Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Who Pays for the Drones? A Two-Part Market for Multi-Provider UAV Edge Computin

Mutaz Al-Tarawneh

mutaz.al-tarawneh@aum.edu.kw

American University of the Middle East https://orcid.org/0000-0002-8894-6946

Research Article

Keywords: UAV-assisted edge computing, mechanism design, reverse auction, capacity market, threshold payments, multi-provider markets

Posted Date: November 4th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11603012/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Who Pays for the Drones? A Two-Part Market for Multi-Provider UAV Edge Computing

Mutaz A. B. Al-Tarawneh[1*]

[1*]College of Engineering and Technology, American University of the Middle East, Egaila, 54200, Kuwait.

Corresponding author(s). E-mail(s): mutaz.al-tarawneh@aum.edu.kw;

Abstract

Uncrewed aerial vehicles (UAVs) can extend edge computing to crowds and areas that ground infrastructure cannot serve. However, a UAV incurs a cost for every minute it is airborne, whether or not it computes. This paper examines how that airtime can be funded when several independent providers sell computation through per-task auctions. It is proved that, when deployed drones compete on price for each task, prices that serve every worthwhile task and never exceed users' valuations cannot cover airtime costs. Simulations confirm this dilemma: pricing tasks at marginal cost causes losses of up to $243 per hour with 20 drones, whereas including airtime in task prices turns away valuable tasks and still causes losses with 15–20 drones. A two-part mechanism is proposed: drone availability is paid in a periodic capacity market, and each task is priced in a second-price auction. Capacity offers are compared by cost per unit of service and paid threshold prices, which makes truthful bidding optimal in each period and keeps payments within the value at stake. In 3,950 simulations, the mechanism keeps providers profitable at every fleet size evaluated and, with 20 drones, yields $113–215 per hour more welfare than full-cost pricing. With an optimally sized fleet, full-cost pricing and an adaptation of a published mechanism are 2–4% more efficient, so the mechanism's value lies in sustainable provider profits and robustness to a fleet size not known in advance. Because users or a sponsor must fund the capacity payments, provider profitability does not imply that users gain.

Keywords: UAV-assisted edge computing, mechanism design, reverse auction, capacity market, threshold payments, multi-provider markets

1 Introduction

Edge computing places computational resources close to users so that latency-sensitive tasks, such as image classification, rendering and interactive media, can be completed within their deadlines [1]. Where ground infrastructure is sparse or overloaded, as at stadiums, festivals or disaster sites, uncrewed aerial vehicles (UAVs) can carry edge servers to the users [2–4]. Most research on UAV-enabled edge computing optimises placement, trajectories and offloading on behalf of a single operator that knows every cost [5]. In practice, aerial computing is likely to be supplied by several independent providers whose costs are private, so that a market is required to determine which provider serves each task and at what price.

A motivating scenario. Consider a stadium during a match. Thousands of spectators share a small number of edge servers whose radio cells cover only parts of the stands, and demand peaks sharply at kick-off, at half-time and at the final whistle. Two operators deploy drones carrying edge servers and offer their capacity to the platform that serves the spectators' applications: short-deadline interactive content, image analysis, rendering and media processing. Each operator knows its own costs, whereas the platform does not. The platform must therefore decide, for each task, which server should execute it and at what price, and, for each period, how many drones should be kept airborne above each part of the stands. In the model considered here, a high-tier drone costs approximately \$20–25 per hour to keep airborne (airframe wear, operator time, batteries and energy), irrespective of whether it computes, whereas a single task is worth between \$0.002 and \$0.008 to its user. Recovering the airtime cost from task payments alone would therefore require several thousand paid tasks per drone and per hour, at prices that many users would decline.

A market of this kind faces a difficulty that ground servers do not present. A rotary-wing UAV consumes most of its power merely to remain airborne [6]: the hovering cost is incurred whether the drone computes or idles, and airframe wear, batteries and operators add further fixed costs. Per-task prices must therefore recover the cost of airtime. If tasks are priced at their marginal cost, that is, the cost of computing them, users obtain inexpensive service but providers incur a loss on every drone deployed. If the airtime cost is spread over the tasks, prices rise until users whose deadlines only a drone could meet decline to pay, and tasks worth serving remain unserved. Competition between providers exacerbates the problem, since it drives prices towards marginal cost. Existing auction mechanisms for edge and aerial computing price computation or task-level service [7–11]; none of the mechanisms reviewed in Section 2 separates the cost of keeping UAV capacity airborne from the cost of executing tasks among competing providers.

This paper addresses the question of who should pay for the drones, and by what mechanism. Three specific questions are considered:

(i) Can a decentralised market in which providers bid on individual tasks allocate them as efficiently as a planner that knows every provider's costs, and how close does either approach come to the best achievable outcome?
(ii) Which pricing schemes keep drone providers financially viable as the fleet grows, and at what cost in efficiency and service quality?

(iii) Can availability be remunerated in a manner that makes truthful cost reporting optimal for providers, and does this property hold in a dynamic market with queues, mobility and demand surges?

The contributions of this paper are fourfold.

1) **A multi-provider market model with explicit cost structures.** The model comprises competing providers that operate edge servers and heterogeneous UAV fleets alongside a remote cloud, users with deadline-constrained tasks and valuations, an air-to-ground channel with shared links, rotary-wing propulsion energy, batteries and the cost of airtime (Section 3).
2) **A deployment dilemma and a mechanism that resolves it.** It is proved that, under first- or second-price competition among deployed UAVs, per-task payments that serve every welfare-improving task and charge users at most their valuation cannot fund UAV airtime, and that, between providers with identical costs and coverage, this deters competing deployments (Proposition 4). A two-part mechanism is then proposed, in which a periodic capacity market remunerates drone availability and a second-price task auction prices computation. Capacity offers are compared by cost per unit of service and paid threshold prices; under this rule, truthful bidding is shown to be a dominant strategy in each capacity period taken in isolation (incentive compatibility across periods is not claimed), truthful providers never incur a loss on a booked period, and payments never exceed the value attributed to a site (Theorems 2 and 3). Edge servers, which operate continuously, retain full-cost prices (Sections 4 and 5).
3) **An informational-efficiency result.** With truthful second-price bids, the decentralised task auction allocates every task as a central scheduler with full knowledge of providers' costs would (Proposition 2); the gap between both and an offline upper bound is quantified.
4) **A large-scale evaluation.** The AirCompSim simulator [12] is extended with provider ownership, bidding, settlement, energy and battery models, and the mechanisms are evaluated, together with adaptations of two published approaches, in 3,950 one-hour simulations with paired seeds (Section 6).

The evaluation supports three principal findings. First, as Proposition 2 predicts, the task auction reproduces the allocation of an informed planner (on-time success differs by at most 0.3 percentage points), which validates the decentralised task stage; both reach 71–86% of the offline upper bound. Second, per-task pricing cannot sustain a drone fleet, whereas the proposed mechanism keeps providers profitable at every fleet size evaluated and degrades gracefully when too many drones are offered: with 20 drones it yields \$113–122 per hour more welfare than full-cost pricing in an event scenario and \$170–215 per hour more with uniformly distributed users. When the fleet size is not known in advance, its average welfare shortfall relative to the best achievable outcome is about half that of full-cost pricing and smaller than that of two published approaches adapted to this setting: pricing hover energy only while serving is 2–4% more efficient at its best fleet but leaves providers with losses from 8–10 drones, and

one-shot capacity procurement keeps providers profitable but falls up to $117 per hour behind with large fleets. Third, threshold payments resist manipulation in the full dynamic market: a provider that overstated its capacity costs earned less in every seed tested, whereas under pay-as-bid payments the same overstatement increased its profit by up to $19 per hour.

A limitation of the mechanism is also identified. When each design is given the fleet size that maximises its welfare, per-task full-cost pricing is 2–3% more efficient than the two-part mechanism, and the optimal fleet is small. At the default costs and valuations, the welfare-maximising choice is to fly few drones or none; whether aerial capacity is worthwhile depends on how users value timely results and on drone costs, and the capacity market is useful precisely because it makes this decision from observed demand. The value of the two-part design therefore lies not in peak efficiency but in sustainable provider profits and in robustness to a fleet size that cannot be known in advance.

The remainder of the paper is organised as follows. Section 2 reviews related work, Section 3 presents the system model, Section 4 describes the market mechanisms and Section 5 analyses their properties. Section 6 reports the evaluation, Section 7 discusses practical implications and limitations, and Section 8 concludes the paper.

2 Related Work

Auction-based offloading is well studied for terrestrial edge computing and increasingly for aerial and space–air–ground networks, but the mechanisms identified in the literature price either computation or task-level service. None separates the cost of keeping UAV capacity airborne from the cost of executing tasks among competing providers. The closest recent work, the iterative double auction of Xu et al. [11], trades divisible computation between UAV buyers and low-Earth-orbit (LEO) satellite sellers through a broker.

2.1 Offloading in UAV-Assisted and Space–Air–Ground Networks

Space–air–ground integrated networks (SAGIN) combine ground users, UAVs and LEO satellites [13]; UAV communication design is surveyed in [4, 14]. UAV-enabled edge computing began with joint bit allocation and path planning for a UAV-mounted cloudlet [2] and now treats UAVs as servers, relays or users [3]. A recent survey classifies resource management into provisioning, offloading and allocation, solved mainly by convex optimization, integer programming and deep reinforcement learning under a single operator with full system knowledge [5]. The SAGIN offloading studies reviewed in [11] follow the same pattern.

Two physical features shape the economics studied here. A rotary-wing UAV draws propulsion power whether or not it computes [6], a cost many studies omit by ignoring hover energy [5]. Coverage depends on altitude through line-of-sight probability [15], which determines which UAVs can bid for a given task.

2.2 Auction Mechanisms for Edge Resources

Auctions let resource owners reveal costs through bids instead of relying on a planner that knows them [1]. Single-round designs include double auctions for sharing cloudlets [16] and for industrial MEC [7], a truthful combinatorial auction with critical-value payments [8], a multiparticipant double auction run by a neutral auctioneer [17], and truthful online combinatorial auctions for offloading [18].

Xu et al. [11]. UAVs submit bid vectors and LEO satellites submit price vectors to a broker, which solves a surrogate welfare problem with logarithmic utilities and quadratic costs. Its optimality conditions match those of the true welfare problem when participants bid their marginal valuations, so the iterations converge to the welfare-maximizing allocation, in about six rounds for three UAVs and three satellites. The authors claim efficiency, individual rationality, incentive compatibility and budget balance, and add an alternating Hedge learner, a multiplicative-weights method [19], so participants can adapt bids without broker guidance.

The design follows Kelly-type proportional allocation [20] and broker-run double auctions for data offloading [21]. Three features limit its use for competing aerial providers:

- **Guarantees rest on price-taking.** Participants who anticipate their effect on prices cause efficiency loss [22], and no bilateral-trade mechanism is efficient, incentive compatible, individually rational and budget balanced in general [23], so dominant-strategy designs give up some efficiency [24]. The authors themselves note that the method needs a central controller and relies on stringent operational assumptions.
- **Divisible resources, no deadlines.** Utility is a concave function of allocated computation; there are no per-task deadlines or queues, and the market clears through repeated rounds.
- **No availability cost.** Seller cost is communication and computation energy plus a constant incurred when a satellite activates service for a task; the cost of keeping a server available does not appear.

Recent aerial auctions are multi-stage. Liu et al. [25] allocate high-altitude-platform (HAP) resources and then UAV resources in two truthful single-round stages, paying UAVs a uniform unit price per resource-second used. Lin et al. [26] combine a double Dutch auction, with clock steps tuned by reinforcement learning, and a Vickrey auction in which UAVs buy edge-server resources. Tan et al. [10] pay UAV sellers threshold prices per processed CPU cycle inside broker coalitions and, as in this work, limit their incentive guarantees to the task level because coupling across tasks rules out global strategy-proofness. This paper uses per-task second-price reverse auctions, which make truthful cost bidding dominant for each task in isolation [27]; incentive compatibility over a whole operating period is not claimed.

2.3 Competition and Deployment Incentives

Competition among providers is usually modeled as a pricing game. Su et al. [28] treat offloading providers as Stackelberg leaders competing on price. Asheralieva and Niyato study offloading in UAV-enabled MEC with multiple service providers through hierarchical games and reinforcement learning [29], and pricing across several infrastructure providers with incomplete information [30]. Apostolopoulos et al. [31] price UAV-mounted servers to shape how users compete for them.

A second group pays UAVs to participate: an online reverse auction that remunerates working UAVs [32], forward contracts between a UAV and an edge server [33], and a reverse auction that buys federated-learning services from UAV sellers [34]. Two recent designs come closest to procuring capacity. TRAIM [9] pays each winning base-station owner a critical value for its whole resource bundle, selected greedily to cover a known batch of tasks at minimum social cost. It is a one-shot procurement of participation, without user valuations or a time dimension. Tan et al. [10] count hover energy only for the time a UAV spends serving tasks, so idle airtime is never compensated.

Surveys name the remaining gap. Most UAV edge-computing studies assume one provider, although several may cover the same location with their own economic models [5], and a survey of satellite networks notes that high deployment costs require mechanisms that sustain provider revenue [35]. Electricity markets met the same problem, since energy prices near marginal cost do not fund capacity that is needed but rarely used, and answered it with capacity markets [36, 37]. This work's two-part mechanism applies that idea: a periodic, budget-limited capacity stage pays for UAV availability with threshold payments [38], and a per-task auction prices drone computation at marginal cost, while edge servers, outside the capacity market, keep full-cost prices.

2.4 Online Allocation and Dynamic Pricing

Tasks arrive continuously and must be committed on arrival. Online auctions for cloud and edge resources combine truthfulness with competitive-ratio guarantees [39, 40], usually through primal–dual analysis in which resource prices rise as capacity is consumed [41]. Revenue management uses the same idea as bid-price control, accepting a request only if its price covers the opportunity cost of the capacity it uses [42]. The opportunity-cost pricing evaluated in this paper follows that line, updating each server's capacity price from blocked demand and idle time, unlike the Hedge learner of [11], which adapts bids across repeated rounds of one market.

2.5 Simulation Tools

Evaluating markets for aerial computing requires a simulator that combines three elements that existing tools model separately: the computation and offloading of deadline-constrained tasks, the radio channel and energy of UAVs, and the economics of independent providers.

Cloud, edge and fog simulators. CloudSim [43] established discrete-event simulation of data-centre resources and provisioning policies, including the accounting of resource costs. iFogSim [44] extends this approach to Internet-of-Things, fog and edge

environments and evaluates the latency, network usage, energy and operational cost of application placement; iFogSim2 [45] adds device mobility, the dynamic clustering of fog nodes and microservice orchestration. EdgeCloudSim [46] models wireless and wide-area network delays and the nomadic mobility of users for offloading studies. In these tools the infrastructure is treated as belonging to a single operator: costs are accounted for, but servers neither belong to competing providers nor bid for tasks, and aerial platforms with their air-to-ground channels and propulsion energy are not modelled. VFogSim [47] adds mobile fog nodes carried by vehicles and data-driven modelling of 5G vehicle-to-everything links, and was designed for techno-economic capacity planning; its mobile nodes are ground vehicles, however, and cost enters as a deployment cost rather than through prices formed in a market.

UAV and network simulators. Joint UAV and network simulators, such as FlyNetSim [48], which couples the ArduPilot autopilot with the ns-3 network simulator, and simulators built on AirSim [49], reproduce flight dynamics, control and wireless communication in detail. They target networking and autonomy rather than the offloading of computation, and their fidelity carries a computational cost that is unnecessary when thousands of one-hour market scenarios must be simulated.

Air-computing simulators. AirCompSim [12] is a discrete-event simulator for the air computing paradigm [50]. It models users with different mobility types, edge servers, UAVs and a remote cloud, offloads each task by default to the server with the shortest queue, and provides heuristic UAV flight policies, an event interface for policies based on deep reinforcement learning, and scenarios of dynamic capacity enhancement. AirFogSim [51, 52] targets UAV-integrated vehicular fog computing, with modules for computation, communication and energy, and use cases that include the deployment of aerial base stations, UAV trajectory planning, vehicle-to-everything task offloading and resource allocation. Neither represents independent providers with private costs, bidding or monetary settlement, and neither accounts for the cost of keeping a UAV airborne.

The simulator used in this work. AirCompSim was adopted as the basis because its discrete-event core, server abstractions and offloading pipeline match the per-task decisions studied here. It was extended with (i) provider ownership, private costs, bidding, the task auction, the capacity market and settlement; (ii) an air-to-ground channel with shared links; (iii) rotary-wing propulsion energy, batteries and battery swaps; (iv) the accounting of airtime, compute energy and edge capital costs; and (v) the offline upper bound of Section 5. The extensions were validated against analytic results, namely the air-to-ground elevation and coverage relations of [15], the rotary-wing power curve of [6] and the Pollaczek–Khinchine waiting time of a single queue, and every theorem of Section 5 was checked numerically against the simulator's own allocation and payment code (Appendix A).

2.6 Summary and Research Gap

Competition, deadlines and truthful payments are each studied already; what is missing is a payment for UAV availability that coexists with competitive task pricing. Table 1 compares the closest works.

Table 1: Comparison with representative studies.

Study	Who trades	Task model	UAV cost treatment	Incentive guarantee	Payment for availability
Xu et al. [11]	UAVs buy computation from LEO satellites via a broker	Divisible computation, concave utility, no deadlines	No UAV costs; seller pays energy plus a per-task activation constant	Four properties at the price-taking equilibrium	None
Liu et al. [25]	Users buy from HAPs, then from UAVs	Resource bundles with delay limits	Unit cost per resource-second; energy only as a constraint	Truthful for users and UAVs, single round	None; UAVs paid per resource-second used
Lin et al. [26]	UAVs buy from edge servers	Resource bundles, video-quality utility	UAVs are buyers, so aerial supply costs do not arise	Argued informally, checked empirically	Not applicable
Wang et al. [9]	Base-station owners sell to a platform	Known batch; every task must be covered	No UAVs	Truthful and individually rational	One-shot payment for a station's resource bundle
Tan et al. [10]	Brokers recruit UAV sellers and compete for task contracts	Deadlines and reliability within 1-s epochs	Hover energy counted only while serving	Task-level threshold payments	None for idle airtime
This work	**Competing providers with edge servers and UAV fleets, plus the cloud, sell to users**	**Poisson tasks, hard deadlines of 0.3–3 s, FIFO queues, shared links**	**Propulsion, batteries, swaps and airtime operating costs**	**Truthful per task (second price) and per capacity period (threshold)**	**Periodic capacity payments, limited by demand-derived site value**

The gap is therefore the coordination of online task pricing with paid UAV availability among competing providers in a physically constrained aerial network. Per-task payments alone cannot fund airtime under competition, while folding airtime into every bid prices out tasks worth more than their marginal cost; this work separates the two and evaluates the consequences.

3 System Model

A service area is considered in which users generate deadline-constrained computation tasks and several providers sell computation on ground edge servers and on UAVs; a remote cloud is always available as a fallback. Time is continuous and the system is simulated by discrete events; Table 2 lists the default parameters, and Table 3 summarises the main notation used in the paper.

Figure 1 shows the physical infrastructure. Users are concentrated in hotspots, one of which lies outside the radio cells of the ground edge servers. Drones of two providers hover above demand that the edge servers cannot serve; each drone serves the users inside its coverage footprint and returns to a depot to swap batteries. Delay-tolerant tasks can also reach a remote cloud over the cellular network and a wide-area link. The market platform does not carry task data: it exchanges only bids, allocations and payments with the servers, while the input and the result of each task travel directly between the user and the server that executes it.

Table 2 Default parameters.

Parameter	Value
Area, edge servers	400×400 m; 4 servers of 10 GHz, Wi-Fi 5 GHz, 40 MHz
UAV tiers (low / medium / high)	altitude 50 / 85 / 120 m; compute 2.5 / 3.75 / 5 GHz; bandwidth 5 / 10 / 20 MHz; battery 80 / 110 / 150 Wh; hover power 168 / 224 / 307 W; capital cost \$4,000 / \$7,000 / \$12,000
UAV radio	3.5 GHz, 30 dBm, coverage threshold 15 dB SNR, dense-urban air-to-ground model
Providers	two: budget fleet (50/30/20% low/medium/high, markup 0.10) and premium fleet (20/30/50%, markup 0.25)
Cloud	$\$10^{-14}$ per cycle, \$0.09/GB egress; cellular link 5/30 Mbit/s per user (event), 20/100 Mbit/s (baseline)
Operations	placement period 100 s, idle recall below 2% utilisation, battery swap 180 s
Costs	electricity \$0.15/kWh, operator \$8/h per UAV, airframe amortised over 1,000 flight hours, edge capital \$10,000 over 3 years plus \$300/month site cost
Run	1 h, first 300 s excluded from task metrics, 10 seeds per configuration

Figure 2 shows the corresponding market structure. Users submit tasks, each with a deadline and a valuation, to the platform. Providers, each of which operates edge servers and a UAV fleet, bid in two markets: once per placement period for keeping drones available at candidate sites, and for every task they can serve in time. The cloud quotes a posted price. The platform runs both markets, assigns each task to a server or to the cloud, and settles all payments. The remainder of this section models the users, servers, radio channel, energy and costs; the two markets are specified in Section 4.

3.1 Users and applications

N users move in a 400 m $\times$ 400 m area. In the *event* scenario, 70% of users stay near one of two hotspots, one of which lies outside edge coverage, and roam within 60 m of their home point. Demand surges three times during the one-hour run, modelling arrival, half-time and departure: the arrival rate is multiplied by 2.5 during minutes 10–20, by 3 during minutes 30–40 and by 2 during minutes 50–60, with 60 s linear ramps, and the whole profile is rescaled so that the mean arrival rate over the run equals that of steady demand. During surges the cellular link to the cloud is congested further: its rates are divided by the demand multiplier. In the *baseline* scenario users are spread uniformly and demand is steady. Each user runs four applications (Table 4); application a generates tasks as a Poisson process. Task i of application a, created at t_i, has CPU demand c_i (log-normal around the application mean), input size d_i, result size $r_i = 0.1\, d_i$, deadline D_a and a valuation $v_i = v_a$: the most the user is willing to

Table 3 Main notation.

Symbol	Meaning
i, a, t_i	task, its application, its creation time
v_i, D_a	task valuation, application deadline
c_i, d_i, r_i	CPU cycles, input size, result size of task i
s, F_s, p_{is}	server, its capacity, processing time c_i/F_s
L_{is}, u_{is}, q_{is}, w_{is}	latency of i on s, and its upload, queueing and return components
k, m_k	provider, its markup
κ^m_{is}, φ_u	marginal (compute-energy) cost of i on s; airborne cost rate of UAV u
b_k, τ	task bid of provider k; second-price payment
P, j, V_j	placement period, candidate site, its value pool
β_u, w_u, r_u	capacity bid of UAV u, its capacity weight, its cost per unit of capacity β_u/w_u
θ_u	threshold payment to an accepted UAV
λ	share of lost value counted in a site's value pool
λ_s	shadow price of server s (opportunity-cost pricing)
$\mathcal{W}$, US, PP, CP	welfare, user surplus, provider profit, cloud profit

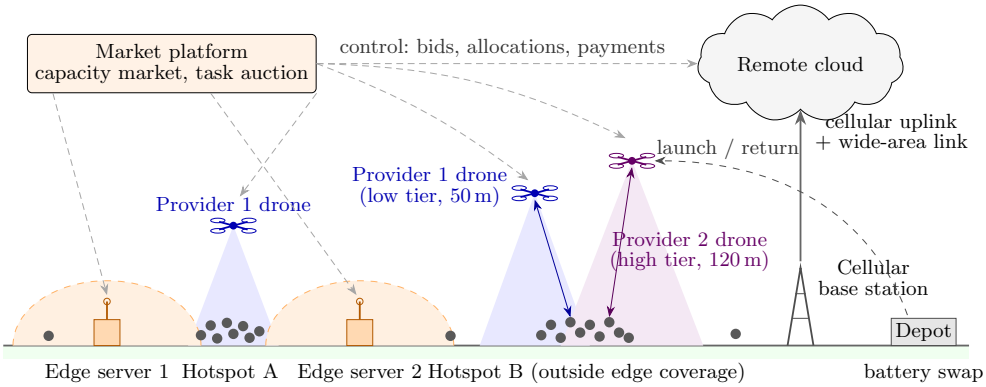


Fig. 1 Physical infrastructure (side view, not to scale). Users cluster in two hotspots, one outside the cells of the ground edge servers. Drones from two providers hover over demand, each serving the users inside its coverage footprint, and return to a depot to swap batteries. Delay-tolerant tasks can reach the remote cloud over the cellular network. The market platform exchanges bids, allocations and payments with all servers (dashed); task data flows directly between users and the server that runs the task.

pay for the result arriving within the deadline. A task completed after its deadline has no value.

3.2 Servers

Edge servers. Four edge servers with 10 GHz of compute sit on a regular grid and serve users through 5 GHz Wi-Fi access points (40 MHz, cells of about 97 m). **UAVs.**

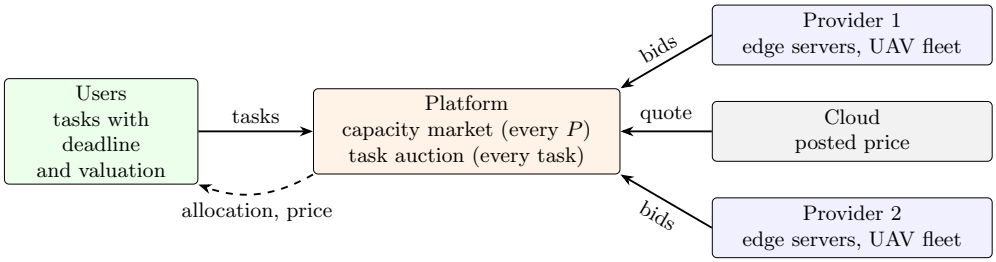


Fig. 2 Market structure. Providers bid for drone availability at candidate sites in a periodic capacity market and for individual tasks in a per-task auction; the cloud quotes a posted price. The platform allocates and settles payments.

Table 4 Applications (per user).

Application	CPU (Gcycles)	Input (Mbit)	Deadline (s)	Mean interarrival (s)	Value ($)
Entertainment	1.0	10	0.3	10	0.004
Multimedia	1.0	20	3.0	10	0.002
Rendering	2.0	5	1.0	20	0.006
Image classification	6.0	5	1.0	20	0.008

Each UAV belongs to one of three tiers (Table 2) that differ in altitude, onboard compute, channel bandwidth, battery and cost; a UAV serves users within the cell its altitude and radio allow. **Cloud.** The cloud has effectively unlimited capacity and charges a posted price per cycle plus egress and request fees, but it is reached over the cellular network and a wide-area link, so only delay-tolerant tasks can use it.

Every server processes tasks one at a time in first-in, first-out order. Server s has capacity F_s, so task i needs $p_{is} = c_i/F_s$ seconds of processing. Its latency is

$$L_{is} = u_{is} + q_{is} + p_{is} + w_{is}, \tag{1}$$

the upload time, the waiting time in the queue, the processing time and the time to return the result. Upload and download share the cell's link among the users transmitting at the same time.

When a task arrives, the platform predicts L_{is} for every candidate server from the server's current backlog and the user's current channel, assuming no further contention. Because later arrivals can only add to a first-in, first-out queue and to link sharing, this prediction never overestimates the true latency; a server predicted to miss the deadline is therefore excluded safely, while a server predicted to meet it may still miss it if the user moves or the link becomes more congested. Results that arrive after a drone has left the user's coverage are relayed through the nearest edge server, which adds delay.

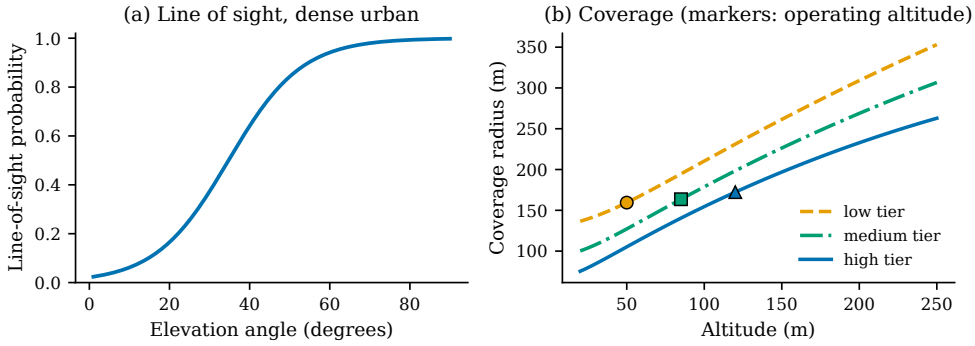


Fig. 3 Radio model. (a) Probability of line of sight as a function of the elevation angle in the dense-urban air-to-ground model. (b) Coverage radius (signal-to-noise ratio of at least 15 dB) as a function of altitude for the three UAV tiers; dots mark the tiers' operating altitudes.

3.3 Radio channel

UAV links follow the air-to-ground model of Al-Hourani et al. [15] for a dense urban environment. A user at elevation angle θ (degrees) sees the UAV in line of sight with probability

$$P_{\mathrm{LoS}}(\theta) = \frac{1}{1 + a\exp\big(-b(\theta - a)\big)}, \qquad (a, b) = (12.08,\ 0.11), \tag{2}$$

and the mean path loss is free-space loss plus an excess loss of 1.6 dB in line of sight and 23 dB otherwise, weighted by P_{LoS}. A user is covered when its signal-to-noise ratio is at least 15 dB; the rate is given by Shannon's formula over the UAV's bandwidth, shared among active users. Edge cells use a log-distance model with exponent 3. Altitude therefore trades coverage against link quality: a higher UAV is seen in line of sight from farther away but suffers more free-space loss. Figure 3 illustrates both effects: line of sight becomes more likely than not only above an elevation of about 35 degrees, and each tier's coverage radius grows with altitude, reaching 159–172 m at the tiers' operating altitudes.

3.4 UAV energy and operations

A rotary-wing UAV flying at speed V draws propulsion power [6]

$$P(V) = P_0\Big(1 + \frac{3V^2}{U_{\mathrm{tip}}^2}\Big) + P_i\Big(\sqrt{1 + \tfrac{V^4}{4v_0^4}} - \tfrac{V^2}{2v_0^2}\Big)^{1/2} + \tfrac{1}{2}d_0\rho s A V^3, \tag{3}$$

with the parameters of [6] scaled by a tier weight factor. Hovering ($V = 0$) costs $P_0 + P_i$: 168, 224 and 307 W for the three tiers, an order of magnitude more than the onboard computer (15–40 W). Figure 4 shows the resulting curves: cruising at 10 m/s draws 25–31% less power than hovering, power is lowest at 10–12 m/s, and it exceeds

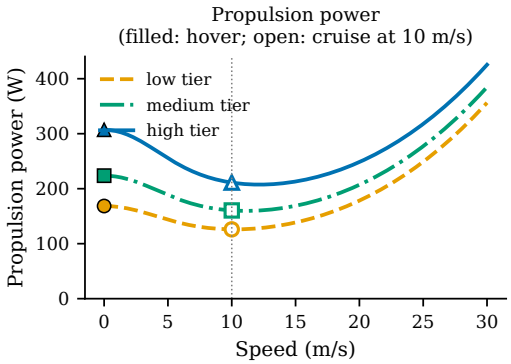


Fig. 4 Propulsion power of the three UAV tiers as a function of speed (rotary-wing model of [6]); filled markers show hovering, open markers the cruise speed of 10 m/s, at which power is 25–31% below the hover power; above 19–24 m/s it exceeds it.

the hover power only above 19–24 m/s, depending on the tier. The hover-based cost rate therefore bounds a UAV's cost in flight in the modelled operating regime, not in general.

Each UAV flies from its provider's depot to a site, serves, and returns to swap batteries when the state of charge falls below 25%, or aborts its queue if it would otherwise run out.

Every placement period ($P = 100\,\mathrm{s}$), each provider places its available UAVs using the location selection index (LSI) of AirCompSim [12], extended to several providers. Candidate sites are the four edge-server locations and the centres of a 3×3 grid. Each user is assigned to the nearest candidate site within coverage, and the provider assumes it will capture a share $1/|\mathcal{K}|$ of the demand there. For site j, let Λ_j be that share of the CPU demand rate (cycles per second), C_j the corresponding CPU work, $\bar{D}_j$ the mean deadline of the users' applications and F_j^{edge} the edge capacity covering the site. The index $\mathrm{LSI}_j = C_j/(F_j^{\mathrm{edge}} - \Lambda_j)$ estimates the delay the edge alone would impose; if the edge is overloaded ($F_j^{\mathrm{edge}} \le \Lambda_j$) or $\mathrm{LSI}_j > \bar{D}_j$, the site needs $\left(C_j/\bar{D}_j + \Lambda_j - F_j^{\mathrm{edge}}\right)/\bar{F}^{\mathrm{UAV}}$ drones, where $\bar{F}^{\mathrm{UAV}}$ is the provider's mean UAV capacity. Drones are then assigned one at a time to the site with the largest remaining need. A UAV whose utilisation stays below 2% is recalled. Figure 5 shows the service area at minute 35 of a run, during the largest demand surge: the capacity market has booked all 20 offered drones, 10 from each provider, at the candidate site in the centre of the area, which lies outside the cells of the four edge servers. This concentration is a property of the layout rather than of the snapshot. Across 10 seeds in each of three configurations (event scenario with 150 users and 10 or 20 drones offered; uniform scenario with 100 users and 10 drones), 99.8–100% of the drone-periods booked after the first period were at the central site, including in the uniform scenario without hotspots. The cause is the providers' placement heuristic rather than the geometry alone. In every configuration the placement index finds unmet demand at 12–13 candidate sites, but its estimate of the drones needed at the busiest site, 54–62 per provider at the central

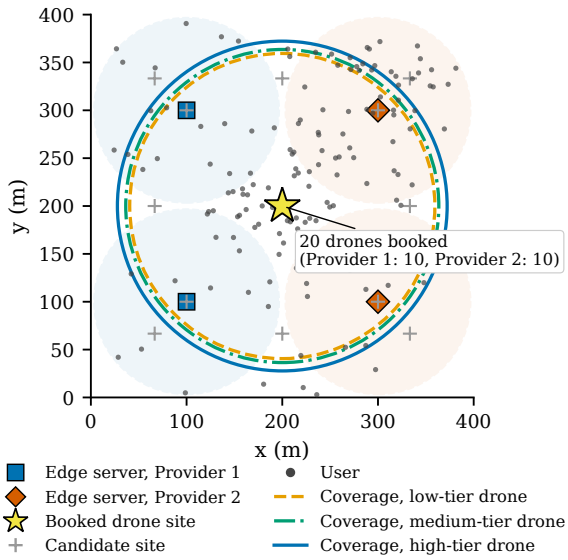


Fig. 5 Snapshot of the service area at minute 35 of a run during a demand surge (proposed mechanism, 150 users, 20 drones offered): users, the cells of the four edge servers, the candidate sites, and the site where the capacity market has booked drones, with the coverage of each drone tier.

site in the event scenario and 12–16 in the uniform scenario, exceeds the 5–10 drones each provider offers. Because the heuristic, inherited from AirCompSim, sends each drone to the site with the largest remaining need and lowers that need by one drone at a time, each provider's whole fleet goes to a single site. In these experiments the capacity market therefore decides how many drones, and of which tiers, serve that site, not where they fly, and the separation of value pools across sites is not exercised.

3.5 Costs and valuations

The *airborne cost rate* of UAV u,

$$\varphi_u = \alpha_u + o + P_u(0)\,\pi, \tag{4}$$

is what it costs per second to keep the UAV in the air: airframe and compute amortisation α_u (capital cost over 1,000 flight hours), operator cost o ($40 per hour per operator supervising five UAVs) and hover energy at electricity price $\pi = \$0.15/\text{kWh}$; battery wear is charged per charge cycle. Computing a task adds its compute energy, the *marginal cost* $\kappa_{is}^{m} = P_s^{\text{comp}}\, p_{is}\, \pi$. Edge servers have capital and site costs that accrue whether or not they compute. The cloud's cost of a task is its posted price c_i^{cloud}.

Welfare. Over a run of length T, with $z_i = 1$ if task i succeeds, welfare is the value of tasks completed on time minus all real costs:

$$\mathcal{W} = \sum_i v_i z_i - \sum_k C_k(T) - \sum_{i \to \text{cloud}} c_i^{\text{cloud}}, \tag{5}$$

where $C_k(T)$ is provider k's total operating cost. Payments are transfers: Proposition 1 shows that $\mathcal{W}$ equals user surplus plus provider profit plus cloud profit. Welfare is therefore the right yardstick for comparing mechanisms that allocate the same tasks but distribute payments differently: a mechanism that only transfers surplus from users to providers leaves it unchanged.

3.6 Information and behaviour

The following assumptions are made regarding the information available to each party and its strategic behaviour.

- **Providers** know their own costs and choose their bids and markups to maximise profit. Their costs are private: the platform does not observe airframe prices, operator wages or energy contracts. The tier of each drone (altitude, compute, bandwidth) is public and verifiable, for example through registration, so it may be used by the mechanism as a capacity weight.
- **Users** attach a value to each task completed within its deadline and are charged at most that value; strategic misreporting of valuations is not modelled and could be handled by a separate demand-side mechanism.
- **The platform** is trusted. It observes which server completed which task and when, and whether a drone booked in the capacity market was actually at its site, using telemetry. It keeps records of delivered and lost value, from which it estimates the value at stake at each site.
- **The cloud** is not strategic: it quotes a posted price and has effectively unlimited capacity.

These assumptions match settings in which a city, a venue or a network operator runs the platform and contracts with independent drone operators.

3.7 Problem statement

The platform requires a mechanism, that is, rules for allocation and payment, for two kinds of decision. At the start of each placement period it decides, for every candidate site j, which offered UAVs to book (the set A_j) and the capacity payment θ_u of each booked UAV. When task i arrives, it decides which server $s(i)$ executes the task, or whether the task falls back to the cloud, and the payment p_i. Both decisions are taken online, without knowledge of future tasks, and providers' costs are private. The objective is to maximise welfare (5) subject to:

- *Feasibility:* a task is assigned only to an available server that covers the user and whose predicted latency meets the deadline, $\hat{L}_{is(i)} \leq D_a$; servers process tasks in first-in, first-out order; UAVs respect their battery limits.
- *User individual rationality:* $p_i \leq v_i$ for every served task; whether users gain overall depends on how the capacity payments are funded (Section 6.5).
- *Provider individual rationality:* a provider that reports its true costs never incurs a loss on a served task or a booked period, $p_i \geq \kappa^m_{is(i)}$ and $\theta_u \geq \varphi_u P$.
- *Incentive compatibility:* reporting the true cost is a dominant strategy for each task and for each capacity period.
- *Budget:* capacity payments at a site never exceed the value at stake there, $\sum_{u \in A_j} \theta_u \leq V_j$.

Proposition 4 shows that per-task price competition alone cannot satisfy user individual rationality, efficient allocation and the funding of UAV airtime simultaneously under competition; this motivates separating the two decisions, and their payments, in the two-part mechanism of Section 4.

4 Market Mechanisms

The platform runs two kinds of market. A *task auction* allocates each task when it arrives; in the two-part mechanism, a *capacity market* additionally decides, every placement period, which offered drones are paid to be available. The pricing of tasks by providers is described first, followed by the two markets and the mechanisms compared in this study.

4.1 Quotes and the cost basis of prices

For a task i, provider k considers each of its servers s that covers the user, is available, and whose predicted latency meets the deadline. Its quote for s is

$$q_{is} = \left(\kappa^m_{is} + \gamma\, f_{is}\right)\left(1 + w_c\, B_s/\Delta\right)\left(1 + m_k\right), \tag{6}$$

where κ^m_{is} is the marginal (compute-energy) cost, f_{is} the server's fixed cost spread over its expected load, $\gamma \in \{0, 1\}$ selects the cost basis, B_s the server's backlog, $w_c = 0.5$ and $\Delta = 10\,\text{s}$ set a congestion surcharge, and m_k is the provider's markup; UAVs low on battery add a further surcharge. Under *full-cost* pricing ($\gamma = 1$) a drone's quote includes its airborne cost rate φ_u divided by its expected utilisation; under *marginal* pricing ($\gamma = 0$) it does not. Providers bid with their cheapest feasible server. Markups either adapt or are fixed. Adaptive markups follow a hill-climbing rule on profit: every 60 s each provider changes its markup by about 0.05, with ±50% random variation, in the same direction as long as its profit over the last interval improved, and reverses the direction when profit fell; markups stay within $[0, 2]$. Markups are set to zero unless stated otherwise, the configuration in which truthful bidding is optimal.

4.2 Task auction

Bids above the task's valuation v_i are discarded, which guarantees that no user pays more than the task is worth. Among the remaining bids, the lowest wins. Under *first-price* payment the winner is paid its bid; under *second-price* payment it is paid the second-lowest qualifying bid, or v_i if it is the only bidder. The cloud always quotes its posted price. If no bid qualifies, the task is sent to the cloud on a best-effort basis. Two further formats are evaluated in Section 6.9: a multi-attribute auction, in which each bid is scored by the equally weighted sum of its price and its predicted latency, both normalised to $[0, 1]$ over the bids for the task, and the lowest score wins; and a fairness auction, in which bids are ranked after multiplying each by $1 + \sigma_k$, where σ_k is the provider's share of all tasks won so far. In both formats the winner is paid its own bid.

Example. A rendering task worth $v_i = \$0.006$ arrives. A drone can meet the deadline and bids \$0.001, an edge server bids \$0.004 and the cloud cannot finish in time (illustrative values). The drone wins under either rule. Under first-price payment it receives \$0.001; under second-price payment it receives the next-best qualifying bid, \$0.004, so it would have gained nothing by bidding above its true cost: up to \$0.004 it wins at the same price, and above it loses. If the edge server could not meet the deadline either, the drone would be the only bidder and would receive the reserve, \$0.006, the whole value of the task. This last case, a single feasible server, is common in crowded hotspots and explains why second-price payments shift surplus from users to providers (Section 6.9). Figure 6 shows the complete sequence of messages for this example.

4.3 Capacity market

At the start of each placement period, every provider offers the drones it wants to keep at each candidate site, and the platform decides which offers to accept.

Value at stake. The platform estimates the value pool V_j of site j from its own records of the previous period: the value delivered by UAVs within coverage of the site plus a share $\lambda = 0.5$ of the value of tasks that failed there. To prevent double counting, sites claim the tasks they cover in decreasing order of their total, so each task counts for one site only; the order depends on demand, never on bids. V_j is a proxy for the value of serving the demand at the site, not the incremental value of one additional drone; the acceptance rule (8) shares it among the booked drones in proportion to their capacity.

Allocation. Offer u carries a bid β_u for being airborne for the period and a public capacity weight $w_u > 0$: the service rate of its tier for an average task, relative to the fastest tier,

$$w_u = \frac{\rho_u}{\max_{u'} \rho_{u'}}, \qquad \rho_u = \left(\frac{\bar{c}}{F_u} + \frac{\bar{b}}{B_u \log_2(1 + \gamma_{\min})} \right)^{-1}, \tag{7}$$

where $\bar{c}$ and $\bar{b}$ are the arrival-rate-weighted mean CPU demand and data volume (input plus result) of a task, F_u and B_u the UAV's compute capacity and bandwidth, and $\gamma_{\min}$ the coverage threshold of 15 dB, so that transfers are valued at the rate of

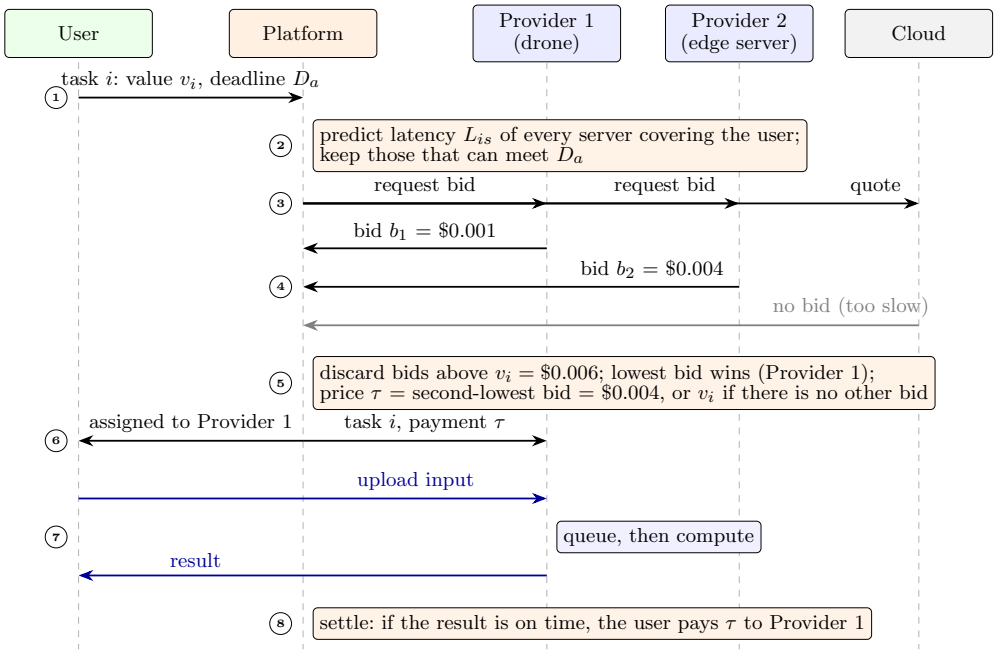


Fig. 6 The task auction for one task, with the illustrative values of the example in Section 4. Steps 2–6 take a negligible time compared with the task itself. If no bid qualifies, the task is sent to the cloud on a best-effort basis. The capacity market (Fig. 7) runs separately, once per placement period, and decides which drones are airborne to bid in the first place.

a user at the cell edge. The weights depend only on public tier specifications and the application mix. With the defaults, w_u is 0.40, 0.67 and 1.0 for the low, medium and high tier. Offers are sorted by cost per unit of capacity $r_u = \beta_u / w_u$ and the k-th is accepted while

$$r_{(k)} \le \frac{V_j}{W_k}, \qquad W_k = w_{(1)} + \cdots + w_{(k)}, \tag{8}$$

that is, while each accepted drone's capacity-proportional share of the pool covers its bid. Accepted drones fly to or stay at the site; rejected ones are not launched or are sent home.

Payment. Under *pay-as-bid*, accepted drones receive their bid. Under *threshold* payments, drone u receives

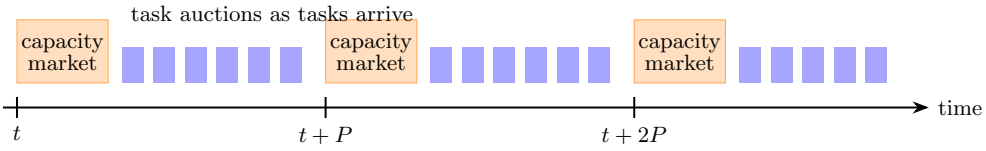
$$\theta_u = w_u \max_{1 \le l \le K_{-u}+1} \min\left(\frac{V_j}{w_u + \sum_{m<l} w_{(m)}},\ r_{(l)} \right), \tag{9}$$

where the other offers are sorted by r, $r_{(l)} = \infty$ beyond their number, and K_{-u} is the number of them accepted without u. θ_u is the highest bid at which u would still have been accepted. Section 5 shows why this payment makes bidding the true cost optimal. Algorithm 1 summarises one period of the capacity market at one site.

Algorithm 1 Capacity market at site j for one placement period

Require: value pool V_j; offers u with bids β_u and public weights w_u

sort offers by cost per unit of capacity $r_u = \beta_u / w_u$, ascending
$W \leftarrow 0$; $A \leftarrow \emptyset$
for each offer u in that order **do**
    if $r_u \le V_j/(W + w_u)$ **then** ▷ its share of the pool covers its bid
        $A \leftarrow A \cup \{u\}$; $W \leftarrow W + w_u$
    else
        break ▷ no later offer can pass (Lemma 1)
    end if
end for
for each $u \in A$ **do**
    compute K_{-u} by running the loop above without u
    $\theta_u \leftarrow w_u \max_{1 \le l \le K_{-u}+1} \min\big(V_j/(w_u + \sum_{m<l} w_{(m)}),\ r_{(l)}\big)$ ▷ threshold payment
end for
return accepted set A and payments θ_u



Capacity market: offers sorted by cost per unit of service; accepted while V_j covers each drone's share; accepted drones are paid θ_u for the period. Task auction: second-price; drones quote marginal cost, edge servers full cost; bids above v_i are discarded.

Fig. 7 The two-part mechanism. Availability is paid once per placement period; each task is priced separately.

Example. A site has a value pool of $V_j = \$8$ for the coming period and receives two offers: a low-tier drone with cost \$3 and weight 0.4, and a high-tier drone with cost \$5 and weight 1.0. Compared by cost per drone, the low-tier drone is cheaper and is accepted ($3 \le 8$); the high-tier drone would then need $5 \le 8/2$, which fails, so only the low-tier drone flies, and its threshold payment is \$5. Compared by cost per unit of service, the high-tier drone costs $5/1.0 = 5$ per unit and the low-tier drone $3/0.4 = 7.5$. The high-tier drone is accepted ($5 \le 8/1$); the low-tier drone would need $7.5 \le 8/1.4 \approx 5.7$, which fails. The high-tier drone flies and is paid $\theta = 1.0 \cdot \min(8/1, 7.5) = \$7.5$, the highest bid at which it would still have been accepted; it earns \$2.50 whatever it bids up to \$7.50, so it has no reason to misstate its cost. The weighted rule buys two and a half times the service for \$2.50 more.

4.4 Mechanisms compared

Four designs are compared; all use the task auction described above with second-price payments and zero markups unless stated otherwise.

- **Full cost (FC):** every server quotes full cost; there is no capacity market.
- **Marginal (MC):** every server quotes marginal cost; there is no capacity market.
- **Two-part, marginal edge (TP-M):** capacity market with threshold payments; every server quotes marginal cost in the task auction.
- **Two-part (TP), proposed:** capacity market with threshold payments (Fig. 7); drones quote marginal cost, because the capacity market pays for their airtime, while edge servers, which are always on and outside the capacity market, quote full cost.

Existing approaches. Two published mechanisms are adapted to this setting for comparison. *Hover cost while serving*, after Tan et al. [10]: drones are paid per task with threshold (second-price) payments, and their quotes add hover energy, including battery wear, only for the time spent processing the task, so idle airtime is never priced; the broker coalitions of the original are omitted. *One-shot procurement*, after TRAIM [9]: at the start of the run the platform buys UAV capacity for the whole run in a reverse auction that accepts offers by cost per unit of CPU capacity until its estimate of the capacity required at each site is covered, and pays critical values, capped at a reserve equal to the airborne cost of the most expensive UAV; procured drones keep their sites and return to them after battery swaps. The original procures base-station resource bundles for a known batch of tasks; here the requirement is estimated with the placement index of Section 3. In both adaptations edge servers quote full cost, as in the proposed mechanism, so that only the treatment of drones differs. The procurement rule was verified to be truthful and individually rational against about 105,000 random misreports.

TP-M isolates the role of edge pricing; the capacity rule without weights, used as an ablation, compares offers by cost per drone (all $w_u = 1$).

4.5 Design rationale

Each element of the proposed mechanism addresses a specific failure of simpler designs.

- **A separate payment for availability.** Airtime is a fixed cost of being ready to serve, not a cost of any particular task. Charging it per task either prices out valuable tasks or leaves it unpaid (Propositions 3 and 4). Electricity markets face the same problem with plants that are needed only at peak demand and address it with capacity payments [36, 37].
- **Value pools estimated from the platform's own records.** The platform pays for a drone at a site only up to the value it expects the drone to create there, so it never commits more than the demand justifies. Because the pools are computed before bids are seen and never overlap, the payment guarantees of Section 5 hold site by site.
- **Threshold rather than pay-as-bid payments.** With pay-as-bid, a provider gains by asking for more than its cost up to the acceptance limit; with threshold payments its payment does not depend on its own bid, so the cost is the best bid to report (Theorems 2 and 3). The price is an information rent paid by users (Section 6.7).

- **Weights by service capacity.** Drones differ in how much they can serve. Comparing cost per drone favours the cheapest, slowest airframes; comparing cost per unit of service buys capacity where it is cheapest. The weights depend only on public tier specifications, so they cannot be manipulated.
- **Drones at marginal cost, edge servers at full cost, in the task auction.** Once a drone's airtime is paid, the only cost a task adds is compute energy, so marginal-cost quotes allocate drone capacity efficiently. Edge servers receive no capacity payment and must recover their fixed costs through task prices; full-cost quotes also steer delay-tolerant tasks towards the cloud, keeping edge capacity free.
- **Second-price task payments.** They make truthful cost bids optimal for each task in isolation (Theorem 1), so the auction reproduces the allocation of an informed planner (Proposition 2).

Computational cost. A task auction compares at most one bid per provider and the cloud, so its cost grows linearly with the number of providers. At a site with n offers, the capacity market sorts the offers in $O(n \log n)$ time and computes each threshold payment by one further pass, $O(n^2)$ in total; with at most a few dozen offers per site, a period is cleared in milliseconds. Computing the value pools takes one pass over the tasks of the previous period for each candidate site.

4.6 Opportunity-cost pricing

As an option, each server adds a shadow price $\lambda_s \geq 0$ per CPU-second to its quotes, $q_{is} \leftarrow q_{is} + \lambda_s p_{is}$, so that a task is served on s only if its value covers both its cost and the capacity it displaces. The platform updates λ_s every second by a projected subgradient step: it rises with the CPU-seconds of valuable tasks lost because they were blocked by s's queue, and falls with s's idle fraction. This is the bid-price control of revenue management [42]; Proposition 7 relates it to the offline optimum.

4.7 Reference policies

To quantify what the market achieves, three non-market policies are also run on the same workloads. *Central scheduler* knows every server's true marginal cost and sends each task to the cheapest server that meets its deadline and value. *Earliest feasible* sends each task to the server predicted to finish first within its deadline. *Least queue*, the default policy of the original AirCompSim [12], picks the server with the shortest queue. Finally, an *offline upper bound* solves a linear-programming relaxation of the problem with all tasks known in advance (Theorem 4).

5 Analysis

This section establishes the properties of the mechanisms: what the task auction achieves, why no per-task mechanism can fund drone airtime, and what the capacity market guarantees. A provider's utility from a task or a period is its payment minus its cost if it is selected, and zero otherwise; ties between equal bids are broken by a

fixed rule and ignored in the proofs. Every result below was also checked numerically against the simulator's own payment and allocation functions (Appendix A).

5.1 Accounting

Proposition 1 (Welfare decomposition) *Let $Q = \sum_k Q_k$ be the total capacity payments and p_i the task payments. Welfare (5) equals* $\mathrm{US} + \mathrm{PP} + \mathrm{CP}$*, where* $\mathrm{US} = \sum_i (v_i z_i - p_i) - Q$ *is user surplus,* $\mathrm{PP} = \sum_k \left(\sum_{i \to k} p_i + Q_k - C_k(T) \right)$ *provider profit, and* $\mathrm{CP} = \sum_{i \to \mathrm{cloud}} (p_i - c_i^{\mathrm{cloud}}) \geq 0$ *the cloud's profit.*

Proof Every task payment goes to a provider or to the cloud, so in US+PP+CP the payment terms cancel and Q appears once with each sign, leaving (5). □

The cloud's profit is positive under second-price payments, when the cloud is the only bidder and is paid the user's valuation; counting it as a transfer, not a cost, is what makes welfare independent of the payment rule for a fixed allocation.

5.2 The task auction

Theorem 1 (Second-price reverse auction with reserve [27]) *Consider one task with reserve v. Among bids $b_k \leq v$ the lowest wins and is paid $\tau = \min(\min_{j \neq w,\, b_j \leq v} b_j,\ v)$. For each provider k with cost κ_k: (i) bidding $b_k = \kappa_k$ is weakly dominant; (ii) truthful bidding gives non-negative utility; (iii) under truthful bidding, the task is served by the cheapest feasible provider whenever that cost is at most v, which maximises the task's welfare.*

Proof Fix the other bids and let $\tau_k = \min(\min_{j \neq k,\, b_j \leq v} b_j,\ v)$. Provider k wins if and only if $b_k < \tau_k$, and is then paid τ_k, which does not depend on b_k. Its utility is $(\tau_k - \kappa_k)\mathbb{1}[b_k < \tau_k]$, maximised by $b_k = \kappa_k$ whether $\kappa_k < \tau_k$ (win, positive utility) or $\kappa_k \geq \tau_k$ (lose, zero). This gives (i) and (ii); (iii) follows because the lowest truthful bid among those at most v wins. □

The intuition is the familiar one for second-price auctions: the bid determines only *whether* a provider wins, not *what* it is paid. Overstating can only turn a profitable win into a loss; understating can only turn a loss into an unprofitable win. The reserve v plays the role of an extra competitor that bids the user's valuation, which is why a sole bidder is paid v.

Proposition 2 (The auction equals the informed scheduler) *With zero markups, marginal-cost quotes and no surcharges, in every state the auction and the central scheduler of Section 4 choose a server with the same cost; without exact cost ties their sample paths coincide.*

Proof Both consider the same candidates and the same feasibility test, and each provider's bid equals the true marginal cost the scheduler uses. The auction keeps bids at most v_i and picks the minimum, which is the scheduler's choice; both fall back to the cloud exactly when no candidate qualifies. By induction over events, identical decisions in identical states produce identical states. □

Corollary 1 (Informational efficiency) *With second-price payments, truthful bidding is dominant for each task and the resulting allocation is the one a planner who knows every provider's costs would choose. The market's contribution is informational: it reaches the informed planner's allocation without knowing private costs, and it cannot do better than that planner.*

These guarantees are per task. With queues, a provider may profit from losing one task to keep its server free for a later, more lucrative one, so truthful bidding is not dominant over a sequence of tasks (a two-task counterexample is given in Appendix A).

5.3 The deployment dilemma

Proposition 3 (Full-cost prices reject welfare-improving tasks) *Let a UAV be airborne for the period regardless of a given task and be the only server that can meet its deadline. If the task's value lies between the UAV's marginal cost and its full-cost bid, $\kappa^m < v_i < b^f$, the full-cost auction rejects it, although serving it would raise welfare by $v_i - \kappa^m > 0$.*

Proof The bid exceeds v_i and is discarded, so no feasible offer remains. Because the airborne cost is incurred either way, serving the task would change welfare by exactly $v_i - \kappa^m$. □

With the defaults, a 1-Gcycle task on a high-tier UAV has a marginal cost of about $\$2 \times 10^{-5}$ and a full-cost bid of about $\$2.6 \times 10^{-3}$; the Multimedia valuation ($0.002) lies in this gap.

Proposition 4 (Per-task price competition cannot fund airtime) *Consider a UAV with airborne cost $F > 0$ for a period in which it is the only feasible server for tasks $j = 1, \ldots, n$ with values v_j and marginal costs κ_j, and let $S = \sum_{j:v_j \geq \kappa_j} (v_j - \kappa_j)$ be the surplus it can create. Consider per-task payments that are* efficient*, serving every task with $v_j \geq \kappa_j$, and* user-individually rational*, $p_j \leq v_j$. (i) The provider's profit is at most $S - F$. (ii) If a second provider has deployed a UAV with identical costs that can serve the same tasks, and prices are set by first-price (Bertrand) competition or by a second-price auction among the deployed UAVs, every task's price equals its marginal cost and each deployed UAV's profit is $-F < 0$. (iii) If providers first decide whether to deploy and then compete as in (ii), no equilibrium has two UAVs serving the same tasks: a UAV is deployed only where its provider expects to be the sole feasible server, and it then recovers F only by charging users up to their valuations.*

Proof (i) Revenue is at most $\sum_{j \text{ served}} v_j$; serving tasks with $v_j < \kappa_j$ only lowers profit, so profit is at most $\sum_{v_j \geq \kappa_j} (v_j - \kappa_j) - F = S - F$. (ii) Under second-price payments the winner is paid the rival's truthful bid, κ_j (Theorem 1); under first-price bidding any price above κ_j is undercut. Either way $p_j = \kappa_j$ and profit is $-F$. (iii) By (ii), deploying alongside a rival yields $-F < 0$, whereas not deploying yields 0; hence at most one of the two deploys in any equilibrium, and the profit of a sole deployer is bounded by (i), which reaches $S - F$ only if prices extract the users' entire surplus. □

The result concerns the price-competition models used in the task auction, not every conceivable per-task mechanism, and it is stated for the stylised symmetric case of providers with identical costs covering the same tasks. With heterogeneous costs,

partially overlapping coverage or differentiated service quality, a deployed drone may retain a margin over its rivals, and competing deployments are not ruled out; the proposition identifies a structural problem under the stated conditions rather than the impossibility of competitive deployment in general. It separates competition among drones that are already airborne, part (ii), from the decision to deploy, part (iii): per-task competition either drives prices to marginal cost, so that no provider recovers its airtime, or deters the competing deployment, so that the remaining drone prices as a monopolist. Deploying the UAV is socially worthwhile whenever $S \geq F$, yet neither outcome both funds the UAV and leaves users their surplus. A separate payment for availability is needed: the capacity market.

To see the size of the problem, take a high-tier drone with an airborne cost of about \$23 per hour that is the only feasible server for tasks worth \$0.004 each with negligible marginal cost. Even if it extracts each user's full valuation, it must complete almost 6,000 tasks per hour, more than one every second for the whole hour, to break even. If a second provider's drone covers the same users, competition drives each price towards the marginal cost and the drone loses its whole airtime cost, however busy it is. Part (ii) is the analogue of the classic observation that marginal-cost pricing cannot recover fixed costs in markets with competition.

5.4 The capacity market

A single site with value pool V and the allocation rule (8) is considered. Offers with equal cost per unit of capacity are ordered by a fixed rule that does not depend on the bids (the order of their identifiers), so the sorted order, and with it every threshold below, is well defined; the arguments that follow use only this order.

Lemma 1 (Prefix structure and monotonicity) *(i) The accepted offers are a prefix of the order by r_u. (ii) If offer u is accepted with bid β, it is accepted with any lower bid, the other bids fixed.*

Proof (i) The bound V/W_k decreases and $r_{(k)}$ increases along the order, so after the first failure no later offer can pass. (ii) A lower bid moves u earlier; the offers ahead of it are a subset of those ahead before, whose tests are unchanged, and u's own bound V/W can only grow because fewer weights precede it. □

Proposition 5 (Pay-as-bid is budget-feasible but manipulable) *Under pay-as-bid payments, total payments at a site are at most V, but truthful bidding is not an equilibrium: an accepted provider can raise its bid up to the acceptance threshold and keep the difference.*

Proof With unit weights, $\sum_{l \leq K} \beta_{(l)} \leq K\beta_{(K)} \leq V$; the weighted case is analogous. For manipulability, take one offer with cost 1 and $V = 10$: bidding 1 earns 0, bidding 9 earns 8, and both are accepted. □

Theorem 2 (Threshold payments) *With unit weights, pay each accepted offer u its threshold $\theta_u = \max_{1 \le l \le K_{-u}+1} \min(V/l,\ c_{(l)})$, where $c_{(1)} \le c_{(2)} \le \dots$ are the other bids ($c_{(l)} = \infty$ beyond their number) and K_{-u} the number of them accepted without u. Then, in a single period: (i) bidding the true cost is weakly dominant; (ii) truthful bidders have non-negative utility; (iii) $\theta_u \le V/K$ for every accepted offer, so total payments are at most V.*

Proof Threshold. With bid b, offer u takes position $l(b) = 1 + \#\{c_{(m)} < b\}$ and is accepted if and only if the offers ahead of it pass, which holds exactly when $l(b) - 1 \le K_{-u}$, and $b \le V/l(b)$. On $(c_{(l-1)}, c_{(l)}]$ the position is l, so the accepted bids are $\bigcup_{l \le K_{-u}+1} (c_{(l-1)}, \min(c_{(l)}, V/l)]$; an empty interval contributes a value no larger than the preceding term, so the supremum is the stated maximum. (i)–(ii) By Lemma 1(ii) the accepted bids form an interval up to θ_u, and θ_u does not depend on u's bid. Utility is $(\theta_u - \kappa_u)\mathbb{1}[\beta_u \le \theta_u]$, maximised by $\beta_u = \kappa_u$ at value $\max(\theta_u - \kappa_u, 0) \ge 0$. This is Myerson's characterisation of truthful single-parameter mechanisms [38]: a monotone allocation with threshold payments. (iii) Removing an accepted u moves later offers forward, loosening their tests, so $K_{-u} \ge K - 1$. Terms with $l \ge K$ are at most $V/l \le V/K$; for $l \le K - 1$, $c_{(l)}$ is an accepted bid, at most $\beta_{(K)} \le V/K$. Hence $\theta_u \le V/K$ and $\sum \theta_u \le V$. □

Theorem 3 (Threshold payments with capacity weights) *With public weights $w_u > 0$, pay each accepted offer θ_u as in (9). Then truthful bidding is weakly dominant, truthful bidders have non-negative utility, and $\sum_{u \text{ accepted}} \theta_u \le V$.*

Proof The argument of Theorem 2 carries over with positions replaced by prefix weights. Lemma 1 gives the prefix and monotonicity, and the same supremum argument yields (9). For the budget, let K offers with total weight W_K be accepted. Since $K_{-u} \ge K - 1$, the prefix ending at u in any term with $l \ge K$ contains at least the $K - 1$ other accepted offers, so its weight is at least W_K and the term is at most V/W_K; for $l \le K - 1$, $r_{(l)}$ belongs to an accepted offer and is at most $r_{(K)} \le V/W_K$. Hence $\theta_u \le w_u V/W_K$ and $\sum_u \theta_u \le V$. □

Because the value pools are fixed before bids are seen and do not overlap, the theorems apply site by site and the total capacity payments of a period never exceed the total value at stake.

In words: an accepted drone is paid the most it could have asked and still been accepted. That amount is set by the other offers and by the value pool, never by the drone's own bid, so asking for more than its cost cannot raise its payment and can only cause it to be rejected. The budget bound follows because each accepted drone's threshold is at most its capacity-proportional share of the pool. The worked example in Section 4.3 illustrates both properties: the high-tier drone earns \$2.50 for any bid up to \$7.50, and its payment of \$7.50 stays within the pool of \$8.

Proposition 6 (Cost recovery) *Assume that (a) flight between sites, including climbing, draws at most the hover power and (b) battery wear is charged per unit of energy, so that it is included in φ_u. Under the two-part mechanism with truthful threshold bids and task payments at or above marginal cost, every UAV's profit satisfies $\Pi_u \ge -\varphi_u A_u^{\text{unpaid}}$, where A_u^{unpaid}*

is its airborne time outside accepted periods (transit to and from sites, and draining after a rejection).

Proof By (a) and (b), the UAV's airborne cost is at most φ_u per second. Each accepted period pays at least its truthful bid $\varphi_u P$, and each task at least its compute energy. Only airtime outside accepted periods remains unpaid. □

Both assumptions hold in the simulated operating regime: in the model of [6], cruising at 10 m/s draws 25–31% less power than hovering and propulsion power stays below the hover power up to 19–24 m/s, depending on the tier; vertical flight is charged at hover power; and the energy price of a UAV includes battery wear per joule. Assumption (a) is a property of this regime, not a general physical property: faster flight, or climbing that costs more than hovering, as real UAVs may experience, would add the excess to the unpaid term.

5.5 Opportunity-cost pricing and the upper bound

The offline bound relaxes the allocation problem, with all tasks of a run known in advance, to a linear program. Tasks that the cloud can finish within their deadlines are counted as successful and removed. For every other task i, let S_i be the set of servers (edge servers, and the UAVs serving at the task's creation) on which it would meet its deadline in isolation, $u^0_{is} + p_{is} + w^0_{is} \le D_a$, where u^0_{is} and w^0_{is} are the upload and return times without contention at the user's position at creation. Time is divided into windows of length L in two staggered families $f \in \{0, 1\}$, offset by 0 and $L/2$; let $\mathcal{T}_{sfk}$ be the tasks whose lifetime $[t_i,\, t_i + D_a]$ lies inside window k of family f and that may use server s. With weights ω_i equal to 1 (for the number of successful tasks) or v_i (for their value), the bound solves

$$
\begin{aligned}
\max_{x} \quad & \sum_{i} \sum_{s \in S_i} \omega_i \, x_{is} \\
\text{s.t.} \quad & \sum_{s \in S_i} x_{is} \le 1 && \forall i, \\
& \sum_{i \in \mathcal{T}_{sfk}} p_{is} \, x_{is} \le L, \quad \sum_{i \in \mathcal{T}_{sfk}} u^0_{is} \, x_{is} \le L, \quad \sum_{i \in \mathcal{T}_{sfk}} w^0_{is} \, x_{is} \le L && \forall s, f, k, \\
& 0 \le x_{is} \le 1,
\end{aligned}
\tag{10}
$$

with $L = 10$ s: within any window, a server's processor, uplink and downlink can each be busy for at most L seconds. The bound adds the optimum of (10) to the tasks the cloud alone can complete. A variant with weights $v_i - \kappa_{is}$ is used in Proposition 7 to interpret shadow prices; because it omits the fixed cost of keeping servers airborne, it is not an upper bound on welfare, and no welfare bound is reported.

Validity. The bound holds for every schedule the simulator can produce, under the assumption of Theorem 4 that users do not move during a task. Tasks that the cloud could complete are counted as successes whichever server handles them, and their use

of edge or UAV capacity is dropped, which only relaxes the program. Contention on a shared link delays transfers but cannot make them cheaper: when simultaneous users share a link's bandwidth, each transfer requires a share of the link's bandwidth-time equal to its isolated transfer time, and the shares of all transfers within a window cannot exceed L. Tasks whose lifetimes cross a window boundary are omitted from that window's constraints, which again only relaxes them; the staggered second family of windows restores most of the omitted constraints. UAVs are admitted only at their positions when a task is created, which is when the simulator allocates it. User mobility during a task changes link rates and is the only approximation, which is why the bound was also checked against every simulated run. In addition, the bound was compared with the exact optimum of 3,000 small workloads (five to seven tasks, two servers), found by simulating every assignment of tasks to servers with shared links and first-in, first-out processing, including workloads concentrated around window boundaries; it was never below the optimum.

Proposition 7 (Shadow prices implement the LP's admission rule) *Consider the LP* (10) *with task–server weights* $\omega_{is} = v_i - \kappa_{is}$*, and let* $\rho^{\mathrm{cpu}}_{sfk}, \rho^{\mathrm{up}}_{sfk}, \rho^{\mathrm{dn}}_{sfk} \geq 0$ *be optimal dual prices of its processor, uplink and downlink constraints. Define the opportunity cost of serving task* i *on server* s *as*

$$\pi_{is} = \sum_{(f,k):\, i \in \mathcal{T}_{sfk}} \left(\rho^{\mathrm{cpu}}_{sfk}\, p_{is} + \rho^{\mathrm{up}}_{sfk}\, u^0_{is} + \rho^{\mathrm{dn}}_{sfk}\, w^0_{is} \right). \tag{11}$$

Then a task served on s *in an optimal solution satisfies* $v_i - \kappa_{is} \geq \pi_{is}$*, and a task with* $v_i - \kappa_{is} < \pi_{is}$ *for every* $s \in S_i$ *is served in no optimal solution.*

Proof The dual constraint of x_{is} is $\mu_i + \pi_{is} + \nu_{is} \geq \omega_{is}$, where $\mu_i \geq 0$ is the price of the constraint $\sum_s x_{is} \leq 1$ and $\nu_{is} \geq 0$ that of $x_{is} \leq 1$. Complementary slackness makes it an equality whenever $x^*_{is} > 0$, so $\omega_{is} - \pi_{is} = \mu_i + \nu_{is} \geq 0$. If $\omega_{is} < \pi_{is}$, equality would require $\mu_i + \nu_{is} < 0$, which is impossible; hence $x^*_{is} = 0$ in every optimal solution. □

The opportunity cost combines the prices of every resource a task would occupy, in every window that contains its lifetime. The online opportunity-cost pricing of Section 4 approximates it with a single price per server and CPU-second, ignoring the link prices; this simplification is exact only when processing is the sole binding resource. The online rule therefore has no worst-case guarantee; it can only help when tasks differ in value per unit of the resources they use, and its value is measured in Section 6.8.

Theorem 4 (Upper bound) *If users do not move during a task, then for any policy using the edge servers, the UAVs serving at a task's creation and the cloud, the number (value) of successful tasks is at most the number (value) the cloud alone could finish in time plus the LP optimum.*

Proof Tasks the cloud could finish are counted in the bound. For every other successful task set $x_{is} = 1$ for its server. Because queues only delay tasks, each latency component is at least

Table 5 Properties of the mechanisms. P: proved in this section; C: fails, shown by counterexample; –: not applicable. Myopic: per task or per capacity period in isolation.

Property	Per task, first price	Per task, second price	Two-part, pay-as-bid	Two-part, threshold
User pays at most valuation	P	P	P	P
Truthful bidding optimal, myopic	– (profit only with markup)	P (Thm. 1)	C (Prop. 5)	P (Thms. 2, 3)
Truthful across tasks or periods	C	C	C	C
Allocation of an informed planner, given the deployment	P at zero markup	P (Prop. 2)	P (task stage)	P (task stage)
Fixed-cost recovery for UAVs	C (Prop. 4)	C (Prop. 4)	up to unpaid airtime	up to unpaid airtime (Prop. 6)
Capacity payments within value at stake	–	–	P	P

its isolated value, so (i, s) is a feasible option; and because a CPU or a channel serves one task at a time, the times assigned to s within any window that contains the tasks' lifetimes sum to at most L. Hence x is LP-feasible. □

Users move slightly during tasks in the simulator, so the bound is approximate; it was not violated in any of the runs.

5.6 Summary of properties

Table 5 collects the properties established in this section. The guarantees are *myopic*: they hold for each task, and for each capacity period, taken in isolation. No mechanism here has a constant competitive ratio against the offline optimum, since a task accepted now can block a more valuable one that arrives later.

6 Evaluation

The mechanisms are evaluated in simulation with respect to five questions. (Q1) How close does the decentralised market come to an informed planner and to the offline optimum? (Q2) Who can fund a drone fleet, and how does the answer depend on the fleet size? (Q3) Does the two-part mechanism resist manipulation? (Q4) Under which conditions are drones economically viable, and when does dynamic pricing help? (Q5) How robust are the conclusions to the auction format and to modelling choices? Table 6 summarises the answers.

Table 6 Summary of findings.

Question	Finding	Evidence
Q1: Market versus planner	The task auction matches an informed central scheduler to within 0.3 percentage points of on-time success and \$2.7 per hour of welfare; both reach 71–86% of the offline upper bound.	Section 6.2
Q2: Funding the fleet	Per-task pricing incurs losses as the fleet grows (marginal pricing from 6 drones, full-cost pricing from 15–20). The proposed mechanism is profitable at every fleet size and yields \$113–215 per hour more welfare than full-cost pricing with 20 drones. At the optimal fleet, full-cost pricing is 2–3% more efficient; under uncertainty about the fleet size, the proposed mechanism halves the average welfare shortfall. Capacity payments of \$100–274 per hour must be funded by a user levy or a sponsor; each provider's drone business is profitable in every run. Pricing hover energy only while serving (after Tan et al.) is 2–4% more efficient at its best fleet but loses money from 8–10 drones; one-shot procurement (after TRAIM) is profitable but up to \$117 per hour less efficient with large fleets; the proposed mechanism has the smallest average shortfall of all designs.	Sections 6.3–6.6
Q3: Incentives	Overstating capacity costs lowers the deviating provider's profit under threshold payments (by \$6.8–24.2 per hour, 10/10 seeds) but raises it by up to \$18.9 per hour under pay-as-bid.	Section 6.7
Q4: Economic viability and dynamic pricing	Ten drones approximately break even at the default parameters and become worthwhile if drone costs halve or valuations double. Opportunity-cost pricing raises welfare by \$31–157 per hour only when task values are heterogeneous.	Section 6.8
Q5: Robustness of the conclusions	Second-price payment shifts \$208–292 per hour from users to providers without changing the allocation; the fairness auction improves on first-price payment. A free-space channel overstates success by 15.6 points. The proposed mechanism is insensitive to the number of providers (2–4) and to placement periods of 30–100 s.	Section 6.9

6.1 Methodology

The simulator extends AirCompSim [12] with provider ownership, bidding, settlement, the radio, energy and battery models of Section 3, and the mechanisms of Section 4. It was validated against the analytic air-to-ground elevation angles of [15], the rotary-wing power curve of [6] and the Pollaczek–Khinchine waiting time of a single queue. Every configuration was run with 10 seeds; a seed fixes users, tasks and demand, so mechanisms are compared on identical workloads. Differences are reported as the mean of per-seed differences ± the 95% confidence half-width (Student t, 9 degrees of freedom), together with the number of seeds in which the difference has the stated sign. In total the study comprises 3,950 one-hour runs. Unless stated otherwise, runs use the event scenario, second-price task auctions, zero markups and the congestion surcharge of Section 4. Providers then bid their true costs, the setting of the incentive results of Section 5, so these runs evaluate the mechanism under truthful bidding. Strategic behaviour is examined separately: deliberate overstatement of capacity costs in Section 6.7, and adaptive markups in the competition experiments of Section 6.9. Welfare and all monetary quantities are expressed in dollars per one-hour run.

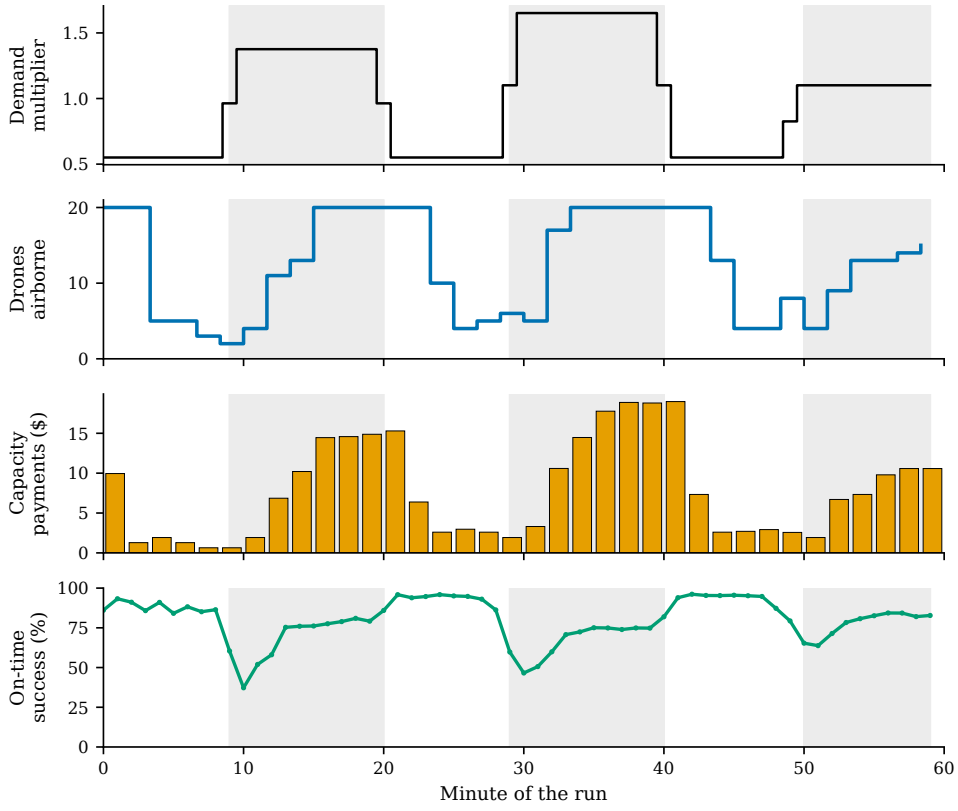


Fig. 8 The proposed mechanism in operation over one run (event scenario, 150 users, 20 drones offered, seed 1): demand multiplier, drones airborne at each placement period, capacity payments per period, and on-time success per minute. Shaded: demand surges.

The mechanism in operation. Figure 8 traces one run of the proposed mechanism with 20 drones offered. In the first period every planned drone is accepted, because the platform has no demand records yet; afterwards the capacity market books 3–6 drones between surges and up to all 20 during them, and the capacity payments per period follow the same pattern. At the start of each surge on-time success dips, to 37–65%, and it recovers as more drones are booked in the following periods; after the surge they are released.

6.2 Market versus planner and the offline bound (Q1)

Figure 9 compares the auction (marginal-cost quotes, no surcharge, the setting of Proposition 2) with the informed central scheduler, two non-market policies and the offline upper bound. The auction and the central scheduler are indistinguishable: in all ten configurations their on-time success differs by at most 0.3 points and their welfare by at most $2.7 per hour, as Proposition 2 predicts. This agreement is expected by

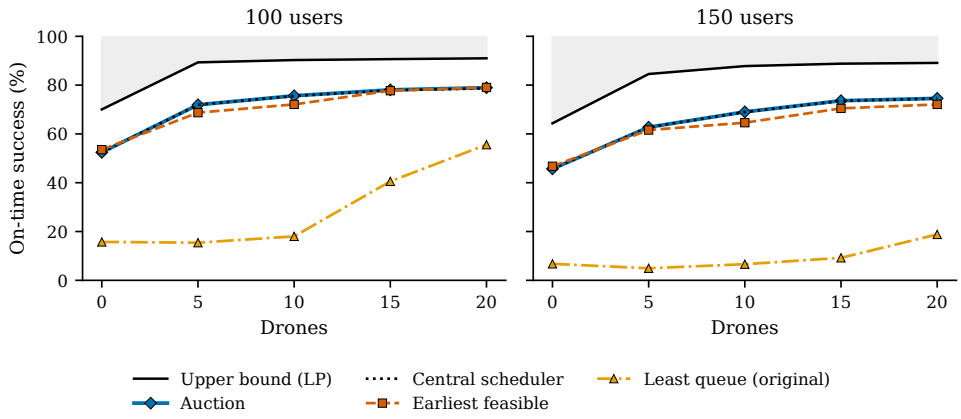


Fig. 9 On-time success of the auction, the informed central scheduler, two reference policies and the offline upper bound (event scenario, mean of 10 seeds). The auction and the central scheduler overlap.

construction and validates the decentralised task stage; it says nothing about deployment or welfare over time, which the following subsections address. The bound shows how much drones raise what is achievable, from 64–70% of tasks on time without drones to 85–91% with five or more; the auction reaches 71–86% of it, a share that grows with the fleet.

Against the deadline-greedy policy (earliest feasible), the auction completes 1.2–4.4 more points of tasks with 150 users whenever drones are present (10/10 seeds). With 10–20 drones and 100 users, however, its welfare is \$4–42 per hour lower (10/10): by routing tasks to the cheapest drone it keeps more drones busy, fewer are recalled, and they fly 1–2.5 more hours. Per-task prices do not account for the cost of keeping a drone airborne, a first sign of the dilemma of Proposition 4. The original AirCompSim policy, which ignores transfer times and deadlines, completes only 5–56% of tasks on time.

6.3 Who pays for the drones (Q2)

Figures 10 and 11 show welfare and provider profit as the number of drones offered grows from 0 to 20, for the four mechanisms. Per-task pricing cannot sustain a fleet. Under marginal pricing, providers incur losses from 6 drones onwards (100 users), reaching \$237–243 per hour with 20 drones. Under full-cost pricing they incur losses from 15 drones with 100 users and from 20 with 150, and the price of drone service turns away tasks worth serving (Proposition 3), so success stays at 62–72%. Welfare under both per-task schemes declines sharply as the fleet grows, because providers keep flying drones whose task revenue does not cover their airtime.

The two-part mechanism keeps providers profitable at every fleet size evaluated (\$131–319 per hour with drones) and degrades slowly. From 6 drones upward with 100 users, and from 8 upward with 150, it outperforms full-cost pricing at the same fleet size by \$11–122 per hour (10/10 seeds), and by \$113–122 per hour with 20 drones.

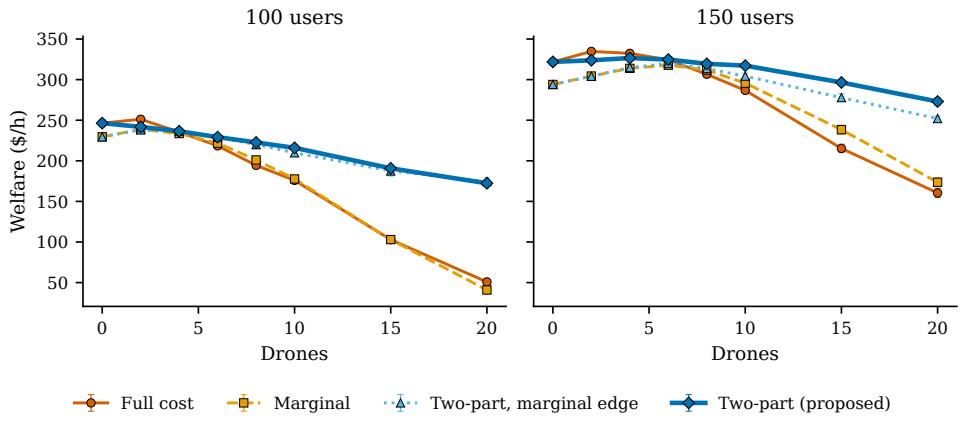


Fig. 10 Welfare as a function of the number of drones offered (event scenario; mean and 95% confidence interval over 10 seeds).

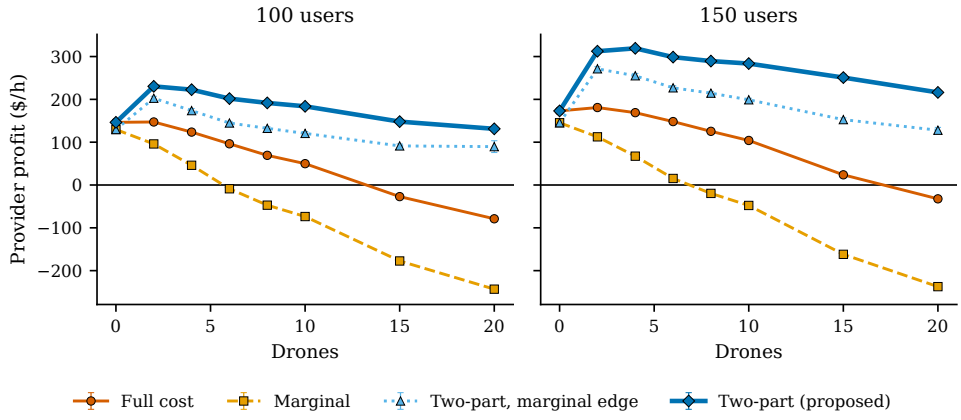


Fig. 11 Provider profit as a function of the number of drones offered (event scenario). Both two-part variants stay profitable at every fleet size.

Pricing edge servers at full cost is a significant design choice: compared with TP-M, the proposed design raises welfare by up to $28 per hour and success by up to 5.5 points, and with no drones it coincides with full-cost pricing, whereas TP-M trails it by $17–28 per hour. Comparing capacity offers by cost per unit of service is equally important: with cost per drone instead, the market booked an increasing share of low-cost, low-capacity drones as the fleet grew (63% of airtime from low-tier drones at 20 drones), and the weighted rule raised success by 0.5–7.6 points and welfare by $2–28 per hour (9/10 or 10/10 seeds).

Each mechanism at its best fleet. Table 7 gives each mechanism's welfare-maximising fleet. Here per-task full-cost pricing is slightly more efficient: paired at the

Table 7 Each mechanism at its welfare-maximising fleet size (event scenario, mean of 10 seeds; monetary values in $/h).

Users	Mechanism	Best fleet	Welfare	Success (%)	Provider profit	User surplus
100	Full cost	2	251.2	62.9	147.2	69.7
100	Marginal	2	238.7	63.1	95.9	110.5
100	Two-part, marginal edge	2	238.1	62.6	202.6	3.5
100	Two-part (proposed)	0	246.4	55.7	146.3	64.9
150	Full cost	2	334.7	54.9	180.9	96.9
150	Marginal	6	317.8	63.8	15.3	248.6
150	Two-part, marginal edge	6	319.8	62.8	226.9	41.7
150	Two-part (proposed)	4	326.6	59.8	319.2	−35.3

Table 8 Welfare shortfall ($/h) relative to the best mechanism at its best fleet size, over fleet sizes 0–20 (event scenario, mean of 10 seeds per configuration). HS: hover cost while serving; OS: one-shot procurement.

Users	Shortfall	FC	MC	TP-M	HS	OS	TP
100	Mean	68.9	72.6	37.7	55.6	60.1	**33.9**
100	Worst	202.5	212.4	**78.9**	186.0	197.8	81.0
150	Mean	52.4	56.5	40.1	32.3	34.4	**24.8**
150	Worst	177.3	164.1	85.6	132.6	119.2	**64.7**

respective best fleets, the proposed mechanism trails it by $\$4.8 \pm 0.7$ per hour with 100 users and $\$8.1 \pm 1.6$ with 150 (0/10 seeds better), about 2–3%. It outperforms marginal pricing by $7.7–8.8 per hour (10/10) and TP-M by $6.8–8.3 per hour. The optimal fleets are small: two drones for full-cost pricing and, at 100 users, no drones for the proposed mechanism. Since the optimal fleet cannot be known in advance and demand varies over time, the gradual decline of the proposed mechanism in Fig. 10 is considered more relevant in practice than the small gap at the optimum.

Robustness to the fleet size. This argument can be quantified from the same runs. For each demand level, let $\mathcal{W}^*$ be the highest welfare achieved by any of the six designs evaluated at any fleet size; the hover-cost design of Section 6.6 attains it at both levels. The *shortfall* of a mechanism at a given fleet size is $\mathcal{W}^*$ minus its welfare there. Table 8 reports the shortfall averaged over the eight fleet sizes evaluated, from 0 to 20 drones, and its maximum. If the fleet size is not known in advance, the proposed mechanism loses on average $33.9 and $24.8 per hour, about half the shortfall of full-cost pricing ($68.9 and $52.4) and less than either existing approach, and its worst case is 60–64% smaller than that of full-cost pricing. The advantage concerns uncertainty about the fleet size rather than about demand: when a fleet sized for 100 users serves 150, or the reverse, full-cost pricing loses nothing, because two drones are optimal for it at both demand levels, whereas the proposed mechanism loses $4.7–9.9 per hour relative to its own optimum.

Table 9 Baseline scenario: uniform users, uncongested cellular link (mean of 10 seeds; monetary values in $/h).

		Welfare			Success (%)		Profit
Users	Drones	FC	MC	TP	FC	TP	FC / TP
60	5	193.3	170.5	247.0	95.7	91.5	−10.0 / 58.7
60	10	141.8	111.2	237.0	96.1	92.0	−62.8 / 55.7
60	20	19.0	−14.9	233.9	96.3	91.5	−188.2 / 53.4
100	5	366.4	322.2	377.2	94.8	93.1	25.4 / 103.5
100	10	316.6	259.1	359.1	95.7	93.4	−26.3 / 86.0
100	20	196.4	137.9	365.9	96.1	91.6	−151.7 / 78.6

A second scenario. With users spread uniformly and an uncongested link to the cloud (Table 9), the proposed mechanism has the highest welfare in every configuration: $11–215 per hour above full-cost pricing and $55–249 above marginal pricing (10/10 seeds, except 8/10 at 100 users and 5 drones), and it is the only design in which providers profit at every fleet size. Full-cost pricing serves 1.8–4.9 points more tasks on time; pricing edge servers at full cost accounts for most of this difference, since TP-M trails full-cost pricing by 9–16 points.

6.4 Where tasks run, and why (Q2, continued)

The mechanisms differ less in how many tasks they serve than in where they send them. Figure 12 shows the destinations of tasks with 150 users. Under every mechanism 26–46% of tasks fall back to the cloud because no acceptable bid exists; this share falls as drones are added. The proposed mechanism moves work from edge servers to drones: with 20 drones, 42% of tasks run on drones and 19% on edge servers, against 17% and 33% under full-cost pricing. Drones quote marginal cost, because the capacity market pays their airtime, while edge servers keep full-cost quotes, so drones win more auctions.

Table 10 breaks success down by application. Full-cost pricing completes only 78% of multimedia tasks against about 96% under the other mechanisms: multimedia's valuation ($0.002) lies between a drone's marginal cost and its full-cost quote, exactly the gap of Proposition 3. The proposed mechanism completes the most image-classification tasks (55%), whose large compute demand suits drones. It also has a cost: it completes only 37% of entertainment tasks, whose 0.3 s deadline only an edge server or a nearby drone can meet, against 57% under full-cost pricing. Priced at marginal cost, drones attract tasks that would otherwise occupy edge servers, and the edge capacity that remains is contested by more delay-tolerant work. Reserving edge capacity for the most urgent applications is a natural extension (Section 7). Aggregate welfare therefore does not settle whether the mechanism suits a given deployment. Where the most urgent applications matter most, for example interactive services bound by strict service-level agreements, a loss of 20 points may be unacceptable, and full-cost pricing or a reserved share of edge capacity is preferable; where compute-heavy or delay-tolerant applications dominate, the trade-off favours the proposed mechanism.

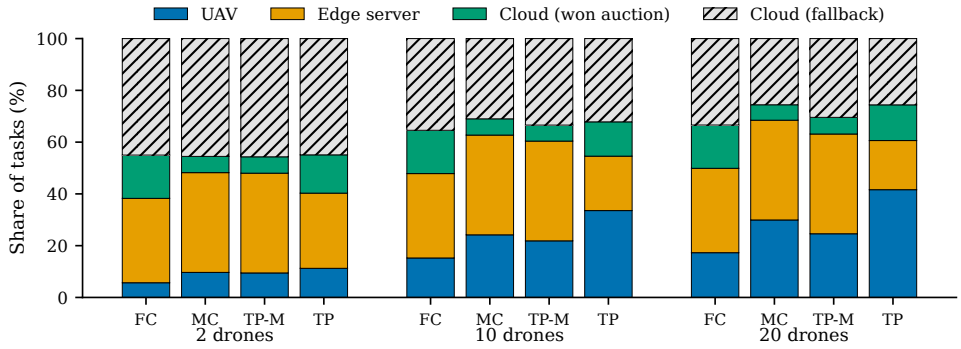


Fig. 12 Where tasks are executed (event scenario, 150 users, share of all tasks over 10 seeds). "Fallback" tasks had no acceptable bid and were sent to the cloud on a best-effort basis.

Table 10 On-time success by application (%), event scenario, 150 users, 10 drones, 10 seeds.

Application (deadline)	FC	MC	TP-M	TP (proposed)
Entertainment (0.3 s)	**57.1**	41.8	36.8	36.5
Image classification (1 s)	44.1	50.0	48.6	**54.9**
Rendering (1 s)	72.3	**89.0**	85.6	86.6
Multimedia (3 s)	78.2	95.5	95.7	**95.9**

Which drones the capacity market books. Table 11 shows the share of airtime flown by each drone tier with 100 users, under the capacity rule with and without service weights (both with marginal edge prices). Compared by cost per drone, the market books ever more of the cheapest, slowest drones as the fleet grows: low-tier drones fly 63% of airtime with 20 drones, and drones serve fewer tasks in total although they fly longer. Compared by cost per unit of service, high-tier drones fly 67% of airtime and drones serve 60% more tasks with less airtime.

6.5 Who funds availability, and who profits (Q2, continued)

The platform passes every task payment from the user to the winning server and pays the capacity payments $Q = \sum_j \sum_{u \in A_j} \theta_u$ from a capacity fund. Its cash flow per period is therefore balanced,

$$\underbrace{\textstyle\sum_i p_i}_{\text{from users}} + \underbrace{Q}_{\text{capacity fund}} = \underbrace{\textstyle\sum_i p_i}_{\text{to servers}} + \underbrace{\textstyle\sum_u \theta_u}_{\text{to booked UAVs}} , \tag{12}$$

Table 11 Airtime share by drone tier (%), flight hours and tasks served by drones per run, with and without service weights in the capacity market (event scenario, 100 users, TP-M, 10 seeds).

Capacity rule	Drones	Low	Medium	High	Flight hours	Tasks on drones
Cost per drone	5	24.5	41.9	33.6	3.5	18,305
	10	55.6	28.6	15.7	5.6	15,402
	20	62.5	25.4	12.1	7.4	12,324
Cost per unit of service	5	15.6	37.2	47.2	3.5	18,804
	10	25.4	31.8	42.8	5.3	20,602
	20	6.8	26.4	66.8	6.4	19,690

Table 12 Money flows under the proposed mechanism ($/h, event scenario, mean of 10 seeds). Task surplus is the value of tasks completed on time minus task payments; user surplus additionally deducts the capacity payments (levy funding). Airborne drones is the mean number of drones in the air over the run.

Users	Drones offered	Value delivered	Task payments	Task surplus	Capacity payments	User surplus (levy)	Airborne drones
100	0	252.0	187.1	64.9	0.0	64.9	0.0
100	2	276.3	192.5	83.8	100.3	−16.5	1.6
100	6	310.9	170.1	140.8	137.1	3.7	4.2
100	10	330.6	165.2	165.4	155.6	9.8	6.0
100	20	343.6	151.1	192.5	173.9	18.6	8.2
150	0	328.9	238.3	90.6	0.0	90.6	0.0
150	2	359.4	246.7	112.8	149.3	−36.5	1.6
150	6	415.9	225.9	190.0	202.7	−12.6	4.6
150	10	451.6	218.6	232.9	234.4	−1.5	7.1
150	20	498.4	202.0	296.4	274.1	22.4	11.4

and the question is who supplies the capacity fund. The budget bound of Theorem 3 limits capacity payments by the estimated value at stake, but that value is not collected from anyone, so the bound does not by itself make the market cash-balanced. Two funding models are therefore distinguished. *User-funded procurement*: the capacity fund is raised from users, for example as a levy per task or a subscription; this is the accounting used for welfare and user surplus throughout this paper (Proposition 1). Per-task individual rationality, $p_i \leq v_i$, then no longer implies that users gain from participating: their participation additionally requires that the levy not exceed their task surplus, which the mechanism as specified does not guarantee. *Sponsor-funded procurement*: a venue operator, a municipality or a service operator funds availability independently of task prices. Users pay only task prices, so per-task individual rationality guarantees that they gain, and the sponsor's payments are bounded by the value pools. Table 12 separates the flows for the proposed mechanism.

Capacity payments are substantial, $100–274 per hour. Under levy funding, users' surplus is negative with small fleets (down to −$36.5 per hour) and, at every fleet size, lower than without drones ($64.9 and $90.6 per hour): most of the value that drones create accrues to providers, through threshold payments and the share λ of lost value counted in the pools. Under sponsor funding, users keep their task surplus, which rises with the fleet, from $64.9 to $192.5 per hour with 100 users and from $90.6

Table 13 Profit by provider and business ($/h, event scenario, 150 users, mean of 10 seeds).

		Provider 1		Provider 2	
Mechanism	Drones	Edge	Drones	Edge	Drones
FC	10	74.1	−34.0	88.8	−24.9
FC	20	73.2	−91.1	88.2	−102.6
TP	10	60.8	57.9	85.5	79.4
TP	20	53.2	31.5	77.8	53.7

to $296.4 with 150, while the sponsor pays capacity payments that exceed the users' gain (for example, $155.6 per hour for a gain of $100.5 with 100 users and 10 drones). The mechanism therefore requires an explicit decision on who funds availability, and a sponsor that funds it should expect part of its payment to remain with providers as the price of truthful offers. The table also shows that offered and deployed fleets differ: of 20 drones offered, only 8.2 (100 users) and 11.4 (150 users) are airborne on average, because the capacity market declines drones whose expected value does not cover their cost. This, rather than any change in how tasks are allocated, is the main source of the mechanism's robustness to the fleet size.

Profit by provider and by business. Table 13 splits each provider's profit into its edge and drone business. Under full-cost pricing the drone business loses money in nearly every configuration, for both providers; providers remain profitable with small fleets only because their edge servers subsidise their drones. Under the proposed mechanism, the drone business of each provider is profitable in every one of the 140 runs with drones, with mean profits of $7.8–69.4 per hour for Provider 1 and $18.5–100.5 for Provider 2 across configurations; edge profits are somewhat lower than under full-cost pricing, because drones take over part of the edge servers' work.

6.6 Comparison with existing approaches (Q2, continued)

Figure 13 compares the proposed mechanism with the two adapted approaches and with full-cost pricing. With small fleets, pricing hover energy while serving yields the highest welfare of all designs: $253.3 and $337.7 per hour at its best fleet (2 and 6 drones), which is $7.0±0.5 and $11.1±2.7 per hour, or 2–4%, more than the proposed mechanism at its best fleet (10/10 seeds). Charging for hover energy during service recovers part of the airtime cost while keeping prices low, but idle airtime remains unpaid: providers lose money in every run from 8 drones (100 users) and 10 drones (150 users) upward, down to −$205 per hour with 20 drones, and welfare falls as the fleet grows, so that the proposed mechanism is ahead by $66.7 and $105.0 per hour with 15 and 20 drones and 100 users, and by $34.4 and $67.9 with 150 users (10/10). One-shot procurement keeps providers profitable at every fleet size and completes the most tasks on time with large fleets (84.4% and 81.0% with 20 drones), because it buys all the capacity its demand estimate calls for and keeps it for the whole run. It does not test whether that capacity is worth its cost: with 20 drones, user surplus under levy funding is −$199.4 and −$90.3 per hour, and welfare is $116.8 and $54.5 per

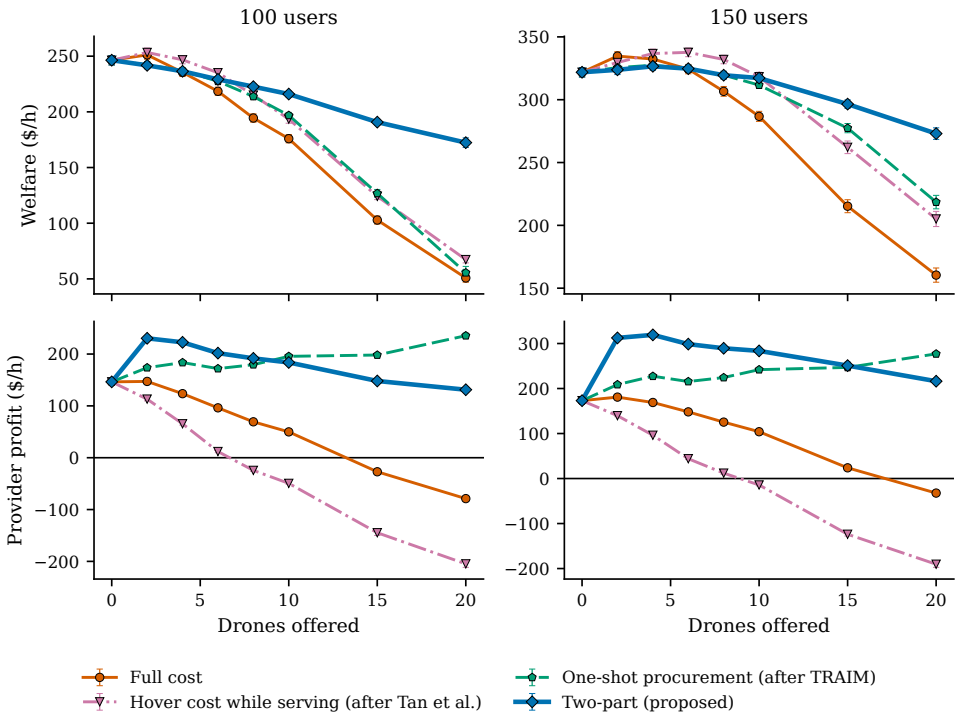


Fig. 13 Comparison with existing approaches adapted to this setting: welfare (top) and provider profit (bottom) as a function of the number of drones offered (event scenario; mean and 95% confidence interval over 10 seeds).

hour below the proposed mechanism (10/10). With small fleets, the two procurement designs perform alike (within $2 per hour).

With uniformly distributed users and an uncongested cellular link, the proposed mechanism has the highest welfare of the three in all six configurations: $11.0–215.7 per hour above the hover-cost design and $25.1–288.9 above one-shot procurement (10/10 seeds, except 8/10 against the hover-cost design with 100 users and 5 drones). Under the hover-cost design providers lose money in every configuration, and under one-shot procurement user surplus is negative with 10 or more drones at 60 users and with 20 drones at 100 users; both complete 0.9–4.9 points more tasks on time.

The proposed mechanism thus lies between the two approaches. Like one-shot procurement, it pays for availability and keeps providers profitable; like per-task pricing, it does not keep capacity whose expected value does not cover its cost. A design that leaves idle airtime unpaid is the most efficient when the fleet is small enough for its drones to be busy, and a design that buys capacity without a value test serves the most tasks, at a cost that users may not be willing to bear. Including both approaches, the proposed mechanism retains the smallest average welfare shortfall over fleet sizes (Table 8). These results compare the adaptations described in Section 4; they do not

Table 14 Change in provider profit ($/h) when Provider 1 multiplies its capacity bids by the factor shown, relative to truthful bidding (proposed mechanism, 150 users, 10 drones; 10/10 seeds unless marked).

Payment rule	Provider	×1.25	×1.5	×2	×3
Pay-as-bid	1 (inflates)	$+1.3 \pm 1.0$ (9/10)	$+8.2 \pm 0.7$	$+18.9 \pm 1.1$	$+18.1 \pm 2.2$
Pay-as-bid	2 (truthful)	+35.3	+42.0	+42.0	+48.6
Threshold	1 (inflates)	-6.8 ± 1.2	-9.1 ± 1.4	-12.2 ± 1.4	-24.2 ± 1.7
Threshold	2 (truthful)	+36.8	+46.2	+52.2	+74.9

establish superiority over the complete original mechanisms, which were designed for different settings.

6.7 Incentives in the capacity market (Q3)

Theorems 2 and 3 concern a single period with fixed value pools. To test them in the full dynamic market, Provider 1 was made to multiply its capacity bids by a factor while Provider 2 bid truthfully (150 users, 10 drones, proposed mechanism). Table 14 shows the change in each provider's profit. Under threshold payments, overstating always lowered the deviating provider's profit, by $6.8–24.2 per hour in 10/10 seeds at every level. Under pay-as-bid, overstating paid: by $8.2 per hour at ×1.5 and $18.9 at ×2, in 10/10 seeds, as Proposition 5 allows. In both cases the truthful rival gained $35–75 per hour, largely at users' expense. With marginal edge prices (TP-M) the gain from overstating under pay-as-bid was smaller (at most $3.7 per hour), because inflated offers were often rejected; with the proposed mechanism that deterrent largely disappears, so threshold payments are needed to make truthful offers optimal in each period.

The payment rules allocate identically under truthful bids, so the difference between them is a transfer: threshold payments move $21–125 per hour from users to providers across the configurations of Section 6.3 (TP-M), and $114 per hour in this experiment. This is the information rent that makes truthful bidding optimal in each period.

6.8 Economic viability of drones and the value of dynamic pricing (Q4)

Figure 14 varies user valuations and drone costs (hotspot scenario without surges, 150 users, proposed mechanism) and shows the welfare of 10 drones relative to none. At the default parameters ten drones approximately break even ($-\$10.2 \pm 3.7$ per hour); they become worthwhile if drone costs halve (+$54.5) or users value timely results twice as much (+$127.9). Meanwhile they raise on-time success by 14.6–20.9 points in every cell except that combining the lowest valuation with the highest drone cost (+2.4), where the capacity market books few drones. TP-M gives the same pattern ($-\$7.6 \pm 1.4$ at the defaults). Whether drones are worth flying is therefore an economic question, and the capacity market answers it from demand.

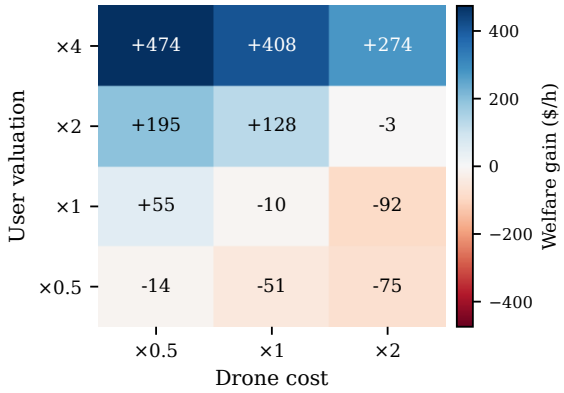


Fig. 14 Welfare with 10 drones minus welfare with none ($/h) as user valuations and drone costs are scaled (hotspot scenario, 150 users, mean of 10 seeds). Every cell holds in 10/10 seeds except valuation ×2 with cost ×2 (4/10 positive), which is break-even.

The share λ of lost value that counts towards a site's pool sets a trade-off (proposed mechanism, 150 users). With 10 drones, raising it from 0.25 to 1 raises success from 66.2% to 68.8% and welfare from $314 to $319 per hour, but shifts so much to providers that user surplus falls from $42 to −$82 per hour; at the default 0.5 it is about zero (−$1.5). With 20 drones a higher share raises success (to 76.4%) but lowers welfare, from $278 to $265 per hour. The default 0.5 is a compromise, not an optimum.

Dynamic pricing under surges. Opportunity-cost pricing helps only when task values differ. With heterogeneous values (entertainment $0.02, image classification $0.002, multimedia $0.001 per task) it raised welfare by $31–157 per hour in every configuration (10/10 seeds) while serving 3.8–6.5 points fewer tasks: it turns away low-value tasks during surges so that valuable ones find room, as Proposition 7 suggests. With the default, fairly similar values it lowered welfare by up to $15.5 per hour. A fixed congestion surcharge never helped reliably.

Figure 15 shows how this happens over a run. Outside surges both variants complete over 90% of tasks. During each surge, opportunity-cost pricing completes fewer tasks than no dynamic pricing (about 50% against 55–60%) but delivers more value per minute (up to $24 against $19), because the rising shadow price turns away low-value tasks so that valuable ones find room. After a surge the price falls back and both variants recover within a minute or two.

6.9 Auction formats, competition and modelling choices (Q5)

Formats. Compared with first-price payment (hotspot scenario, full-cost quotes, zero markups), second-price payment allocated identically but shifted $208–292 per hour from users to providers: when a single provider can serve a task, it is paid the user's full valuation. This makes providers profitable through monopoly extraction, not cost recovery, which is a further argument for paying availability explicitly. The fairness auction improved on first-price in every metric (success +0.2 to +2.5 points, welfare

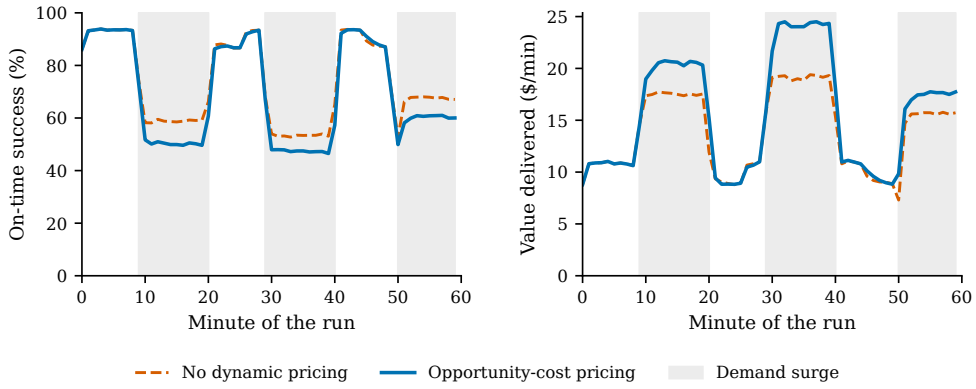


Fig. 15 Minute-by-minute on-time success and value delivered with heterogeneous task values (event scenario, 150 users, 10 drones, mean of 10 seeds). Shaded: demand surges.

+$1.6 to +$34 per hour) by spreading load and reducing queueing misses; the multi-attribute auction did worse (welfare −$7.5 to −$13).

Competition with strategic markups. With 2, 3 or 4 providers sharing the same fleet (150 users), welfare changed by −$4 to +$21 per hour, learned markups settled at 0.15–0.24 with no downward trend, and providers incurred losses in every run (−$70 to −$194 per hour). Valuations already cap prices, so additional sellers have little to undercut, and more competition does not solve the funding problem.

Modelling choices (150 users, 10 drones; Table 15). The channel model is by far the most consequential: free-space propagation instead of the air-to-ground model overstates success by 15.6 points and welfare by about a third, because it ignores the loss of line of sight at low elevation. Simulators with free-space links are therefore likely to overstate the value of drones. Idle recall saves 1.2 flight hours and $17 per hour without affecting service, and ignoring link sharing would flatter every mechanism by about 1.5 points.

Sensitivity of the proposed mechanism (150 users). Its results do not hinge on the placement period of 100 s. A period of 60 s raises welfare by $6.5±1.8 and $5.7±3.2 per hour with 10 and 20 drones, and 30 s gives +$5.0 and −$6.2; longer periods react too slowly to demand surges and lower welfare by $7.9–15.7 per hour at 200 s and $13.0–34.0 at 300 s (10/10 seeds). With three or four providers sharing the same fleet instead of two, welfare changes by +$0.3 to +$4.4 per hour, and every provider is profitable in every run, with mean profits of at least $39 per hour.

6.10 Threats to validity

Internal validity. All comparisons are paired by seed, so differences between mechanisms are not driven by differences in workload. The share of seeds in which each difference has the stated sign is reported, and most differences hold in all ten. The simulator's channel, energy and queueing models were checked against analytic results, and every theoretical property was checked against the simulator's own allocation

Table 15 Effect of modelling choices relative to the reference (event scenario, 150 users, 10 drones, full-cost pricing; reference: 64.5% success, \$286.7/h welfare). All non-zero differences hold in at least 9/10 seeds.

Variant	Success (points)	Welfare (\$/h)	Provider profit (\$/h)
Free-space channel instead of air-to-ground	$+15.6 \pm 0.7$	$+90.6 \pm 4.6$	-132.4 ± 4.2
No idle recall	-0.1 ± 0.1	-16.9 ± 1.5	-16.8 ± 1.3
No shared radio links	$+1.5 \pm 0.2$	$+9.1 \pm 1.0$	$+8.5 \pm 0.9$
Central instead of provider-aware placement	0.0	-0.1 ± 0.5	0.0

and payment code (Appendix A). The contribution of the individual design elements, namely the comparison of capacity offers by cost per unit of service and full-cost prices for edge servers, is assessed against variants that omit them (the unweighted capacity rule and TP-M), so that the effect of each element can be judged separately.

Construct validity. Welfare counts payments as transfers and charges every real cost, including the cloud's posted price and the airtime of idle drones; user surplus and provider profit are reported separately so that they can be weighted differently. On-time success treats a late result as worthless, which suits interactive applications but overstates losses for applications with soft deadlines.

External validity. Costs and valuations are assumptions, and the break-even map shows that conclusions about whether drones are economically viable depend on them. The geography (one 400×400 m area, four edge servers), the two demand patterns and the two providers are a deliberately small setting; the mechanism's guarantees do not depend on them, but the size of the welfare differences may. Strategic behaviour is limited to markups and capacity-bid inflation; richer strategies, such as manipulating value pools over several periods, are not simulated.

7 Discussion and Limitations

7.1 Implications for practice

For a platform operator. The results support the following configuration: a capacity market is run in each placement period, in which drones are compared by cost per unit of service and paid threshold prices; tasks are priced in a second-price auction; and edge servers retain full-cost quotes. Under this configuration the operator does not need to predetermine the number of drones in service, since drones are booked whenever their expected value, estimated from the platform's own demand records, covers their cost; the outcome still depends on the drones offered, their placement and the demand estimates. The only parameter that requires a policy decision is the share λ of lost value counted towards a site's pool: a higher share books more drones and improves service, but transfers a larger share of the surplus from users to providers.

Financial sustainability. Provider profitability is not the same as the financial sustainability of the market as a whole. Capacity payments of \$100–274 per hour must be funded by someone: if users fund them through a levy, they can be worse off than without drones (Section 6.5). A deployment should therefore either secure a sponsor, such as a venue operator or a public body, or cap the levy at the users' task surplus.

For drone providers. Under per-task pricing, a provider's best response to competition is to fly fewer drones or none, and the market under-provides aerial capacity exactly when it is most needed. Under the two-part mechanism a truthful provider never incurs a loss on a booked period, although airtime outside booked periods remains unpaid (Proposition 6), and is paid for availability, which makes investment in a fleet predictable.

For venues and public authorities. Where drones serve crowds at events or disasters, the capacity market makes the cost of aerial coverage explicit and limits it by the value that coverage creates. A public body subsidising coverage could do so by topping up value pools at chosen sites, without changing the rest of the mechanism.

7.2 Deployment considerations

A real deployment needs elements that the simulation takes for granted. Booked drones must prove that they were at their site for the period, which telemetry or remote identification can provide; payments for a period should be settled only after this check. The placement period trades responsiveness against overhead: a shorter period follows demand more closely, but makes drones fly between sites more often and gives the value-pool estimate less data. Value pools are computed from the platform's records, which providers could try to influence over several periods; auditing pools against long-run averages, or capping how fast they may change, would limit this. Finally, the task auction requires latency predictions for every candidate server, which edge orchestration systems already compute for offloading decisions.

7.3 Limitations

What the two-part mechanism offers. The mechanism does not make a perfectly sized fleet more efficient; it makes any fleet financially sustainable and keeps welfare high when the fleet is too large. Both matter because the right fleet depends on demand, valuations and costs that change and that no operator knows in advance. The capacity market answers the question of which drones are worth flying from the platform's own records of demand, and threshold payments make truthful offers optimal for providers in each period.

The price of truthfulness. Threshold payments transfer \$21–125 per hour from users to providers relative to pay-as-bid with marginal edge prices, and \$114 per hour with the proposed mechanism (150 users, 10 drones). Under the proposed mechanism, users keep little surplus with small fleets, and with 2–10 drones and 150 users their surplus is slightly negative (−\$36 to −\$2 per hour), because they pay full-cost edge prices as well as capacity payments. A reserve price, a cap on capacity payments, or funding by a sponsor (Section 6.5) is a natural refinement.

Urgent applications. Pricing drones at marginal cost draws work away from edge servers, which helps compute-heavy tasks but cost the most urgent application (0.3 s deadline) 20 points of success relative to full-cost pricing. Reserving a share of edge capacity for tasks that only the edge can serve in time, or prioritising by deadline within the task auction, would address this without changing the capacity market.

Booking efficiency. When a provider overstated its capacity costs, total welfare rose slightly (from \$317 to \$323 per hour) while success fell, which suggests that the default pool share books somewhat more drones than is efficient. The pool share λ is a design parameter that trades success against welfare and user surplus.

Parameters and scope. At the default costs and valuations, ten drones approximately break even, so any conclusion on the economic viability of drones is conditional on these parameters; the break-even map of Fig. 14 is the more reliable result. All runs use one 400×400 m area with four edge servers and two demand patterns. Because the providers' placement heuristic sends each provider's whole fleet to the site of largest need, drones are booked almost exclusively at one site, so site selection and the separation of value pools across sites are barely exercised; testing them requires a placement rule that spreads drones across sites.

Incentives over time. The guarantees hold per task and per capacity period. Over several periods a provider could try to influence the value pools, and over several tasks it could trade one task for another (Appendix A); a fully dynamic incentive-compatible mechanism remains open. The fairness auction performed best among the task-auction formats, but its incentive properties are not analysed here.

The upper bound. The offline bound is a linear relaxation that assumes users stay in place during a task, so the achievable optimum may be lower than the bound and the 71–86% share reached by the market is conservative. Finally, the two published approaches were adapted to this setting through their pricing, selection and payment rules, not reimplemented in full, so the comparison does not establish superiority over the complete originals, and the placement period of 100 s is a reasonable rather than an optimal default (60 s performed slightly better).

8 Conclusion

A UAV incurs a cost for every minute it is airborne, and per-task prices cannot recover that cost when several providers compete. It was proved that per-task price competition among deployed drones cannot fund their airtime without giving up efficient allocation or users' surplus, and that, between identical providers, it deters competing deployments, and losses of up to \$243 per hour were observed in simulation. A two-part market was proposed that remunerates availability in a periodic capacity auction, in which offers are compared by cost per unit of service and paid threshold prices, and that prices tasks in a second-price auction, with drones quoting marginal cost and edge servers quoting full cost. Truthful bidding was shown to be a dominant strategy in each stage taken separately, capacity payments were shown never to exceed the value at stake, and the per-task auction was shown to reach the allocation of an informed central scheduler.

In 3,950 simulations, the mechanism kept providers profitable at every fleet size evaluated, an empirical finding rather than a guarantee, since airtime outside booked periods remains unpaid (Proposition 6), and yielded \$113–215 per hour more welfare than full-cost pricing with 20 drones across two scenarios; when the fleet size was not known in advance, its average welfare shortfall was about half that of full-cost pricing and smaller than that of adaptations of two published approaches, one of which leaves

idle airtime unpaid and the other buys capacity without a value test. A provider that overstated its capacity costs always earned less under threshold payments, but up to \$19 per hour more under pay-as-bid payments. When every design is given its optimal fleet, full-cost pricing and the hover-cost design remain 2–4% more efficient; the contribution of the two-part mechanism is therefore sustainability and robustness rather than peak efficiency. Provider profitability does not by itself make the market financially sustainable: the capacity payments must be funded by users or by a sponsor, and user funding can leave users worse off than without drones. Future work should evaluate the mechanism on measured demand traces and larger service areas, introduce a reserve price or a cap to limit the information rent borne by users, and develop incentive guarantees that hold across periods.

Appendix A Numerical Verification and a Counterexample

Numerical checks. Each result of Section 5 is tested against the simulator's own allocation and payment functions in its test suite:

- Theorem 1: 3,000 random single-task instances with up to five bidders and a random reserve, against about 170,000 random and near-threshold misreports; no profitable deviation and no negative truthful utility.
- Proposition 2: the auction and the central scheduler, run on the same seed, made the same placement and success decision for 99.98% of 13,158 tasks; the remainder stem from exact cost ties between identical drones of different providers.
- Theorem 2: 2,000 random sites with up to six offers, against about 84,000 misreports; Theorem 3: 2,000 random sites with random weights in $[0.1, 1]$, against about 85,000 misreports. Both confirm truthfulness, non-negative utility, the budget bound and the prefix structure.
- Proposition 5: the budget bound on 2,000 random sites, and the manipulation example.
- Proposition 7: for 800 random instances of the LP with welfare weights, the solver's dual prices of all processor, uplink and downlink constraints were used to compute each task's opportunity cost (11); none of 16,496 served task–server pairs had a net value below its opportunity cost, and no task priced out on every server was served.
- Proposition 1: the identity in full simulations, and equal welfare under first- and second-price payments for the same allocation.
- Theorem 4: no simulated policy exceeded the bound in any run. For 3,000 small random workloads (five to seven tasks, two servers; arrivals within 2 s, spread over 25 s, or clustered around a window boundary), every assignment of tasks to servers was simulated with processor-shared links and first-in, first-out processing; in all 6,000 comparisons (task count and value) the LP bound was at least the exact optimum, and capacity was binding in 32% of them.

Truthfulness fails across tasks. Provider A has one server with a processing time of 1 s per task; provider B has a server with a higher marginal cost; the cloud

can meet neither deadline. Task 1 arrives at $t = 0$ with deadline 1 s; A's cost is κ, B's is $\kappa + \delta$. Task 2 arrives at $t = 0.1$ with deadline 1 s and value v_2, and only A covers its user. If A bids truthfully on task 1, it wins and is paid $\kappa + \delta$, but its server is busy until $t = 1$, so it cannot serve task 2 in time: its total is δ. If it bids above $\kappa + \delta$ on task 1, B wins task 1, and A, as the only bidder on task 2, is paid the reserve v_2: its total is $v_2 - \kappa$. For $v_2 - \kappa > \delta$ (e.g. $\kappa = 1$, $\delta = 0.5$, $v_2 = 4$) the deviation is profitable.

Appendix B Reproducibility

All experiments were run on the extended simulator described in Section 3. Each of the 57 experiment batches records its command, swept and fixed settings, and the simulator code version; a batch runner reproduces all of them, and an analysis script computes the reported statistics and figures from the per-run results.

References

[1] Qiu, H., Zhu, K., Luong, N.C., Yi, C., Niyato, D., Kim, D.I.: Applications of auction and mechanism design in edge computing: A survey. IEEE Transactions on Cognitive Communications and Networking **8**(2), 1034–1058 (2022)

[2] Jeong, S., Simeone, O., Kang, J.: Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning. IEEE Transactions on Vehicular Technology **67**(3), 2049–2063 (2018)

[3] Zhou, F., Hu, R.Q., Li, Z., Wang, Y.: Mobile edge computing in unmanned aerial vehicle networks. IEEE Wireless Communications **27**(1), 140–146 (2020)

[4] Mozaffari, M., Saad, W., Bennis, M., Nam, Y.-H., Debbah, M.: A tutorial on UAVs for wireless networks: Applications, challenges, and open problems. IEEE Communications Surveys & Tutorials **21**(3), 2334–2360 (2019)

[5] Xia, X., Fattah, S.M.M., Babar, M.A.: A survey on UAV-enabled edge computing: Resource management perspective. ACM Computing Surveys (2024) https://doi.org/10.1145/3626566

[6] Zeng, Y., Xu, J., Zhang, R.: Energy minimization for wireless communication with rotary-wing UAV. IEEE Transactions on Wireless Communications **18**(4), 2329–2345 (2019)

[7] Sun, W., Liu, J., Yue, Y., Zhang, H.: Double auction-based resource allocation for mobile edge computing in industrial internet of things. IEEE Transactions on Industrial Informatics **14**(10), 4692–4701 (2018)

[8] Su, Y., Fan, W., Liu, Y., Wu, F.: A truthful combinatorial auction mechanism towards mobile edge computing in industrial internet of things. IEEE Transactions on Cloud Computing **11**(2), 1678–1691 (2023)

[9] Wang, X., Ma, Y., Qin, Z., Xu, J., Zhao, J., Zeng, R., Song, Y., He, Q.: Truthful reverse auction-based incentive mechanisms for task offloading in mobile edge computing. Computer Networks **273**, 111767 (2025) https://doi.org/10.1016/j.comnet.2025.111767

[10] Tan, L., Lu, Z., Gu, S., Wang, Y., Song, Z., Cheng, P.: A group-selling auction for deadline-constrained computation collaborative offloading in multi-UAV edge networks. Ad Hoc Networks **194**, 104394 (2027) https://doi.org/10.1016/j.adhoc.2026.104394

[11] Xu, S., Wang, Y., Li, Q., Yao, H., Wang, Z., Mai, T.: Task offloading for space–air–ground integrated network based on iterative double auction algorithm. IEEE Transactions on Vehicular Technology (2026) https://doi.org/10.1109/TVT.2026.3728136 . Early access

[12] Yamansavascilar, B., Ozgovde, A., Ersoy, C.: AirCompSim: A discrete event simulator for air computing. In: Proc. IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6 (2025)

[13] Liu, J., Shi, Y., Fadlullah, Z.M., Kato, N.: Space-air-ground integrated network: A survey. IEEE Communications Surveys & Tutorials **20**(4), 2714–2741 (2018)

[14] Zeng, Y., Zhang, R., Lim, T.J.: Wireless communications with unmanned aerial vehicles: Opportunities and challenges. IEEE Communications Magazine **54**(5), 36–42 (2016)

[15] Al-Hourani, A., Kandeepan, S., Lardner, S.: Optimal LAP altitude for maximum coverage. IEEE Wireless Communications Letters **3**(6), 569–572 (2014)

[16] Jin, A., Song, W., Zhuang, W.: Auction-based resource allocation for sharing cloudlets in mobile cloud computing. IEEE Transactions on Emerging Topics in Computing **6**(1), 45–57 (2018)

[17] Huang, J., Li, S., Yang, L., Si, J., Ma, X., Wang, S.: Multiparticipant double auction for resource allocation and pricing in edge computing. IEEE Internet of Things Journal **11**(8), 14007–14016 (2024)

[18] Wang, X., Wang, X., Wang, C., Zeng, R., Ma, L., He, Q., Huang, M.: Truthful online combinatorial auction-based mechanisms for task offloading in mobile edge computing. IEEE Transactions on Mobile Computing **24**(7), 6488–6502 (2025)

[19] Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences **55**(1), 119–139 (1997)

[20] Kelly, F.: Charging and rate control for elastic traffic. European Transactions on Telecommunications **8**(1), 33–37 (1997)

[21] Iosifidis, G., Gao, L., Huang, J., Tassiulas, L.: A double-auction mechanism for mobile data-offloading markets. IEEE/ACM Transactions on Networking **23**(5), 1634–1647 (2015)

[22] Johari, R., Tsitsiklis, J.N.: Efficiency loss in a network resource allocation game. Mathematics of Operations Research **29**(3), 407–435 (2004)

[23] Myerson, R.B., Satterthwaite, M.A.: Efficient mechanisms for bilateral trading. Journal of Economic Theory **29**(2), 265–281 (1983)

[24] McAfee, R.P.: A dominant strategy double auction. Journal of Economic Theory **56**(2), 434–450 (1992)

[25] Liu, X., Liu, J., Zhu, C.: Truthful two-stage auctions for task offloading in the cooperation of UAVs and HAPs. Alexandria Engineering Journal **144**, 143–157 (2026) https://doi.org/10.1016/j.aej.2026.04.041

[26] Lin, L., Jiang, H., Xiong, J., Zhao, J., Li, P., Li, X., Lin, L., Zeng, D.: Fair and efficient resource allocation in UAV-enhanced edge computing: A dual-layer auction approach. IEEE Transactions on Mobile Computing **25**(11), 21408–21425 (2026) https://doi.org/10.1109/TMC.2026.3715203

[27] Vickrey, W.: Counterspeculation, auctions, and competitive sealed tenders. The Journal of Finance **16**(1), 8–37 (1961)

[28] Su, Y., Fan, W., Liu, Y., Wu, F.: Game-Based Pricing and Task Offloading in Mobile Edge Computing Enabled Edge-Cloud Systems. arXiv:2101.05628 (2021)

[29] Asheralieva, A., Niyato, D.: Hierarchical game-theoretic and reinforcement learning framework for computational offloading in UAV-enabled mobile edge computing networks with multiple service providers. IEEE Internet of Things Journal (2019) https://doi.org/10.1109/JIOT.2019.2923702

[30] Asheralieva, A., Niyato, D.: Distributed dynamic resource management and pricing in the IoT systems with blockchain-as-a-service and UAV-enabled mobile edge computing. IEEE Internet of Things Journal **7**(3), 1974–1993 (2020)

[31] Apostolopoulos, P.A., Fragkos, G., Tsiropoulou, E.E., Papavassiliou, S.: Data offloading in UAV-assisted multi-access edge computing systems under resource uncertainty. IEEE Transactions on Mobile Computing **22**(1), 175–190 (2023)

[32] Zhou, R., *et al.*: Incentive mechanisms for online task offloading with privacy-preserving in UAV-assisted mobile edge computing. IEEE/ACM Transactions on Networking (2024) https://doi.org/10.1109/TNET.2024.3364141

[33] Liwang, M., Gao, Z., Wang, X.: Let's trade in the future! A futures-enabled fast resource trading mechanism in edge computing-assisted UAV networks. IEEE

Journal on Selected Areas in Communications **39**(11), 3252–3270 (2021)

[34] Cheng, Z., Liwang, M., Xia, X., Min, M., Wang, X., Du, X.: Auction-promoted trading for multiple federated learning services in UAV-aided networks. IEEE Transactions on Vehicular Technology **71**(10), 10960–10974 (2022)

[35] Luong, N.C., Sui, Z., Le, D.V., Cao, J., Ma, B., Hai, N.D., Zhang, R., Quang, V.V., Niyato, D., Feng, S.: Incentive Mechanism Design for Resource Management in Satellite Networks: A Comprehensive Survey. arXiv:2601.03757 (2026)

[36] Cramton, P., Stoft, S.: A capacity market that makes sense. The Electricity Journal **18**(7), 43–54 (2005)

[37] Cramton, P., Ockenfels, A., Stoft, S.: Capacity market fundamentals. Economics of Energy & Environmental Policy **2**(2), 27–46 (2013)

[38] Myerson, R.B.: Optimal auction design. Mathematics of Operations Research **6**(1), 58–73 (1981)

[39] Shi, W., Zhang, L., Wu, C., Li, Z., Lau, F.C.M.: An online auction framework for dynamic resource provisioning in cloud computing. In: Proceedings of ACM SIGMETRICS (2014)

[40] Zhang, D., Tan, L., Ren, J., Awad, M.K., Zhang, S., Zhang, Y., Wan, P.-J.: Near-optimal and truthful online auction for computation offloading in green edge-computing systems. IEEE Transactions on Mobile Computing **19**(4), 880–893 (2020)

[41] Buchbinder, N., Naor, J.: The design of competitive online algorithms via a primal–dual approach. Foundations and Trends in Theoretical Computer Science **3**(2–3), 93–263 (2009)

[42] Talluri, K.T., Ryzin, G.J.: The Theory and Practice of Revenue Management. Springer, New York, NY, USA (2004)

[43] Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A.F., Buyya, R.: CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience **41**(1), 23–50 (2011)

[44] Gupta, H., Dastjerdi, A.V., Ghosh, S.K., Buyya, R.: iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments. Software: Practice and Experience **47**(9), 1275–1296 (2017)

[45] Mahmud, R., Pallewatta, S., Goudarzi, M., Buyya, R.: iFogSim2: An extended iFogSim simulator for mobility, clustering, and microservice management in edge

and fog computing environments. Journal of Systems and Software **190**, 111351 (2022) https://doi.org/10.1016/j.jss.2022.111351

[46] Sonmez, C., Ozgovde, A., Ersoy, C.: EdgeCloudSim: An environment for performance evaluation of edge computing systems. Transactions on Emerging Telecommunications Technologies **29**(11), 3493 (2018)

[47] Akgül, Ö.U., Mao, W., Cho, B., Xiao, Y.: VFogSim: A data-driven platform for simulating vehicular fog computing environment. IEEE Systems Journal **17**(3), 5002–5013 (2023) https://doi.org/10.1109/JSYST.2023.3286329

[48] Baidya, S., Shaikh, Z., Levorato, M.: FlyNetSim: An open source synchronized UAV network simulator based on ns-3 and ArduPilot. In: Proc. 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), pp. 37–45 (2018). https://doi.org/10.1145/3242102.3242118

[49] Shah, S., Dey, D., Lovett, C., Kapoor, A.: AirSim: High-fidelity visual and physical simulation for autonomous vehicles. In: Field and Service Robotics, pp. 621–635. Springer, Cham (2018)

[50] Yamansavascilar, B., Ozgovde, A., Ersoy, C.: Air computing: A survey on a new generation computation paradigm. Computer Networks **251**, 110653 (2024)

[51] Wei, Z., Huang, C., Li, B., Zhao, Y., Cheng, X., Yang, L., Zhang, R.: AirFogSim: A Light-Weight and Modular Simulator for UAV-Integrated Vehicular Fog Computing. arXiv:2409.02518 (2024)

[52] Wei, Z., Li, B., Zhang, R.: AirFogSim: A python package for benchmarking collaborative intelligence in low-altitude vehicular fog computing. Journal of Open Source Software **10**(111), 8267 (2025) https://doi.org/10.21105/joss.08267