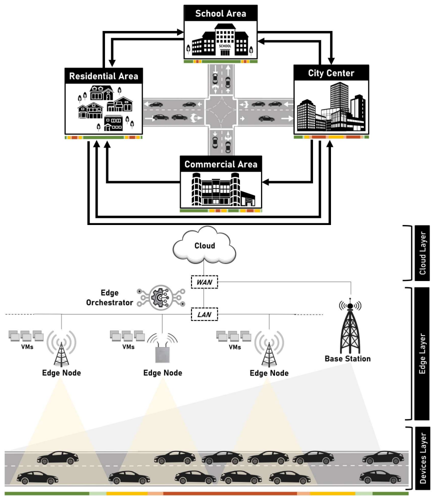
Prediction Module
→ **Long Short Memory** (LSTM) + **Deep Neural Networks** (DNN)
→ Predicts: Next Tasks, Future Edge Node Loads and Next Vehicle Location.

Decision Module
→ **DRL Agent** + **Double Deep Q Network** (Double DQN)
→ Determines the Best Offloading Locations for the APPs by:
  – Performing Automatic Intersection of Several Input Variables — *APP's Requirements*, *DRL Rewards*, *Environment State*, *Predictions (Prediction Module)*
  – Using Environment Historical Records to Predict DRL Rewards.

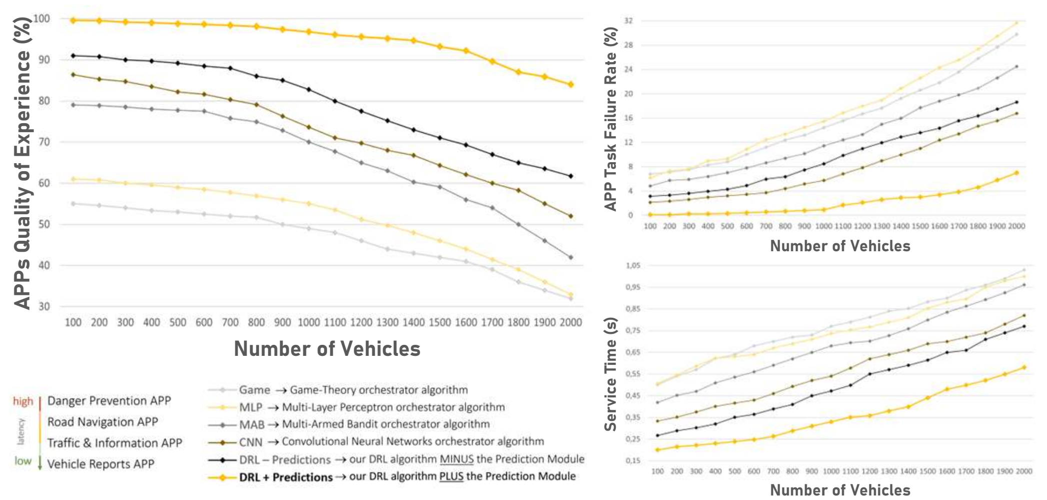
3. Adaptive Network Edge Orchestrator Architecture



4. Simulation & Tests Scenario



5. Tests Results on Performance: APP QoE, Task Failure & Service Time



6. Conclusions & Future Work

→ The Prediction Module (using LSTM with DNNs) significantly improves the training speed and accuracy of the model, increasing performance.
→ Achieves quicker computation by reducing offloading locations and decision time.
→ Makes good offloading decisions using the Double Q Network automatic intersection of several input features (Predictions, Environment, Rewards, APPs Requirements.

Next Steps: Develop a task migration mechanism, to migrate tasks partially/fully between edge or cloud, taking into mind the application's task dependency.

TÉCNICO LISBOA

Professor Fernando Mira da Silva

PhD in Electrical and Computer Engineering

phdopendays.tecnico.ulisboa.pt