

Adaptive Decentralized Queue Disclosure for Impatient Tenants in Edge and Non-terrestrial Systems

Anthony Kiggundu*, Bin Han[†] and Hans D. Schotten*[†]

*German Research Center for Artificial Intelligence (DFKI), Germany

[†]RPTU University of Kaiserslautern-Landau, Germany

Abstract—We study how queue-state information disclosures affect impatient tenants in multi-tenant edge systems. We propose an information-bulletin strategy in which each queue periodically broadcasts two Markov models. One is a model of steady-state service-rate behavior and the other a model of the queue length inter-change times. Tenants autonomously decide to renege or jockey based on this information. The queues observe tenant responses and adapt service rates via a learned, rule-based predictive policy designed for decentralized, partially-observed, and time-varying environments. We compare this decentralized, information-driven policy to the classical, centralized MDP hedging-point policy for $M/M/2$ systems. Numerical experiments quantify the tradeoffs in average delay, impatience and robustness to stale information. Results show that when full, instantaneous state information and stationarity hold, the hedging-point policy yields less impatience but this diminishes as information becomes partial or stale. The rule-based predictive policy on the other hand is more robust to staleness in dispatched information, making it conducive for conditions typical of edge cloud and non-terrestrial deployments.

Index Terms—6G, Queuing theory, Jockeying and Reneging, Behavioral modeling, Performance evaluation

I. INTRODUCTION

In heterogeneous 5G/6G deployments, multi-tenant resource sharing will be driven not only by static provisioning but increasingly by tenants' autonomous decisions (e.g., whether to offload a task to a remote queue or process it locally). In queuing systems, disclosures of queue state (such as queue lengths, waiting-time estimates or service statistics) can materially alter tenants' behavior and induce competition for resources via jockeying and reneging [1], [2], [3]. Queueing control in multi-server systems can be cast either as a centralized optimization problem or as a decentralized behavioral problem where tenants make local decisions using locally available information. [4] generalized the centralized optimal joint routing, service and jockeying policy to have a hedging-point structure. In contrast, modern Multi-access Edge Computing (MEC) and Non Terrestrial Networks (NTN) settings are decentralized, suffer partial and stale status broadcasts, and exhibit time-varying channels and mobility. In such environments it is unrealistic to assume a single central controller with instantaneous, global state. Moreover, existing disclosure rules and heuristics often developed for static or lightly mobile settings do not, however, answer three fundamental questions that are central

to decentralized control: *what* state information should be shared, *how much* it should be represented, and *how frequently* updates should be dispatched [5], [6], [7]. These questions are especially salient in self-organizing 6G settings where device heterogeneity, strict QoS requirements, and multi-slice subscriptions complicate tenants' preferences and render one-size-fits-all disclosure policies ineffective [8], [9].

We propose and evaluate an *information bulletin* framework in which queues periodically broadcast two Markov models: (i) a service-rate steady-state model (captures processing capability statistics) and (ii) an inter-change time model (captures dynamics of queue length changes) [10]. Upon receiving a bulletin (possibly stale, depending on dispatch interval), a tenant decides either to remain, jockey to another queue, or renege and process locally; these micro decisions create observable feedback that queues use to update and learn service-rate policies. Our objective is to minimize a composite measure of system performance that penalizes mean delay, jockeying events and reneging (also referred to as tenant impatience). We show that the resulting optimization formulation is nonconvex and analytically intractable.

Rather than relying on ad-hoc heuristics [2] for to optimize the system, we design a rule-based *predictive* policy that (a) estimates service-rate vectors from the two broadcast models, (b) prescribes when and what to broadcast, and (c) adaptively reconfigures queue service rates to regulate impatience. We validate the approach through extensive numerical experiments and sensitivity studies. Our main contributions are:

- 1) We introduce the *information bulletin* concept for multi-tenant queues and formalize two Markov descriptors (service-rate distribution and inter-change time) as dispatchable state summaries suited for resource-constrained control channels.
- 2) We derive closed-form expressions for jockeying and reneging probabilities under these descriptors, and formulate the joint impatience-minimization problem that optimally trades off delay, jockeying and reneging. We show this optimization is analytically intractable.
- 3) We propose a practical, rule-based predictive policy that learns service-rate vectors from tenant responses and adapts the service rates online.
- 4) Our extensive numerical evaluations quantify the value of different bulletin models and dispatch intervals.

We also demonstrate robustness of the learned policy across heterogeneous workloads and compare the decentralized, information-driven predictive policy with the hedging-point MDP policy [4].

In Section II, we characterize for the information models and the impatience arising from the dispatched information models. We also formulate for a service rate optimization problem and describe our empirical experiments. We then document the numerical evaluations of the information models on the impatience in Section III. We conclude with some open questions and some discussions around possible future work in Section IV.

II. SYSTEM MODELING

We assumed an $M/M/2$ setup, that is queues i and j . Either queue always processes jobs faster than the other. New arrival follow a Poisson distribution with rates $\lambda = \lambda_i + \lambda_j$. The expressions $\mu_i = \frac{\lambda + \delta\lambda}{2}, \mu_j = \frac{\lambda - \delta\lambda}{2}$ ensure that the rates μ_i and μ_j at which these requests get served vary following a First Come First Server (FCFS) service discipline. Each queue dispatches information about its status to tenants at intervals r in seconds introducing some staleness. We characterize for the effect arising from this information staleness using a drift sensitivity η .

A. Markov Models of the Queue Status Information

Our behavioral model is built around two types of information, the Markovian model of the service rates of the queues and the Markovian model of the how the queues change in size, i.e. Inter-Changing Time Distribution (ICD). A round-robin selection technique decides which information model to dispatch during an iteration.

1) *Markov Model of the Service rates*: Fundamentally, setting high service rates for either queues i or j should lead to the queue shrinking in size faster than when the rates are low which leads to longer queues. The heterogeneity in the service rates also introduces state dependent drifts to yield a scenario of fast and slow service. These interstate visits can result in some bursts in the queue sizes over time. In equilibrium however, the size of any of these $M/M/1$ queues i or j follows a birth-death Continuous Time Markov Chain (CTMC) with K states and rates $\{\mu_i\}_{i=1}^K, \{\mu_j\}_{j=1}^K$ respectively. We let X and Y denote these two stationary service rate distributions over $\{\mu_1 < \dots < \mu_K\}$ with steady state probabilities $\pi_i^X = Pr\{\mu = \mu_i\}$ and $\pi_j^Y = Pr\{\mu = \mu_j\}$. Then the effective service rates as averaged over the entire Markov chain are defined by (1).

$$\bar{\mu}_X = \sum_{i=1}^K \pi_i^X \mu_i, \quad \bar{\mu}_Y = \sum_{j=1}^K \pi_j^Y \mu_j. \quad (1)$$

2) *Markov Model of the Queue Length Dynamics - Inter-Changing Time Distribution (ICD)*: This measure basically quantifies how frequently transitions occur in a queuing system and can be used to analyze the stability or efficiency of a queuing system. If the queue lengths changes often, then intuitively that queue should be the better one. Fewer perturbations in

the system on other hand suggest stability with measurably equal departures and admissions or lower throughput. But low interchange time in the sizes of the queues can also imply underlying performance bottlenecks. For the queue therefore in state n , when $n = 0$ (empty) only an arrival changes the state while when $n \geq 1$ an arrival or departure event can occur to end the queue in state $\lambda_i + \mu_i$. The Markovian model that encapsulates how frequently the queue length changes in steady state given these events is defined by (2).

$$R_i = \sum_{n=0}^{\infty} \pi_{i,n} (\lambda_i + \mu_i \cdot 1_{n \geq 1}). \quad (2)$$

where $1_{n \geq 1}$ is an indicator function for measurability around $n \geq 1$ since a departure event only occurs when $n \geq 1$. However, (2) can be simplified as

$$\sum_{n=0}^{\infty} \pi_{i,n} (\lambda_i + \mu_i \cdot 1_{n \geq 1}) = \pi_{i,0} \lambda_i + \sum_{n=1}^{\infty} \pi_{i,n} (\lambda_i + \mu_i), \quad (3)$$

where $\pi_{i,0} = 1 - \rho_i$ and $\sum_{n=1}^{\infty} \pi_{i,n} = \rho_i$ and eventually:

$$R_i = (1 - \rho_i) \lambda_i + \rho_i (\lambda_i + \mu_i) = \lambda_i + \rho_i \mu_i. \quad (4)$$

Since $\rho_i \mu_i = \lambda_i$,

$$R_i = \lambda_i + \lambda_i = 2\lambda_i \quad (5)$$

is a count of the number of changes introduced by these events. We can therefore express for the expected time between successive changes in the sizes of the queue i or j using (6).

$$T_{ICD}^i = \frac{1}{R_i} = \frac{1}{2\lambda_i} \quad (6)$$

3) *First-Order Stochastic Dominance (FSD)*: To find the better queue given its service rate distribution, we need to characterize for the CDFs $F_X(\mu_k)$ and $F_Y(\mu_k)$ ($k = 1, \dots, K$) of X and Y from the steady state probabilities π_i^X, π_j^Y above using 7.

$$F_X(\mu_k) = \sum_{i=1}^K \pi_i^X, \quad F_Y(\mu_k) = \sum_{j=1}^K \pi_j^Y \quad (7)$$

Then, according to [11], if $P[X > x] \geq P[Y > x] \forall x \in R$ then X is said to first order stochastically dominate Y and the conditions defined by (8) must hold.

$$F_X(\mu_k) \leq F_Y(\mu_k) \quad \forall k = 1, \dots, K \\ \exists k \quad F_X(\mu_k) < F_Y(\mu_k) \quad (8)$$

Therefore, the rationale to abandon the system or switch queues given the comparison of the the service rates distributions is defined around the stochastic dominance of one distribution against the other as conditioned by (9).

$$\sum_{i=1}^K \pi_i^X \leq \sum_{i=1}^K \pi_i^Y, \quad \forall k, \quad \exists k : F_X(\mu_k) < F_Y(\mu_k). \quad (9)$$

where $k = 1, \dots, K$

B. Reneging Behavior

The decision to abandon the queue is premised on the condition that the local processing T_{local} (deterministic and no waiting time involved) is less than the estimated waiting time in either queues, such that jockeying to the alternative queue is not an option. However, FSD is a comparison between two distributions of the service rate vectors of queues i, j . From

(1) and (7) therefore, queue i FSD-dominates queue j when $F_i(\bar{\mu}_i) \geq F_j(\bar{\mu}_j) \quad \forall t \geq 0$. Given the better service rate vector, we need to redefine its corresponding CDF in terms of the waiting times distribution $W_i \sim f_{w_i}(t)$ or $W_j \sim f_{w_j}(t)$ in steady-state $\pi_j(n)$ before direct comparison to T_{local} is made. In equilibrium, for a buffered request in queue i at position ℓ , we express for the expected remaining time until that request gets served using (10).

$$\mathbb{E}[W_i|\ell] = \sum_{i=0}^{\ell} \frac{1}{\mu_i} = \frac{\ell}{\mu_i} = \sum_{i=0}^{\ell} \pi_{\ell} \frac{\ell}{\mu_i} \quad (10)$$

And the CDF of this remaining time generally obeys the Erlang distribution, $W_i|l \sim \text{Erlang}(l, \mu_i)$ whose definition takes the form of (11).

$$F_{W_i|\ell}(t) = \mathbb{P}(W_i \leq t | \ell) = 1 - \sum_{v=0}^{\ell-1} \frac{[(\mu_i - \lambda_i)t]^v}{v!} e^{-(\mu_i - \lambda_i)t} \quad (11)$$

But the renegeing tenant in queue i first weighs the option whether to jockey and land at position k in the other queue j . The expected sojourn time of a jockeyed request to queue j is also generalized to an Erlang distribution of order k ($W_j \sim \text{Erlang}(k, 2\mu_j)$) and its CDF is expressed for by (12).

$$F_{W_j|k}(t) = \mathbb{P}(W_j \leq t | k) = 1 - \sum_{v=0}^{k-1} \frac{[(2\mu_j - \lambda_j)t]^v}{v!} e^{-(2\mu_j - \lambda_j)t} \quad (12)$$

Then taking the mean over all n requests under stationary conditions π_n we redefine the CDFs of the expected waiting times of either queues as:

$$\begin{aligned} F_{W_i}(t) &= \sum_{\ell=0}^{\infty} \pi(n) \cdot \mathbb{P}(W_i \leq t | \ell) \\ F_{W_j}(t) &= \sum_{n=0}^{\infty} \pi(n) \cdot \mathbb{P}(W_j \leq t | n) \end{aligned} \quad (13)$$

Therefore based on the comparison $F_{W_i}(t) \geq F_{W_j}(t) \quad \forall t \geq 0$ and assuming queue i is the better queue at pose ℓ , (14) characterizes the decision to renege to local processing as a final comparison to the local processing T_{local} otherwise jockey to the better queue.

$$P_{reng}^{\text{FSD}}(\ell) = \mathbb{P}(W_i > T_{local} | \ell) = 1 - F_{W_i|\ell}(T_{local}) \quad (14)$$

Factoring in the interval r at which the status is dispatched and the staleness factor η , the renegeing probability P_{reng}^{FSD} given the FSD of the better of two queues in comparison to local processing is then:

$$\begin{aligned} \mathbb{P}_{reng}^{\text{FSD}}(\ell) &= 1 - \left[1 - \sum_{v=0}^{\ell-1} \frac{[(\mu_i - \lambda_i) \Delta]^v}{v!} e^{-(\mu_i - \lambda_i) \Delta} \right] \\ &= \sum_{v=0}^{\ell-1} \frac{[(\mu_i - \lambda_i) \Delta]^v}{v!} e^{-(\mu_i - \lambda_i) \Delta}, \end{aligned} \quad (15)$$

where $\Delta = T_{local} - \eta r$, r as the dispatch interval, $\eta \in [0, 1]$ denoting how degraded the status is, i.e. perfect state versus staleness.

And the rate at which requests that renege from a particular queue when consuming from the FSD information source is

calculated using (17).

$$\begin{aligned} R_{reng}^{\text{FSD}}(\ell) &= \lambda_i \sum_{\ell=0}^{\infty} \pi_{\ell} P_{reng}^{\text{FSD}}(\ell) \\ &= \lambda_i \sum_{\ell=0}^{\infty} \pi_{\ell} \sum_{v=0}^{\ell-1} \frac{[\mu_i (T_{local} - \eta r)]^v}{v!} e^{-\mu_i (T_{local} - \eta r)} \end{aligned} \quad (16)$$

On the other hand, dispatching the interchanging times of the queue lengths implies the rationale to abandon the current queue i or j for local processing follows from a comparison of how often the sizes of both queues change. The comparison $T_{\text{ICD}}^i < T_{\text{ICD}}^j$ gives the queue whose length is changing often. This could imply more impatient tenants or more requests being processed. Assuming $T_{\text{ICD}}^i < T_{\text{ICD}}^j$ is true, then the tenant compares the remaining time at position ℓ in the current queue i defined by (10) to T_{local} . Such that, the probability $P_{reng}^{\text{ICD}}(\ell)$ to renege to local processing is similarly defined using (15). And the percentage R_{reng}^{ICD} of requests that renege from the queue given this kind of information is also defined using

$$R_{reng}^{\text{ICD}} = \lambda_i \sum_{n=1}^{\infty} \pi_n \cdot P_{reng}^{\text{ICD}}(n, \mu_i, (T_{local} - \eta r)) \quad (18)$$

C. Jockeying Behavior

From (5), it is known that the rate at which the sizes of either queues changes is characterized by the arrivals and departures. We take the case of independence of both events here and denote as M_i, M_j the number of such events in queue i, j respectively. Then a request switches from queues i to j given the comparison between the interchanging times T_{ICD}^i and T_{ICD}^j as defined by (6). The probability $P_{i \rightarrow j}^{\text{ICD}}$ therefore to jockey from queue $i \rightarrow j$ and vice-versa is defined around this difference in the number of events in either queues as:

$$P_{i \rightarrow j}^{\text{ICD}} = \sigma(d[M_i - M_j]) \quad (19)$$

where $\sigma = \frac{1}{1+e^{-x}}$ is a sigmoid function that tunes how steep the probability rises with the drift difference and $d > 0$ is a decision function that regulates the switching behavior given this difference.

From (5), we can redefine $M_i = 2\lambda_i e^{-\eta r}$ and $M_j = 2\lambda_j e^{-\eta r}$, then based on the difference $M_j - M_i = 2(\lambda_j - \lambda_i)e^{-\eta r}$ the jockeying probability is characterized for using (20)

$$\begin{aligned} P_{i \rightarrow j}^{\text{ICD}} &= \sigma(d[2\lambda_i e^{-\eta r} - 2\lambda_j e^{-\eta r}]) \\ &= \frac{1}{1 + e^{-2de^{-\eta r}(\lambda_i - \lambda_j)}} \end{aligned} \quad (20)$$

since $\lambda_i = \mu_i \rho_i, R_i = 2\lambda_i = 2\mu_i \rho_i$.

The jockeying rate within the system in this case is then defined for by:

$$R_{i \rightarrow j}^{\text{ICD}} = \lambda_i P_{i \rightarrow j}^{\text{ICD}} + \lambda_j P_{j \rightarrow i}^{\text{ICD}} \quad (21)$$

In the case of the FSD-based waiting time distribution, (11) characterizes for the CDF of the remaining waiting time $W_i|l$ at position l . Similarly, (12) is definitive of the expected waiting time when this request is migrated to queue j . The

decision to move from $i \rightarrow j$ is therefore pegged on the FSD comparative rule $F_{W_i}(t) \geq F_{W_j}(t) \quad \forall t$.

$$\mathbb{P}_{i \rightarrow j}^{\text{FSD}}(\ell) = \mathbb{P}\{W_{j,k} < W_{i,\ell}\} = \int_0^\infty f_{W_{j,k}}(t) [1 - F_{W_{i,\ell}}(t)] dt, \quad (22)$$

where $f_{W_{j,k}}(t) = \frac{d}{dt} F_{W_{j,k}}(t)$ is the probability density function in j and $[1 - F_{i,\ell}(t)]$ corresponds to $\mathbb{P}(W_i > t)$

But because the states are dispatched every r seconds, the degradation in updates suffers a staleness. And the probability that the request is moved to from queue $i \rightarrow j$ is defined using (23)

$$\mathbb{P}_{i \rightarrow j}^{\text{FSD}}(\ell) = \int_0^\infty \frac{[\xi_j(t - \eta r)]^{k-1}}{(k-1)!} \xi_j e^{-\xi_j(t - \eta r)} \times \sum_{m=0}^{\ell-1} \frac{[\xi_i(t - \eta r)]^m}{m!} e^{-\xi_i(t - \eta r)} dt, \quad (23)$$

Here, $\xi_i = 2\mu_i - \lambda_i$ and $\xi_j = 2\mu_j - \lambda_j$ and (23) which can be expressed as:

$$\mathbb{P}_{i \rightarrow j}^{\text{FSD}}(\ell) = \sum_{k=0}^{\ell-1} \frac{\xi_j^\ell}{(\xi_i + \xi_j)^{\ell+k}} \frac{(\ell+k-1)!}{(\ell-1)! k!} \quad (24)$$

And in stable conditions, (25) characterizes for the rate at which requests are moved around in time.

$$R_{i \rightarrow j}^{\text{FSD}} = \lambda_i P_{i \rightarrow j}^{\text{FSD}} + \lambda_j P_{j \rightarrow i}^{\text{FSD}} \quad (25)$$

Given these information models, how can the queue adapt to the impatient tenant's behavior and learn to recalibrate its service rate vector $\bar{\mu}$ to minimize average delay and the overall impatience among the tenants. This translates into an optimization problem, where we can jointly choose the service-rate vector $\bar{\mu}$ in (1) that satisfies the Karush–Kuhn–Tucker (KKT) conditions. We limit the scope of the formulation to the ICD information (although the characterization for the FSD would adopt similar form) model and (26) is definitive of the corresponding optimization problem.

$$\min_{\mu_i, \mu_j} \tau [W_i(\mu_i) + W_j(\mu_j)] + \varphi [R_i^{\text{renew}}(\mu_i) + R_j^{\text{renew}}(\mu_j)] + \psi [R_{i \rightarrow j}^{\text{jockey}}(\mu_i, \mu_j) + R_{j \rightarrow i}^{\text{jockey}}(\mu_j, \mu_i)] \quad (26)$$

$$\text{s.t. } \mu_{i,\min} \leq \mu_i < \mu_{i,\max}, \quad \mu_i > \lambda_i,$$

$$\mu_{j,\min} \leq \mu_j < \mu_{j,\max}, \quad \mu_j > \lambda_j$$

where $W_i(\mu_i) = \frac{\rho_i}{\mu_i - \lambda_i}$

The corresponding optimization function to this problem is given by (27)

$$f(\mu_i, \mu_j) = \tau \left(\frac{\rho_i}{\mu_i - \lambda_i} + \frac{\rho_j}{\mu_j - \lambda_j} \right) + \varphi (\lambda_i R_{i,\text{renew}}^{\text{ICD}}(\mu_i) + \lambda_j R_{j,\text{renew}}^{\text{ICD}}(\mu_j)) + \psi (\lambda_i \sigma (d[2\lambda_i e^{-\eta r} - 2\lambda_j e^{-\eta r}]) + \lambda_j \sigma (d e^{-\eta r} [(\mu_i - \lambda_i) - (\mu_j - \lambda_j)])) \quad (27)$$

where in equilibrium π_2 denotes the first state in which both queues are occupied.

When there are $n = 0$ requests in the system, the independence between arrival and a departure events implies $\epsilon_{i \neq j} = \mu_i - \lambda_i$ since the two events happening simultaneously ends the queue in an empty state. We take the independence of events to simplify the third term in (27) as $z = d e^{-\eta r} [\epsilon_j - \epsilon_i]$, such

that $\sigma'(z) = \sigma(z)[1 - \sigma(z)]$. We then define the inequality constraints $g_{1,i}(\mu_i) = \mu_{i,\min} - \mu_i \leq 0$, $g_{2,i}(\mu_i) = \mu_i - \mu_{i,\max} \leq 0$, $g_{3,i}(\mu_i) = \lambda_i - \mu_i \leq 0$ for both i and j . Equation (28) expresses for the corresponding Lagrange Function with multipliers $\gamma_{m,i} \geq 0$ for each of $m = 3$ constraints $g_{m,i} \leq 0$ and $g_{m,j} \leq 0$.

$$\mathcal{L}(\mu_i, \mu_j, \{\gamma_{k,i}\}) = f(\mu_i, \mu_j) + \sum_{i=1}^2 [\gamma_{1,i}(\mu_{i,\min} - \mu_i) + \gamma_{2,i}(\mu_i - \mu_{i,\max}) + \gamma_{3,i}(\lambda_i - \mu_i)]. \quad (28)$$

In equilibrium, $\nabla_{\mu_i, \mu_j} \mathcal{L} = 0$, which implies that for both i and j , we express for the derivatives of each term in (27) as:

$$\frac{\partial f}{\partial \mu_i} - \gamma_{1,i} + \gamma_{2,i} + \gamma_{3,i} = 0 \quad (29)$$

$$\frac{\partial f}{\partial \mu_j} - \gamma_{1,j} + \gamma_{2,j} + \gamma_{3,j} = 0$$

Such that (30) and (31) characterize for the KKT stationarity conditions.

$$0 = -\frac{\lambda_i(2\mu_i - \lambda_i)}{\mu_i^2(\mu_i - \lambda_i)^2} + \varphi \lambda_i \sum_{v=0}^{\ell_i-1} \left[\frac{v}{\epsilon_i} - \Delta \right] \frac{(\epsilon_i \Delta)^v}{v!} e^{-\epsilon_i \Delta} + \psi d e^{-\eta r} \sigma'(z) (\lambda_i + \lambda_j) - \gamma_{1,i} + \gamma_{2,i} + \gamma_{3,i}, \quad (30)$$

and

$$0 = -\frac{\lambda_j(2\mu_j - \lambda_j)}{\mu_j^2(\mu_j - \lambda_j)^2} + \varphi \lambda_j \sum_{v=0}^{\ell_j-1} \left[\frac{v}{\epsilon_j} - \Delta \right] \frac{(\epsilon_j \Delta)^v}{v!} e^{-\epsilon_j \Delta} - \psi d e^{-\eta r} \sigma'(z) (\lambda_i + \lambda_j) - \gamma_{1,j} + \gamma_{2,j} + \gamma_{3,j}. \quad (31)$$

Essentially, from these formulations, we can randomly select an active-set of multipliers, resolve the nonlinear equations (30),(31) and perform both complementary slackness and feasibility tests for i, j using

$$\gamma_{1,i}(\mu_{i,\min} - \mu_i) = 0, \quad \gamma_{2,i}(\mu_i - \mu_{i,\max}) = 0, \quad \gamma_{3,i}(\lambda_i - \mu_i) = 0, \quad \gamma_{1,i} \geq 0, \gamma_{2,i} \geq 0, \gamma_{3,i} \geq 0 \quad (32)$$

Any $(\mu_i^*, \mu_j^*, \{\gamma_{j,i}^*\})$ satisfying all of these is a candidate local minimum. If the solution lies outside the bound defined by (32), we validate the KKT conditionality under defined constraints after a projection to boundaries. However, f must be jointly convex in (μ_i, μ_j) when we verify that its Hessian matrix H_{μ_i, μ_j} is positive semi-definite for all feasible μ_i, μ_j . This is however not the case since no closed-form solution for μ_i^*, μ_j^* exists. And to find this optimality, we numerically evaluate our rule-based policy as a solution to the KKT system under varying μ_i, μ_j .

D. Rule-based Queue Policy for Predictive Modeling

The rule-based policy is basically queue knowledge abstracted as a behavioral model of impatience. From the status information models dispatched and the corresponding tenant reactions evolves a predictive model that encapsulates the probability that a user makes the optimal decision (renege, jockey). Such that, the queue iteratively adjusts its service rates to minimize the impatience and overall delay in the system. And these performance measures characterize the utility function. To maximize this utility, the rule-based queue policy determines the next calibration in processing rates at a given

interval given input from the predictive model. Essentially, the output from the predictive model are used by the queue to generate action probabilities given the current state in that dispatch interval. And it each iteration, the reaction from the tenants is the input to the predictive model for improvement. We compare our rule-based policy to a hedging point that maximizes the expected discounted return as derived by [4]. The Markov Decision Process is solved numerically by value iteration on a suitably truncated state space. Here, jockeying is defined by monotone switching curves that partition the state space (based on the queue lengths) into regions where the queue serves its own queue or serves a jockeyed job. Essentially, a table of optimal actions (or of the switching curves) is precomputed and the lookup applied at runtime.

III. NUMERICAL RESULTS

We dispatch the information at defined intervals $r \in \{3, 5, 7, 9\}$ seconds. For each dispatch interval, we simulated over 300 runs for varying configuration pairs of μ_i, μ_j with different measures of the arrival rates $\lambda : 3 \leq \lambda \leq 17$.

Figure 1 is illustrative of when no server policy is embedded. Here, the chaotic and variability in the profile of the reneging and jockeying rates is evidence of instability. Even under this volatility, it is evident that the Markov model of the service rates yields less impatience in comparison to the Markov model of the changes in the queue length sizes. This is logical since with the interchanging times dispatches, requests receive a naive snapshot of the entire dynamics leading to overreacting. This indirectly yields jockeying back and forth to find the short-lived pool. On the other hand, the service rates information provides a direct mapping to how fast jobs are processed in a queue to control the impatience. Figure 2 shows the optimized policy from the predictive modeling where this impatience is more controllable. The frequency of dispatching appears to lead to optimality between the intervals 5 to 7 where the impatience is minimized and stability, particularly when requests get the service rates information. Figure 3 compares the hedging-point policy to the rule-based policy. Although the hedge-point jockeying policy here seems more stable, higher renege and jockeying rates are observed. The Markov-based policies on the other hand are more volatile, but potentially record lower rates especially in lower intervals. Increasing the dispatch interval tends to reduce rate volatility and brings all policies closer together, though the hedge-point policy remains smoother. Ideally, this evaluation reveals that to minimize the volatility and achieve predictable system behavior, the hedge-point policy seems more suitable. But for optimality in the service or lower rates, Markov-based policies may be preferable, especially at lower intervals.

Figure 4 illustrates the overall effect of the information model dispatches on the waiting time of the impatient tenant. When no policy exists in the system, requests that abandon or switch tend to wait for longer as observed from the median measures. The predictive model on the contrary encapsulates useful queue descriptor knowledge which yields optimal waiting times for tenants. The waiting times record lower medians

for both reneging and jockeying requests when we embed the policy but more optimality is observable when the Markov model of the service rates is dispatched.

Table I provides a quantitative summary of Figure 5. Here, the optimized policy is statistically robust and yields lower yet more consistent objective values (mean = 0.53) in comparison to no optimization at all (mean = 1.78). The effect of the intervals is evident from the distance between optimal and non-optimal endpoints as the interval increases (from 3s to 9s). The optimized trajectory in each interval achieves a lower objective value consistently as the landscape changes, finding better solutions, while the non-optimized path sometimes does not. The path is visibly more oriented towards the valleys or optimal regions in the surface unlike in the non optimal case. And the variability is smaller for optimized results (Std. Dev = 0.15) than for non-optimized (Std. Dev. = 0.97), indicating more reliability in performance.

TABLE I: Optimized vs Non-Optimized Objective Values and Statistical Summary

Interval	Opt. Obj.	Non-Opt. Obj.	Opt. (μ_1, μ_2)	Non-Opt. (μ_1, μ_2)
3s	0.39	1.62	(14.50, 15.00)	(4.50, 2.50)
5s	0.77	3.40	(7.00, 8.00)	(2.50, 0.50)
7s	0.749	0.84	(8.00, 8.50)	(6.50, 5.50)
9s	0.46	1.27	(9.50, 10.00)	(8.50, 6.50)
Mean	0.53	1.78	Avg. Impr.: 1.26	
Std Dev	0.15	0.97		
Min	0.39	0.84		
Max	0.77	3.40		

IV. CONCLUSION AND OUTLOOK

The role of system status information in shaping the decisions of impatient tenants is understudied, yet it is central to realizing data-driven, decentralized resource allocation in next-generation networks. In this work we introduced an *information-bulletin* framework based on two queue-descriptor models. We analyzed how the periodic dispatch of such queue state information affects tenant jockeying and reneging. We then developed a lightweight rule-based predictive policy that learns and adapts service rates from tenant responses under potentially stale information. Our numerical study compares a scenario where no policy (baseline) is embedded against the hedging-point MDP benchmark, and the bulletin-driven rule-based policy and quantifies tradeoffs in mean delay, jockeying and reneging rates. The results indicate that both informed policies substantially reduce tenant impatience relative to the no-policy baseline; the hedging-point policy appearing more suitable for stabilizing the system, while the rule-based policy is more robust to information staleness and the nonstationary service dynamics typical of edge cloud environments. Future work would include more information models that abstract the subscription costs and replacing the rule-based heuristics with reinforcement learning techniques and validate the approach on prototype MEC testbeds with realistic delay traces.

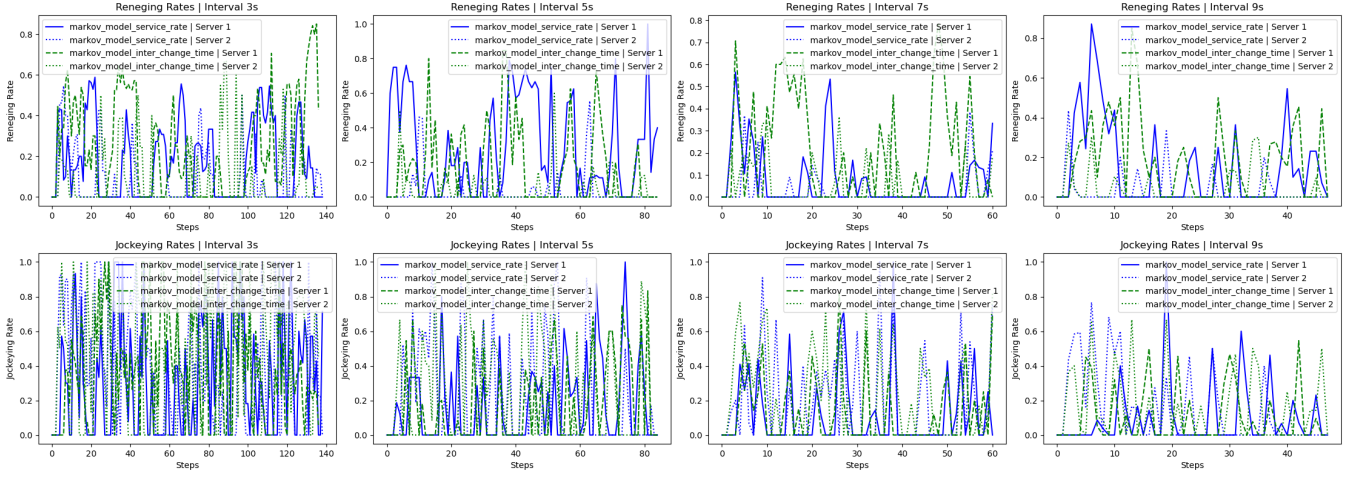


Fig. 1: When no policy is embedded, the resulting phenomena here is instability regardless of the Markov model of information dispatched. This volatility however as expected reduces when the dispatch intervals are increased.

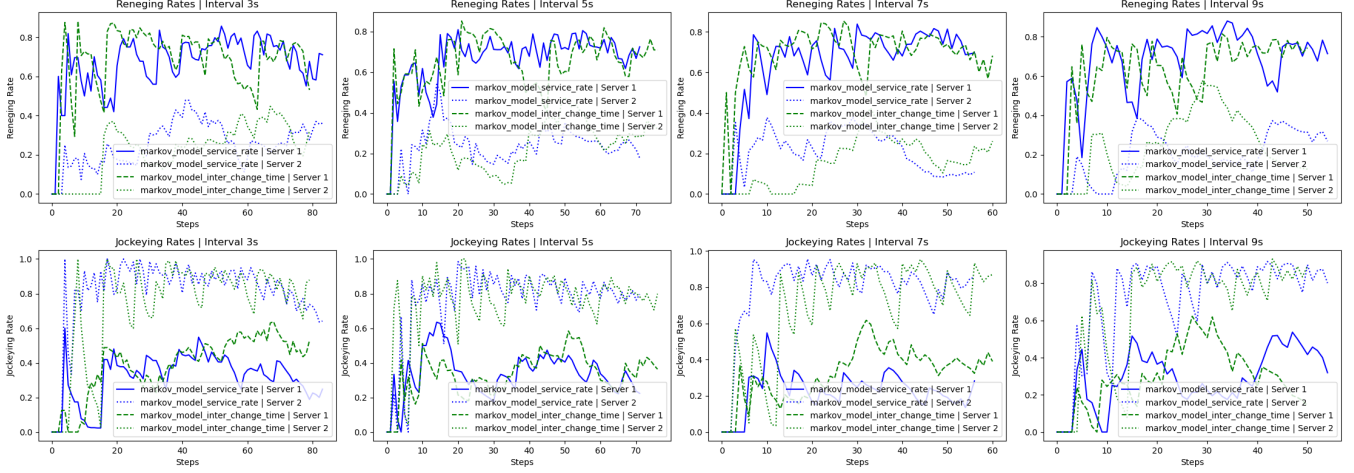


Fig. 2: The rule-based queue policy that learns the dispatching and service rates manages to regulate the impatience with the Markov model of the service rates proving more vital especially in keeping the jockeying minimal.

ACKNOWLEDGMENT

This work is supported in fully by the German Federal Ministry of Research, Technology and Space (BMFTR) within the Open6GHub project under grant numbers 16KISK003K and 16KISK004.

REFERENCES

- [1] A. Kiggundu, B. Han, D. Krummacker, and H. D. Schotten, "Chronicles of jockeying in queuing systems," *ArXiv*, vol. abs/2402.11061, 2024. [Online]. Available: <https://api.semanticscholar.org/CorpusID:267751329>
- [2] Y. Ouyang and D. Teneketzis, "Signaling for decentralized routing in a queueing network," *Annals of Operations Research*, vol. 317, no. 2, pp. 737–775, 2022.
- [3] A. Kiggundu, B. Han, D. Krummacker, and H. D. Schotten, "Resource allocation in mobile networks: A decision model of jockeying in queues," in *2024 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, 2024, pp. 311–316.
- [4] B. Lin, Y. Lin, and R. Bhatnagar, "Optimal policy for controlling two-server queueing systems with jockeying," *Journal of Systems Engineering and Electronics*, vol. 33, no. 1, pp. 144–155, 2022.
- [5] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proceedings of the 31st IEEE International Conference on Computer Communications (INFOCOM)*, Orlando, FL, USA, 2012, pp. 2731–2735.
- [6] T. Choudhury, G. Joshi, W. Wang, and S. Shakkottai, "Job dispatching policies for queueing systems with unknown service rates," ser. *MobiHoc '21*. Association for Computing Machinery, 2021, p. 181–190. [Online]. Available: <https://doi.org/10.1145/3466772.3467047>
- [7] R. Hassin and M. Haviv, "Equilibrium strategies and the value of information in a two line queueing system with threshold jockeying," *Communications in Statistics. Stochastic Models*, vol. 10, no. 2, pp. 415–435, 1994.
- [8] "3GPP TS 28.531 V16.14.1 (2023-01): Management and Orchestration; Provisioning," 3rd Generation Partnership Project (3GPP), Technical Specification TS 28.531, January 2023, version 16.14.1, Release 16. [Online]. Available: <https://www.3gpp.org/dynareport/28531.htm>
- [9] K. Gardner, J. Abdul Jaleel, A. Wickeham, and S. Doroudi, "Scalable load balancing in the presence of heterogeneous servers," *Performance Evaluation*, vol. 145, p. 102151, 10 2020.
- [10] S. R. Mahabhashyam and N. Gautam, "On queues with markov modulated service rates," *Queueing Systems*, vol. 51, pp. 89–113, 2005.
- [11] R. Torres, "Stochastic dominance," Discussion Paper, Tech. Rep., 1990.
- [12] A. Kiggundu, B. Han, and H. D. Schotten, "Information bulletin strategy in impatient queueing," *arXiv preprint arXiv:2508.04241*, 2025.

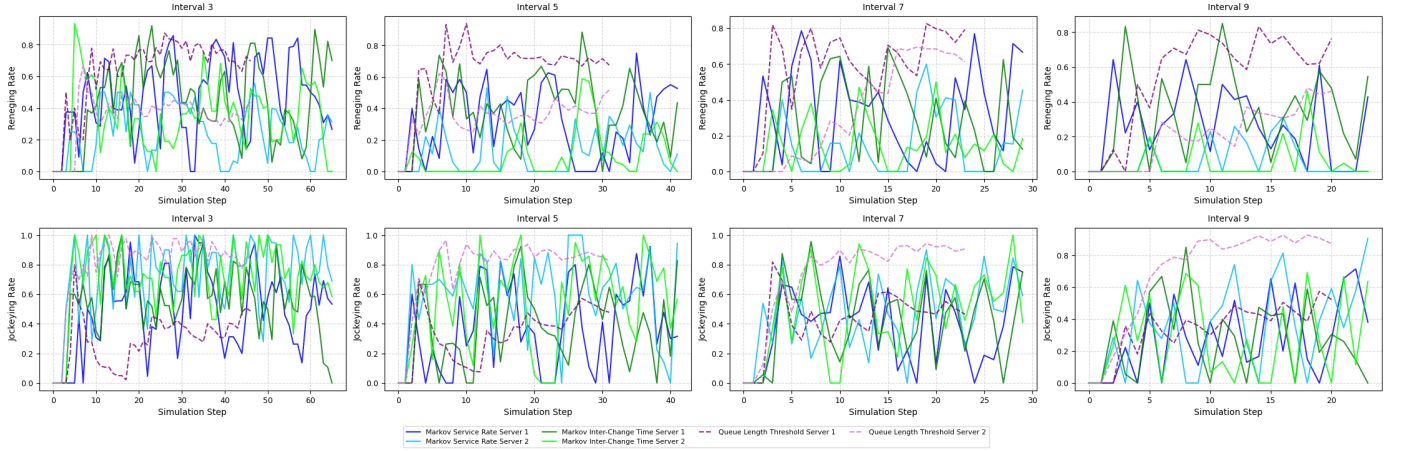


Fig. 3: Across all intervals, slightly lower and more stable renege rates are observed for both policies, though the hedge-point policy remains relatively higher. The jockeying rates for all policies begin to converge at higher dispatch intervals and show less volatility.

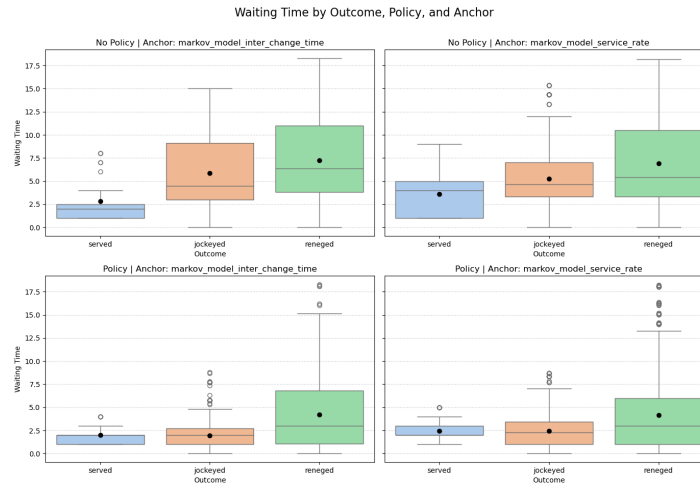


Fig. 4: The box plot showing the median waiting time when a policy is embedded versus otherwise for both Markov information models. The waiting time and spread here are high when no policy is embedded for both impatience behavior [12].

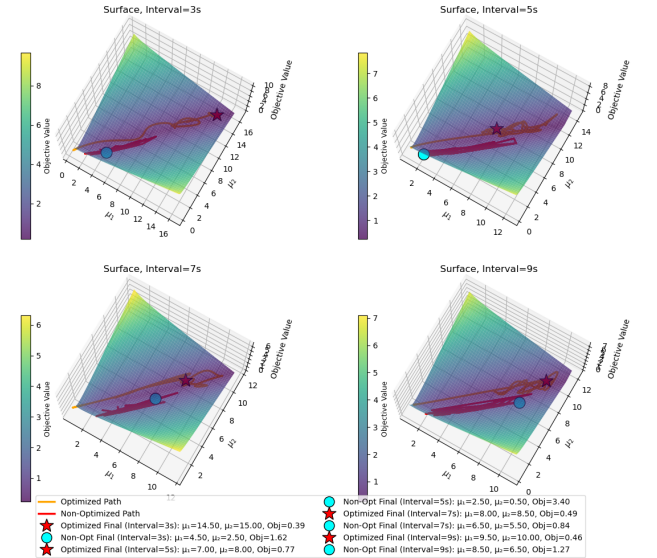


Fig. 5: Here, the rule-based policy leads to substantially better (lower) objective values in comparison to when no policy is embedded. Our policy finds the near-optimum despite changes in the interval and impatience. Absence of a control policy on the other hand is consistently suboptimal under the behavioral changes since it cannot adapt.