VECMAN: A Framework for Energy-Aware Resource Management in VEC Systems

{Tayebeh Bahreini, Marco Brocanelli, and Daniel Grosu } Wayne State University

Introduction

Vehicular Edge Computing (VEC): Process massive amounts of workload of connected electric vehicles at the edge of the network.

Computational nodes in VEC:

- Static: Cell tower, RSUs.
- Dynamic: Vehicles.

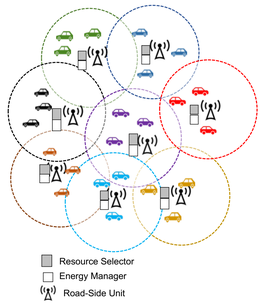
Challenges:

- Static nodes: scalability.
- Dynamic nodes: limited energy budget, unreliable connectivity, unfair workload distribution.

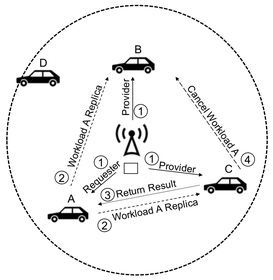
Our method:

- Improve scalability: rely on the computational capacity of vehicles.
- Improve reliability: speculative execution.
- Improve fairness: apply a selection algorithm.
- Manage energy budget: exploit the discrete power settings of CPU/GPU.
  - CPUs have a fixed number of selectable configurations for voltage-frequency levels.
  - Selected default configuration may not be fully utilized by the local work load.

System model



VEC Network model.



Resource management in the coverage area.

Resource selector (RSP):

- Determine the length of resource selection period (T).
- Determine the subset of vehicles V that with a risk factor less than α stay in the coverage area for T units of time.
- Objective: maximize the total amount of computing resources for T units of time.

Energy manager (ERMP):

- Determine the vehicles' state (i.e., requester or provider).
- Determine the number of replicas for each requester's workload.
- Allocate replica on provider vehicles.
- Objective: minimize the maximum energy balance over all vehicles.

VECMAN Framework

VECMAN Framework:

1. Run I-Selector to determine the set of participating vehicles as well as the duration of resource selection period.
2. Within resource selection period, run the energy manager, G-ERMP, periodically.

I-Selector:

1. Start with a small value of T.
2. Increase this value iteratively until obtain a length of period that maximizes the amount of available computing resources.

G-ERMP:

1. **Phase 1:** Pick a random sample of scenarios generated based on the probable locations of vehicles and solve the deterministic version of ERMP (GD-ERMP) for each scenario.
2. **Phase 2:** Based on the assignment obtained for each scenario, determine the number of replicas for each vehicle as well as the replica assignment.

G-ERMP Algorithm

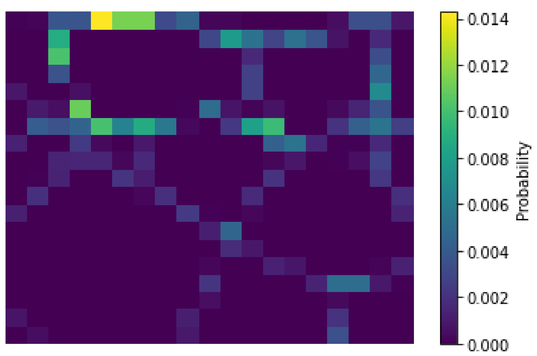
Phase 1: GD-ERMP

1. Pick a vehicle with the minimum energy balance as the first provider.
2. For each vehicle that is not selected as a provider, find the nearest provider that has enough capacity.
3. If the current set of providers is not enough to satisfy all the requests, add the next provider and continue the procedure.

Phase 2: G-ERMP

1. Create a graph based on the allocation obtained by the deterministic algorithm.
   - Vertex: a vehicle
   - Edge: requester i to provider j assignment, weighted by the number of scenarios in which a request of i has been allocated to provider j.
2. Partition the graph into the set of providers and the set of requesters.
3. Determine the number of replicas for each requester.

Experimental setup



Distribution of vehicles over the most congested area (2 km by 2 km).

1. Fairness metric: Coefficient of Variation (CV) over the energy balance,

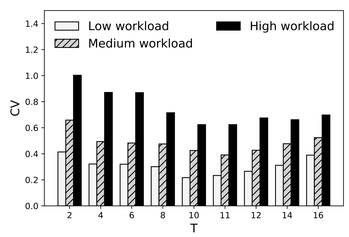
$$CV = \frac{\sqrt{\frac{1}{V}\sum_{i=1}^{V}(E_i^{blnc} - \bar{E})^2}}{\bar{E}}$$

   - A lower value of CV means a more fair distribution of requests.

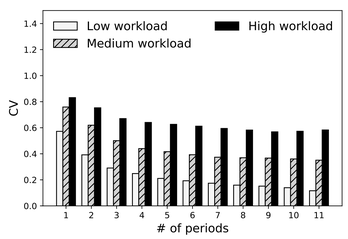
2. Performance metric: the percentage of the total energy saved,

$$\text{ES}(\%) = 100 \cdot \frac{\sum_{i \in \mathcal{V} \setminus P}(\vartheta_i \cdot f_i + \theta_i) \cdot n_i}{\sum_{i \in \mathcal{V}}(\vartheta_i \cdot f_i + \theta_i) \cdot n_i}$$

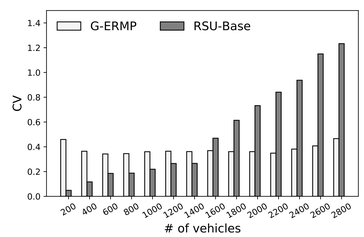
Experimental results



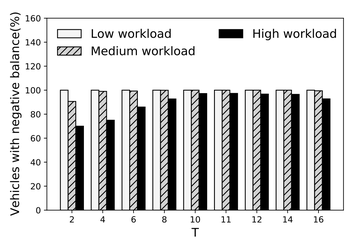
CV vs. length of resource selection period



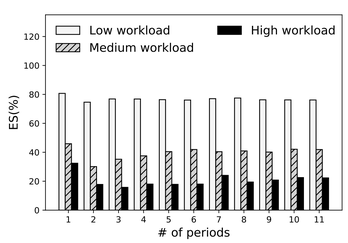
CV over periods



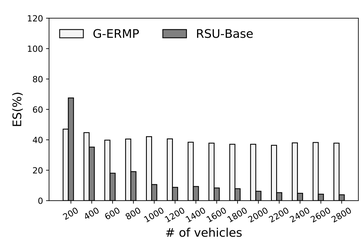
CV vs. number of vehicles



Vehicles with negative energy balance vs. length of resource selection period



Energy saving over periods



Energy saving vs. number of vehicles