Radio and Computation Resource Management in Unmanned Vehicles with Edge Computing

Giuseppe Baruffa, Luca Rugini, Fabrizio Frescura, Paolo Banelli
Dept. of Engineering, University of Perugia, Perugia, Italy
{giuseppe.baruffa, luca.rugini, fabrizio.frescura, paolo.banelli}@unipg.it

***Abstract*—This work deals with the resource management for fleets of unmanned vehicles (UV), both ground-based and aerial, which offload computation tasks to remote services. The UVs are equipped with onboard sensors (camera, etc.), have scarce computation resources, and exploit multiple networks with different radio-access technologies, for requesting additional computation resources during their mission. The UVs are so entitled to offload intensive CPU tasks, such as object detection, to computational nodes located in the edge/cloud. The aim is to optimize the average latency, taking into account also throughput and handover rate. We consider that UVs have to manage their resources without cooperation or the help of a control server. To this end, we propose and compare decentralized algorithms, also based on reinforcement learning techniques.**

***Index Terms*—Edge computing, computational offloading, multi-armed bandit, unmanned vehicles.**

I. Introduction

According to [1], key performance indicators for efficient offloading in edge computing include latency, data rate, energy consumption, seamless access, and network connectivity, which are typically traded off with throughput, service blockage, handover, and computation accuracy. The benefits of multi-radio access technology (RAT) solutions are outlined in [2]. Indeed, when a specific RAT does not provide the desired performance, there may be another RAT that offers better traffic conditions [3]. The use case investigated in this work entails a fleet of unmanned vehicles (UV) that can be coordinated to exploit services in external computational nodes (CN). Each UV is connected to the network via wireless links to access points (AP) or base stations (BS), possibly multi-RAT. Each AP is connected to one or more CNs, via high-speed wired backhauls. Depending on network topology, the link latency time may be low for edge CNs, and high for cloud CNs. We consider a scenario that is populated by a fleet of aerial (i.e., drones) or ground UVs, which fulfill a delivery or exploratory mission by visiting a number of waypoints [4]. This scenario includes obstacles, which have to be recognized and avoided using onboard sensors and (possibly offloaded) object detection services. This work introduces novel algorithms, which target the optimization of radio and computational resources usage, also based on reinforcement learning [5] via multi-armed bandit techniques [6].

This work was supported by the European Union - Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUP E83C22004640001, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART").

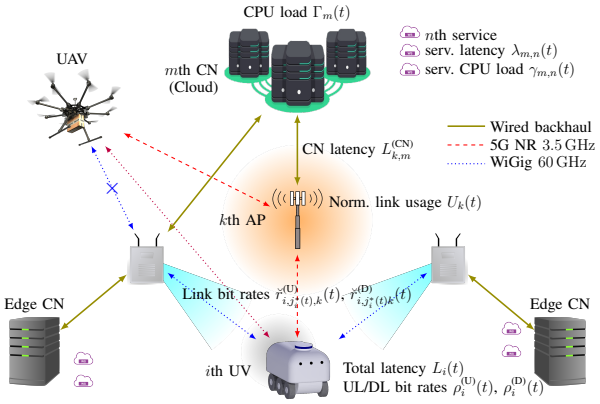


Fig. 1. Simulated scenario: ground and aerial UVs, multi-RAT APs, edge and cloud CNs, remote services.

II. System Model

With reference to Fig. 1, N_{UV}, N_{AP}, and N_{CN} are, respectively, the number of UVs, APs, and CNs. Differently from [6], our model uses more realistic path losses, a different duplex technique, and standard-based radio parameters.

A. UV Motion Model

The mission of the ith UV consists of $N_i^{(\text{W})}$ waypoints with 3D coordinates $\mathbf{w}_{i,l}$: specifically, at time t, the UV is located at coordinates $\mathbf{u}_i(t)$, and is moving between the waypoints with indexes $l = b_i(t)$ (base waypoint) and $l+1$ (target waypoint), at a velocity $v_i(t)$ along the direction $\boldsymbol{\nu}_i(t)$, with $\|\boldsymbol{\nu}_i(t)\|_2^2 = 1$, as shown in Fig. 2. We assume that the UVs move from one waypoint to the next, exploiting some feedback information from the network CNs, such as, for instance, obstacle detection. Considering uniform linear motion with time step Δ_{t}, the UV position at time $t+\Delta_{\text{t}}$ is obtained as

$$\mathbf{u}_i(t+\Delta_{\text{t}}) = \mathbf{u}_i(t) + \Delta_{\text{t}} v_i(t) \boldsymbol{\nu}_i(t). \tag{1}$$

The starting point of the ith UV is $\mathbf{u}_i(0) = \mathbf{w}_{i,0}$. The UV velocity depends on a "throttle" parameter $\tau_{\text{S}}(t) \geq 0$, as

$$v_i(t) = \max\left(\hat{v}_i, \check{v}_i \min\left(1, \tau_{\text{S}}(t)\right)\right), \tag{2}$$

where $\hat{v}_i$ and $\check{v}_i$ are the minimum and maximum speeds, respectively. For instance, when feedback from a remote service returns quickly, the UV speed will increase up to $\check{v}_i$. Differently, if responses are exceedingly delayed, the UV will be driven slowly down to $\hat{v}_i$. A UV mission is completed at

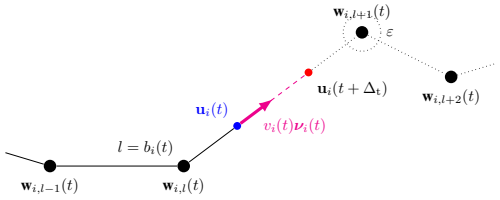


Fig. 2. Motion of UVs along the mission waypoints.

time T_i when the final waypoint is reached within a desired positioning accuracy, that is, when $\|\mathbf{u}_i(T_i) - \mathbf{w}_{i,N_i^{(\mathrm{W})}}\| < \varepsilon$.

B. Radio Communication Model

We assume time-division duplex for all RATs, so that uplink (UL) and downlink (DL) share the same frequency band, and that each UV exclusively uses a single radio interface per time instant. The kth AP is located at coordinates $\mathbf{a}_k$ and receives from the jth radio interface of the ith UV an UL power

$$P_{(i,j)\to k}(t) = \frac{P_{i,j} G_{i,j}^{(\mathrm{T})}(\mathbf{a}_k; t) G_k^{(\mathrm{R})}(\mathbf{u}_i(t))}{Z_k(\mathbf{u}_i(t))}, \tag{3}$$

where $P_{i,j}$ is the transmitted UL power, $G_{i,j}^{(\mathrm{T})}(\mathbf{x}; t)$ is the UV transmission gain in the direction of $\mathbf{x}$, $G_k^{(\mathrm{R})}(\mathbf{x})$ is the AP reception gain from the direction of $\mathbf{x}$, and $Z_k(\mathbf{x})$ is the AP propagation loss toward the point $\mathbf{x}$. The two gains include the radiation pattern of the antennas, while the propagation loss depends on line-of-sight (LOS) or non-LOS (NLOS) visibility condition. A dual expression, $P_{(i,j)\leftarrow k}$, stands for the DL.

Both UL and DL bit rates depend on the received signal strength $P_{\leftrightarrow}$ through the RAT modulation and coding set (MCS) sensitivity $S_{\mathrm{rat},\mu}$ and bit rate $R_{\mathrm{rat},\mu}$ tables [7], where μ is the MCS index. The optimal MCS index (selected from the set $\mathcal{M}$) is given by

$$\mu^* = \underset{\mu\,:\,P_{\leftrightarrow} - S_{\mathrm{rat},\mu} \geq 0}{\arg\max} S_{\mathrm{rat},\mu}. \tag{4}$$

The link bit rate is $\breve{r}_b = R_{\mathrm{rat},\mu^*}$, or $\breve{r}_b = 0$ if $P_{\leftrightarrow} < S_{\mathrm{rat},\mu}$, $\forall \mu$. The UV radio interface and the AP are connected if both UL $\breve{r}_{i,j,k}^{(\mathrm{U})}$ and DL $\breve{r}_{i,j,k}^{(\mathrm{D})}$ bit rates are not null, with a connection status $D_{i,j,k} = 1$ if $\breve{r}_{i,j,k}^{(\mathrm{U})} \breve{r}_{i,j,k}^{(\mathrm{D})} > 0$, or $D_{i,j,k} = 0$ if $\breve{r}_{i,j,k}^{(\mathrm{U})} \breve{r}_{i,j,k}^{(\mathrm{D})} = 0$. Thus, for the kth AP, we can define the normalized channel usage, as

$$U_k(t) = \sum_{i=1}^{N_{\mathrm{UV}}} D_{i,j_i(t),k}(t) \left(\frac{\rho_i^{(\mathrm{U})}(t)}{\breve{r}_{i,j_i(t),k}^{(\mathrm{U})}(t)} + \frac{\rho_i^{(\mathrm{D})}(t)}{\breve{r}_{i,j_i(t),k}^{(\mathrm{D})}(t)} \right), \tag{5}$$

where $\rho_i^{(\mathrm{U})}(t)$ and $\rho_i^{(\mathrm{D})}(t)$ are the bit rates to be sent on the links, and $j_i(t)$ is the selected radio interface index. Note that when $U_k(t) > 1$, the channel is considered over capacity, and a proportional rate reduction is expected for all UVs.

C. Service Model

We model the backhaul links between the kth AP and the mth CN with a time-invariant connection status $K_{k,m} = 1$ and round-trip time latency $L_{k,m}^{(\mathrm{CN})}$; when a backhaul link is not present, $K_{k,m} = 0$ and $L_{k,m}^{(\mathrm{CN})} = +\infty$. Each CN has $N_m^{(\mathrm{C})}$ CPUs and $N_m^{(\mathrm{S})}$ services. Using the nth service of the mth CN gives a typical end-to-end latency $\lambda_{m,n}(t)$, which considers the input/output data processing, with a load of $\gamma_{m,n}(t)$ CPUs. The total CPU load requested on the mth CN is

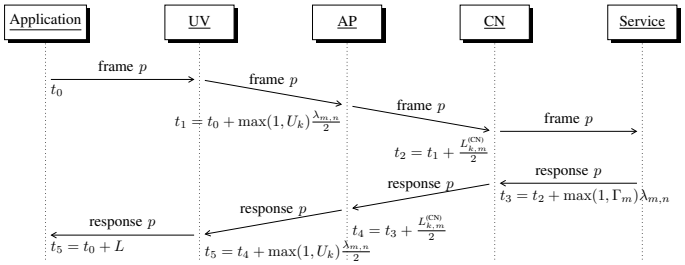


Fig. 3. Sequence diagram detailing the latency composition.

$$\Gamma_m(t) = \frac{1}{N_m^{(\mathrm{C})}} \sum_{n=1}^{N_m^{(\mathrm{S})}} z_{m,n}(t) \gamma_{m,n}(t), \tag{6}$$

where $z_{m,n}(t)$ denotes the number of UVs that are using the nth service of the mth CN. Note that when $\Gamma_m(t) > 1$, the CPUs are overloaded and a proportional decrease in offered performance is expected. We assume that each UV produces a video stream that requires a service, either internal or external. Each response of the external service experiences a latency $L_i(t)$. As illustrated in Fig. 3, the pth video frame is generated at time t_0 and sent through the UV radio interface. In the UV-AP radio link, the typical service delay $\lambda_{m,n}/2$ may increase by $U_k(t)$ times if the channel usage is $U_k(t) > 1$. This model considers TDD, a perfect RAT scheduler, and that the latency is equally split between UL and DL. In the AP-CN link, if the CPUs are overloaded, we assume that the response delay increases $\Gamma_m(t) > 1$ times. The pth response proceeds back to the application, and the experienced latency is $L_i(t) = t_5 - t_0$. Hence, the effective service latency depends on the AP link usage $U_k(t)$, on the CN CPU load $\Gamma_m(t)$, on the typical service latency $\lambda_{m,n}(t)$, and on the AP-CN link delay $L_{k,m}^{(\mathrm{CN})}$ as

$$L_i(t) = \lambda_{m_i,n_i}(t)\ \max\{1, U_{k_i}(t)\}\ \max\{1, \Gamma_{m_i}(t)\} + L_{k_i,m_i}^{(\mathrm{CN})}, \tag{7}$$

where $k_i = k_i(t)$, $m_i = m_i(t)$, and $n_i = n_i(t)$ represent the chosen AP, CN, and service indexes of the ith UV (we reserve the index 0 to represent the internal service). When the AP and CN are overloaded, the latency increases by $U_{k_i}(t) > 1$ and $\Gamma_{m_i}(t) > 1$, respectively. Differently, it remains at the typical value. The "throttle" parameter in (2) is $\tau_{\mathrm{S}}(t) = \breve{L}_i / L_i(t)$, where $\breve{L}_i$ is the maximum latency accepted by the UV.

D. Target Performance Parameters

The ith UV performance parameters are: the *handover rate*

$$H_i = \mathrm{E}_t\{(k_i(t) \neq k_i(t - \Delta_t)) \wedge (k_i(t) k_i(t - \Delta_t) \neq 0)\}, \tag{8}$$

where E_t denotes the time average; the external *service blockage rate* (the UV cannot exploit offloading)

$$B_i^{(\mathrm{S})} = \mathrm{E}_t\{n_i(t) = 0\}; \tag{9}$$

the *average latency*

$$\bar{L}_i = \mathrm{E}_{t:n_i(t)>0}\{L_i(t)\}; \tag{10}$$

and the *average UL bit rate*

$$\bar{r}_i^{(\mathrm{U})} = \mathrm{E}_{t:n_i(t)>0}\{\breve{r}_{i,j_i(t),k_i(t)}^{(\mathrm{U})}\}. \tag{11}$$

We also define the corresponding parameters averaged over the N_{UV} UVs, denoted by H, $B^{(\mathrm{S})}$, $\bar{L}$, $\bar{r}^{(\mathrm{U})}$, respectively.

III. Resource Management: Proposed Algorithms

A resource management algorithm can select in a decentralized way the AP, CN, and service configuration, to minimize the handover rate H, or the latency $\bar{L}$, or the service blockage rate $B^{(\mathrm{S})}$, or to maximize the UL bit rate $\bar{r}^{(\mathrm{U})}$. The following algorithms are proposed: heuristic allocation of resources with memory (HARM), capacious AP, quick CN, first service (CQF), minimum latency (MLAT), and multi-armed bandit reinforcement learning (MABRL). The proposed algorithms employ selfish strategies: while HARM, CQF, and MLAT require some prior knowledge, such as the latency between CNs and APs, MABRL does not require any prior knowledge, as it learns over time, during operation.

A. Heuristic Allocation of Resources with Memory (HARM)

The purpose of the HARM algorithm (Fig. 4) is to minimize configuration changes and, if necessary, to perform random assignment of valid configurations. HARM tries to request the AP, CN, and service so as to balance the loads $\Gamma_m(t)$ and $U_k(t)$ on the whole network of APs and CNs. If the current AP is not congested (it is not consuming all the channel resources, $U_k(t) \leq 1$) and if the current CN is not exhausted (the CPU is not fully loaded, $\Gamma_m(t) \leq 1$), HARM keeps the configuration $(k_i(t-\Delta_{\mathrm{t}}), m_i(t-\Delta_{\mathrm{t}}), n_i(t-\Delta_{\mathrm{t}}))$ used at time $t-\Delta_{\mathrm{t}}$. Otherwise, HARM finds a new configuration $(k_i(t), m_i(t), n_i(t))$ by randomly exploring the APs, CNs, and services that guarantee sufficient resources, e.g., free AP channel usage $U_k(t)$ and low CN loads $\Gamma_m(t)$, and chooses the one which provides the largest margin in terms of available AP UL bit rate and unassigned CN CPU resource. By keeping the old configuration $(k_i(t-\Delta_{\mathrm{t}}), m_i(t-\Delta_{\mathrm{t}}), n_i(t-\Delta_{\mathrm{t}}))$, HARM avoids unnecessary handovers (thus minimizing the handover rate H_i), and with its random configuration exploration, HARM assigns the service by distributing the load and throughput in a balanced way among CNs and APs. For real operation, HARM considers the most recent known AP channel usage, $U_k(t-\Delta_{\mathrm{t}})$, and CN load, $\Gamma_m(t-\Delta_{\mathrm{t}})$, as well as the current AP access bit rate, $\breve{\rho}_{i,j,k}^{(\mathrm{U})}(t)$.

B. Capacious AP, Quick CN, First Service (CQF)

The CQF algorithm (Fig. 5) tries to maximize the average access bit rate $\mathrm{E}_t\{\breve{\rho}_{i,j,k}^{(\mathrm{U})}(t)\}$ and, as a secondary goal, to minimize the experienced latency $\bar{L}_i$. CQF tries to request an AP, CN, and service independently of the previous configuration $(k_i(t-\Delta_{\mathrm{t}}), m_i(t-\Delta_{\mathrm{t}}), n_i(t-\Delta_{\mathrm{t}}))$: first, CQF chooses, among the associated APs, the one providing the largest access bit rate, $\max_{j,k:D_{i,j,k}=1} \breve{\rho}_{i,j,k}^{(\mathrm{U})}(t)$; then, CQF chooses, among all

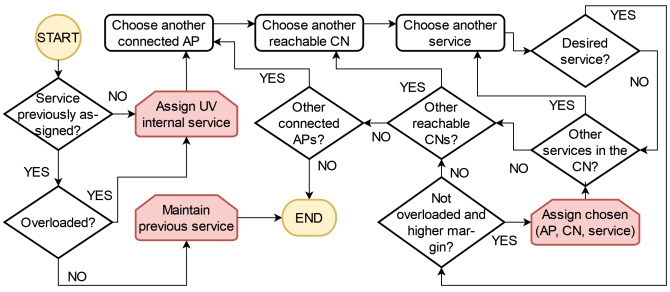


Fig. 4. The HARM algorithm.

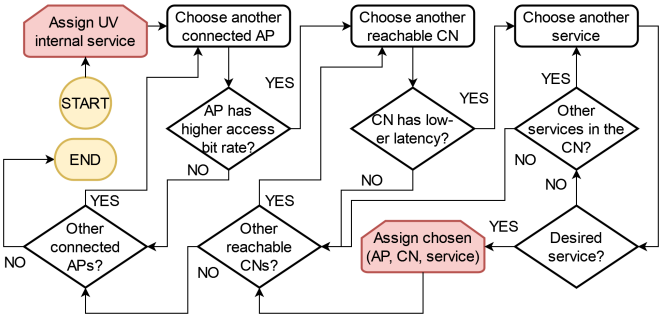


Fig. 5. The CQF algorithm.

the CNs connected to the best AP, the one with the minimum latency, $\min_{m:K_{k,m}=1} L_{k,m}^{(\mathrm{CN})}$. Finally, CQF chooses the desired service in the selected best CN. The CQF algorithm is greedy, as it can overload the CNs and the APs, and it also requires to know the latency $L_{k,m}^{(\mathrm{CN})}$ between the APs and the CNs.

C. Minimum Latency (MLAT)

Differently from CQF, the MLAT algorithm (Fig. 6) tries to minimize the latency $\bar{L}_i$ only. MLAT attempts to request an AP, CN, and service independently of the previous configuration $(k_i(t-\Delta_{\mathrm{t}}), m_i(t-\Delta_{\mathrm{t}}), n_i(t-\Delta_{\mathrm{t}}))$, by trying a connection with every AP. Then, MLAT chooses, among all CNs connected to the AP, the one with the minimum latency, $\min_{m:K_{k,m}=1} L_{k,m}^{(\mathrm{CN})}$. Finally, MLAT chooses the desired service in the selected best CN. Also the MLAT algorithm is resource greedy, and it requires the knowledge of the latency $L_{k,m}^{(\mathrm{CN})}$ between the APs and the CNs.

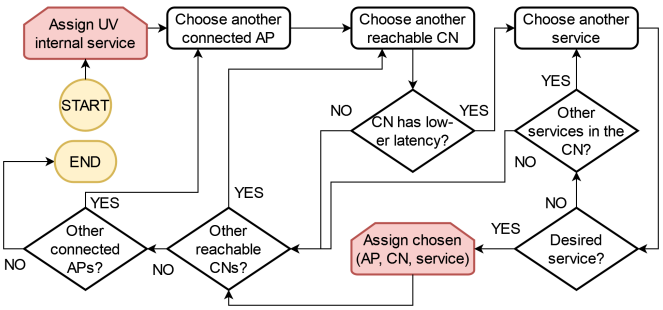


Fig. 6. The MLAT algorithm.

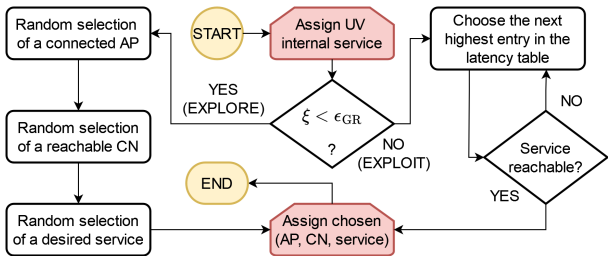


Fig. 7. The MABRL algorithm.

D. Multi-Armed Bandit Reinforcement Learning (MABRL)

In contrast to the previous algorithms, MABRL (Fig. 7) uses reinforcement learning techniques, implemented with a multi-armed bandit heuristic algorithm [6], [8]. In general, MABRL tries to minimize the average latency $\bar{L}_i$, by exploiting a table containing the average latency (experienced so far) associated to a specific AP, CN, and service configuration. MABRL chooses between *exploration* (random configuration of AP, CN, and service) and *exploitation* (current best configuration of AP, CN, and service in the table, based on the minimum experienced latency). The bandit reward function is defined as

$$Q_{i,k,m,n}(t) = \frac{1}{\mathrm{E}_{t' \in \mathcal{T}_{i,k,m,n}(t)}\{L_i(t')\}}, \tag{12}$$

where $\mathcal{T}_{i,k,m,n}(t) = \{t' : t - W \leq t' < t_{\mathrm{tr}}, k_i(t') = k, m_i(t') = m, n_i(t') = n\}$ is the set of time instants when the triple (k, m, n) is selected, $W > 0$ is the time window duration where the stored latencies are searched, and $t_{\mathrm{tr}} \geq 0$ is the training time duration.

For the bandit algorithm, we use the ε-greedy strategy: given a uniformly distributed random variable $\xi \sim \mathcal{U}(0,1)$, MABRL decides between exploration and exploitation based on the *greediness* parameter $\varepsilon_{\mathrm{gr}} \in [0,1]$. When $\xi \leq \varepsilon_{\mathrm{gr}}$, exploration takes place, and the assigned triple is $(k_i, m_i, n_i)(t) = (k', m', n')$, where k' is chosen randomly among the AP connected to the ith UV at time t, m' is chosen randomly among the CNs that are connected to the AP k', and n' is chosen randomly among the services of CN m'. When $\xi > \varepsilon_{\mathrm{gr}}$, exploitation is performed, and the assigned triple is

$$(k_i, m_i, n_i)(t) = \arg\max_{k,m,n} Q_{i,k,m,n}(t). \tag{13}$$

Using an exhaustive search in the reward table, the complexity of MABRL is upper bounded by $O(N_{\mathrm{CF}})$, where the number of configurations $N_{\mathrm{CF}} = N_{\mathrm{RAT}} N_{\mathrm{APRAT}} N_{\mathrm{CN}} N_{\mathrm{CNSVC}}$ considers N_{RAT} RATs used, N_{APRAT} APs per RAT, N_{CN} CNs, and N_{CNSVC} services per CN. However, location-based prefiltering of table entries reduces N_{CF}, and table ordering with cost $O(N_{\mathrm{CF}} \log N_{\mathrm{CF}})$ can reduce the search time.

IV. Simulated Performance Results

For the simulations, we assume that each UV is equipped with a front-facing depth camera and has simple onboard hit detection. More complex tasks, such as object detection, are offloaded to the CNs, or internally processed in case of service blockage. Common simulation parameters are reported in Tab. I. The typical latency $\lambda_{m,n}$ of each service is fixed, while the requested CPU load is a bounded normal random variable, $\gamma_{m,n} \sim \mathcal{BN}(\bar{\gamma}, \sigma_\gamma^2, \gamma_{\mathrm{MIN}}, \gamma_{\mathrm{MAX}})$. The bit rate of the UL requests is variable, depending on the video resolution (fixed), frame rate (fixed), and compression ratio (random variable). The services are deployed to all CNs.

TABLE I
Common Simulation Parameters

Parameter	5G NR value	WiGig value
Carrier frequency (GHz)	3.5	60
Bandwidth (MHz)	20	2100
AP (UV) TX power (dBm)	46 (23)	27 (27)
AP (UV) antenna gain (dB)	18 (6)	13 (13)
AP/UV vertical beam ($\lvert\alpha\rvert \leq 90°$)	$\cos^2 \alpha$	$\cos^2 \alpha$
AP horizontal beamwidth (°)	360	180
UV horizontal beamwidth (°)	360	60
UV UL bit rate $\rho_i^{(\mathrm{U})}$	25 ± 7 Mbit/s	
AP/edge CN latency $L^{(\mathrm{CN})}$	5 ms - 6 ms	
AP/cloud CN latency $L^{(\mathrm{CN})}$	50 ms	
Typical service latency L	33.3 ms	
Service CPU load γ	$1000 \pm 50 \in$ [800,1200] mcpu	

TABLE II
Results for *simple* Scenario

Parameter	HARM	CQF	MLAT	MABRL
Serv. blockage rate $B^{(\mathrm{S})}$ (%)	7.0	3.2	3.2	4.4
UL bit rate $\bar{r}^{(\mathrm{U})}$ (Mbit/s)	1421.8	**1951.3**	1767.7	1697.4
Handover rate H (%)	**0.3**	13.8	27.3	7.2
Latency $\bar{L}$ (ms)	89.7	**68.9**	**68.7**	81.6

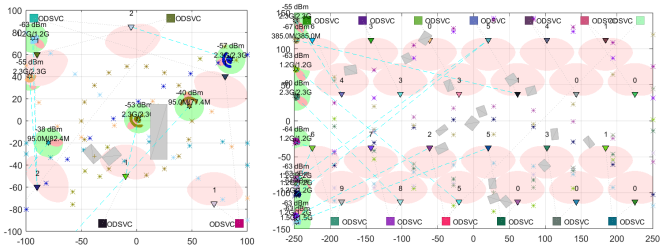
The RATs adopted in this context are the 60 GHz IEEE 802.11ad (WiGig) and the 3.5 GHz 3GPP 5G New Radio (NR) standards. WiGig links are simulated by an attenuation with either CI [9] or 5GCM UMi Open Square LOS [10] models, and 40 dB additional loss in case of NLOS. 5G NR links, instead, are simulated by an attenuation with either UMa LOS or UMa NLOS models [10]. For both RATs, we do not account for path loss variations due to shadowing effects.

We consider two different scenarios, named *simple* and *high_density*, and each UV hosts the two RATs interfaces. Moreover, for MABRL, we consider $\varepsilon_{\mathrm{gr}} = 0.07$, $W = +\infty$, and $t_{\mathrm{tr}} = +\infty$. Each scenario is simulated three times with different randomness seeds and $\Delta_{\mathrm{t}} = 0.25$ s, and the average values of the target parameters are taken.

A. Results for the simple Scenario

This scenario (Fig. 8a) consists of a 200 m×200 m area with 3 buildings, 6 WiGig APs, 1 5G NR BS, and 6 UVs (4 ground robots and 2 drones). The WiGig APs are connected to 4 CNs (2 APs to 2 CNs each, the remaining 4 to 1 CN each), 1 of which is cloud-like and has 8 CPUs, while the remaining are edge-like with 4 CPUs each. The WiGig APs are scattered in the area, while the 5G NR BS is about 900 m off-center.

Tab. II shows the results of the target parameters obtained for this scenario. HARM obtains the smallest handover rate,



(a) (b)

Fig. 8. The two scenarios: *simple* (a) and *high_density* (b).

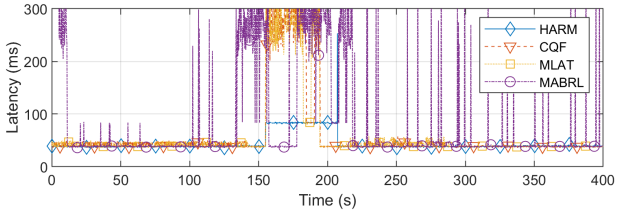


Fig. 9. Latency vs. time for UV1 in the *simple* scenario.

CQF obtains a high UL access bit rate, and MLAT obtains the lowest latency. MABRL does not achieve the best latency, because, differently from the other algorithms, it has no side information available and discovers the best solution by learning. Fig. 9 shows how the experienced latency varies along time for a ground robot (UV1): in the MABRL case, the exploration phases cause latency spikes, which increase the average value. Fig. 10a reveals that CQF and MLAT have comparable performance for all UVs; one mobile robot (UV4) experiences high latency with all algorithms, due to a long service blockage caused by absence of radio coverage.

B. Results for the high_density Scenario

This wider scenario ($500\,\mathrm{m} \times 300\,\mathrm{m}$) has 16 buildings, 24 WiGig APs, 1 5G BS, and 10 ground robots moving from the left to the right side (Fig. 8b). The WiGig APs are connected to 12 edge-like CNs with 8 CPUs (each CN connects to 2 APs), while the 5G BS is connected to a cloud-like CN with

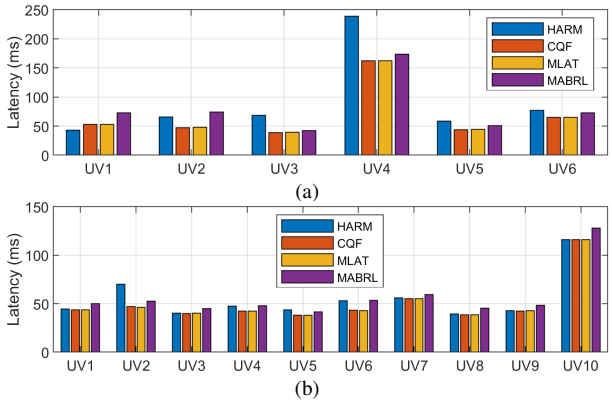


Fig. 10. Latency vs. UV for the *simple* (a) and *high_density* (b) scenarios.

TABLE III
RESULTS FOR *high_density* SCENARIO

Parameter	HARM	CQF	MLAT	MABRL
Serv. blockage rate $B^{(S)}$ (%)	2.8	2.0	2.0	3.0
UL bit rate $\bar{r}^{(U)}$ (Mbit/s)	1709.5	**2069.7**	1551.5	1583.6
Handover rate H (%)	**0.4**	32.8	61.4	11.1
Latency $\bar{L}$ (ms)	56.7	**50.9**	**51.0**	57.7

8 CPUs. The 5G BS is off-center like in the *simple* scenario, but the WiGig APs are uniformly placed in the area.

Tab. III shows the obtained results. HARM achieves the smallest handover rate, CQF yields the highest UL access bit rate, and MLAT produces one of the lowest latencies. MABRL does not obtain the lowest latency, but the gap with the other algorithms is smaller than in the *simple* scenario. CQF, MLAT, and MABRL provide a higher handover rate than in the *simple* scenario, because in the *high_density* scenario there are more APs, and all algorithms (except HARM) do not try to keep the previous AP. For the same reason the service blockage rate is lower for all algorithms. From Fig. 10b, we note that CQF and MLAT provide the best latencies for all UVs, with MABRL giving a close performance: one mobile robot (UV10) has a larger latency than the others, due to limited radio coverage.

V. CONCLUSION

This paper has presented radio and service selection algorithms used for unmanned vehicles assisted by task offloading and edge computing. The HARM, CQF, and MLAT algorithms need prior information on the radio network, computation nodes, and services. Differently, the MABRL algorithm learns over time the best service to use for computation offloading. While MABRL is not optimal as the other ones, it yields an effective performance in realistic simulated scenarios.

REFERENCES

[1] K. C.-J. Lin, H.-C. Wang, Y.-C. Lai, and Y.-D. Lin, "Communication and computation offloading for multi-RAT mobile edge computing," *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 180–186, Dec. 2019.

[2] A. Ali and K. Ali, "Multi-radio parallel offloading in multi-access edge computing: Optimizing load shares, scheduling and capacity," *IEEE Internet Things J.*, pp. 1–17, 2023.

[3] G. Baruffa *et al.*, "AI-driven ground robots: Mobile edge computing and mmWave communications at work," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 3104–3119, 2024.

[4] J. Chen, X. Zhang, B. Xin, and H. Fang, "Coordination between unmanned aerial and ground vehicles: A taxonomy and optimization perspective," *IEEE Trans. Cybernetics*, vol. 46, no. 4, pp. 959–972, 2015.

[5] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artificial Intell. Res.*, vol. 4, pp. 237–285, 1996.

[6] X. Han and X. Xu, "Multi-armed bandit algorithm for online offloading and scheduling in edge computing environment," in *Int. Conf. Comput. Netw. Commun. (ICNC 2024)*, Big Island, Hawaii, Feb. 2024, pp. 81–87.

[7] N.-N. Dao, M. Park, J. Kim, and S. Cho, "Adaptive MCS selection and resource planning for energy-efficient communication in LTE-M based IoT sensing platform," *PloS one*, vol. 12, no. 8, pp. 1–20, Aug. 2017.

[8] V. Kuleshov and D. Precup, "Algorithms for multi-armed bandit problems," *arXiv preprint arXiv:1402.6028*, 2014.

[9] C. de Souza *et al.*, "A study on propagation models for 60 GHz signals in indoor environments," *Frontiers Commun. Netw.*, vol. 2, 2022.

[10] T. S. Rappaport *et al.*, "Overview of millimeter wave communications for fifth-generation (5G) wireless networks—With a focus on propagation models," *IEEE Trans. Ant. Prop.*, vol. 65, no. 12, Dec. 2017.