Supplementary Materials

EC.1. Further Discussion on the Limitations of Existing Approaches

In addition to the discussions in Section 2.1, we provide an additional overview of related problems in the literature and discuss why they cannot be used to solve the problem that we introduced in Section 2.

First, the work in (Johari et al. 2016, 2017) also studies how to maximize the average payoff for a system with unknown client types. (The notion of "type" in (Johari et al. 2016, 2017) is comparable to "class" in this paper, while "worker" is comparable to "client" in this paper.) However, there are a number of crucial differences. The main policies proposed in (Johari et al. 2016, 2017) assume that the system operator knows the "shadow prices," which captures the long-term congestion level of each server. This shadow price is the solution to an optimization problem that depends on a number of system parameters, such as the payoff vector of each type, the probability distribution of the types, the arrival rate of each type, and the expected number of tasks of each client. Thus, the model in (Johari et al. 2016, 2017) has to assume that all of these system parameters are known in advance.

In contrast, in our model, neither the payoff vector for each type (i.e., each "class" in this paper) nor the client arrival/departure statistics are known to the system. As a result, the theoretic policies in (Johari et al. 2016, 2017) cannot be used or easily adapted in our setting. Although (Johari et al. 2017) also provides a heuristic policy that uses queue length to replace the shadow price, no theoretical performance guarantee is proved. Moreover, this heuristic policy requires that the unused service of the servers can be queued and utilized later. Since our model does not allow such queueing of unused service, we cannot use this heuristic policy either. One could argue that, after the system has been operated in a stationary setting for a long time, the operator may eventually be able to learn the arrival/departure statistics and the class-dependent payoff vectors, in which case the theoretic policies in (Johari et al. 2016, 2017) could then be used (although acquiring such

knowledge would require highly non-trivial learning procedures for estimating both the clients' payoff vectors and their underlying classes.) However, this argument will become problematic when the system runs only for a limited amount of time. In contrast, in this paper, we will develop analytical results that hold for a finite time-horizon. As a result, the setting studied in this paper, as well as the adaptive algorithm that we will develop, will be useful both for the early phase of the system operation, and when the class composition or arrival/departure statistics experience recent changes. Finally, note that the work in (Massoulie and Xu 2016) also addresses uncertainty in both client features and client arrival rates. However, it still assumes the knowledge of the payoff vectors for all types in advance and divides exploration and exploitation into distinct stages. Further, (Massoulie and Xu 2016) does not aim to maximize the system payoff as in this paper. Thus, it is unclear how the approach there can be used in our setting.

Myopic Matching: The second line of related work is the multi-player MAB formulation in (Kalathil et al. 2014, Nayyar et al. 2016, Gai et al. 2012, Anandkumar et al. 2011, Lai et al. 2011, Lelarge et al. 2013, Combes and Proutiere 2015, Shahrampour et al. 2017, Buccapatnam et al. 2015), which is also a common way in the literature to study online learning in a two-sided system, where clients and servers with unknown payoffs need to be matched to maximize the system payoff. However, this class of work assumes that both the client population and the server population are fixed. In contrast, in our model, the client population is constantly changing. Thus, the multi-player MAB solution can at best be viewed as a *myopic* solution for a snapshot of our system in time. Indeed, as illustrated at the end of Section 2, such a myopic matching strategy (for each snapshot in time) is sub-optimal in the long run even when the payoff vectors are known in advance.

Queue-Length Based Control: The above example clearly illustrates the need to cater to uncertain client dynamics in the system. In the literature, there is a third body of work based on max-weight algorithms to handle uncertainty in client dynamics (Tassiulas and Ephremides 1992, Neely et al. 2005, Chiang 2004, Eryilmaz and Srikant 2005, Lin et al. 2006). While some of them aim for throughput-optimality (i.e., to stabilize the system at the maximum-possible offered load), others

aim for maximizing the long-term payoff as in our problem. However, this line of work usually assumes that each client's payoff vector is known. As a result, we may not be able to use many of these algorithms when the payoff vectors are unknown (e.g., algorithms that define weights as the number of clients of each class cannot be used when the class index of each client is unknown). One particular algorithm that is closely related to our problem setting is the following queue-length based policy. By building up queues of unserved tasks at the servers, one can use the queue length q_j at server j as a "shadow price" that captures the congestion level at the server. The system operator then adjusts each client's payoff parameter C^*_{ij} to $C^*_{ij} - q_j/V$ with a proper choice of V and assigns the next task to the server with the highest adjusted payoff. When the payoff vector of each client is known in advance, this type of queue-length based control can be shown to attain near-optimal system payoff when the parameter V is large (Tan and Srikant 2012), although the length of the server-side queues also grows with V.

However, when the payoff vector of each client is unknown, such a queue-length based control will no longer produce high long-term payoff (see simulation results in Fig. 3(a) and Section 5.2). The underlying reason is that, when V is large, such a queue-length based policy produces large queues, which leads to a vicious cycle between queueing and learning. (See also Remark 2 in Section 2.) Indeed, as we will demonstrate in Section 5, a straightforward version of such queue-length based control will lead to a lower system payoff when combined with online learning.

Online Matching: There is a fourth line of related work on online matching that also aims to address uncertainty in future client dynamics (albeit in an adversarial setting) (Alaei et al. 2013, Chakrabarty et al. 2008, Feldman et al. 2009, Zheng et al. 2013). However, similar to queue-length based control, such studies typically do not deal with payoff uncertainty either.

Finally, although (Bimpikis and Markakis 2015) and (Shah et al. 2017) also study the integration of learning and control, they either allow only a small system with two types of clients and two servers (Bimpikis and Markakis 2015), or have to use an exponential number of virtual queues to approach optimality (Shah et al. 2017). In contrast, in this work, we aim to develop computationally-efficient algorithms that can be implemented even for large systems.

In summary, it remains an open question how to seamlessly integrate online learning with adaptive control when there is uncertainty in both client dynamics and payoff vectors, and how to account for the complex closed-loop interactions between queueing and learning. Below, we will propose our new algorithm and analytical techniques that address these difficulties.

EC.2. Missing Proof of Theorem 2

EC.2.1. Proof of Lemma 1

We first note that

$$n_i(t+1) = n_i(t) + U_i(t+1) - D_i(t),$$

where $U_i(t+1)$ and $D_i(t)$ are the number of client arrivals at the beginning of time $t+1$ and the number of departures at the end of time t, respectively, of class i. Note that $U_i(t)$ is a Bernoulli random variable with mean λ_i/N. For $i \notin \mathcal{I}(t)$, i.e., $n_i(t) = 0$, we have $D_i(t) = 0$ and $n_i(t+1) = U_i(t+1)$. It follows that

$$\mathbb{E}[n_i^2(t+1) - n_i^2(t) \mid n_i(t) = 0] = \mathbb{E}\left[U_i^2(t+1)\right] = \frac{\lambda_i}{N}. \tag{EC.1}$$

For $i \in \mathcal{I}(t)$, the expected time-difference of $n_i^2(t)$ is bounded by

$$\begin{aligned}
&\mathbb{E}[n_i^2(t+1) - n_i^2(t) \mid \mathbf{n}(t), \mathbf{p}(t)] \\
&= \mathbb{E}\left[2n_i(t)\left(U_i(t+1) - D_i(t)\right) + \left(U_i(t+1) - D_i(t)\right)^2 \mid \mathbf{n}(t), \mathbf{p}(t)\right] \\
&\le 2n_i(t)\left(\frac{\lambda_i}{N} - \mathbb{E}\left[D_i(t) | \mathbf{n}(t), \mathbf{p}(t)\right]\right) + \frac{\lambda_i}{N} + \mathbb{E}\left[D_i^2(t) | \mathbf{n}(t), \mathbf{p}(t)\right] \\
&\le \frac{2n_i(t)}{N}\left(\lambda_i - \sum_{k=1}^{n_i(t)} \sum_{j=1}^{J} p_{ij}^k(t)\right) + \frac{\lambda_i}{N} + \frac{\mu^2}{N}\left(1 + \frac{\gamma^2}{(\gamma-1)^2 N}\right),
\end{aligned} \tag{EC.2}$$

where in the last step we have used the following bounds shown in Lemma EC.7 in Appendix EC.4:

$$\mathbb{E}\left[D_i(t) | \mathbf{n}(t), \mathbf{p}(t)\right] \ge \frac{1}{N} \sum_{k=1}^{n_i(t)} \sum_{j=1}^{J} p_{ij}^k(t) - \frac{\gamma^2 \mu^2}{2(\gamma-1)^2 n_i(t) N^2}, \tag{EC.3}$$

and $\mathbb{E}\left[D_i^2(t) \mid \mathbf{n}(t), \mathbf{p}(t)\right] \le \frac{\mu^2}{N}$.

Combining (EC.1) and (EC.2), the expected Lyapunov drift is then bounded by

$$\begin{aligned}
&\mathbb{E}\left[L((\mathbf{n}(t+1))) - L(\mathbf{n}(t)) \mid \mathbf{n}(t), \mathbf{p}(t)\right] \\
&\leq \frac{1}{N} \sum_{i \in \mathcal{I}(t)} \frac{n_i(t)}{\lambda_i} \left(\lambda_i - \sum_{k=1}^{n_i(t)} \sum_{j=1}^{J} p_{ij}^k(t) \right) + \frac{c_1 + c_2}{N} \\
&= \frac{1}{N} \sum_{i \in \mathcal{I}(t)} \frac{n_i(t)}{\lambda_i} \sum_{k=1}^{n_i(t)} \sum_{j=1}^{J} \left(\frac{\lambda_i p_{ij}^*}{n_i(t)} - p_{ij}^k(t) \right) + \frac{c_1 + c_2}{N}, \qquad \text{(EC.4)}
\end{aligned}$$

where c_1 and c_2 are defined in the lemma, and we use $N \geq 1$ in the inequality. Adding $\frac{V}{N}\Delta(t)$ in (18), the result then follows. Note that we need to single out the first term of (18) corresponding to $n_i(t) = 0$, which produces the second term of (19).

EC.2.2. Proof of Lemma 2

Let $\widetilde{\mathbf{p}}(t)$ denote the vector $[\widetilde{p}_{ij}^k(t)]$ such that

$$\widetilde{p}_{ij}^k(t) = \frac{\lambda_i p_{ij}^*}{n_i(t)}$$

for all $i \in \mathcal{I}(t)$, $j = 1, ..., J$ and $k = 1, ..., n_i(t)$. Using the concavity of the function f and the fact that $\frac{n_i(t)}{\lambda_i} + V(W_{ij}^k - \gamma)$ is the partial derivative of f at $\widetilde{\mathbf{p}}(t)$, we can prove the following auxiliary lemma.

LEMMA EC.1. *For any* $\mathbf{W} = [W_{ij}^k]$ *and* $\boldsymbol{\pi} = [\pi_{ij}^k]$, *it holds that*

$$\begin{aligned}
&\sum_{i \in \mathcal{I}(t)} \sum_{j=1}^{J} \sum_{k=1}^{n_i(t)} \left(\frac{n_i(t)}{\lambda_i} + V(W_{ij}^k - \gamma) \right) \left(\frac{\lambda_i p_{ij}^*}{n_i(t)} - \pi_{ij}^k \right) \\
&\leq f(\widetilde{\mathbf{p}}(t) | \mathbf{n}, \mathbf{W}) - f(\boldsymbol{\pi} | \mathbf{n}, \mathbf{W}). \qquad \text{(EC.5)}
\end{aligned}$$

Proof of Lemma EC.1. For any given $\mathbf{n}$ and $\mathbf{W} = [W_{ij}^k]$, the gradient of $f(\boldsymbol{\pi}|\mathbf{n}, \mathbf{W})$ (defined in (25)) with respect to $\boldsymbol{\pi}$ is given by $\nabla f(\boldsymbol{\pi}|\mathbf{n}, \mathbf{W}) = \left[\frac{\partial f}{\partial \pi_{ij}^k} \right]$ with

$$\frac{\partial f}{\partial \pi_{ij}^k} = \frac{1}{\sum_{j=1}^{J} \pi_{ij}^k} + V(W_{ij}^k - \gamma). \qquad \text{(EC.6)}$$

Since the function f is concave in π_{ij}^k, we thus have, for any $\widetilde{\mathbf{p}}$,

$$f(\boldsymbol{\pi}|\mathbf{n}, \mathbf{W}) - f(\widetilde{\mathbf{p}}|\mathbf{n}, \mathbf{W}) \leq [\nabla f(\widetilde{\mathbf{p}}|\mathbf{n}, \mathbf{W})]'(\boldsymbol{\pi} - \widetilde{\mathbf{p}}). \qquad \text{(EC.7)}$$

Using the value of $\widetilde{\mathbf{p}}(t)$ given in the lemma, the partial derivative of f at $\widetilde{\mathbf{p}}(t)$ is equal to

$$\left.\frac{\partial f}{\partial \pi_{ij}^k}\right|_{\pi_{ij}^k=\frac{\lambda_i p_{ij}^*}{n_i(t)}} = \frac{n_i(t)}{\lambda_i} + V(W_{ij}^k - \gamma). \tag{EC.8}$$

Therefore, for any $\boldsymbol{\pi}$, we have,

$$\begin{aligned} &f(\boldsymbol{\pi}|\mathbf{n},\mathbf{W}) - f(\widetilde{\mathbf{p}}(t)|\mathbf{n},\mathbf{W}) \\ &\le \sum_{i\in\mathcal{I}(t)}\sum_{k=1}^{n_i(t)}\sum_{j=1}^{J}\left(\frac{n_i(t)}{\lambda_i} + V(W_{ij}^k-\gamma)\right)\left(\pi_{ij}^k - \frac{\lambda_i p_{ij}^*}{n_i(t)}\right). \end{aligned}$$

The result of the lemma then follows. □

The significance of this lemma is as follows. Let $\mathbf{C}^*$ denote the vector $[W_{ij}^k]$ such that $W_{ij}^k = C_{ij}^*$ for all $i \in \mathcal{I}(t)$, $j = 1,...,J$ and $k = 1,...,n_i(t)$. If we choose $\mathbf{W} = \mathbf{C}^*$ and $\boldsymbol{\pi} = \mathbf{p}(t)$ in Lemma EC.1, then $A_1(t)$ given in (20) is simply the left-hand-side of (EC.5). Applying Lemma EC.1, we then have

$$A_1(t) \le f(\widetilde{\mathbf{p}}(t)|\mathbf{n},\mathbf{C}^*) - f(\mathbf{p}(t)|\mathbf{n},\mathbf{C}^*). \tag{EC.9}$$

Next, recall that $\widehat{\mathbf{p}}(t) = \left[\widehat{p}_{ij}^k(t)\right]$ is the maximizer of $f(\boldsymbol{\pi}|\mathbf{n},\mathbf{C}^*)$ over the constraint (15). Since $\widetilde{\mathbf{p}}(t)$ also satisfies the constraint (15), we have $f(\widetilde{\mathbf{p}}(t)|\mathbf{n},\mathbf{C}^*) \le f(\widehat{\mathbf{p}}(t)|\mathbf{n},\mathbf{C}^*)$. Combining with (EC.9), we get the desired (26):

$$A_1(t) \le f(\widehat{\mathbf{p}}(t)|\mathbf{n},\mathbf{C}^*) - f(\mathbf{p}(t)|\mathbf{n},\mathbf{C}^*).$$

Finally, recalling the definition of $A_2(t)$ and $A_3(t)$ given in (28), we have

$$\begin{aligned} &f(\widehat{\mathbf{p}}(t)|\mathbf{n},\mathbf{C}^*) = f(\widehat{\mathbf{p}}(t)|\mathbf{n},\mathbf{C}(t)) + A_3(t) \\ &\le f(\mathbf{p}(t)|\mathbf{n},\mathbf{C}(t)) + A_3(t) \qquad \text{(by the optimality of } \mathbf{p}(t)) \\ &= [f(\mathbf{p}(t)|\mathbf{n},\mathbf{C}^*) + A_2(t)] + A_3(t). \end{aligned} \tag{EC.10}$$

Combining (26) and (EC.10), we thus have $A_1(t) \le A_2(t) + A_3(t)$, completing the proof of Lemma 2.

EC.2.3. Proof of Lemma 3

In this subsection, we bound $\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[A_2(t)]$ for a given T. Recall that Λ is a particular realization of the sequence of client arrival-times up to time slot T. We use $\mathbb{E}_\Lambda$ to denote the conditional expectation given Λ. Given Λ, we slightly abuse notation and use the index k now to denote the k-th client of class i that arrives to the system. Let $t_1(k)$ denote the arrival time of this client, and $t_2(k)$ denote the *minimum* of her departure time and T.

Before the proof, we first give an intuitive interpretation of Lemma 3.

REMARK EC.1. The result of Lemma 3 can be interpreted as follows. Suppose that $p_{ij}^k(s)=1$ for all time s. Then, after s time-slots, the gap $C_{ij}^k(s)-C_{ij}^*$ is within the order of $\Theta(\sqrt{\log s/s})$ with high probability by the Chernoff-Hoeffding bound (see Lemma EC.8 in Appendix EC.5). Summing over all $1\le s\le a_k$, we get an expression on the order of the first term in (29). Of course, under our algorithm $p_{ij}^k(t)$ is random. Fortunately, here the loss given by the left-hand-side of (29) accumulates at the same rate as the probability $p_{ij}^k(t)$ of choosing a server (which will not be the case for the proof of Lemma 4 in Appendix EC.2.4). Thus, we may view the time as being slowed down at the rate of $\sum_{j=1}^{J}p_{ij}^k(t)$, and expect the bound in (29) to hold. However, the tricky part is that the value of $p_{ij}^k(t)$ also depends on previous values of $C_{ij}^k(s)$, $s<t$. Our proof uses a delicate martingale argument to take care of such dependency.

Next we explain how to derive the upper bound to $\mathbb{E}[A_2(t)]$ in (31) from Lemma 3.

REMARK EC.2. First, given a sequence of arrival times Λ, by re-indexing the clients in the order of their arrival times, we have

$$\sum_{t=1}^{T}A_2(t)=V\sum_{i=1}^{I}\sum_{k=1}^{m_i(T)}\sum_{t=t_1(k)}^{t_2(k)}\sum_{j=1}^{J}\left(C_{ij}^k(t)-C_{ij}^*\right)p_{ij}^k(t),$$

where $m_i(T)$ is the total number of arrivals of class i up to time slot T. Since Lemma 3 holds for any client k, we have

$$\sum_{t=1}^{T}\mathbb{E}_\Lambda\left[A_2(t)\right]\le V\sum_{i=1}^{I}\sum_{k=1}^{m_i(T)}\left(4\sqrt{2Ja_k\log a_k}+\left(4\sqrt{2\log a_k}+3\right)\mu\right).$$

Note that a_k has mean N, and further both $\sqrt{x\log x}$ and $\sqrt{\log x}$ are concave functions. Moreover, $\mathbb{E}[m_i(T)] = \lambda_i T/N$. Taking the expectation of a_k and then Λ over both sides of the last displayed equation and applying Jensen's inequality, we get the desired (31):

$$\frac{1}{TV}\sum_{t=1}^{T}\mathbb{E}[A_2(t)] \le \frac{\lambda}{N}\left(4\sqrt{2JN\log N} + \left(4\sqrt{2\log N}+3\right)\mu\right).$$

Finally, we present the proof of Lemma 3.

Proof of Lemma 3. Fix k, i. Without loss of generality, we can take $t_1(k) = 1$, i.e., we label the first time-slot as the time-slot when this particular client arrives to the system. Recall that $h_{ij}^k(t)$ is the number of tasks from this particular client that have been assigned to server j by the end of time t, and $h_i^k(t) = \sum_{j=1}^{J} h_{ij}^k(t)$. For $t = 0$, we take $h_{ij}^k(0) = h_i^k(0) = 0$. In the definition of the event $F_j(t)$ below, in order to avoid the difficulty of division by zero, we further define $\widehat{h}_{ij}^k(t) = \max\{h_{ij}^k(t), 1\}$ and $\widehat{h}_i^k(t) = \max\{h_i^k(t), 1\}$. Then, for each $j = 1, \ldots, J$, define the event

$$F_j(t) = \left\{\overline{C}_{ij}^k(t-1) - C_{ij}^* \le \sqrt{\frac{2\log \widehat{h}_i^k(t-1)}{\widehat{h}_{ij}^k(t-1)}}\right\},$$

where $\overline{C}_{ij}^k(t-1)$ is the empirical average of the received payoffs at the end of time slot $t-1$. We let $Q_{ij}^k(t) = C_{ij}^k(t) - C_{ij}^* \le 1$. Then

$$\sum_{s=1}^{t_2(k)} Q_{ij}^k(s)p_{ij}^k(s) \le \sum_{s=1}^{t_2(k)} Q_{ij}^k(s)p_{ij}^k(s)1_{F_j(s)} + \sum_{s=1}^{t_2(k)} p_{ij}^k(s)1_{F_j^c(s)}. \tag{EC.11}$$

We first bound the expectation of the first term in (EC.11). By the definition of the event $F_j(t)$ and the definition of the UCB estimate $C_{ij}^k(t)$,

$$\begin{aligned}\sum_{s=1}^{t_2(k)} Q_{ij}^k(s)p_{ij}^k(s)1_{F_j(s)} &\le \sum_{s=1}^{t_2(k)} 2\sqrt{\frac{2\log \widehat{h}_i^k(s-1)}{\widehat{h}_{ij}^k(s-1)}}p_{ij}^k(s)\\ &\le 2\sqrt{2\log a_k}\sum_{s=1}^{t_2(k)}\sqrt{\frac{1}{\widehat{h}_{ij}^k(s-1)}}p_{ij}^k(s).\end{aligned} \tag{EC.12}$$

Recall that $Y_{ij}^k(t)$ is the actual number of tasks served by server j at time slot t. We now show that

$$\mathbb{E}_\Lambda\left[\sum_{s=1}^{t_2(k)}\sqrt{\frac{1}{\widehat{h}_{ij}^k(s-1)}}Y_{ij}^k(s)\right] = \mathbb{E}_\Lambda\left[\sum_{s=1}^{t_2(k)}\sqrt{\frac{1}{\widehat{h}_{ij}^k(s-1)}}p_{ij}^k(s)\right]. \tag{EC.13}$$

To see this, let $\mathcal{F}_t$ denote the filtration (which contains all the system information) up to the end of time-slot t. In particular, $Y_{ij}^k(s)$ and $p_{ij}^k(s)$ are measurable with respect to $\mathcal{F}_t$ for all $s \le t$. Let

$$M_t \triangleq \sum_{s=t_1(k)}^{t} \sqrt{\frac{1}{\widehat{h}_{ij}^k(s-1)}}(Y_{ij}^k(s) - p_{ij}^k(s)).$$

Since $\mathbb{E}_\Lambda[Y_{ij}^k(s+1)|p_{ij}^k(s+1)] = p_{ij}^k(s+1)$, we have $\mathbb{E}_\Lambda[Y_{ij}^k(s+1) - p_{ij}^k(s+1)|\mathcal{F}_s] = 0$. It then follows that M_t is martingale. Further, note that $t_2(k)$ is the minimum between T and the first time that $h_i^k(t)$ exceeds a_k. Hence, $t_2(k)$ is a stopping time with respect to the filtration $\mathcal{F}_t$ and is upper bounded by T. Invoking the Optional Stopping Theorem (Hajek 2015, Section 10.4), we then have $\mathbb{E}[M_{t_2(k)}] = 0$, which is precisely (EC.13).

By definition, $Y_{ij}^k(t) = h_{ij}^k(t) - h_{ij}^k(t-1)$. It follows that

$$\begin{aligned}\sum_{s=1}^{t_2(k)} \sqrt{\frac{1}{\widehat{h}_{ij}^k(s-1)}} Y_{ij}^k(s) &= \sum_{s=1}^{t_2(k)} \sqrt{\frac{1}{\widehat{h}_{ij}^k(s-1)}} \left(h_{ij}^k(s) - h_{ij}^k(s-1)\right) \\ &\le 2\mu_j + \int_1^{h_{ij}^k(t_2(k))-1} \frac{1}{\sqrt{x}} \mathrm{d}x \\ &\le 2\mu_j + 2\sqrt{h_{ij}^k(t_2(k))},\end{aligned}$$

where the second-to-last step follows because whenever $h_{ij}^k(t) = 0$, we must have $h_{ij}^k(t) - h_{ij}^k(t-1) = 0$ and we always have $h_{ij}^k(t) - h_{ij}^k(t-1) \le \mu_j$. Therefore, summing the last displayed equation from $j = 1$ to $j = J$, we get that

$$\sum_{j=1}^{J} \sum_{s=1}^{t_2(k)} \sqrt{\frac{1}{\widehat{h}_{ij}^k(s-1)}} Y_{ij}^k(s) \le 2\mu + 2\sum_{j=1}^{J} \sqrt{h_{ij}^k(t_2(k))} \le 2\mu + 2\sqrt{J\sum_{j=1}^{J} h_{ij}^k(t_2(k))} = 2\mu + 2\sqrt{Ja_k},$$

where the second inequality holds due to the Cauchy–Schwarz inequality.

Combining the last displayed equation with (EC.12) and (EC.13), we have

$$\mathbb{E}_\Lambda\left[\sum_{j=1}^{J}\sum_{s=1}^{t_2(k)} Q_{ij}^k(s) p_{ij}^k(s) 1_{F_j(s)}\right] \le 4\sqrt{2Ja_k \log a_k} + 4\mu\sqrt{2\log a_k}. \tag{EC.14}$$

To bound the expected value of the second term in (EC.11), we first note that

$$\mathbb{E}_\Lambda\left[\sum_{s=1}^{t_2(k)} Y_{ij}^k(s) 1_{F_j^c(s)}\right] = \mathbb{E}_\Lambda\left[\sum_{s=1}^{t_2(k)} p_{ij}^k(s) 1_{F_j^c(s)}\right], \tag{EC.15}$$

which can be derived similarly to (EC.13). For each integer $0 \le a \le a_k$, define

$$\tau_a = \min\left\{T+1, \inf\left\{s \ge 1 \mid h_{ij}^k(s-1) \ge a\right\}\right\}.$$

Note that $1 = \tau_0 \le \tau_1 \le \cdots \le \tau_{a_k} = t_2(k)+1$. Then,

$$\begin{aligned}
\sum_{s=1}^{t_2(k)} Y_{ij}^k(s) 1_{F_j^c(s)} &= \sum_{a\in[0,a_k-1]:\tau_{a+1}>\tau_a} \sum_{s=\tau_a}^{\tau_{a+1}-1} Y_{ij}^k(s) 1_{F_j^c(s)} \\
&= \sum_{a\in[0,a_k-1]:\tau_{a+1}>\tau_a} Y_{ij}^k(\tau_{a+1}-1) 1_{F_j^c(\tau_{a+1}-1)} \\
&\le \mu_j + \mu_j \sum_{a\in[1,a_k-1]:\tau_{a+1}>\tau_a} 1_{F_j^c(\tau_{a+1}-1)}, \qquad \text{(EC.16)}
\end{aligned}$$

where the second equality holds because across all $s \in [\tau_a, \tau_{a+1}-1]$, the value of $Y_{ij}^k(s)$ can be non-zero only at $\tau_{a+1}-1$ (otherwise, $h_{ij}^k(s)$ will be at least $a+1$ before $\tau_{a+1}-1$, contradicting the definition of τ_{a+1}). Combining (EC.15) and (EC.16) yields that

$$\begin{aligned}
&\mathbb{E}_\Lambda\left[\sum_{s=1}^{t_2(k)} p_{ij}^k(s) 1_{F_j^c(s)}\right] \\
&\le \mu_j + \mu_j \sum_{a\in[1,a_k-1]:\tau_{a+1}>\tau_a} \mathbb{P}_\Lambda\left\{F_j^c(\tau_{a+1}-1)\right\}. \qquad \text{(EC.17)}
\end{aligned}$$

Fix an integer a such that $\tau_{a+1} > \tau_a$. It follows that the number $h_{ij}^k(\tau_a - 1)$ of tasks assigned to server j by the end of time-slot $\tau_a - 1$ must be exactly equal to a. Note that $\tau_a - 1 \le \tau_{a+1} - 2 < \tau_{a+1} - 1$. Hence, we must also have $h_{ij}^k(\tau_{a+1}-2) = a$ and thus the empirical average $\overline{C}_{ij}^k(\tau_{a+1}-2)$ is the average of exactly a i.i.d. Bernoulli random variables $X(1), \ldots, X(a)$ with mean C_{ij}^*. Moreover, $a \le h_i^k(\tau_{a+1}-2) \le a_k$. Therefore,

$$F_j^c(\tau_{a+1}-1) \subset \cup_{\eta=a}^{a_k}\left\{\frac{1}{a}\sum_{u=1}^{a} X(u) - C_{ij}^* > \sqrt{\frac{2\log\eta}{a}}\right\}.$$

By the union bound,

$$\begin{aligned}
\mathbb{P}_\Lambda\left\{F_j^c(\tau_{a+1}-1)\right\} &\le \sum_{\eta=a}^{a_k} \mathbb{P}\left\{\frac{\sum_{u=1}^{a} X(u)}{a} - C_{ij}^* > \sqrt{\frac{2\log\eta}{a}}\right\} \\
&\le \sum_{\eta=a}^{a_k} \frac{1}{\eta^4} \le \frac{1}{a^4} + \int_a^\infty \frac{1}{\eta^4} d\eta \le \frac{4}{3a^3},
\end{aligned}$$

where the second inequality is due to Lemma EC.8. Substituting it into (EC.17), we then have

$$\mathbb{E}_\Lambda\left[\sum_{s=1}^{t_2(k)} p_{ij}^k(s) 1_{F_j^c(s)}\right] \le \mu_j + \mu_j \sum_{a=1}^{\infty} \frac{4}{3a^3} = 3\mu_j. \tag{EC.18}$$

The final result follows by combining (EC.14) and (EC.18) with (EC.11). □

EC.2.4. Proof of Lemma 4.

We now bound the contribution to $\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[A_3(t)]$ by the k-th arriving client of class i, using similar ideas as Appendix EC.2.3. Here, however, we face a major new difficulty related to the issue of "learning slow-down" discussed in Section EC.1. Note that the rate with which tasks of this client are assigned to the servers at time t is given by $\sum_{j=1}^{J} p_{ij}^k(t)$, which may decrease as there are more clients in the system. We thus refer to this value as the "learning rate," since it determines how quickly (or slowly) the system can receive payoff feedback for this client and improve her payoff estimate. However, the loss in $A_3(t)$ accumulates at a different rate $\widehat{p}_{ij}^k(t)$. We therefore refer to $\sum_{j=1}^{J}\widehat{p}_{ij}^k(t)$ as the "loss-accummulation rate." If the learning rate is small and the loss-accummulation rate is large, it is possible that the total loss may grow unboundedly because the system does not learn as quickly as it incurs losses. The following lemma thus becomes crucial. It shows that, thanks to our choice in Algorithm 1 that all the UCB estimates are no greater than 1 and $\gamma > 1$, the learning rate for any client will be at most a constant factor away from the loss-accummulation rate. We will then use this lemma to show that $\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[A_3(t)]$ must then be upper-bounded by a constant.

LEMMA EC.2. *At any time t, suppose that $[\widehat{p}_{ij}^k(t)]$ and $[p_{ij}^k(t)]$ are the maximizers of $f(\boldsymbol{\pi}|\mathbf{n}(t), \mathbf{C}^*)$ and $f(\boldsymbol{\pi}|\mathbf{n}(t), \mathbf{C}(t))$ defined in (25), respectively, subject to the constraint (15). Then, for all clients k of all classes i, the following holds:*

$$\frac{\sum_{j=1}^{J}\widehat{p}_{ij}^k(t)}{\sum_{j=1}^{J} p_{ij}^k(t)} \le \left(\frac{\gamma}{\gamma-1}\right)^2.$$

The main idea behind Lemma EC.2 is as follows. Note that for any $\mathbf{W}$, the maximizer of $f(\boldsymbol{\pi}|\mathbf{n}(t), \mathbf{W})$ subject to server capacity constraint (15) must satisfy the KKT condition:

$$\sum_{j=1}^{J} \pi_{ij}^k(t) = \frac{1}{V \min_j\{q_j(t) + \gamma - W_{ij}^k\}},$$

where $q_j(t) \geq 0$ is the optimal dual variable corresponding to server j's capacity constraint. For either $\mathbf{W} = \mathbf{C}^*$ or $\mathbf{W} = \mathbf{C}(t)$, the value of W_{ij}^k cannot differ by more than 1. Hence, the value of $q_j(t) + \gamma - W_{ij}^k$ cannot differ by more than a factor $\frac{\gamma}{\gamma-1}$. This implies that the value of $\sum_{j=1}^J p_{ij}^k(t)$ or $\sum_{j=1}^J \widehat{p}_{ij}^k(t)$ across different clients cannot differ much either. The result of Lemma EC.2 can then be readily shown. Below we give the formal proof.

Proof of Lemma EC.2. Fix t and client k of class i. Let $\mathbf{n} = \mathbf{n}(t)$ and recall that $C_{ij}^k(t) \leq 1$. First, consider $[\widehat{p}_{ij}^k(t)]$, which is the maximizer of $f(\mathbf{p}|\mathbf{n}, \mathbf{C}^*))$ over the constraint (15). Let $q_j(t) \geq 0$ be the optimal dual variable corresponding to server j's capacity constraint in (15). By the KKT condition, for any client k of any class i, we have

$$\sum_{j=1}^J \widehat{p}_{ij}^k(t) = \frac{1}{V \min_j\{q_j(t) + \gamma - C_{ij}^*\}} \leq \frac{1}{V(\gamma-1)}, \tag{EC.19}$$

where the inequality holds because $q_j(t) \geq 0$ and $C_{ij}^* \leq 1$. We now compare the pair (i,k) with another pair $(i',k') \neq (i,k)$. Note that for any $c, c' \in [0,1]$, we have

$$q_j(t) + \gamma - c \geq q_j(t) + \gamma - 1 \geq \frac{\gamma-1}{\gamma}\left(q_j(t) + \gamma - c'\right). \tag{EC.20}$$

Applying this relationship to the KKT condition (EC.19), we have

$$\begin{aligned}\sum_{j=1}^J \widehat{p}_{ij}^k(t) &= \frac{1}{V \min_j\{q_j(t) + \gamma - C_{ij}^*\}} \\ &\leq \frac{\gamma/(\gamma-1)}{V \min_j\{q_j(t) + \gamma - C_{i'j}^*\}} = \frac{\gamma}{\gamma-1}\sum_{j=1}^J \widehat{p}_{i'j}^{k'}(t).\end{aligned}$$

By the capacity constraint (15),

$$\sum_{j=1}^J \widehat{p}_{ij}^k(t) + \sum_{(i',k')\neq(i,k)} \sum_{j=1}^J \widehat{p}_{i'j}^{k'}(t) \leq \mu,$$

where $\mu \triangleq \sum_{j=1}^J \mu_j$. Combining the above two inequalities, we have

$$\sum_{j=1}^J \widehat{p}_{ij}^k(t) \leq \frac{\mu}{1 + (n(t)-1)(\gamma-1)/\gamma} \leq \frac{\gamma\mu}{(\gamma-1)n(t)} \triangleq P_A,$$

where $n(t) = \sum_{i=1}^I n_i(t)$. Combining the last displayed equation with (EC.19), we have

$$\sum_{j=1}^J \widehat{p}_{ij}^k(t) \leq \min\left\{P_A, \frac{1}{V(\gamma-1)}\right\}. \tag{EC.21}$$

Next, we consider $[p_{ij}^k(t)]$, which is the maximizer of $f(\mathbf{p}|\mathbf{n},\mathbf{C}(t)))$ over the constraint (15). Let $q_j(t)$ now denote the optimal dual variable for this optimization problem. Using the KKT condition again, we have

$$\sum_{j=1}^{J} p_{ij}^k(t) = \frac{1}{V \min_j\{q_j(t)+\gamma-C_{ij}^k(t)\}}.$$

Combining the last displayed equation with (EC.20) again, we have that for any pairs (i,k) and (i',k'),

$$\sum_{j=1}^{J} p_{ij}^k(t) \geq \frac{(\gamma-1)/\gamma}{V\gamma \min_j\{q_j(t)+\gamma-C_{i'j}^{k'}(t)\}} = \frac{\gamma-1}{\gamma}\sum_{j=1}^{J} p_{i'j}^{k'}(t).$$

Now, consider the following two sub-cases. In sub-case 1, suppose that $q_j(t)>0$ for all j, which means that all servers are fully utilized. Then, we get

$$\sum_{j=1}^{J}\mu_j = \sum_{j=1}^{J} p_{ij}^k(t) + \sum_{(i',k')\neq(i,k)}\sum_{j=1}^{J} p_{i'j}^{k'}(t).$$

Hence, it follows from the last two displayed equations that

$$\sum_{j=1}^{J} p_{ij}^k(t) \geq \frac{\mu}{1+\gamma(n(t)-1)/(\gamma-1)} \geq \frac{(\gamma-1)\mu}{\gamma n(t)} \triangleq P_B.$$

In sub-case 2, suppose that there is some server j_0 such that $q_{j_0}(t)=0$. Then

$$\begin{aligned}\sum_{j=1}^{J} p_{ij}^k(t) &= \frac{1}{V\min_j\{q_j(t)+\gamma-C_{ij}^k(t)\}}\\ &\geq \frac{1}{V(q_{j_0}(t)+\gamma-C_{i,j_0}^k)} \geq \frac{1}{V\gamma}.\end{aligned}$$

Combining these two sub-cases together, we have

$$\sum_{j=1}^{J} p_{ij}^k(t) \geq \min\left\{P_B, \frac{1}{V\gamma}\right\}. \tag{EC.22}$$

Combining (EC.21) and (EC.22), we have

$$\frac{\sum_{j=1}^{J}\widehat{p}_{ij}^k(t)}{\sum_{j=1}^{J} p_{ij}^k(t)} \leq \frac{\min\{P_A, \frac{1}{V(\gamma-1)}\}}{\min\{P_B, 1/(V\gamma)\}} \leq \left(\frac{\gamma}{\gamma-1}\right)^2.$$

□

Next, we give the intuition behind the proof of Lemma 4.

REMARK EC.3. Recall that $h_{ij}^k(t-1)$ is the number of tasks from this particular client that have been assigned to server j by the end of time $t-1$, and $h_i^k(t-1)=\sum_{j=1}^J h_{ij}^k(t-1)$. Further, let $h_{ij}^k(0)=h_i^k(0)=0$. For each time-slot s and $1\le j\le J$, define the "good" event

$$G_j(s)=\left\{h_{ij}^k(s-1)\ge 1, C_{ij}^*-\overline{C}_{ij}^k(s-1)\le\sqrt{\frac{2\log h_i^k(s-1)}{h_{ij}^k(s-1)}}\right\}$$
$$\cup\left\{h_{ij}^k(s-1)=0\right\},$$

where $\overline{C}_{ij}^k(s-1)$ is the empirical average of the received payoffs at the end of time slot $s-1$. Let $G(s)=\cap_{j=1}^J G_j(s)$. Note that on the event $G(s)$, the UCB estimates satisfy $C_{ij}^k(s)\ge C_{ij}^*$ and thus the terms $\left(C_{ij}^*-C_{ij}^k(s)\right)\widehat{p}_{ij}^k(s)$ on the left hand side of (30) are negative. Hence, to prove the lemma, it suffices to bound the sum due to the probability of the "bad" event $G^c(s)$ (i.e., the complement of $G(s)$), which accumulates at the rate of $\sum_{j=1}^J\widehat{p}_{ij}^k(s)$. Thanks to Lemma EC.2, this loss-accumulation rate is upper bounded by a constant multiplied by the learning rate $\sum_{j=1}^J p_{ij}^k(s)$. Finally, using concentration bounds and a delicate martingale argument, we can show that the probability of the bad event $G^c(s)$ decreases polynomially fast as the learning rate accumulates, resulting to the bound in (30).

Proof of Lemma 4. Without loss of generality, we take $t_1(k)=1$, i.e., we label the first time-slot as the time-slot when this particular client arrives to the system. Let $c_\gamma=\gamma^2/(\gamma-1)^2$. Let $1_{G(s)}$ denote the indicator variable which is 1 if $G(s)$ holds and 0 otherwise. Then

$$\begin{aligned}
&\sum_{s=1}^{t_2(k)}\left(C_{ij}^*-C_{ij}^k(s)\right)\widehat{p}_{ij}^k(s)\\
&=\sum_{s=1}^{t_2(k)}\left(C_{ij}^*-C_{ij}^k(s)\right)\widehat{p}_{ij}^k(s)1_{G(s)}+\sum_{s=1}^{t_2(k)}\left(C_{ij}^*-C_{ij}^k(s)\right)\widehat{p}_{ij}^k(s)1_{G^c(s)}\\
&\le\sum_{s=1}^{t_2(k)}\widehat{p}_{ij}^k(s)1_{G^c(s)},
\end{aligned}$$

where the last inequality holds because $C_{ij}^*\le 1$ and $C_{ij}^*\le C_{ij}^k(s)$ on the event $G(s)$. Summing over all server j, we obtain

$$\sum_{j=1}^J\sum_{s=1}^{t_2(k)}\left(C_{ij}^*-C_{ij}^k(s)\right)\widehat{p}_{ij}^k(s)\le\sum_{j=1}^J\sum_{s=1}^{t_2(k)}\widehat{p}_{ij}^k(s)1_{G^c(s)}$$

$$\leq c_\gamma \sum_{s=1}^{t_2(k)} 1_{G^c(s)} \sum_{j=1}^{J} p_{ij}^k(s),$$

where the last inequality holds because $1_{G^c(s)}$ does not depend on the index j, and in view of Lemma EC.2. Taking the conditional expectation given Λ (i.e., the realization of arriving times) over both sides of the last displayed equation, we obtain

$$\mathbb{E}_\Lambda\left[\sum_{j=1}^{J}\sum_{s=1}^{t_2(k)}\left(C_{ij}^*-C_{ij}^k(s)\right)\widehat{p}_{ij}^k(s)\right] \leq c_\gamma \mathbb{E}_\Lambda\left[\sum_{s=1}^{t_2(k)} 1_{G^c(s)}\sum_{j=1}^{J} p_{ij}^k(s)\right]. \tag{EC.23}$$

Similar to $Y_l^j(t)$ in Section 3, denote $Y_{ij}^k(t)$ as the actual number of tasks from client k of class i served by server j at time slot t. We now show that

$$\mathbb{E}_\Lambda\left[\sum_{s=1}^{t_2(k)} 1_{G^c(s)}\sum_{j=1}^{J} p_{ij}^k(s)\right] = \mathbb{E}_\Lambda\left[\sum_{s=1}^{t_2(k)} 1_{G^c(s)}\sum_{j=1}^{J} Y_{ij}^k(s)\right]. \tag{EC.24}$$

To see this, let $\mathcal{F}_t$ denote the filtration (which contains all the system information) up to the end of time-slot t. In particular, $Y_{ij}^k(s)$ and $1_{G^c(s+1)}$ are measurable with respect to $\mathcal{F}_t$ for all $s \leq t$. Let $M_t \triangleq \sum_{s=1}^{t} 1_{G^c(s)} \sum_{j=1}^{J}\left(Y_{ij}^k(s)-p_{ij}^k(s)\right)$. Since $\mathbb{E}_\Lambda[Y_{ij}^k(s+1)-p_{ij}^k(s+1)|\mathcal{F}_s]=0$, it follows that M_t is martingale. Further, note that $t_2(k)$ is the minimum between T and the first time that $h_i^k(t)$ exceeds a_k. Hence, $t_2(k)$ is a stopping time with respect to the filtration $\mathcal{F}_t$ and is upper bounded by T. Invoking the Optional Stopping Theorem (Hajek 2015, Section 10.4), we then have $\mathbb{E}[M_{t_2(k)}]=0$, which is precisely (EC.24). In view of (EC.23) and (EC.24), to prove the lemma it suffices to show that

$$\mathbb{E}_\Lambda\left[\sum_{s=1}^{t_2(k)} 1_{G^c(s)}\sum_{j=1}^{J} Y_{ij}^k(s)\right] \leq 3J\mu. \tag{EC.25}$$

Towards this end, for each integer $a=0,1,...,a_k$, define

$$\tau_a = \min\left\{T+1, \inf\left\{s\geq 1 \mid h_i^k(s-1)\geq a\right\}\right\}.$$

In other words, τ_a is the first time-slot s such that the number of client k's tasks already assigned to servers at the end of the previous time-slot $s-1$ is at least a. Note that $1=\tau_0\leq\tau_1\leq\cdots\leq\tau_{a_k}=t_2(k)+1$. Then, we have

$$\sum_{s=1}^{t_2(k)} 1_{G^c(s)}\sum_{j=1}^{J} Y_{ij}^k(s) = \sum_{a\in[0,a_k-1]:\tau_{a+1}>\tau_a}\ \sum_{s=\tau_a}^{\tau_{a+1}-1} 1_{G^c(s)}\sum_{j=1}^{J} Y_{ij}^k(s)$$

$$= \sum_{a\in[0,a_k-1]:\tau_{a+1}>\tau_a} 1_{G^c(\tau_{a+1}-1)} \sum_{j=1}^{J} Y_{ij}^k(\tau_{a+1}-1)$$

$$\le \mu + \mu \sum_{a\in[1,a_k-1]:\tau_{a+1}>\tau_a} 1_{G^c(\tau_{a+1}-1)}, \tag{EC.26}$$

where the second equality holds because across all $s\in[\tau_a,\tau_{a+1}-1]$, the value of $\sum_{j=1}^{J} Y_{ij}^k(s)$ can be non-zero only at the last time-slot $\tau_{a+1}-1$ (otherwise, $h_i^k(s)$ will be at least $a+1$ before $\tau_{a+1}-1$, contradicting the definition of τ_{a+1}); and the last inequality holds because $\sum_{j=1}^{J} Y_{ij}^k(s)\le\mu$ and it can also be non-zero at most once across all $s\in[\tau_0,\tau_1-1]$.

Now, fix an integer a such that $\tau_{a+1}>\tau_a$. It follows that the number $h_i^k(\tau_a-1)$ of tasks assigned by the end of time-slot τ_a-1 must be exactly equal to a. Note that $\tau_a-1\le\tau_{a+1}-2<\tau_{a+1}-1$. Hence, we must also have $h_i^k(\tau_{a+1}-2)=a$. By the union bound,

$$\mathbb{P}_\Lambda\left\{G^c(\tau_{a+1}-1)\right\} \le \sum_{j=1}^{J} \mathbb{P}_\Lambda\left\{G_j^c(\tau_{a+1}-1)\right\}. \tag{EC.27}$$

Fix a $1\le j\le J$. When $h_{ij}^k(\tau_{a+1}-2)\ge 1$, the empirical average $\overline{C}_{ij}^k(\tau_{a+1}-2)$ is the average of $h_{ij}^k(\tau_{a+1}-2)$ i.i.d. Bernoulli random variables $X(1),X(2),\ldots$ with mean C_{ij}^*. Although $h_{ij}^k(\tau_{a+1}-2)$ is random, it holds that $h_{ij}^k(\tau_{a+1}-2)\le h_i^k(\tau_{a+1}-2)=a$. Therefore,

$$\begin{aligned}
&\mathbb{P}_\Lambda\left\{G_j^c(\tau_{a+1}-1)\right\}\\
&\le \mathbb{P}_\Lambda\left\{\forall 1\le r\le a: \frac{1}{r}\sum_{u=1}^{r} X(u) - C_{ij}^* < -\sqrt{\frac{2\log a}{r}}\right\}\\
&\le \sum_{r=1}^{a} \mathbb{P}\left\{\frac{1}{r}\sum_{u=1}^{r} X(u) - C_{ij}^* < -\sqrt{\frac{2\log a}{r}}\right\} \le \sum_{r=1}^{a}\frac{1}{a^4} \le \frac{1}{a^3},
\end{aligned} \tag{EC.28}$$

where the second inequality is due to the Chernoff-Hoeffding Bound (see Lemma EC.8 in Appendix EC.5). Combining (EC.26)-(EC.28) yields

$$\mathbb{E}_\Lambda\left[\sum_{s=1}^{t_2(k)} 1_{G^c(s)} \sum_{j=1}^{J} Y_{ij}^k(s)\right] \le \mu + \mu J \sum_{a=1}^{\infty}\frac{1}{a^3} \le \mu + 3\mu J/2 \le 3\mu J,$$

which is exactly (EC.25). The result of the lemma then follows. □

EC.3. Proof of Theorem 1

To prove Theorem 1, we first show that when the number of users in the system is large enough, all the servers must be busy (Lemma EC.3). Then, we show that when the number of users in the system is sufficiently large, the total number of departures will first-order stochastically dominate a Binomial random variable with mean $(\mu+\lambda)/(2N)$ (Lemma EC.4). Finally, we construct a coupling between the system and a Geom/Geom/μ queue to bound $n(t)$ and obtain the final result (Lemma EC.5 and Lemma EC.6).

LEMMA EC.3. *If $n(t) \geq \mu V\gamma$, then for every server j,*

$$\sum_{i\in I(t)}\sum_{k=1}^{n_i(t)} p_{ij}^k(t) = \mu_j.$$

Proof of Lemma EC.3. We prove it by contradiction. Suppose that the conclusion does not hold. Then there must exist a server j_0 such that $q_{j_0}(t)=0$. Since $[p_{ij}^k(t)]$ solves (14), it follows from the KKT condition that for every user k of class i,

$$\sum_{j=1}^{J} p_{ij}^k(t) = \frac{1}{V\min_j\{q_j(t)+\gamma-C_{ij}^k(t)\}} \geq \frac{1}{V\gamma}.$$

Hence, we have

$$\sum_{i\in I(t)}\sum_{k=1}^{n_i(t)}\sum_{j=1}^{J} p_{ij}^k(t) \geq \frac{n(t)}{V\gamma} \geq \mu = \sum_{j=1}^{J}\mu_j,$$

where the second inequality holds due to the assumption that $n(t)\geq \mu V\gamma$. On the other hand, due to the server capacity constraint,

$$\sum_{i\in I(t)}\sum_{k=1}^{n_i(t)} p_{ij}^k(t) \leq \mu_j.$$

Combining the last two displayed equations yields that

$$\sum_{i\in I(t)}\sum_{k=1}^{n_i(t)} p_{ij}^k(t) = \mu_j$$

for all j, which leads to a contradiction. □

Next, let

$$\tau^* = \mu\gamma\max\left\{V, \frac{2\mu^2}{(\mu-\lambda)(\gamma-1)}\right\}. \tag{EC.29}$$

We present the following lemma which is the key to bound Theorem 1.

LEMMA EC.4. *When the number of clients in the system is sufficiently large, i.e., if*

$$n(t) \geq \tau^*, \tag{EC.30}$$

then the number of total departures at time t, $D(t)$, first-order stochastically dominates W, where

$$W \sim \text{Binom}\left(\mu, \frac{\mu+\lambda}{2N\mu}\right),$$

i.e., $\mathbb{P}\{D(t) \geq x\} \geq \mathbb{P}\{W \geq x\}$ for all $x \geq 0$.

Proof of Lemma EC.4. According to Algorithm 1 (lines 11 to 14), each server has μ_j capacity and thus makes μ_j independent selections of users to serve. In total, there are μ such selections at every time-slot. Let us index these selections by $1, 2, \ldots, \mu$. Note that different selections may select the same user. Let S_ℓ denote the set of users (i,k) who has been selected before the ℓ-th selection. Let $X_\ell = 1$ if the ℓ-th selected user is not in S_ℓ, and $X_\ell = 0$, otherwise. Suppose that server j makes the ℓ-th selection. Then, conditioned on S_ℓ, $X_\ell(t)$ is a Bernoulli random variable with mean

$$\begin{aligned}
\mathbb{P}\{X_\ell(t) = 1 | S_\ell\} &= \frac{1}{\mu_j} \sum_{i \in I(t)} \sum_{k=1}^{n_i(t)} p_{ij}^k(t) \mathbf{1}_{\{(i,k) \notin S_\ell\}} \\
&= 1 - \frac{1}{\mu_j} \sum_{i \in I(t)} \sum_{k=1}^{n_i(t)} p_{ij}^k(t) \mathbf{1}_{\{(i,k) \in S_\ell\}} \\
&\geq 1 - \mu \max_{i \in I(t)} \max_{1 \leq k \leq n_i(t)} p_{ij}^k(t),
\end{aligned}$$

where the second equality holds due to Lemma EC.3, and the last inequality holds because $\mu_j \geq 1$ and $|S_\ell| \leq \mu$. Using (EC.40) in the proof of Lemma EC.7, we have

$$\sum_{j=1}^{J} p_{ij}^k(t) \leq \frac{\gamma\mu}{(\gamma-1)n(t)}, \quad \forall (i,k).$$

Combining the last two displayed equations yields that

$$\begin{aligned}
\mathbb{P}\{X_\ell(t) = 1 | S_\ell\} &\geq 1 - \frac{\gamma\mu^2}{(\gamma-1)n(t)} \\
&\geq \frac{\mu+\lambda}{2\mu},
\end{aligned} \tag{EC.31}$$

where the last inequality holds due to the assumption (EC.30). Let

$$Y_\ell \overset{\text{i.i.d.}}{\sim} \text{Bern}\left(\frac{\mu+\lambda}{2\mu}\right), \quad \ell = 1, \ldots, \mu. \tag{EC.32}$$

Then we can couple X_ℓ and Y_ℓ such that $X_\ell \geq Y_\ell$ for all $1 \leq \ell \leq \mu$ in the following way by starting from $\ell = 1$.

If $X_\ell = 1$, i.e., a user $(i,k) \notin S_\ell$ is chosen at the ℓ-th selection, then independently of everything else generate $Y_\ell = 1$ with probability

$$\frac{\mu+\lambda}{2\mu\mathbb{P}\{X_\ell = 1|S_\ell\}} \overset{(EC.31)}{\leq} 1$$

and $Y_\ell = 0$ otherwise. If $X_\ell = 0$, i.e., a user $(i,k) \in S_\ell$ is chosen at the ℓ-th selection, then let $Y_\ell = 0$. Note that by construction,

$$\mathbb{P}\{Y_\ell = 1|S_\ell\} = \mathbb{P}\{X_\ell = 1|S_\ell\}\frac{\mu+\lambda}{2\mu\mathbb{P}\{X_\ell = 1|S_\ell\}} = \frac{\mu+\lambda}{2\mu}.$$

Hence, Y_ℓ is independent of S_ℓ. As a consequence,

$$\begin{aligned}
&\mathbb{P}\{Y_\ell = 1|Y_1, \ldots, Y_{\ell-1}, Y_{\ell+1}, Y_\mu\} \\
&= \mathbb{E}_{S_\ell}\left[\mathbb{P}\{Y_\ell = 1|S_\ell, Y_1, \ldots, Y_{\ell-1}, Y_{\ell+1}, Y_\mu\}\right] \\
&= \mathbb{E}_{S_\ell}\left[\mathbb{P}\{Y_\ell = 1|S_\ell\}\right] = \frac{\mu+\lambda}{2\mu}.
\end{aligned}$$

Thus, Y_ℓ's are independently and identically distributed as specified in (EC.32). Moreover, $Y_\ell \leq X_\ell$. Let $Z_\ell \overset{\text{i.i.d.}}{\sim} \text{Bern}(1/N)$, which are independent of Y_ℓ's. For each ℓ such that $Y_\ell = 1$ or $X_\ell = 1$, the user selected at the ℓ-th selection has not been selected before. Thus, it runs out of tasks and leaves the system immediately with probability equal to $1/N$, independently of everything else. Thus, we can further construct a coupling such that the ℓ-th selected user leaves the system immediately after the ℓ-th selection if and only if $Z_\ell = 1$. Hence, the total number of departures $D(t)$ satisfies that

$$D(t) \geq \sum_{\ell=1}^{\mu} Y_\ell Z_\ell.$$

The proof is complete by letting

$$W \triangleq \sum_{\ell=1}^{\mu} Y_\ell Z_\ell \sim \text{Binom}\left(\mu, \frac{\mu+\lambda}{2N\mu}\right).$$

□

To complete the proof of Theorem 1, we first establish a coupling between $n(t)$ and $m(t)$, which is a Geom/Geom/μ queue with dynamics given by

$$m(t+1) = [m(t) + U(t+1) - D'(t)]_+, \tag{EC.33}$$

where $[x]_+ = \max\{x, 0\}$ and

$$D'(t) \sim \text{Binom}\left(\mu, \frac{\mu+\lambda}{2\mu N}\right).$$

Then, we prove that $\mathbb{E}[m(t)]$ is bounded and $\mathbb{E}[n(t)] \le \mathbb{E}[m(t)] + \tau^* + 1$ for all t in Lemma EC.5 and Lemma EC.6, respectively.

LEMMA EC.5. *The queue $m(t)$ following (EC.33) is stable and satisfies*

$$\mathbb{E}[m(t)] \le \mathbb{E}[m(\infty)] \le \frac{\lambda+\mu}{\mu-\lambda}.$$

where $\mathbb{E}[m(\infty)]$ denotes the mean queue length in steady state.

Proof of Lemma EC.5. We first show that $m(t)$ is stable and bound $\mathbb{E}[m(\infty)]$ from the above. Let

$$b_j = \mathbb{P}\{D'(t) = j\}, \quad \forall j \ge 0.$$

and $\bar{b} = \mathbb{E}[D'(t)]$. Note that $m(t)$ is a discrete-time Markov chain with transition matrix given by

$$P = \begin{bmatrix} 1-a & a & & & & \\ c_1 & d_1 & d_0 & & & \\ c_2 & d_2 & d_1 & d_0 & & \\ c_3 & d_3 & d_2 & d_1 & d_0 & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

where

$$a = \lambda/N,\ d_0 = ab_0,\ d_j = ab_j + (1-a)b_{j-1},\ j \ge 1$$

$$c_j = 1 - \sum_{k=0}^{j} d_k,\ j \ge 1.$$

Since $a < \bar{b}$, it is shown in Neuts (1981) (also cf. (Alfa 2016, Section 4.8.1)) that P is positive recurrent and the steady-state distribution is given by

$$\pi_0 = \frac{b_0(1-r)}{b_0(1-r)+r}, \quad \pi_k = \frac{r^k(1-r)}{b_0(1-r)+r}, \quad \forall k \geq 1,$$

where r is the unique solution in $(0,1)$ of

$$\sum_{k=0}^{\infty} d_k r^{k-1} = 1.$$

Plugging in the expression of d_k, we find that

$$\begin{aligned}\sum_{k=0}^{\infty} d_k r^{k-1} &= \left(1 + a\frac{1-r}{r}\right) \mathbb{E}\left[r^{D'(t)}\right] \\ &= \left(1 + a\frac{1-r}{r}\right)\left(1 - \frac{\bar{b}(1-r)}{\mu}\right)^{\mu},\end{aligned}$$

where the last equality holds in view of the moment generating function of Binomial distribution. Combining the last two displayed equations gives

$$\left(1 + a\frac{1-r}{r}\right)\left(1 - \frac{\bar{b}(1-r)}{\mu}\right)^{\mu} = 1.$$

Using the expression $1 + x \leq e^x$ for all $x \in \mathbb{R}$, we get that

$$e^{a(1-r)/r - \bar{b}(1-r)} \geq 1,$$

which further implies that $r \leq a/\bar{b}$. Hence, the mean queue length in steady state satisfies

$$\begin{aligned}\mathbb{E}\left[m(\infty)\right] &= \sum_{k \geq 1} k\pi_k = \frac{r}{(b_0(1-r)+r)(1-r)} \\ &\leq \frac{1}{1-r} \leq \frac{\bar{b}}{\bar{b}-a} = \frac{\lambda+\mu}{\mu-\lambda}.\end{aligned}$$

Finally, we complete the proof by showing that $\mathbb{E}\left[m(t)\right]$ is non-decreasing in t. Let $Z(t-1) = U(t) - D'(t-1)$. By recursively applying the Lindely's recursion (Ganesh et al. 2004, Lemma1.1):

$$m(t) = \left[m(t-1) + Z(t-1)\right]_+,$$

we get that

$$m(t) = \left[\sum_{s=0}^{t-1} Z(s), \sum_{s=1}^{t-1} Z(s), \ldots, Z(t-1)\right]_+.$$

Let $X(0)=0$ and $X(s)=Z(0)+\ldots+Z(s-1)$. Since $Z(t)$ are *i.i.d.* across t,

$$\begin{aligned} m(t) &\stackrel{d}{=} [Z(0), Z(0)+z(1), \ldots, Z(0)+\ldots+Z(t-1)]_+ \\ &\stackrel{(d)}{=} \max_{0\le s\le t}\{X(s)\}, \end{aligned}$$

where $\stackrel{d}{=}$ means "equal in distribution." Thus,

$$m(t+1) \stackrel{(d)}{=} \max_{0\le s\le t+1}\{X(s)\}$$

first-order stochastically dominates $m(t)$. As a consequence, $\mathbb{E}[m(t+1)] \ge \mathbb{E}[m(t)]$. □

Next, we construct the coupling between $n(t)$ and $m(t)$ and show that $n(t)$ is upper bounded by $m(t)$ plus a constant. Recall that

$$n(t+1) = n(t) + U(t+1) - D(t),$$

where $U(t) \sim \text{Bern}(\lambda/N)$. In view of Lemma EC.4, when $n(t) \ge \tau^*$, we can couple $D(t)$ and $D'(t)$ such that $D(t) \ge D'(t)$. We than have the following lemma.

LEMMA EC.6. *For any time t, the total number of users $n(t)$ in the system is bounded by*

$$n(t) \le m(t) + \tau^* + 1, \tag{EC.34}$$

where $m(t)$ is the Geom/Geom/μ queue defined in (EC.33).

Proof of Lemma EC.6. We prove this by induction. When $t=0$, (EC.34) holds trivially. Suppose (EC.34) holds for t, we prove that it also holds for $t+1$. In particular, if $n(t) \le \tau^*$, since $U(t+1) \le 1$, it holds that $n(t+1) \le \tau^*+1$ and thus (EC.34) holds. If $n(t) \ge \tau^*$, since $D'(t) \le D(t)$, it follows that

$$\begin{aligned} n(t+1) &= n(t) + U(t+1) - D(t) \\ &\le n(t) + U(t+1) - D'(t) \\ &\le m(t) + U(t+1) - D'(t) + \tau^* + 1 \\ &\le m(t+1) + \tau^* + 1. \end{aligned}$$

Therefore, (EC.34) holds for $t+1$ in all cases. □

It immediately follows from Lemma EC.6 and Lemma EC.5 that

$$\begin{aligned}\mathbb{E}[n(t)] &\le \mathbb{E}[m(t)] + \tau^* + 1 \\ &\le \frac{\lambda+\mu}{\mu-\lambda} + \tau^* + 1 \\ &= \frac{2\mu}{\mu-\lambda}\left(1 + \frac{\mu^2\gamma}{\gamma-1}\right) + \mu\gamma V,\end{aligned}$$

where the last equality follows from (EC.29).

EC.4. Bounds on the Expected Number of Client Departures

LEMMA EC.7. *Given* $\mathbf{n}(t) = [n_i(t)]$, *let* $\mathbf{p}(t) = [p_{ij}^k(t)]$ *denote the vector of assignment probabilities computed in Step 2 of the proposed Algorithm 1 at time-slot* t. *Fix a class* i *with* $n_i(t) \ge 1$. *Let* $D_i(t)$ *denote the number of client departures of class* i *at time-slot* t. *Conditioned on* $\mathbf{n}(t)$ *and* $\mathbf{p}(t)$, *the expectation of* $D_i(t)$ *satisfies the following bounds:*

$$\mathbb{E}[D_i(t)|\mathbf{n}(t),\mathbf{p}(t)] \ge \frac{1}{N}\sum_{k=1}^{n_i(t)}\sum_{j=1}^{J} p_{ij}^k(t) - \frac{\gamma^2\mu^2}{2(\gamma-1)^2 n_i(t) N^2}, \tag{EC.35}$$

$$\mathbb{E}[D_i(t)|\mathbf{n}(t),\mathbf{p}(t)] \le \frac{1}{N}\sum_{k=1}^{n_i(t)}\sum_{j=1}^{J} p_{ij}^k(t) \le \frac{\mu}{N}, \tag{EC.36}$$

and

$$\mathbb{E}\left[D_i^2(t) \mid \mathbf{n}(t),\mathbf{p}(t)\right] \le \frac{\mu^2}{N}, \tag{EC.37}$$

where $\mu = \sum_{j=1}^{J}\mu_j$.

The intuition behind this lemma is as follows. Note that in Step 3 of the proposed Algorithm 1, each unit service of server j has $\sum_{k=1}^{n_i(t)} p_{ij}^k(t)/\mu_j$ probability to pick a client from class i. Hence, if all units of the μ_j service rate of all servers j were fully used, the expected number of departures of class i would have been $\frac{1}{N}\sum_{k=1}^{n_i(t)}\sum_{j=1}^{J} p_{ij}^k(t)$, which appears in both (EC.35) and (EC.36). However, note that if a client of class i already runs out of tasks after she is picked by one server, there will be no additional departure when a subsequent server picks this client again. The additional term

$\frac{\gamma^2\mu^2}{2(\gamma-1)^2 n_i(t)N^2}$ in the lower bound (EC.35) accounts for such descrepancy. We also note that this additional term decreases with N and $n_i(t)$, which is true because the larger the values of N and $n_i(t)$, the smaller the departure probability $\sum_{j=1}^{J} p_{ij}^k(t)/(N\mu_j)$ of a particular client, and hence the smaller the discrepancy.

Proof of Lemma EC.7. Denote $E_i^k(t)$ as the event that client k of class i remains in the system at the end of time t, $k=1,...,n_i(t)$. Recall that in Step 3 of our proposed Algorithm 1, each unit service of server j has $p_{ij}^k(t)/\mu_j$ probability of picking client k of class i, independently of other units of service of the same or other servers. Thus, if this client has not run out of tasks yet, it has $p_{ij}^k(t)/(N\mu_j)$ probability to leave the system. We then have

$$\mathbb{P}[E_i^k(t)|\mathbf{n}(t),\mathbf{p}(t)] = \prod_{j=1}^{J}\left(1-\frac{p_{ij}^k(t)}{N\mu_j}\right)^{\mu_j},$$

and

$$\mathbb{E}[D_i(t)|\mathbf{n}(t),\mathbf{p}(t)] = \sum_{k=1}^{n_i(t)}\left(1-\mathbb{P}[E_i^k(t)]\right). \tag{EC.38}$$

Next, we derive the desired lower and upper bounds. In order to derive a lower bound, we use the inequality $1-x\le e^{-x}\le 1-x+x^2/2$ for $x\ge 0$. We have

$$\begin{aligned}1-\mathbb{P}[E_i^k(t)|\mathbf{n}(t),\mathbf{p}(t)] &\ge 1-\exp\left(-\sum_{j=1}^{J}\frac{p_{ij}^k(t)}{N}\right)\\ &\ge \frac{1}{N}\sum_{j=1}^{J}p_{ij}^k(t)-\frac{1}{2N^2}\left(\sum_{j=1}^{J}p_{ij}^k(t)\right)^2.\end{aligned}$$

Hence,

$$\mathbb{E}[D_i(t)|\mathbf{n}(t),\mathbf{p}(t)] \ge \frac{1}{N}\sum_{k=1}^{n_i(t)}\sum_{j=1}^{J}p_{ij}^k(t)-\frac{1}{2N^2}\sum_{k=1}^{n_i(t)}\left(\sum_{j=1}^{J}p_{ij}^k(t)\right)^2. \tag{EC.39}$$

To bound the second term of (EC.39), we now show that, for all $k=1,...,n_i(t)$,

$$\sum_{j=1}^{J}p_{ij}^k(t)\le\frac{\gamma\mu}{(\gamma-1)n(t)}\le\frac{\gamma\mu}{(\gamma-1)n_i(t)}. \tag{EC.40}$$

To see this, recall that $\mathbf{p}(t)$ solves the maximization problem (14) subject to the capacity constraint (15). Let $q_j(t)\ge 0$ be the optimal dual variable corresponding to server j's capacity constraint in

(15). Then, consider any two different clients: client k of class i and client k' of class i', $(i',k') \neq (i,k)$. We have

$$q_j(t)+\gamma-C_{ij}^k(t) \geq q_j(t)+\gamma-1 \geq \frac{\gamma-1}{\gamma}\left(q_j(t)+\gamma-C_{i'j}^{k'}\right),$$

where we have used $C_{ij}^k(t) \leq 1$ and $C_{i'j}^{k'} \geq 0$. Hence, by the KKT condition, we have,

$$\begin{aligned}\sum_{j=1}^J p_{ij}^k(t) &= \frac{1}{V \min_j\{q_j(t)+\gamma-C_{ij}^k(t)\}} \\ &\leq \frac{\gamma/(\gamma-1)}{V \min_j\{q_j(t)+\gamma-C_{i'j}^{k'}\}} = \frac{\gamma}{\gamma-1}\sum_{j=1}^J p_{i'j}^{k'}(t).\end{aligned}$$

By the capacity constraint (15), we have

$$\sum_{i\in\mathcal{I}(t)}\sum_{k=1}^{n_i(t)} p_{ij}^k(t) \leq \mu_j.$$

Thus, summing over all j, we have

$$\begin{aligned}\mu = \sum_{j=1}^J \mu_j &\geq \sum_{i\in\mathcal{I}(t)}\sum_{k=1}^{n_i(t)}\sum_{j=1}^J p_{ij}^k(t) \\ &\geq \sum_{j=1}^J p_{ij}^k(t) + \sum_{\substack{i'\in\mathcal{I}(t),k'=1,\ldots,n_{i'}(t)\\(i',k')\neq(i,k)}}\sum_{j=1}^J p_{i'j}^{k'}(t) \\ &\geq \frac{(\gamma-1)n(t)}{\gamma}\sum_{j=1}^J p_{ij}^k(t).\end{aligned}$$

The inequality (EC.40) then follows. Substituting (EC.40) into (EC.39), we then have,

$$\mathbb{E}[D_i(t)|\mathbf{n}(t),\mathbf{p}(t)] \geq \frac{1}{N}\sum_{k=1}^{n_i(t)}\sum_{j=1}^J p_{ij}^k(t) - \frac{\gamma^2\mu^2}{2(\gamma-1)^2 n_i(t) N^2}.$$

This proves (EC.35).

On the other hand, in order to obtain an upper bound on (EC.38), we iteratively apply the relationship $(1-x)(1-y) \geq 1-(x+y)$ for all $x,y \geq 0$. We then obtain

$$\mathbb{P}[E_i^k(t)|\mathbf{n}(t),\mathbf{p}(t)] \geq 1-\sum_{j=1}^J \mu_j \frac{p_{ij}^k}{N\mu_j} = 1-\frac{1}{N}\sum_{j=1}^J p_{ij}^k.$$

Substituting into (EC.38), the inequality (EC.36) then follows.

Finally, to show (EC.37), we use $D_i(t) \leq \sum_{j=1}^J \mu_j = \mu$. Combining with (EC.36), we then have

$$\mathbb{E}\left[D_i^2(t) \mid \mathbf{n}(t),\mathbf{p}(t)\right] \leq \mathbb{E}\left[D_i(t) \mid \mathbf{n}(t),\mathbf{p}(t)\right]\mu \leq \frac{\mu^2}{N}.$$

□

EC.5. Chernoff-Hoeffding Bound

We cite the following lemma from Fact 1 in Auer et al. (2002a), which is simply the Chernoff-Hoeffding bound.

LEMMA EC.8. *Let $X(s), s=1,2,...$ be a sequence of* i.i.d. *Bernoulli random variables with mean C^*_{ij}. For any given positive integer a and any real number $t \geq 1$, the followings hold:*

$$\mathbb{P}\left\{\frac{\sum_{s=1}^{a} X(s)}{a} - C^*_{ij} \geq \sqrt{\frac{2\log t}{a}}\right\} \leq \frac{1}{t^4},$$

and

$$\mathbb{P}\left\{\frac{\sum_{s=1}^{a} X(s)}{a} - C^*_{ij} \leq -\sqrt{\frac{2\log t}{a}}\right\} \leq \frac{1}{t^4}.$$

EC.6. Derivation of the Continuous-Payoff Oracle Upper Bound

In this section, we derive the oracle upper bound under the following continuous payoff setting. First, assume each client l has a payoff vector $(C^*_{l,1}, C^*_{l,2})$ independently and uniformly distributed over the lower triangular part of the unit square $\Omega = \{(x,y) \in [0,1]^2 : x \geq y\}$. In other words, let X_1 and X_2 denote two independent random variables uniformly distribued over $[0,1]$. Then, $(C^*_{l,1}, C^*_{l,2})$ has the same distribution as $(X_{(1)}, X_{(2)})$, where $X_{(1)} = \max\{X_1, X_2\}$ and $X_{(2)} = \min\{X_1, X_2\}$.

Note that all clients prefer server 1. However, the total arrival rate of tasks is 1.2, while the server 1 has service rate 1. Thus, the system has to assign tasks from some clients to server 2. To maximize the total expected payoff, it is preferable to assign tasks from those clients l with a small gap $C^*_{l,1} - C^*_{l,2}$ to server 2. In particular, the optimal assignment is the following threshold policy. For a threshold τ to be determined, assign client l to server 2 if $C^*_{l,1} - C^*_{l,2} < \tau$; otherwise assign client l to server 1. In other words, assign client l to server 1 if and only if its payoff vector $(C^*_{l,1}, C^*_{l,2})$ falls into the region $\Omega_1 \triangleq \{(x,y) \in [0,1]^2 : x \in [\tau, 1], y \in [0, x-\tau]\}$. Since the area of Ω_1 is $(1-\tau)^2/2$, while the area of Ω is $1/2$, it follows that according to this threshold policy, the fraction of tasks assigned to server 1 is given by $(1-\tau)^2$. Thus, to satisfy the capacity constraint of server 1, we need $(1-\tau)^2 = 1/1.2$. Next, we calculate the total expected payoff under this threshold policy.

Note that since $\mathbb{E}\left[X_{(1)}\right]=2/3$, if all tasks were assigned to server 1, the expected payoff per task would be 2/3. Thus, it suffices to calculate the expected payoff loss per task assigned to server 2, which is given by

$$2\int_{\Omega\setminus\Omega_1}(x-y)dxdy,$$

where the factor of 2 is due to the fact that $(X_{(1)},X_{(2)})$ is uniformly distributed over Ω and hence its probability density function is equal to 2. Note that

$$\begin{aligned}2\int_{\Omega\setminus\Omega_1}(x-y)dxdy&=2\int_0^{\tau}dx\int_0^{x}(x-y)dy+2\int_{\tau}^{1}dx\int_{x-\tau}^{x}(x-y)dy\\&=\int_0^{\tau}x^2dx+\int_{\tau}^{1}\tau^2dx\\&=\frac{1}{3}\tau^3+\tau^2(1-\tau)=\tau^2-\frac{2}{3}\tau^3.\end{aligned}$$

In conclusion, the total expected payoff is given by

$$1.2\times\left(\frac{2}{3}-\tau^2+\frac{2}{3}\tau^3\right)=1.4-\frac{5}{9}\times\sqrt{1.2}\approx 0.7914.$$

In comparision, if a myopic matching strategy is used, with close-to-1 probability two tasks will be assigned at each time, one of which will have to be assigned to server 2. Thus, the overall expected payoff is approximately upper-bounded by

$$1.2\times\left(\mathbb{E}\left[X_{(1)}\right]+\mathbb{E}\left[X_{(2)}\right]\right)/2=1.2\times\left(\frac{2}{3}+\frac{1}{3}\right)/2=0.6,$$

which is consistent with simulation results in Fig. 3(d).

EC.7. Dealing with Server-Side Uncertainty

In the main body of the paper, we have assumed that the service at each server is deterministic. Specifically, each server j serves exactly μ_j tasks per time-slot. While this assumption simplifies the design of Algorithm 1, i.e., the deterministic service capacity μ_j directly goes into the constraint (8), it may also limit the applicability of the algorithm in practical situations where the service of each server may be random and the mean service rate of the server may be unknown. In this

section, we propose a heuristic solution that extends Algorithm 1 to such more general settings with server-side uncertainty.

The intuition behind the proposed heuristics is that, since Algorithm 1 is a convex program, we can develop a dual-based solution, where dual-variables are updated based on the actual number of tasks served in each time-slot. Thus, such an approach can be used even when the service is random and the mean service rate is unknown. We start from the convex program (7) for Algorithm 1, assuming that μ_j is given and $C_j^l(t)$ is fixed for all time t. We associate a dual variable q_j for the constraint (8). Let $\mathbf{q} = [q_j, j \in \mathcal{S}]$. It is easy to show that the following dual-based algorithm will converge to the optimal dual solution as $t \to \infty$. At each step t, based on $\vec{q}(t)$, each client l determines its assignment probabilities so that they satisfy the following:

$$\sum_{j=1}^{J} p_j^l(t) = \frac{1}{V \min_{j\in\mathcal{S}}\{q_j(t) + \gamma - C_j^l(t)\}}, \tag{EC.41}$$

$$p_j^l(t) \geq 0, \text{ if } q_j(t) + \gamma - C_j^l(t) = \min_{j'}\{q_{j'}(t) + \gamma - C_{j'}^l(t)\}, \tag{EC.42}$$

$$p_j^l(t) = 0, \text{ otherwise.} \tag{EC.43}$$

Then, the dual variables are updated by:

$$q_j(t+1) = \left[q_j(t) + \alpha\left(\sum_{l=1}^{n(t)} p_j^l(t) - \mu_j\right)\right]^+, \tag{EC.44}$$

where α is a positive step-size, and $p_j^l(t)$ is the primal variable computed according to (EC.41)-(EC.43). For details of the derivation, see Appendix EC.8.

Clearly, the only part of the above iteration that depends on μ_j is (EC.44). To model the setting with server-side uncertainty, we assume that the number of tasks that server j can complete in time-slot t is a random variable $d_j(t)$ with mean $\mu_j = \mathbb{E}[d_j(t)]$. We assume that $d_j(t)$ is independent across servers and *i.i.d.* in time. Thus, we replace (EC.44) by the following noisy version:

$$q_j(t+1) = \left[q_j(t) + \alpha\left(\sum_{l=1}^{n(t)} Y_j^l(t) - d_j(t)\right)\right]^+, \tag{EC.45}$$

where $Y_j^l(t)$ is the actual number of tasks sent from client l to server j at time-slot t (with $\mathbb{E}\left[Y_j^l(t)\right] = p_j^l(t)$). In this way, the dual update in (EC.45) can be applied even when the service is random and

the mean service rate μ_j is unknown. Further, instead of solving the convex problem (7) exactly (which requires running the dual iterations (EC.45) a large number of times within a time-slot t), we can run only one iteration of (EC.45) per time-slot t, rendering the time complexity even lower. This idea is inspired by the removal of time-scale separation in flow-level congestion control (Lin et al. 2008). There, it was shown that the maximum flow-level stability can still be maintained with only one iteration of the congestion control algorithm implemented in each time-slot. Here, we show through simulations that a similar idea can be applied to task assignment without sacrificing significant payoff loss.

Readers familiar with the discussions in Section EC.1 will immediately notice that the dual variable $q_j(t)$ is closely related to the task queue length at server j. Specifically, if we let

$$Z_j(t+1) = \left[Z_j(t) + \sum_{l=1}^{n(t)} Y_j^l(t) - d_j(t) \right]^+, \tag{EC.46}$$

then $Z_j(t)$ is precisely the length of the task queue at server j at time t. Thus, $q_j(t)$ and $Z_j(t)$ are simply related by $q_j(t) = \alpha Z_j(t)$. Recall from Section EC.1 that this task queue also delays the payoff feedback for learning, which leads to the potentially-detrimental closed-loop interaction between queueing and learning. However, here this feedback delay is better controlled because, whenever $q_j(t)$ gets large, the assignment probabilities $p_j^l(t)$ will become small according to (EC.41). This negative feedback ensures that the delay for learning will likely not be very large. In order to further control this delay, we propose the following "virtual queue" concept. At each time-slot, as new tasks are being assigned from the clients to each server (according to the probabilities $p_j^l(t)$), whenever the task queue at a server reaches a limit $Q_{\max}$, further tasks assigned to the server will be declined and returned to the corresponding client. In this way, the actual task queue at each server will never exceed $Q_{\max}$. Let $r_j^l(t)$ be the number of tasks returned to client l by server j in time-slot t. Then, the "real" queue at each server evolves as

$$Q_j(t+1) = \left[Q_j(t) + \sum_{l=1}^{n(t)} Y_j^l(t) - \sum_{l=1}^{n(t)} r_j^l(t) - d_j(t) \right]^+,$$

while $Z_j(t)$ can be thought of as the "virtual" queue. Note that the assign probabilities are still computed in (EC.41)-(EC.43) based on $q_j(t) = \alpha Z_j(t)$, which corresponds to the virtual queue.

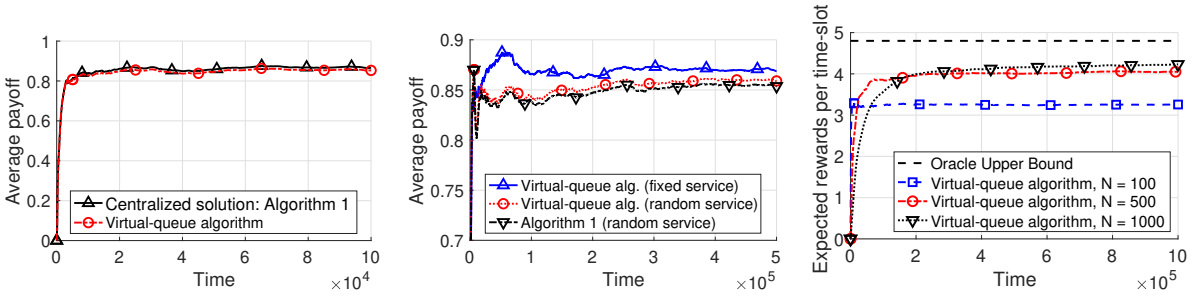
EC.7.1. Simulation Studies of the Virtual-Queue Algorithm

Below, we will use simulation to demonstrate that the above virtual-queue algorithm is computationally efficient and enjoys similar performance as Algorithm 1 for the case when the service is deterministic. In addition, it can achieve comparable performance even when the service is random and the mean service rate is unknown. For simplicity, we will use $\alpha = 1/V$ for the rest of the study in this section.

Payoff Performance: We first evaluate the performance of the virtual-queue algorithm for the case when the service of each server is deterministic, i.e., $d_j(t) = \mu_j$ for all t. In this way, we can directly compare the performance of the virtual-queue algorithm with that of Algorithm 1, so that we can understand how the system performance is affected by the learning delay (due to task queues) and the fact that only one iteration of the dual update is executed in each time-slot. Specifically, we use the same 2-class 2-server setup as in Section 5. For both algorithms, we use $N = 100$, $V = 21$, $\gamma = 1.1$. Further, for the virtual-queue algorithm, we use $Q_{\max} = 100$. In Fig. EC.1(a), we show the average payoff of both Algorithm 1 and the virtual queue algorithm. We observe that the virtual-queue algorithm has a slightly smaller average payoff than Algorithm 1. However, the difference is small, which suggests that the impact of learning delay and one-iteration-per-slot is not significant. We also plot the queue histogram and the feedback delay of the system, and observe that they are not very large under the virtual-queue algorithm. See Fig. EC. 2(a) and more discussions later.

We then simulate the virtual-queue algorithm when the service of each server is random and the mean service rate is unknown to the operator. To simulate random service, we assume that the number of tasks completed by server j in time-slot t is a discrete *i.i.d.* uniform random variable taken values from $\{0, 1, 2\}$. Fig. EC.1(b) shows the payoff performance of the virtual-queue algorithm when the service is random. For comparisons, we also show the performance of the virtual-queue algorithm when the service is deterministic and is equal to the mean 1 at all times. Further, we simulate a simple extension of Algorithm 1 that replaces μ_j in (8) by the current

amount of service $d_j(t)$. We observe a small performance degradation for the virtual-queue algorithm when the service changes from deterministic to random. However, with random service, the virtual-queue algorithm still achieves a slightly higher expected payoff than the simple extension of Algorithm 1, while at the same time enjoys much lower computational complexity and applies to even distributed settings.

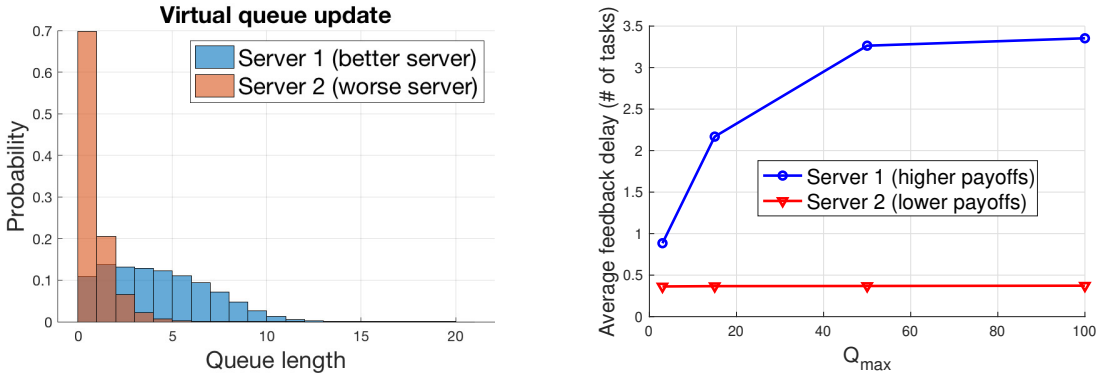


(a) (b) (c)

Figure EC.1 Plot (a): The virtual-queue algorithm only incurs slight performance degradation compared to Algorithm 1. Plot (b): With random service, the virtual-queue algorithm slightly outperforms a simple extension of Algorithm 1 that uses the current amount of service at time t. Plot (c): The proposed low-complexity algorithm can be easily applied to large systems without increasing runtime complexity.

Feedback Delay and Queue Backlog: To further observe the effect of the learning delay, we plot the histogram of the server queue lengths in Fig. EC.2(a). Note that under the oracle algorithm (with exact payoff vectors revealed), server 1 (with higher payoff) and server 2 should be utilized 100% and 20% of the time, respectively. Thus, one may be concerned that the busier server 1 may have a large task queue under the virtual-queue algorithm. From Fig. EC.2(a), we can observe that the length of the task queue at server 1 is not large at all, which limits the feedback delay for learning.

In fact, the queue-occupation probability decays quickly with the queue length. We believe that the reason behind this behavior is the negative feedback introduced by (EC.41)-(EC.43). That is, when the queue at a server increases, the likelihood that more tasks are routed to it also decreases. As a result, the task-queue length at the server can be self-regulated to keep the learning delay



(a) (b)

Figure EC.2 Plot (a): The queue histogram shows that the preferred server 1 does not become very large, which helps to limit feedback delay. Plot (b):The feedback delay in terms of the number of samples waiting for completion for all clients (averaged over time) saturates as $Q_{\max}$ is increased.

of each client to be small. To capture the feedback delay of the system, we collect the number of tasks of a given client queued at the servers at time t. We take the maximum across all clients, and then averaged over all time t. The result is shown in Fig. EC.2(b) with different $Q_{\max}$. The result suggests that the UCB estimates used for task assignment are only delayed by 4 samples on average. Thus, we expect that the task assignment decisions are close to the centralized solution Algorithm 1.

A Larger Example: Finally, note that the virtual-queue algorithm incurs significantly lower complexity than Algorithm 1, which allows us to easily evaluate it for large systems. Here, we simulate a 10-class and 10-server system, where each class has a unique payoff distribution profile for the servers. We assume that tasks arrive at the system with rate 6 (i.e., $\lambda = 6$), and each server has the service capacity $\mu_j = 1$. The probability that a client belongs to each class is equiprobable, and the different classes have different underlying payoff vectors $[C^*_{ij}]$ associate with servers. Specifically,

we use

$$[C^*_{ij}] = \begin{bmatrix} 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\ 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.9 & 0.3 & 0.9 & 0.3 & 0.9 & 0.3 & 0.9 & 0.3 & 0.9 & 0.3 \\ 0.9 & 0.1 & 0.9 & 0.1 & 0.9 & 0.1 & 0.9 & 0.1 & 0.9 & 0.1 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 \\ 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.9 & 0.8 & 0.7 & 0.6 & 0.5 & 0.4 & 0.3 & 0.2 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 \\ 0.0 & 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 & 0.9 \end{bmatrix}.$$

Assuming that the system information is known, we can obtain the payoff upper-bound of 4.8 by solving (4)-(6). This upper bound is shown as the dashed line in Fig. EC.1(c).

The payoff performance of the virtual-queue algorithm is shown in Fig. EC.1(c) for varying N. With $N = 100$, the algorithm can only achieve 68% of the upper bound performance, while in contrast, the 2-class 2-server case studied earlier can achieve 91% of the upper bound under the same parameter $V = 21$ (see Fig. EC.3(a)). The gap from the upper bound becomes much smaller when $N = 500$ and $N = 1000$. This is because, when there are more servers and classes, more tasks are needed for "exploration" before the operator can distinguish between different classes of clients with high confidence.

EC.8. Derivation of the Dual Updates

In order to derive the dual-based heuristics, we start from the convex program (7) for Algorithm 1, assuming that μ_j is given and $C^l_j(t)$ is fixed for all time t. We associate a dual variable q_j for the constraint (8). Letting $\mathbf{q} = [q_j, j \in \mathcal{S}]$, the Lagrangian can then be written as

$$\begin{aligned} L(\mathbf{p}, \mathbf{q}) &= \sum_{l=1}^{n(t)} \left\{ \frac{1}{V} \log \left(\sum_{j=1}^{J} p^l_j \right) + \sum_{j=1}^{J} p^l_j (C^l_j - \gamma) \right\} - \sum_{j=1}^{J} q_j \left(\sum_{l=1}^{n(t)} p^l_j - \mu_j \right) \\ &= \sum_{l=1}^{n(t)} \frac{1}{V} \log \left(\sum_{j=1}^{J} p^l_j \right) - \sum_{l=1}^{n(t)} \sum_{j=1}^{J} p^l_j (q_j - C^l_j + \gamma) + \sum_{j=1}^{J} q_j \mu_j. \end{aligned} \tag{EC.47}$$

Given the current set of dual variables $\mathbf{q}(t) = [q_j(t)]$, maximizing the Lagrangian over the primal variables $\mathbf{p} \geq 0$ yields the objective $D(\mathbf{q}(t))$ of the dual problem. The solution to this maximization step has the following simple structure:

$$\sum_{j=1}^{J} p_j^l(t) = \frac{1}{V \min_{j \in \mathcal{S}}\{q_j(t) + \gamma - C_j^l(t)\}}, \tag{EC.48}$$

$$p_j^l(t) \geq 0, \text{ if } q_j(t) + \gamma - C_j^l(t) = \min_{j'}\{q_{j'}(t) + \gamma - C_{j'}^l(t)\}, \tag{EC.49}$$

$$p_j^l(t) = 0, \text{ otherwise.} \tag{EC.50}$$

Note that given $\mathbf{q}(t)$, each user ℓ can determine its assignment probabilities independently. The corresponding primal solution may not be unique. Specifically, for each user ℓ, if there is more than one server j satisfying the condition in (EC.49), the assigned probabilities break arbitrarily so that (EC.48) is satisfied. Finally, to minimize the dual $D(\mathbf{q}(t))$, we can use the following dual update:

$$q_j(t+1) = \left[q_j(t) + \alpha \left(\sum_{l=1}^{n(t)} p_j^l(t) - \mu_j \right) \right]^+, \tag{EC.51}$$

where α is a positive step-size, and $p_j^l(t)$ is the primal variable computed according to (EC.48)-(EC.50). It is well-known that, for a fixed instance of the optimization problem (7), i.e., if the set of clients and their payoff-estimates $C_j^l(t)$ were fixed for all time t, the above iteration would converge to the optimal dual solution as $t \to \infty$.