UNIVERSIDAD
POLITÉCNICA
DE MADRID

Don't Let Me Down! Offloading Robot VFs Up to the Cloud

June 20, 2023

K. Gillani (UC3M)
J. Martín-Pérez (UPM)
M. Groshev (UC3M)
A. de la Oliva (UC3M)
R. Gazda (ID)

Overview

UNIVERSIDAD POLITÉCNICA DE MADRID

Networked robotics interact with servers for, e.g.:

- remote driving; or
- assembly robots.

A remote driving Service S has VFs:

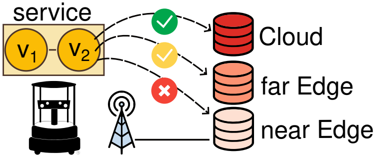
- the robot drivers v_1; and
- the remote controller v_2.

The literature tends to offload – e.g. the remote control v_2 – to Edge, not to Cloud.

The literature tends to offload – e.g. the remote control v_2 – to Edge, not to Cloud.

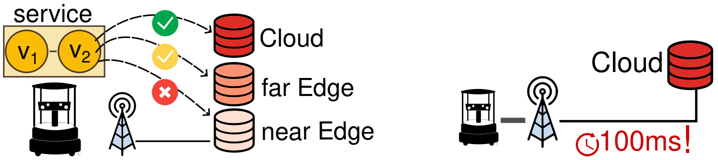
But channel impairments are still there!
Communication between VFs (v_1, v_2) is harnessed.

Introduction: Don't Let Me Down!



(a)

Introduction: Don't Let Me Down!



(a) (b)

UNIVERSIDAD POLITÉCNICA DE MADRID

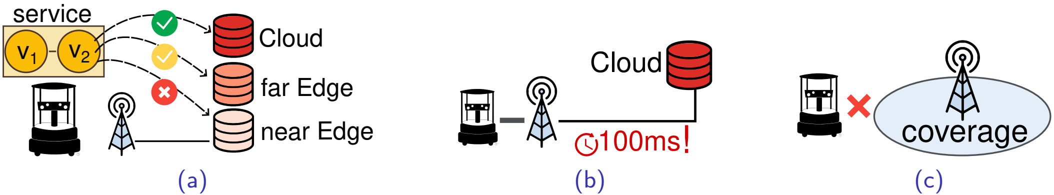


Figure 1: Don't Let Me Down! fosters offloading the robot service VFs up to the cloud (a), yet preventing the cloud large latencies (b) and running out of coverage (c).

Hardware graph G with:

- robots $r \in V(G)$;
- radio PoA $R \in V(G)$;
- switches/routers $w \in V(G)$; and
- server nodes $n \in V(G)$

links are edges:

- wireless links $(r, R) \in E(G)$; and
- wired links $(n_1, n_2) \in E(G)$.

Robotic service graph S:

- VFs $v \in V(S)$; and
- VLs $(v_1, v_2) \in V(S)$.

The service S asks for

- a maximum delay $D(S)$;
- computational resources $C(v)$, $\forall v \in S$; and
- bandwidth resources $\lambda(v_1, v_2)$, $\forall (v_1, v_2) \in E(S)$.

UNIVERSIDAD POLITÉCNICA DE MADRID

The decisions to take are

- assigning VFs to servers $a(n) \subset S$; and
- assigning VLs to links $a(n_1, n_2) \subset E(S)$.

to minimize the deployment cost

$$\min_{a(n)} \sum_{n \in V(G)} \kappa_n |a(n)| \tag{1}$$

with $\kappa_n \in \mathbb{R}^+$ the server n monetary cost, typically satisfying

$$\kappa_{\text{Cloud}} < \kappa_{\text{Edge}} \tag{2}$$

Decide adequate PoA association/handovers $a(r, R) \in E(S)$ as the robot moves.

$$\sum_{(v_1,v_2)\in a(r_i,R_i)} \lambda(v_1, v_2) \leq \mathrm{T}(r_i, R_i), \quad \forall (v_1, v_2) \in a(r_i, R_i) \tag{3}$$

so wireless capacity is enough

$$\mathrm{T}(r_i, R_i) = (1 - \delta(r_i, R_i))\lambda(r_i, R_i) \log_2 \left(1 + \frac{\sigma_{R_i}(r_i)}{N}\right) \tag{4}$$

with $\delta(r_i, R_i)$ the PER and $\sigma_{R_i}(r_i)$ the signal strength.

UNIVERSIDAD POLITÉCNICA DE MADRID

A shortest-path based algorithm whose core idea is

Don't Let the VFs go Down to the Edge!

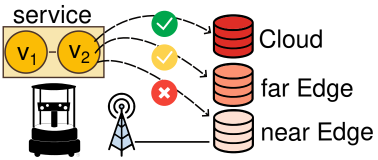


Figure 2: Don't Let Me Down! sends VFs Up to the Cloud.

Don't Let Me Down! is based on the τ metric

$$\begin{aligned} \tau : \quad \mathcal{P}(V(G)) &\longrightarrow \quad \mathbb{R}^+ & (5) \\ \mathcal{P}_0 \quad &\longmapsto \quad \kappa_{\mathcal{P}_0[-1]} + \sum_{(n_1,n_2)\in\mathcal{P}_0} \left(\frac{1}{\lambda(n_1,n_2)} + d(n_1,n_2) \right) & (6) \end{aligned}$$

that fosters the use of

- cheap servers as the Cloud;
- non-congested links; and
- fast links.

Don't Let Me Down! is summarized in three steps:

1. *prune PoAs* without enough capacity $R \notin \{\hat{R}_i\}$

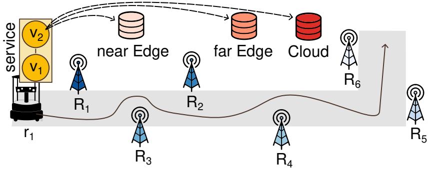
$$\{\hat{R}_i\} = \{R_i : \lambda(v_1, v_2) \leq T(r_i, R_i)\}_i \tag{7}$$

2. find server to *offload VFs* using

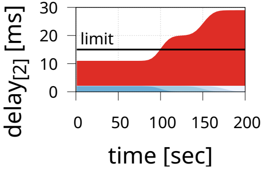
$$\text{Dijkstra}(\textit{source} = r, \text{ weight} = \tau) \tag{8}$$

3. *select PoA* with best free/delay ratio

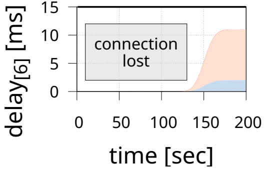
$$\arg\min_{\hat{R}_i} \left\{ \frac{1}{\lambda(r_i, \hat{R}_i)} + d(r_i, \hat{R}_i) \right\} \tag{9}$$



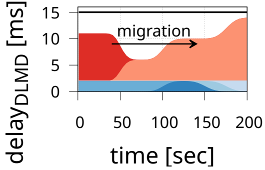
(a)



(b)

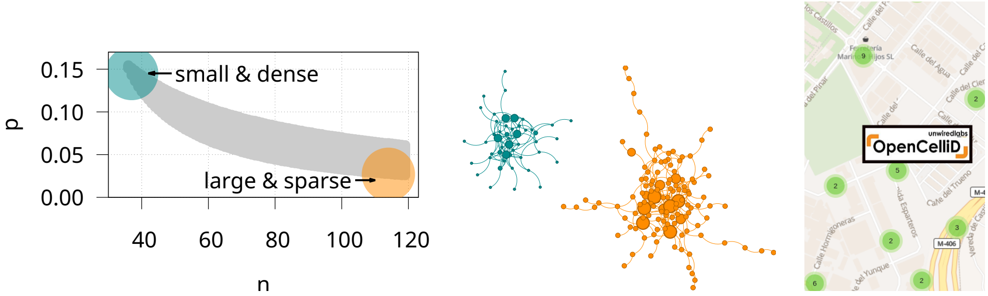


(c)



(d)

Figure 3: Delay experienced by the robot using [1], [2] and Don't Let Me Down (DLMD).



(a) (b) (c)

Figure 4: Erdős–Rényi setups (a) network graphs (b) with PoAs of an industrial area (c).

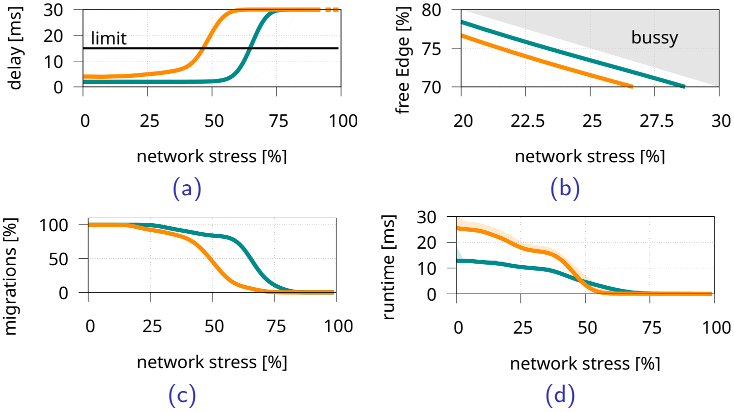


Figure 5: DLMD stress test in Fig. 4 small (green) and large (orange) networks.

Conclusions & Future Work

Don't Let Me Down!

- takes ~ 10 [ms];
- reaches optimality in small scenarios (backup);
- offloads & handovers to meet < 15 [ms]; and
- minimizes Edge consumption.

Still to

- study w/ weighted $\frac{1}{\lambda} + d$ metric for τ;
- include SINR+propagation models; and
- study probabilistic mobility models.

C. Delgado, L. Zanzi, X. Li, and X. Costa-Pérez, "OROS: Orchestrating ROS-driven Collaborative Connected Robots in Mission-Critical Operations," *IEEE*, 2022.

B. Németh, N. Molner, J. Martín-Pérez, C. J. Bernardos, A. de la Oliva, and B. Sonkoly, "Delay and Reliability-Constrained VNF Placement on Mobile and Volatile 5G Infrastructure," *IEEE Transactions on Mobile Computing*, vol. 21, no. 9, 2022.

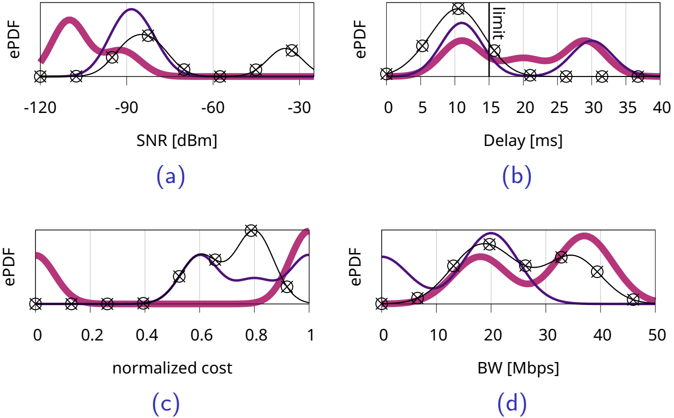


Figure 6: Metrics' ePDFs using DLMD (circle), optimal (cross), [1] (thickest), and [2] (thick)