D-STOP

**Technical Report 127**

# Online Matching with Queueing Dynamics

**Research Supervisor**
Sanjay Shakkottai
Wireless Networking and Communications Group

December 2016

Project Title: *Online Learning for Freight—An Examination of Queueing Regret*

# Data-Supported Transportation Operations & Planning Center (D-STOP)

A Tier 1 USDOT University Transportation Center at The University of Texas at Austin

D-STOP is a collaborative initiative by researchers at the Center for Transportation Research and the Wireless Networking and Communications Group at The University of Texas at Austin.

**Technical Report Documentation Page**

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| D-STOP/2016/127 | | |

4. Title and Subtitle
Online Matching with Queueing Dynamics

5. Report Date
December 2016

6. Performing Organization Code

7. Author(s)
Subhashini Krishnasamy, Rajat Sen, Ramesh Johari (Stanford University), and Sanjay Shakkottai

8. Performing Organization Report No.
Report 127

9. Performing Organization Name and Address
Data-Supported Transportation Operations & Planning Center (D-STOP)
The University of Texas at Austin
1616 Guadalupe Street, Suite 4.202
Austin, Texas 78701

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
DTRT13-G-UTC58

12. Sponsoring Agency Name and Address
Data-Supported Transportation Operations & Planning Center (D-STOP)
The University of Texas at Austin
1616 Guadalupe Street, Suite 4.202
Austin, Texas 78701

13. Type of Report and Period Covered

14. Sponsoring Agency Code

15. Supplementary Notes
Supported by a grant from the U.S. Department of Transportation, University Transportation Centers Program.
Project Title: Online Learning for Freight—An Examination of Queueing Regret

16. Abstract
We consider a variant of the multiarmed bandit problem where jobs queue for service, and service rates of different servers may be unknown. We study algorithms that minimize queue-regret: the (expected) difference between the queue-lengths obtained by the algorithm, and those obtained by a "genie"-aided matching algorithm that knows exact service rates. A naive view of this problem would suggest that queue-regret should grow logarithmically: since queue-regret cannot be larger than classical regret, results for the standard MAB problem give algorithms that ensure queue-regret increases no more than logarithmically in time. Our paper shows surprisingly more complex behavior. In particular, the naive intuition is correct as long as the bandit algorithm's queues have relatively long regenerative cycles: in this case queue-regret is similar to cumulative regret, and scales (essentially) logarithmically. However, we show that this "early stage" of the queueing bandit eventually gives way to a \late stage", where the optimal queue-regret scaling is $O(1=t)$. We demonstrate an algorithm that (order-wise) achieves this asymptotic queue-regret, and also exhibits close to optimal switching time from the early stage to the late stage.

| 17. Key Words | 18. Distribution Statement |
|---|---|
| queueing regret, matching algorithm, bandit methods, multiarmed bandit, queueing bandit | No restrictions. This document is available to the public through NTIS (http://www.ntis.gov):<br>National Technical Information Service<br>5285 Port Royal Road<br>Springfield, Virginia 22161 |

| 19. Security Classif.(of this report) | 20. Security Classif.(of this page) | 21. No. of Pages | 22. Price |
|---|---|---|---|
| Unclassified | Unclassified | | |

## Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation's University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

## Acknowledgements

The authors recognize that support for this research was provided by a grant from the U.S. Department of Transportation, University Transportation Centers.

# Online Matching with Queueing Dynamics

Subhashini Krishnasamy[1], Rajat Sen[1], Ramesh Johari[2], and Sanjay Shakkottai[1]

[1]The University of Texas at Austin
[2]Stanford University

**Abstract**

We consider a variant of the multiarmed bandit problem where jobs *queue* for service, and service rates of different servers may be unknown. We study algorithms that minimize *queue-regret*: the (expected) difference between the queue-lengths obtained by the algorithm, and those obtained by a "genie"-aided matching algorithm that knows exact service rates. A naive view of this problem would suggest that queue-regret should grow logarithmically: since queue-regret cannot be larger than classical regret, results for the standard MAB problem give algorithms that ensure queue-regret increases no more than logarithmically in time. Our paper shows surprisingly more complex behavior. In particular, the naive intuition is correct as long as the bandit algorithm's queues have relatively long regenerative cycles: in this case queue-regret is similar to cumulative regret, and scales (essentially) logarithmically. However, we show that this "early stage" of the queueing bandit eventually gives way to a "late stage", where the optimal queue-regret scaling is $O(1/t)$. We demonstrate an algorithm that (order-wise) achieves this asymptotic queue-regret, and also exhibits close to optimal switching time from the early stage to the late stage.

## 1 Introduction

Stochastic multi-armed bandits (MAB) have a rich history in sequential decision making [1, 2, 3]. In its simplest form, a collection of $K$ arms are present, each having a binary reward (Bernoulli random variable over $\{0, 1\}$) with an unknown success probability[1] (and different across arms). At each (discrete) time, a single arm is chosen by the bandit algorithm, and a (binary-valued) reward is accrued. The MAB problem is to determine which arm to choose at each time in order to minimize the cumulative expected regret, namely, the cumulative loss of reward when compared to a genie that has knowledge of the arm success probabilities.

In this paper, we consider the variant of this problem motivated by *queueing* applications. Formally, suppose that arms are pulled upon arrivals of *jobs*; each arm is now a *server* that can serve the arriving job. In this model, the stochastic reward described above is equivalent to *service*. In other words, if the arm (server) that is chosen results in positive reward, the job is successfully completed and departs the system. However, this basic model fails to capture an essential feature of service in many settings: in a queueing system, *jobs wait until they complete service*. Such systems are *stateful*: when the chosen arm results in zero reward, the job being served remains in the queue, and over time the model must track the remaining jobs waiting to be served. The difference between the cumulative number of arrivals and departures, or the *queue length*, is the most common measure of the quality of the service strategy being employed.

[1]Here, the success probability of an arm is the probability that the reward equals '1'.

Queueing is employed in modeling a vast range of service systems, including supply and demand in online platforms (e.g., Uber, Lyft, Airbnb, Upwork, etc.); order flow in financial markets (e.g., limit order books); packet flow in communication networks; and supply chains. In transportation systems, applications include matching passengers to cars (e.g. Uber, Lyft), or goods from shippers to carrier trucks in a freight matching system. In all of these systems, queueing is an essential part of the model: e.g., in online platforms, the available supply (e.g. available drivers in Uber or Lyft, or available rentals in Airbnb) queues until it is "served" by arriving demand (ride requests in Uber or Lyft, booking requests in Airbnb). Since MAB models are a natural way to capture learning in this entire range of systems, incorporating queueing behavior into the MAB model is an essential challenge.

This problem clearly has the explore-exploit tradeoff inherent in the standard MAB problem: since the success probabilities across different servers are unknown, there is a tradeoff between learning (*exploring*) the different servers and (*exploiting*) the most promising server from past observations. We refer to this problem as the *queueing bandit*. Since the queue length is simply the difference between the cumulative number arrivals and departures (cumulative actual reward; here reward equals job service), the natural notion of regret here is to compare the expected queue length under a bandit algorithm with the corresponding one under a genie policy (with identical arrivals) that however always chooses the arm with the highest expected reward.

Formally, let $Q(t)$ be the queue length at time $t$ under a given bandit algorithm, and let $Q^*(t)$ be the corresponding queue length under the "genie" policy that always schedules the optimal server (i.e. always plays the arm with the highest mean). We define the *queue-regret* as the difference in expected queue lengths for the two policies. That is, the regret is given by:

$$\Psi(t) := \mathbb{E}\left[Q(t) - Q^*(t)\right]. \tag{1}$$

Here $\Psi(t)$ has the interpretation of the traditional MAB regret with caveat that rewards are accumulated only if there is a job that can benefit from this reward. We refer to $\Psi(t)$ as the *queue-regret*; formally, our goal is to develop bandit algorithms that minimize the queue-regret.

To develop some intuition, we compare this to the standard stochastic MAB problem. For the standard problem, well-known algorithms such as UCB, KL-UCB, and Thompson sampling achieve a cumulative regret of $O((K-1)\log t)$ at time $t$ [4, 5, 6], and this result is essentially tight: there exists a lower bound of $\Omega((K-1)\log t)$ over all policies in a reasonable class, so-called $\alpha$-consistent policies [7]. In the queueing bandit, we can obtain a simple bound on the queue-regret by noting that it cannot be any higher than the traditional regret (where a reward is accrued at each time whether a job is present or not). This leads to an upper bound of $O((K-1)\log t)$ for the queue regret.

However, this upper bound does not tell the whole story for the queueing bandit: we show that there are two "stages" to the queueing bandit. In the *early* stage, the bandit algorithm is unable to even stabilize the queue – effectively, the expected success probability (expectation taken over both the arm Bernoulli random variables and the arm selection by the bandit algorithm) is smaller than the arrival probability of a job. Thus, on average, the queue length increases over time and is continuously backlogged (because the arrival rate exceeds the expected service rate), leading to an "unstable" growth of the queue; therefore the queue-regret grows with time, similar to the cumulative regret. Once the algorithm is able to stabilize the queue—the *late* stage—then a dramatic shift occurs in the behavior of the queue regret. A stochastically stable queue goes through **regenerative cycles** – a random cyclical behavior where queues build-up over time, then empty, and the cycle repeats. The associated recurring"zero-queue-length" epochs means that sample-path queue-regret essentially "resets" at (stochastically) regular intervals; i.e., the sample-path queue-regret becomes zero or below zero at these time instants. Thus the queue-regret should fall over time, as the algorithm learns.

Our main results provide lower bounds on queue-regret for both the early and late stages, as well as algorithms that essentially match these lower bounds. We first describe the late stage, and then describe the early stage for a heavily loaded system.

**1. The late stage**. We first consider what happens to the queue regret as $t \to \infty$. As noted above, a reasonable intuition for this regime comes from considering a standard bandit algorithm, but where the sample-path queue-regret "resets" at time points of regeneration.[2] In this case, the queue-regret is approximately a (discrete) *derivative* of the cumulative regret. Since the optimal cumulative regret scales like $\log t$, asymptotically the optimal queue-regret should scale like $1/t$. Indeed, we show that the queue-regret for $\alpha$-consistent policies is at least $C/t$ infinitely often, where $C$ is a constant independent of $t$. Further, we introduce an algorithm called Q-ThS for the queueing bandit (a variant of Thompson sampling with explicit structured exploration), and show an asymptotic regret upper bound of $O\left(\text{poly}(\log t)/t\right)$ for Q-ThS, thus matching the lower bound up to poly-logarithmic factors in $t$. Q-ThS exploits *structured exploration*: we exploit the fact that the queue regenerates regularly to explore more systematically and aggressively.

**2. The early stage**. The preceding discussion might suggest that an algorithm that explores aggressively would dominate any algorithm that balances exploration and exploitation. However, this intuition would be incorrect, because an overly aggressive exploration policy will preclude the queueing system from ever stabilizing, which is *necessary* to induce the regenerative cycles that lead the system to the late stage. As a simple example, it is well known that if the only goal is to identify the best out of two servers as fast as possible, the optimal algorithm is a balanced randomized experiment (with half the trials on one server, and half on the other) [8]. But such an algorithm has positive probability of failing to stabilize the queue, and so the queue-regret will grow over time.

To even enter the late stage, therefore, we need an algorithm that exploits enough to actually stabilize the queue (i.e. choose good arms sufficiently often so that the mean service rate exceeds the expected arrival rate). We refer to the early stage of the system, as noted above, as the period before the algorithm has learned to stabilize the queues. For a *heavily loaded system, where the arrival rate approaches the service rate of the optimal server,* we show a lower bound of $\Omega(\log t/\log\log t)$ on the queue-regret in the early stage. Thus up to a $\log\log t$ factor, the early stage regret behaves similarly to the cumulative regret (which scales like $\log t$). The heavily loaded regime is a natural asymptotic regime in which to study queueing systems, and has been extensively employed in the literature; see, e.g., [9, 10] for surveys.

Perhaps more importantly, our analysis shows that the time to switch from the early stage to the late stage scales at least as $t = \Omega(K/\epsilon)$, where $\epsilon$ is the gap between the arrival rate and the service rate of the optimal server; thus $\epsilon \to 0$ in the heavy-load setting. In particular, we show that the early stage lower bound of $\Omega(\log t/\log\log t)$ is valid up to $t = O(K/\epsilon)$; on the other hand, we also show that, in the heavy-load limit, depending on the relative scaling between $K$ and $\epsilon$, the regret of Q-ThS scales like $O\left(\text{poly}(\log t)/\epsilon^2 t\right)$ for times that are arbitrarily close to $\Omega(K/\epsilon)$. In other words, Q-ThS is nearly optimal in the time it takes to "switch" from the early stage to the late stage.

Our results constitute the first insight into the behavior of regret in this queueing setting; as emphasized, it is quite different than that seen for minimization of cumulative regret in the standard MAB problem. The preceding discussion highlights why minimization of queue-regret presents a

[2] This is inexact since the optimal queueing system and bandit queueing system may not regenerate at the same time point; but the intuition holds.

subtle learning problem. On one hand, if the queue has been stabilized, the presence of regenerative cycles allows us to establish that queue regret must eventually decay to zero at rate $1/t$ under an optimal algorithm (the late stage). On the other hand, to actually have regenerative cycles in the first place, a learning algorithm needs to exploit enough to actually stabilize the queue (the early stage). Our analysis not only characterizes regret in both regimes, but also essentially exactly characterizes the transition point between the two regimes. In this way the queueing bandit is a remarkable new example of the tradeoff between exploration and exploitation.

## 2 Related work

**MAB algorithms**. Stochastic MAB models have been widely used in the past as a paradigm for various sequential decision making problems in industrial manufacturing, communication networks, clinical trials, online advertising and webpage optimization, and other domains requiring resource allocation and scheduling; see, e.g., [1, 2, 3]. The MAB problem has been studied in two variants, based on different notions of optimality. One considers mean accumulated loss of rewards, often called *regret*, as compared to a genie policy that always chooses the best arm. Most effort in this direction is focused on getting the best regret bounds possible at any *finite time* in addition to designing computationally feasible algorithms [3]. The other line of research models the bandit problem as a Markov decision process (MDP), with the goal of optimizing *infinite horizon* discounted or average reward. The aim is to characterize the structure of the optimal policy [2]. Since these policies deal with optimality with respect to infinite horizon costs, unlike the former body of research, they give steady-state and not finite-time guarantees. Our work uses the regret minimization framework to study the queueing bandit problem.

**Bandits for queues**. There is body of literature on the application of bandit models to queueing and scheduling systems [2, 11, 12, 13, 14, 15, 16, 17]. These queueing studies focus on infinite-horizon costs (i.e., statistically steady-state behavior, where the focus typically is on conditions for optimality of index policies); further, the models do not typically consider user-dependent server statistics. Our focus here is different: algorithms and analysis to optimize finite time regret.

## 3 Problem Setting

We consider a discrete-time queueing system with a single queue and $K$ servers. The servers are indexed by $k = 1, \ldots, K$. Arrivals to the queue and service offered by the links are according to product Bernoulli distribution and i.i.d. across time slots. The mean arrival rate is given by $\lambda$ and the mean service rates by the vector $\boldsymbol{\mu} = [\mu_k]_{k\in[K]}$, with $\lambda < \max_{k\in[K]} \mu_k$. In any time slot, the queue can be served by at most one server and the problem is to schedule a server in every time slot. The scheduling decision at any time $t$ is based on past observations corresponding to the services obtained from the scheduled servers until time $t-1$. Statistical parameters corresponding to the service distributions are considered unknown. The queueing system evolution can be described as follows. Let $\kappa(t)$ denote the server that is scheduled at time $t$. Also, let $R_k(t)$ be the service offered by server $k$ and $S(t)$ denote the service offered by server $\kappa(t)$ at time $t$, i.e., $S(t) = R_{\kappa(t)}(t)$. If $A(t)$ is the number of arrivals at time $t$, then the queue-length at time $t$ is given by:

$$Q(t) = (Q(t-1) + A(t) - S(t))^+ .$$

Our goal in this paper is to focus attention on how queueing behavior impacts regret minimization in bandit algorithms. We evaluate the performance of scheduling policies against the policy

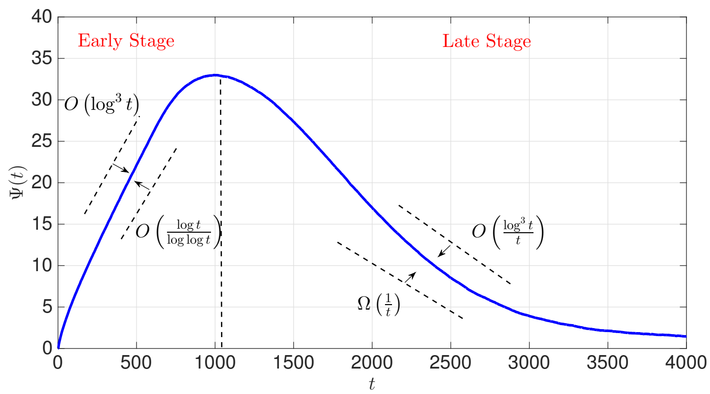



Figure 1: Queue-regret $\Psi(t)$ under Q-ThS in a system with $K = 5$, $\epsilon = 0.1$ and $\Delta = 0.17$

that schedules the (unique) optimal server in every time slot, i.e., the server $k^* := \arg\max_{k \in [K]} \mu_k$ with the maximum mean rate $\mu^* := \max_{k \in [K]} \mu_k$. Let $Q(t)$ be the queue-length vector at time $t$ under our specified algorithm, and let $Q^*(t)$ be the corresponding vector under the optimal policy. We define *regret* as the difference in mean queue-lengths for the two policies. That is, the regret is given by:

$$\Psi(t) := \mathbb{E}\left[Q(t) - Q^*(t)\right].$$

We use the terms *queue-regret* or simply *regret* to refer to $\Psi(t)$.

Throughout, when we evaluate queue-regret, we do so under the assumption that the queueing system starts in the steady state distribution of the system induced by the optimal policy, as follows.

**Assumption 1** (Initial State)**.** *Both $Q(0)$ and $Q^*(0)$ have the same initial state distribution, and this is chosen to be the stationary distribution of $Q^*(t)$; this distribution is denoted $\pi_{(\lambda,\mu^*)}$.*

# 4 The Late Stage

We analyze the performance of a scheduling algorithm with respect to queue-regret as a function of time and system parameters like: *(a)* the load on the system $\epsilon := (\mu^* - \lambda)$, and *(b)* the minimum difference between the rates of the best and the next best servers $\Delta := \mu^* - \max_{k \neq k^*} \mu_k$.

As a preview of the theoretical results, Figure 1 shows the evolution of queue-regret with time in a system with 5 servers under a scheduling policy inspired by Thompson Sampling. Exact details of the scheduling algorithm can be found in Section 4.2. It is observed that the regret goes through a phase transition. In the initial stage, when the algorithm has not estimated the service rates well enough to stabilize the queue, the regret grows poly-logarithmically similar to the classical MAB setting. After a critical point when the algorithm has learned the system parameters well enough to stabilize the queue, the queue-length goes through regenerative cycles as the queue become empty. In other-words, instead of the queue length being continuously backlogged, the queuing system has a stochastic cyclical behavior where the queue builds up, becomes empty, and this cycle recurs. Thus at the beginning of every regenerative cycle, there is no accumulation of past errors and the sample-path queue-regret is at most zero. As the algorithm estimates the parameters better with time, the length of the regenerative cycles decreases and the queue-regret decays to zero.

**Notation**: For the results in Section 4, the notation $f(t) = O(g(K, \epsilon, t))$ for all $t \in h(K, \epsilon)$ (here, $h(K, \epsilon)$ is an interval that depends on $K, \epsilon$) implies that there exist constants $C$ and $t_0$

independent of $K$ and $\epsilon$ such that $f(t) \leq Cg(K, \epsilon, t)$ for all $t \in (t_0, \infty) \cap h(K, \epsilon)$.

### 4.1 An Asymptotic Lower Bound

We establish an asymptotic lower bound on regret for the class of $\alpha$-consistent policies; this class for the queueing bandit is a generalization of the $\alpha$-consistent class used in the literature for the traditional stochastic MAB problem [7, 18, 19]. The precise definition is given below ($\mathbb{1}\{\cdot\}$ below is the indicator function).

**Definition 1.** *A scheduling policy is said to be $\alpha$-consistent (for some $\alpha \in (0, 1)$) if given a problem instance, there holds $(\lambda, \boldsymbol{\mu})$, $\mathbb{E}\left[\sum_{s=1}^{t} \mathbb{1}\{\kappa(s) = k\}\right] = O(t^{\alpha})$ for all $k \neq k^*$.*

Theorem 1 below gives an asymptotic lower bound on the average queue-regret and per-queue regret for an arbitrary $\alpha$-consistent policy.

**Theorem 1.** *For any problem instance $(\lambda, \boldsymbol{\mu})$ and any $\alpha$-consistent policy, the regret $\Psi(t)$ satisfies*

$$\Psi(t) \geq \left(\frac{\lambda}{4} D(\boldsymbol{\mu})(1-\alpha)(K-1)\right)\frac{1}{t}$$

*for infinitely many $t$, where*

$$D(\boldsymbol{\mu}) = \frac{\Delta}{\mathrm{KL}\left(\mu_{min}, \frac{\mu^*+1}{2}\right)}. \tag{2}$$

*Outline for theorem 1.* The proof of the lower bound consists of three main steps. First, in lemma 21, we show that the regret at any time-slot is lower bounded by the probability of a sub-optimal schedule in that time-slot (up to a constant factor that is dependent on the problem instance). The key idea in this lemma is to show the equivalence of any two systems with the same marginal service distributions with respect to bandit algorithms. This is achieved through a carefully constructed coupling argument that maps the original system with independent service across links to another system with service process that is dependent across links but with the same marginal distribution.

As a second step, the lower bound on the regret in terms of the probability of a sub-optimal schedule enables us to obtain a lower bound on the cumulative queue-regret in terms of the number of sub-optimal schedules. We then use a lower bound on the number of sub-optimal schedules for $\alpha$-consistent policies (lemma 19 and corollary 20) to obtain a lower bound on the cumulative regret. In the final step, we use the lower bound on the cumulative queue-regret to obtain an *infinitely often* lower bound on the queue-regret. □

### 4.2 Achieving the Asymptotic Bound

We next focus on algorithms that can (up to a poly log factor) achieve a scaling of $O(1/t)$. A key challenge in showing this is that we will need high probability bounds on the number of times the correct arm is scheduled, and these bounds to hold over the late-stage regenerative cycles of the queue. Recall that these regenerative cycles are random time intervals with $\Theta(1)$ expected length for the optimal policy, and whose lengths are correlated with the bandit algorithm decisions (the queue length evolution is dependent on the past history of bandit arm schedules). To address this, we propose a slightly modified version of the Thompson Sampling algorithm. The algorithm, which we call Q-ThS, has an explicit structured exploration component similar to $\epsilon$-greedy algorithms. This structured exploration provides sufficiently good estimates for all arms (including sub-optimal ones) in the late stage.

We describe the algorithm we employ in detail. Let $T_k(t)$ be the number of times server $k$ is assigned in the first $t$ time-slots and $\hat{\boldsymbol{\mu}}(t)$ be the empirical mean of service rates at time-slot $t$ from past observations (until $t-1$). At time-slot $t$, Q-ThS decides to *explore* with probability $\min\{1, 3K\log^2 t/t\}$, otherwise it *exploits*. When exploring, it chooses a server uniformly at random. The chosen exploration rate ensures that we are able to obtain concentration results for the number of times any link is sampled.[3] When exploiting, for each $k \in [K]$, we pick a sample $\hat{\theta}_k(t)$ of distribution $\text{Beta}\left(\hat{\mu}_k(t)T_k(t-1)+1, \left(1-\hat{\mu}_k(t)\right)T_k(t-1)+1\right)$, and schedule the arm with the largest sample (the standard Thompson sampling for Bernoulli arms [20]). Details of the algorithm are given in Algorithm 1 in the Appendix.

We now show that, for a given problem instance $(\lambda, \boldsymbol{\mu})$ (and therefore fixed $\epsilon$), the regret under Q-ThS scales as $O\left(\text{poly}(\log t)/t\right)$. We state the most general form of the asymptotic upper bound in theorem 2. A slightly weaker version of the result is given in corollary 3. This corollary is useful to understand the dependence of the upper bound on the load $\epsilon$ and the number of servers $K$.

**Notation** : For the following results, the notation $f(t) = O\left(g(K,\epsilon,t)\right)$ for all $t \in h(K,\epsilon)$ (here, $h(K,\epsilon)$ is an interval that depends on $K,\epsilon$) implies that there exist constants $C$ and $t_0$ independent of $K$ and $\epsilon$ such that $f(t) \le Cg(K,\epsilon,t)$ for all $t \in (t_0,\infty) \cap h(K,\epsilon)$.

**Theorem 2.** *Consider any problem instance $(\lambda, \boldsymbol{\mu})$. Let $w(t) = \exp\left(\left(\frac{2\log t}{\Delta}\right)^{2/3}\right)$, $v'(t) = \frac{6K}{\epsilon}w(t)$ and $v(t) = \frac{24}{\epsilon^2}\log t + \frac{60K}{\epsilon}\frac{v'(t)\log^2 t}{t}$. Then, under Q-ThS the regret $\Psi(t)$, satisfies*

$$\Psi(t) = O\left(\frac{Kv(t)\log^2 t}{t}\right)$$

*for all $t$ such that $\frac{w(t)}{\log t} \ge \frac{2}{\epsilon}$, $t \ge \exp\left(6/\Delta^2\right)$ and $v(t)+v'(t) \le t/2$.*

**Corollary 3.** *Let $w(t)$ be as defined in Theorem 2. Then,*

$$\Psi(t) = O\left(K\frac{\log^3 t}{\epsilon^2 t}\right)$$

*for all $t$ such that $\frac{w(t)}{\log t} \ge \frac{2}{\epsilon}$, $\frac{t}{w(t)} \ge \max\left\{\frac{24K}{\epsilon}, 15K^2\log t\right\}$, $t \ge \exp\left(6/\Delta^2\right)$ and $\frac{t}{\log t} \ge \frac{198}{\epsilon^2}$.*

*Outline for Theorem 2.* As mentioned earlier, the central idea in the proof is that the sample-path queue-regret is at most zero at the beginning of regenerative cycles, i.e., instants at which the queue becomes empty. The proof consists of two main parts – one which gives a high probability result on the number of sub-optimal schedules in the exploit phase in the late stage, and the other which shows that at any time, the beginning of the current regenerative cycle is not very far in time.

The former part is proved in lemma 9, where we make use of the structured exploration component of Q-ThS to show that all the links, including the sub-optimal ones, are sampled a sufficiently large number of times to give a good estimate of the link rates. This in turn ensures that the algorithm schedules the correct link in the exploit phase in the late stages with high probability.

For the latter part, we prove a high probability bound on the last time instant when the queue was zero (which is the beginning of the current regenerative cycle) in lemma 15. Here, we make use of a recursive argument to obtain a tight bound. More specifically, we first use a coarse high probability upper bound on the queue-length (lemma 11) to get a first cut bound on the beginning of

[3] The exploration rate could scale like $\log t/t$ if we knew $\Delta$ in advance; however, without this knowledge, additional exploration is needed.

the regenerative cycle (lemma 12). This bound on the regenerative cycle-length is then recursively used to obtain tighter bounds on the queue-length, and in turn, the start of the current regenerative cycle (lemmas 14 and 15 respectively).

The proof of the theorem proceeds by combining the two parts above to show that the main contribution to the queue-regret comes from the structured exploration component in the current regenerative cycle, which gives the stated result. □

# 5 The Early Stage in the Heavily Loaded Regime

In order to study the performance of $\alpha$-consistent policies in the early stage, we consider the *heavily loaded* system, where the arrival rate $\lambda$ is close to the optimal service rate $\mu^*$, i.e., $\epsilon = \mu^* - \lambda \to 0$. This is a well studied asymptotic in which to study queueing systems, as this regime leads to fundamental insight into the structure of queueing systems. See, e.g., [9, 10] for extensive surveys. Analyzing queue-regret in the early stage in the heavily loaded regime has the effect that the the optimal server is the only one that stabilizes the queue. As a result, in the heavily loaded regime, effective learning and scheduling of the optimal server play a crucial role in determining the transition point from the early stage to the late stage. For this reason the heavily loaded regime reveals the behavior of regret in the early stage.

**Notation:** For all the results in this section, the notation $f(t) = O\left(g(K, \epsilon, t)\right)$ for all $t \in h(K, \epsilon)$ ($h(K, \epsilon)$ is an interval that depends on $K, \epsilon$) implies that there exist numbers $C$ and $\epsilon_0$ that depend on $\Delta$ such that for all $\epsilon \geq \epsilon_0$, $f(t) \leq Cg(K, \epsilon, t)$ for all $t \in h(K, \epsilon)$.

Theorem 4 gives a lower bound on the regret in the heavily loaded regime, roughly in the time interval $\left(K^{1/(1-\alpha)}, O\left(K/\epsilon\right)\right)$ for any $\alpha$-consistent policy.

**Theorem 4.** *Given any problem instance $(\lambda, \boldsymbol{\mu})$, and for any $\alpha$-consistent policy and $\gamma > \frac{1}{1-\alpha}$, the regret $\Psi(t)$ satisfies*

$$\Psi(t) \geq \frac{D(\boldsymbol{\mu})}{2}(K-1)\frac{\log t}{\log \log t}$$

*for $t \in \left[\max\{C_1 K^{\gamma}, \tau\}, (K-1)\frac{D(\boldsymbol{\mu})}{2\epsilon}\right]$ where $D(\boldsymbol{\mu})$ is given by equation 2, and $\tau$ and $C_1$ are constants that depend on $\alpha$, $\gamma$ and the policy.*

*Outline for Theorem 4.* The crucial idea in the proof is to show a lower bound on the queue-regret in terms of the number of sub-optimal schedules (Lemma 22). As in Theorem 1, we then use a lower bound on the number of sub-optimal schedules for $\alpha$-consistent policies (given by Corollary 20) to obtain a lower bound on the queue-regret. □

Theorem 4 shows that, for any $\alpha$-consistent policy, it takes at least $\Omega\left(K/\epsilon\right)$ time for the queue-regret to transition from the early stage to the late stage. In this region, the scaling $O(\log t/ \log\log t)$ reflects the fact that queue-regret is dominated by the cumulative regret growing like $O(\log t)$. A reasonable question then arises: after time $\Omega\left(K/\epsilon\right)$, should we expect the regret to transition into the late stage regime analyzed in the preceding section?

We answer this question by studying when Q-ThS achieves its late-stage regret scaling of $O\left(\text{poly}(\log t)/\epsilon^2 t\right)$ scaling; as we will see, in an appropriate sense, Q-ThS is close to optimal in its transition from early stage to late stage, when compared to the bound discovered in Theorem 4. Formally, we have Corollary 5, which is an analog to Corollary 3 under the heavily loaded regime.

**Corollary 5.** *For any problem instance $(\lambda, \boldsymbol{\mu})$, any $\gamma \in (0,1)$ and $\delta \in (0, \min(\gamma, 1-\gamma))$, the regret under Q-ThS satisfies*

$$\Psi(t) = O\left(\frac{K \log^3 t}{\epsilon^2 t}\right)$$

$\forall t \geq C_2 \max\left\{\left(\frac{1}{\epsilon}\right)^{\frac{1}{\gamma-\delta}}, \left(\frac{K}{\epsilon}\right)^{\frac{1}{1-\gamma}}, \left(K^2\right)^{\frac{1}{1-\gamma-\delta}}, \left(\frac{1}{\epsilon^2}\right)^{\frac{1}{1-\delta}}\right\}$, *where $C_2$ is a constant independent of $\epsilon$ (but depends on $\Delta$, $\gamma$ and $\delta$).*

By combining the result in Corollary 5 with Theorem 4, we can infer that in the heavily loaded regime, the time taken by Q-ThS to achieve $O\left(\text{poly}(\log t)/\epsilon^2 t\right)$ scaling is, in some sense, order-wise close to the optimal in the $\alpha$-consistent class. Specifically, for any $\beta \in (0,1)$, there exists a scaling of $K$ with $\epsilon$ such that the queue-regret under Q-ThS scales as $O\left(\text{poly}(\log t)/\epsilon^2 t\right)$ for all $t > (K/\epsilon)^{\beta}$ while the regret under any $\alpha$-consistent policy scales as $\Omega\left(K \log t / \log \log t\right)$ for $t < K/\epsilon$.

We conclude by noting that while the transition point from the early stage to the late stage for Q-ThS is near optimal in the heavily loaded regime, it does does not yield optimal regret performance in the early stage in general. In particular, recall that at any time $t$, the structured exploration component in Q-ThS is invoked with probability $3K \log^2 t/t$. As a result, we see that, in the early stage, queue-regret under Q-ThS could be a $\log^2 t$-factor worse than the $\Omega\left(\log t / \log \log t\right)$ lower bound shown in Theorem 4 for the $\alpha$-consistent class. This intuition can be formalized: it is straightforward to show an upper bound of $2K \log^3 t$ for any $t > \max\{C_3, U\}$, where $C_3$ is a constant that depends on $\Delta$ but is independent of $K$ and $\epsilon$; we omit the details.

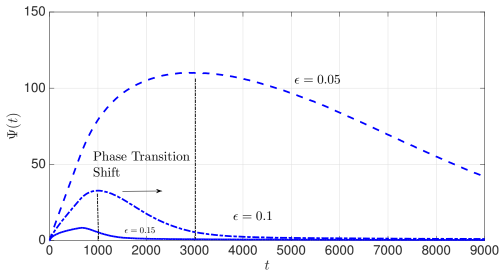
## 6 Simulation Results

In this section we present simulation results of various queueing bandit systems with $K$ servers. These results corroborate our theoretical analysis in Sections 4 and 5. In particular a phase transition from unstable to stable behavior can be observed in all our simulations, as predicted by our analysis. In the remainder of the section we demonstrate the performance of Algorithm 1 under variations of system parameters like the traffic ($\epsilon$), the gap between the optimal and the suboptimal servers ($\Delta$), and the size of the system ($K$). We also compare the performance of our algorithm with versions of UCB-1 [4] and Thompson Sampling [20] without structured exploration.

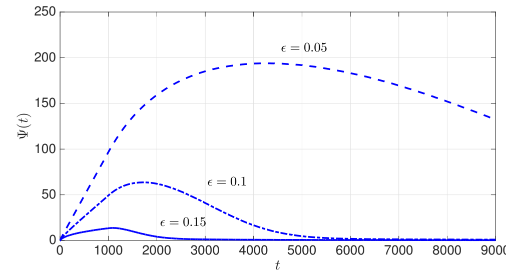
**Variation with $\boldsymbol{\epsilon}$ and** $K$. In Figure 2 we see the evolution of $\Psi(t)$ in systems of size 5 and 7 . It can be observed that the regret decays faster in the smaller system, which is predicted by Theorem 2 in the late stage and Corollary 5 in the early stage. The performance of the system under different traffic settings can be observed in Figure 2. It is evident that the regret of the queueing system grows with decreasing $\epsilon$. This is in agreement with our analytical results (Corollaries 3 and 5). In Figure 2 we can observe that the time at which the phase transition occurs shifts towards the right with decreasing $\epsilon$ which is predicted by Corollaries 3 and 5.

## 7 Discussion and Conclusion

This paper provides the first regret analysis of the queueing bandit problem, including a characterization of regret in both early and late stages, together with analysis of the switching time; and an algorithm (Q-ThS) that is asymptotically optimal (to within poly-logarithmic factors) and also essentially exhibits the correct switching behavior between early and late stages. There remain substantial open directions for future work.




(a) Queue-Regret under Q-ThS for a system with 5 servers with $\epsilon \in \{0.05, 0.1, 0.15\}$




(b) Queue-Regret under Q-ThS for a a system with 7 servers with $\epsilon \in \{0.05, 0.1, 0.15\}$

Figure 2: Variation of Queue-regret $\Psi(t)$ with $K$ and $\epsilon$ under Q-Ths. The phase-transition point shifts towards the right as $\epsilon$ decreases. The efficiency of learning decreases with increase in the size of the system.

*First*, is there a single algorithm that gives optimal performance in *both* early and late stages, as well as the optimal switching time between early and late stages? The price paid for structured exploration by Q-ThS is an inflation of regret in the early stage. An important open question is to find a single, adaptive algorithm that gives good performance over all time. As we note in the appendix, classic (unstructured) Thompson sampling is an intriguing candidate from this perspective.

*Second* the most significant technical hurdle in finding a single optimal algorithm is the difficulty of establishing concentration results for the number of suboptimal arm pulls within a regenerative cycle whose length is dependent on the bandit strategy. Such concentration results would be needed in two different limits: first, as the start time of the regenerative cycle approaches infinity (for the asymptotic analysis of late stage regret); and second, as the load of the system increases (for the analysis of early stage regret in the heavily loaded regime). Any progress on the open directions described above would likely require substantial progress on these technical questions as well.

# Acknowledgements

This work is partially supported by NSF Grants CNS-1161868, CNS-1343383, CNS-1320175, ARO grants W911NF-15-1-0227, W911NF-14-1-0387 and the US DoT supported D-STOP Tier 1 University Transportation Center.

# Appendix

We present our theoretical results in a more general setting where there are $U$ queues and $K$ servers, such that $1 \leq U \leq K$. All the results in the body of the paper become a special case of this setting when $U = 1$. The queues and servers are indexed by $u = 1, \ldots, U$ and $k = 1, \ldots, K$ respectively. Arrivals to queues and service offered by the links are according to product Bernoulli distribution and i.i.d. across time slots. The mean arrival rates are given by the vector $\boldsymbol{\lambda} = (\lambda_u)_{u \in [U]}$ and the mean service rates by the matrix $\boldsymbol{\mu} = [\mu_{uk}]_{u \in [U], k \in [K]}$.

In any time slot, each server can serve at most one queue and each queue can be served by at most one server. The problem is to schedule, in every time slot, a matching in the complete

**Algorithm 1** Q-ThS

At time $t$,
Let $\mathsf{E}(t)$ be an independent Bernoulli sample of mean $\min\{1, 3K\frac{\log^2 t}{t}\}$.
**if** $\mathsf{E}(t) = 1$ **then**
*Explore:*
Schedule a server uniformly at random.
**else**
*Exploit:*
For each $k \in [K]$, pick a sample $\hat{\theta}_k(t)$ of distribution,

$$\hat{\theta}_k(t) \sim \text{Beta}\left(\hat{\mu}_k(t)T_k(t-1) + 1, \left(1 - \hat{\mu}_k(t)\right)T_k(t-1) + 1\right).$$

Schedule a server

$$\kappa(t) \in \arg\max_{k \in [K]} \hat{\theta}_k(t).$$

**end if**

bipartite graph between queues and servers. The scheduling decision at any time $t$ is based on past observations corresponding to the services obtained for the scheduled matchings until time $t-1$. Statistical parameters corresponding to the service distributions are considered unknown. The relevant notation for this system has been provided in Table 1.

The queueing system evolution can be described as follows. Let $\kappa_u(t)$ denote the server that is assigned to queue $u$ at time $t$. Therefore, the vector $\boldsymbol{\kappa}(t) = (\kappa_u(t)_{u \in [U]})$ gives the matching scheduled at time $t$. Let $R_{uk}(t)$ be the service offered to queue $u$ by server $k$ and $S_u(t)$ denote the service offered to queue $u$ by server $\kappa_u(t)$ at time $t$. If $\mathbf{A}(t)$ is the (binary) arrival vector at time $t$, then the queue-length vector at time $t$ is given by:

$$\mathbf{Q}(t) = \left(\mathbf{Q}(t-1) + \mathbf{A}(t) - \mathbf{S}(t)\right)^+.$$

### Regret Against a Unique Optimal Matching

Our goal in this paper is to focus attention on how queueing behavior impacts regret minimization in bandit algorithms. To emphasize this point, we consider a somewhat simplified switch scheduling system. In particular, we assume for every queue, there is a unique optimal server with the maximum expected service rate for that queue. Further, we assume that the optimal queue-server pairs form a matching in the complete bipartite graph between queues and servers, that we call the *optimal matching*; and that this optimal matching stabilizes every queue.

Table 1: General Notation

| Symbol | Description |
|---|---|
| $\lambda_u$ | Expected rate of arrival to queue $u$ |
| $\lambda_{min}$ | Minimum arrival rate across all queues |
| $A_u(t)$ | Arrival at time $t$ to queue $u$ |
| $\mu_{uk}$ | Expected service rate of server $k$ for queue $u$ |
| $R_{uk}(t)$ | Service rate between server $k$ queue $u$ at time $t$ |
| $k_u^*$ | Best server for queue $u$ |
| $\mu_u^*$ | Expected rate of best server for queue $u$ |
| $\mu_{max}$ | Maximum service rate across all links |
| $\mu_{min}$ | Minimum service rate across all links |
| $\Delta$ | Minimum (among all queues) difference between the best and second best servers |
| $\kappa_u(t)$ | server assigned to queue $u$ at time $t$ |
| $S_u(t)$ | Potential service provided by server assigned to queue $u$ at time $t$ |
| $Q_u(t)$ | queue-length of queue $u$ at time $t$ |
| $Q_u^*(t)$ | queue-length of queue $u$ at time $t$ for the optimal strategy |
| $\Psi_u(t)$ | Regret for queue $u$ at time $t$ |

Formally, make the following definitions:

$$\mu_u^* := \max_{k\in[K]} \mu_{uk}, \quad u \in [U]; \tag{3}$$

$$k_u^* := \arg\max_{k\in[K]} \mu_{uk}, \quad u \in [U]; \tag{4}$$

$$\epsilon_u := \mu_u^* - \lambda_u, \quad u \in [U]; \tag{5}$$

$$\Delta_{uk} := \mu_u^* - \mu_{uk}, \quad u \in [U], k \in [K]; \tag{6}$$

$$\Delta := \min_{u\in[U], k\notin k_u^*} \Delta_{uk}; \tag{7}$$

$$\mu_{min} := \min_{u\in[U], k\in[K]} \mu_{uk}; \tag{8}$$

$$\mu_{max} := \max_{u\in[U], k\in[K]} \mu_{uk}; \tag{9}$$

$$\lambda_{min} := \min_{u\in[U]} \lambda_u. \tag{10}$$

The following assumptions will be in force throughout the paper.

**Assumption 2** (Optimal Matching)**.** *There is a* unique optimal matching*, i.e.:*

1. *There is a unique optimal server for each queue: $k_u^*$ is a singleton, i.e., $\Delta_{uk} > 0$ for $k \neq k_u^*$, for all $u$,*
2. *The optimal queue-server pairs for a matching: For any $u' \neq u$, $k_u^* \neq k_{u'}^*$.*

**Assumption 3** (Stability)**.** *The optimal matching stabilizes every queue, i.e., the arrival rates lie within the stability region: $\epsilon_u > 0$ for all $u \in [U]$.*

The assumption of a unique optimal matching essentially means that the queues and servers are solving a pure coordination problem; for example, in the crowdsourcing example described in the introduction, this would correspond to the presence of a unique worker best suited to each type of job. Note that the setting described in Section 3 is equivalent to the unique optimal matching case when $U = 1$. We now describe an algorithm for the unique best match setting which is a more general version of Algorithm 1

---

**Algorithm 2** Q-ThS(match)

---

At time $t$,

Let $\mathsf{E}(t)$ be an independent Bernoulli sample of mean $\min\{1, 3K\frac{\log^2 t}{t}\}$.

**if** $\mathsf{E}(t) = 1$ **then**

*Explore:*

Schedule a matching from $\mathcal{E}$ uniformly at random.

**else**

*Exploit:*

For each $k \in [K], u \in [U]$ , pick a sample $\hat{\theta}_{uk}(t)$ of distribution,

$$\hat{\theta}_{uk}(t) \sim \text{Beta}\left(\hat{\mu}_{uk}(t)T_{uk}(t-1)+1, \left(1-\hat{\mu}_{uk}(t)\right)T_{uk}(t-1)+1\right).$$

Compute for all $u \in [U]$

$$\hat{k}_u(t) := \arg\max_{k\in[K]} \hat{\theta}_{uk}(t)$$

Schedule a matching $\boldsymbol{\kappa}(t)$ such that

$$\boldsymbol{\kappa}(t) \in \arg\min_{\boldsymbol{\kappa}\in\mathcal{M}} \sum_{u\in[U]} \mathbb{1}\left\{\kappa_u \neq \hat{k}_u(t)\right\},$$

i.e., $\boldsymbol{\kappa}(t)$ *is the projection of* $\hat{\mathbf{k}}(t)$ *onto the space of all matchings* $\mathcal{M}$ *with Hamming distance as metric.*

**end if**

---

The notation specific to Algorithm 2 has been provided in Table 2.

# 8 Proofs

We provide details of the proofs for Theorem 2 in Section 8.1 and for Theorems 16 and 17 in Section 8.2. In each section, we state and prove a few intermediate lemmas that are useful in proving the theorems.

## 8.1 Regret Upper Bound for Q-ThS(match)

Theorem 2 is a special case ($U = 1$) of Theorem 6 stated below,

**Theorem 6.** *Consider any problem instance* $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ *which has a single best matching. For any* $u \in [U]$*, let* $w(t) = \exp\left(\left(\frac{2\log t}{\Delta}\right)^{2/3}\right)$, $v'_u(t) = \frac{6K}{\epsilon_u}w(t)$*,* $t \geq \exp\left(6/\Delta^2\right)$ *and* $v_u(t) = \frac{24}{\epsilon_u^2}\log t +$

Table 2: Notation specific to Algorithm 2

| Symbol | Description |
|---|---|
| $\mathsf{E}(t)$ | Indicates if the algorithm schedules a matching through *Explore* |
| $\mathsf{E}_{uk}(t)$ | Indicates if Server $k$ is assigned to Queue $u$ at time $t$ through *Explore* |
| $\mathsf{I}_{uk}(t)$ | Indicates if Server $k$ is assigned to Queue $u$ at time $t$ through *Exploit* |
| $T_{uk}(t)$ | Number of time slots Server $k$ is assigned to Queue $u$ in time $[1, t]$ |
| $\hat{\boldsymbol{\mu}}(t)$ | Empirical mean of service rates at time $t$ from past observations (until $t-1$) |
| $\boldsymbol{\kappa}(t)$ | Matching scheduled in time-slot $t$ |

$\frac{60K}{\epsilon_u}\frac{v'_u(t)\log^2 t}{t}$. *Then, under Q-ThS(match) the regret for queue $u$, $\Psi_u(t)$, satisfies*

$$\Psi_u(t) = O\left(\frac{K v_u(t)\log^2 t}{t}\right)$$

*for all $t$ such that $\frac{w(t)}{\log t} \geq \frac{2}{\epsilon_u}$, $t \geq \exp\left(6/\Delta^2\right)$ and $v_u(t) + v'_u(t) \leq t/2$.*

**Corollary 7.** *Let $w(t) = \exp\left(\left(\frac{2\log t}{\Delta}\right)^{2/3}\right)$. Then,*

$$\Psi_u(t) = O\left(K\frac{\log^3 t}{\epsilon_u^2 t}\right)$$

*for all $t$ such that $\frac{w(t)}{\log t} \geq \frac{2}{\epsilon_u}$, $\frac{t}{w(t)} \geq \max\left\{\frac{24K}{\epsilon_u}, 15K^2\log t\right\}$, and $\frac{t}{\log t} \geq \frac{198}{\epsilon_u^2}$.*

As shown in Algorithm 2, $\mathsf{E}(t)$ indicates whether Q-ThS(match) chooses to explore at time $t$. We now obtain a bound on the expected number of time-slots Q-ThS(match) chooses to explore in an arbitrary time interval $(t_1, t_2]$. Since at any time $t$, Q-ThS(match) decides to explore with probability $\min\{1, 3K\frac{\log^2 t}{t}\}$, we have

$$\mathbb{E}\left[\sum_{l=t_1+1}^{t_2}\mathsf{E}(l)\right] \leq 3K\sum_{l=t_1+1}^{t_2}\frac{\log^2 l}{l} \leq 3K\int_{t_1}^{t_2}\frac{\log^2 l}{l}\,\mathrm{d}l = K\left(\log^3 t_2 - \log^3 t_1\right). \tag{11}$$

The following lemma gives a probabilistic upper bound on the same quantity.

**Lemma 8.** *For any $t$ and $t_1 < t_2$,*

$$\mathbb{P}\left[\sum_{l=t_1+1}^{t_2}\mathsf{E}(l) \geq 5\max\left(\log t, K\left(\log^3 t_2 - \log^3 t_1\right)\right)\right] \leq \frac{1}{t^4}.$$

*Proof.* To prove the result, we will use the following Chernoff bound: for a sum of independent Bernoulli random variables $Y$ with mean $\mathbb{E}Y$ and for any $\delta > 0$,

$$\mathbb{P}\left[[Y \geq (1+\delta)\mathbb{E}Y\right] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mathbb{E}Y}.$$

If $\mathbb{E}Y \geq \log t$, the above bound for $\delta = 4$ gives

$$\mathbb{P}\left[Y \geq 5\mathbb{E}Y\right] \leq \frac{1}{t^4}.$$

Note that $\{\mathsf{E}(l)\}_{l=t_1+1}^{t_2}$ are independent Bernoulli random variables and let $X = \sum_{l=t_1}^{t_2} \mathsf{E}(l)$. Now consider the probability $\mathbb{P}\left[X \geq 5\max\left(\log t, \mathbb{E}X\right)\right]$. If $\mathbb{E}X \geq \log t$, then the result is true from the above Chernoff bound. If $\mathbb{E}X < \log t$, then it is possible to construct a random variable $Y$ which is a sum of independent Bernoulli random variables, has mean $\log t$ and stochastically dominates $X$, in which case we can again use the Chernoff bound on $Y$. Therefore,

$$\mathbb{P}\left[X \geq 5\log t\right] \leq \mathbb{P}\left[Y \geq 5\log t\right] \leq \frac{1}{t^4}.$$

Using inequality (11), we have the required result, i.e.,

$$\mathbb{P}\left[\sum_{l=t_1+1}^{t_2} \mathsf{E}(l) \geq 5\max\left(\log t, K\left(\log^3 t_2 - \log^3 t_1\right)\right)\right] \leq \mathbb{P}\left[X \geq 5\max\left(\log t, \mathbb{E}X\right)\right] \leq 1/t^4.$$

□

Let $w(t) = \exp\left(\left(\frac{2\log t}{\Delta}\right)^{2/3}\right)$. The next lemma shows that, with high probability, Q-ThS(match) does not schedule a sub-optimal matching when it exploits in the late stage.

**Lemma 9.** *For* $t \geq \exp\left(6/\Delta^2\right)$,

$$\mathbb{P}\left[\bigcup_{u\in[U]} \sum_{l=w(t)+1}^{t} \sum_{k\neq k_u^*} \mathsf{I}_{uk}(l) > 0\right] = O\left(\frac{UK}{t^3}\right).$$

*Proof.* Let $X_{uk}(l), u = 1, 2, .., U, k = 1, 2, .., K, l = 1, 2, 3..$ be independent random variables denoting the service offered in the $l^{th}$ assignment of the server $k$ to queue $u$. Consider the events,

$$T_{uk}(w(t)) \geq \frac{1}{2}\log^3(w(t)), \;\forall k \in [K], u \in [U] \tag{12}$$

$$\theta_{uk_u^*}(s) > \mu_u^* - \sqrt{\frac{\log^2(s)}{T_{uk_u^*}(s)}}, \;\forall s, \text{ s.t. } w(t)+1 \leq s \leq t, u \in [U] \tag{13}$$

and

$$\theta_{uk}(s) \leq \mu_u^* - \sqrt{\frac{\log^2(s)}{T_{uk_u^*}(s)}}, \;\forall s, k \text{ s.t. } w(t)+1 \leq s \leq t, k \neq k_u^*, u \in [U] \tag{14}$$

It can be seen that, given the above events, Q-ThS(match) schedules the optimal matching in all time-slots in $(w(t), t]$ in which it decides to exploit, i.e., $\sum_{l=w(t)+1}^{t} \sum_{k \neq k_u^*} \mathsf{I}_{uk}(l) = 0$ for all $u \in [U]$. We now show that the events above occur with high probability.

Note that, since the matchings in $\mathcal{E}$ cover all the links in the system, $T_{uk}(w(t)) \leq \frac{1}{2}\log^3(w(t))$ for some $u, k$ implies that $\sum_{l=1}^{w(t)} \mathbb{1}\{\boldsymbol{\kappa}(t) = \boldsymbol{\kappa}\} \leq \frac{1}{2}\log^3(w(t))$ for some $\boldsymbol{\kappa} \in \mathcal{E}$. Since $\sum_{l=1}^{w(t)} \mathbb{1}\{\boldsymbol{\kappa}(t) = \boldsymbol{\kappa}\}$ is a sum of i.i.d. Bernoulli random variables with mean $\log^3(w(t))$, we use Chernoff bound to prove that event (12) occurs with high probability.

$$\begin{aligned}
\mathbb{P}[(12) \text{ is false}] &\leq \sum_{\boldsymbol{\kappa} \in \mathcal{E}} \mathbb{P}\left[\sum_{l=1}^{w(t)} \mathbb{1}\{\boldsymbol{\kappa}(t) = \boldsymbol{\kappa}\} \leq \frac{1}{2}\log^3(w(t))\right] \\
&\leq K \exp\left(-\frac{1}{8}\log^3(w(t))\right) \\
&= K \exp\left(-\frac{1}{8}\left(\frac{2\log t}{\Delta}\right)^2\right) = o\left(\frac{K}{t^4}\right). \qquad (15)
\end{aligned}$$

In order to prove high probability bounds for the other two events, we define $U_s$ to be a sequence of i.i.d uniform random variables taking values in $[0, 1]$ for $s = w(t)+1, ..., t$. Let us also define $\Sigma_{u,k,l} = \sum_{r=1}^{l} X_{uk}(r)$. In what follows let $F^{\text{Beta}}_{a,b}$ denote the c.d.f of the Beta$(a, b)$ distribution while $F^{\text{B}}_{n,p}$ denotes the c.d.f. of a Binomial$(n, p)$ distribution. Let $S_{uk}(t) = \hat{mu}_{uk}(t)T_{uk}(t)$ for all $u \in [U], k \in [K]$.

$$\begin{aligned}
\mathbb{P}[(13) \text{ is false}] &\leq \sum_{u\in[U]} \sum_{s=w(t)+1}^{t} \mathbb{P}\left[\theta_{uk_u^*}(s) \leq \mu_u^* - \sqrt{\frac{\log^2(s)}{T_{uk_u^*}(s)}}\right] \\
&= \sum_{u\in[U]} \sum_{s=w(t)+1}^{t} \mathbb{P}\left[U_s \leq F^{\text{Beta}}_{S_{uk_u^*}(s)+1, T_{uk_u^*}(s)-S_{uk_u^*}(s)+1}\left(\mu_u^* - \sqrt{\frac{\log^2(s)}{T_{uk_u^*}(s)}}\right)\right] \\
&\overset{(i)}{\leq} \sum_{u\in[U]} \sum_{s=w(t)+1}^{t} \mathbb{P}\left[\exists l \in \left\{\frac{1}{2}\log^3(s), ..., s\right\} : F^{B}_{l+1, \mu_u^* - \sqrt{\frac{\log^2(s)}{l}}}\left(\Sigma_{u,k_u^*,l}\right) \leq U_s \middle| (12) \text{ is true}\right] \\
&\quad + o\left(\frac{UK}{t^3}\right) \\
&\leq \sum_{u\in[U]} \sum_{s=w(t)+1}^{t} \sum_{l=\frac{1}{2}\log^3(s)}^{s} \mathbb{P}\left[\Sigma_{u,k_u^*,l} \leq (F^B)^{-1}_{l+1, \mu_u^* - \sqrt{\frac{\log^2(s)}{l}}}(U_s)\right] + o\left(\frac{UK}{t^3}\right)
\end{aligned}$$

In $(i)$ we use the well-known Beta-Binomial trick [] and the fact that given (12) is true, $uk_u^*$ has been scheduled enough number of times. Now the term $(F^B)^{-1}_{l+1, \mu_u^* - \sqrt{\frac{\log^2(s)}{l}}}(U_s)$ can be thought of as the sum of $l+1$ i.i.d Bernoulli random variables with mean $\mu_u^* - \sqrt{\frac{\log^2(s)}{l}}$. Let $Z_r$ be a sequence of i.i.d random variable with mean $\sqrt{\frac{\log^2(s)}{l}}$. Therefore we have,

$$\begin{aligned}
\mathbb{P}\left[\Sigma_{u,k,l} \leq (F^B)^{-1}_{l+1, \mu_u^* - \sqrt{\frac{\log^2(s)}{l}}}(U_s)\right] &\leq \mathbb{P}\left[\sum_{r=1}^{l} Z_r \leq 1\right] \\
&\overset{(ii)}{\leq} e^{-\frac{\log^2(s)}{3}} \qquad (16)
\end{aligned}$$

Here, $(ii)$ is due to Chernoff-Hoeffding's inequality. Therefore we have,

$$\begin{aligned}
\mathbb{P}\left[(13) \text{ is false}\right] &\leq U \sum_{s=w(t)+1}^{t} \sum_{l=\frac{1}{2}\log^3(s)}^{s} \exp\left(-\frac{\log^2(s)}{3}\right) + o\left(\frac{UK}{t^3}\right) \\
&\leq U \exp\left(-\frac{1}{3}\log^2(w(t)) + 2\log t\right) + o\left(\frac{UK}{t^3}\right) \\
&= U \exp\left(-\frac{1}{3}\left(\frac{2\log t}{\Delta}\right)^{4/3} + 2\log t\right) + o\left(\frac{UK}{t^3}\right) = o\left(\frac{UK}{t^3}\right).
\end{aligned}$$

$$\begin{aligned}
\mathbb{P}\left[(14) \text{ is false}\right] &\leq \sum_{u\in[U],k\neq k_u^*} \sum_{s=w(t)+1}^{t} \mathbb{P}\left[\theta_{uk}(s) > \mu_u^* - \sqrt{\frac{\log^2(s)}{T_{uk_u^*}(s)}}\right] \\
&\leq \sum_{u\in[U],k\neq k_u^*} \sum_{s=w(t)+1}^{t} \mathbb{P}\left[\theta_{uk}(s) > \mu_u^* - \sqrt{\frac{\log^2(s)}{T_{uk_u^*}(s)}} \middle| (12) \text{ is true}\right] + o\left(\frac{UK}{t^3}\right) \\
&\overset{(iii)}{\leq} \sum_{u\in[U],k\neq k_u^*} \sum_{s=w(t)+1}^{t} \mathbb{P}\left[\theta_{uk}(s) > \mu_u^* - \sqrt{\frac{2}{\log(s)}} \middle| (12) \text{ is true}\right] + o\left(\frac{UK}{t^3}\right) \\
&\overset{(iv)}{\leq} \sum_{u\in[U],k\neq k_u^*} \sum_{s=w(t)+1}^{t} \mathbb{P}\left[\theta_{uk}(s) > \mu_{uk} + \frac{\Delta}{2} \middle| (12) \text{ is true}\right] + o\left(\frac{UK}{t^3}\right) \\
&\overset{(v)}{\leq} \sum_{u\in[U],k\neq k_u^*} \sum_{s=w(t)+1}^{t} \mathbb{P}\left[\exists l \in \left\{\frac{1}{2}\log^3(s), ..., s\right\} : \Sigma_{u,k,l} \geq (F^B)^{-1}_{l+1,\mu_{uk}+\frac{\Delta}{2}}(U_s)\right] + o\left(\frac{UK}{t^3}\right) \\
&\overset{(vi)}{\leq} o\left(\frac{UK}{t^3}\right)
\end{aligned}$$

We observe that given (12) is true, we have scheduled $uk_u^*$ enough number of times in order to get $(iii)$. In $(iv)$ we use that fact that $t \geq \exp\left(6/\Delta^2\right)$. $(v)$ is due to the Beta-Binomial trick while $(vi)$ is a result of applying the Chernoff-Hoeffding bound to the first term in $(v)$ in a manner similar to that of (16). □

For any time $t$, let

$$B_u(t) := \min\{s \geq 0 : Q_u(t-s) = 0\}$$

denote the time elapsed since the beginning of the current regenerative cycle for queue $u$. Alternately, at any time $t$, $t - B_u(t)$ is the last time instant at which queue $u$ was zero.

The following lemma gives an upper bound on the sample-path queue-regret in terms of the number of sub-optimal schedules in the current regenerative cycle.

**Lemma 10.** *For any $t \geq 1$,*

$$Q_u(t) - Q_u^*(t) \leq \sum_{l=t-B_u(t)+1}^{t} \left(\mathsf{E}(l) + \sum_{k\neq k_u^*} \mathsf{I}_{uk}(l)\right).$$

*Proof.* If $B_u(t) = 0$, i.e., if $Q_u(t) = 0$, then the result is trivially true.

Consider the case where $B_u(t) > 0$. Since $Q_u(l) > 0$ for all $t - B_u(t) + 1 \le l \le t$, we have

$$Q_u(l) = Q_u(l-1) + A_u(l) - S_u(l) \quad \forall t - B_u(t) + 1 \le l \le t.$$

This implies that

$$Q_u(t) = \sum_{l=t-B_u(t)+1}^{t} A_u(l) - S_u(l).$$

Moreover,

$$Q_u^*(t) = \max_{1\le s\le t} \left( Q_u^*(0) + \sum_{l=s}^{t} A_u(l) - S_u^*(l) \right)^+ \ge \sum_{l=t-B_u(t)+1}^{t} A_u(l) - S_u^*(l).$$

Combining the above two expressions, we have

$$\begin{aligned}
Q_u(t) - Q_u^*(t) &\le \sum_{l=t-B_u(t)+1}^{t} S_u^*(l) - S_u(l) \\
&= \sum_{l=t-B_u(t)+1}^{t} \sum_{k\in[K]} \left(R_{uk_u^*}(l) - R_{uk}(l)\right) \left(\mathsf{E}_{uk}(l) + \mathsf{I}_{uk}(l)\right) \\
&\le \sum_{l=t-B_u(t)+1}^{t} \sum_{k\neq k_u^*} \left(\mathsf{E}_{uk}(l) + \mathsf{I}_{uk}(l)\right) \\
&\le \sum_{l=t-B_u(t)+1}^{t} \left( \mathsf{E}(l) + \sum_{k\neq k_u^*} \mathsf{I}_{uk}(l) \right),
\end{aligned}$$

where the second inequality follows from the assumption that the service provided by each of the links is bounded by 1, and the last inequality from the fact that $\sum_{k\in[K]} \mathsf{E}_{uk}(l) = \mathsf{E}(l)\ \forall l, \forall u \in [U]$. □

In the next lemma, we derive a coarse high probability upper bound on the queue-length. This bound on the queue-length is used later to obtain a first cut bound on the length of the regenerative cycle in Lemma 12.

**Lemma 11.** *For any $l \in [1, t]$,*

$$\mathbb{P}\left[Q_u(l) > 2Kw(t)\right] = O\left(\frac{UK}{t^3}\right)$$

*$\forall t$ s.t. $\frac{w(t)}{\log t} \ge \frac{2}{\epsilon_u}$ and $t \ge \exp\left(6/\Delta^2\right)$.*

*Proof.* From Lemma 10,

$$Q_u(t) - Q_u^*(t) \le \sum_{l=t-B_u(t)+1}^{t} \left( \mathsf{E}(l) + \sum_{k\neq k_u^*} \mathsf{I}_{uk}(l) \right) \le \sum_{l=1}^{t} \left( \mathsf{E}(l) + \sum_{k\neq k_u^*} \mathsf{I}_{uk}(l) \right).$$

Since $Q_u^*(t)$ is distributed according to $\pi_{(\lambda_u,\mu_u^*)}$,

$$\mathbb{P}\left[Q_u^*(t) > w(t)\right] = \frac{\lambda_u}{\mu_u^*} \left( \frac{\lambda_u\left(1-\mu_u^*\right)}{\left(1-\lambda_u\right)\mu_u^*} \right)^{w(t)} \le \exp\left( w(t) \log\left( \frac{\lambda_u\left(1-\mu_u^*\right)}{\left(1-\lambda_u\right)\mu_u^*} \right) \right) \le \frac{1}{t^3}$$

if $\frac{w(t)}{\log t} \geq \frac{2}{\epsilon_u}$. The last inequality follows from the following bound –

$$\begin{aligned}\log\left(\frac{(1-\lambda_u)\,\mu_u^*}{\lambda_u\,(1-\mu_u^*)}\right) &= \log\left(1+\frac{\epsilon_u}{\lambda_u\,(1-\mu_u^*)}\right)\\ &\geq \log\left(1+4\epsilon_u\right) \quad \text{since } \left(\lambda_u\,(1-\mu_u^*) < 1/4\right)\\ &\geq \frac{3}{2}\epsilon_u.\end{aligned}$$

Moreover, from Lemma 8, we have

$$\mathbb{P}\left[\sum_{l=1}^{t} \mathsf{E}(l) > Kw(t)\right] = o\left(\frac{1}{t^3}\right).$$

Now, note that

$$\sum_{l=1}^{t}\sum_{k\neq k_u^*} \mathsf{I}_{uk}(l) \leq (K-1)w(t) + \sum_{l=w(t)+1}^{t}\sum_{k\neq k_u^*} \mathsf{I}_{uk}(l).$$

Therefore,

$$\mathbb{P}\left[\sum_{l=1}^{t}\sum_{k\neq k_u^*} \mathsf{I}_{uk}(l) > (K-1)w(t)\right] \leq \mathbb{P}\left[\sum_{l=w(t)+1}^{t}\sum_{k\neq k_u^*} \mathsf{I}_{uk}(l) > 0\right] = O\left(\frac{UK}{t^3}\right)$$

from Lemma 9. Using the inequalities above, we have

$$\begin{aligned}\mathbb{P}\left[Q_u(t) > 2Kw(t)\right] &\leq \mathbb{P}\left[Q_u^*(t) > w(t)\right] + \mathbb{P}\left[\sum_{l=1}^{t} \mathsf{E}(l) > Kw(t)\right]\\ &\quad + \mathbb{P}\left[\sum_{l=1}^{t}\sum_{k\neq k_u^*} \mathsf{I}_{uk}(l) > (K-1)w(t)\right]\\ &\leq \frac{1}{t^3} + O\left(\frac{UK}{t^3}\right)\\ &= O\left(\frac{UK}{t^3}\right).\end{aligned}$$

□

**Lemma 12.** *Let* $v_u'(t) = \frac{6K}{\epsilon_u}w(t)$ *and let* $v_u$ *be an arbitrary function. Then,*

$$\mathbb{P}\left[B_u\left(t - v_u(t)\right) > v_u'(t)\right] = O\left(\frac{UK}{t^3}\right)$$

$\forall t$ *s.t.* $\frac{w(t)}{\log t} \geq \frac{2}{\epsilon_u}$,$t \geq \exp\left(6/\Delta^2\right)$ *and* $v_u(t) + v_u'(t) \leq t/2$.

*Proof.* Let $r(t) := t - v_u(t)$. Consider the events

$$Q_u(r(t) - v_u'(t)) \leq 2Kw(t), \tag{17}$$

$$\sum_{l=r(t)-v_u'(t)+1}^{r(t)} A_u(l) - R_{uk_u^*}(l) \leq -\frac{\epsilon_u}{2}v_u'(t), \tag{18}$$

$$\sum_{l=r(t)-v_u'(t)+1}^{r(t)} \mathsf{E}(l) + \sum_{k\neq k_u^*} \mathsf{I}_{uk}(l) \leq Kw(t). \tag{19}$$

By the definition of $v'_u(t)$,

$$2Kw(t) - \frac{\epsilon_u}{2} v'_u(t) \leq -Kw(t).$$

Given Events (17)-(19), the above inequality implies that

$$\begin{aligned} Q_u(r(t) - v'_u(t)) + \sum_{l=r(t)-v'_u(t)+1}^{r(t)} A_u(l) &\leq \sum_{l=r(t)-v'_u(t)+1}^{r(t)} R_{uk_u^*}(l) - \left( \mathsf{E}(l) + \sum_{k \neq k_u^*} \mathsf{I}_{uk}(l) \right) \\ &\leq \sum_{l=r(t)-v'_u(t)+1}^{r(t)} S_u(l), \end{aligned}$$

which further implies that $Q_u(l) = 0$ for some $l \in [r(t) - v'_u(t) + 1, r(t)]$. This gives us that $B_u(r(t)) \leq v'_u(t)$.

We now show that each of the events (17)-(19) occur with high probability. Consider the event (18) and note that $A_u(l) - R_{uk_u^*}(l)$ are i.i.d. random variables with mean $-\epsilon_u$ and bounded between $-1$ and 1. Using Chernoff bound for sum of bounded i.i.d. random variables, we have

$$\mathbb{P}\left[ \sum_{l=r(t)-v'_u(t)+1}^{r(t)} A_u(l) - R_{uk_u^*}(l) > -\frac{\epsilon_u}{2} v'_u(t) \right] \leq \exp\left( -\frac{\epsilon_u^2}{8} v'_u(t) \right) \leq \frac{1}{t^3}$$

since $v'_u(t) \geq \frac{6K}{\epsilon_u} w(t) \geq \frac{24}{\epsilon_u^2} \log t$.

By Lemmas 11, 9 and 8, the probability that any of the events (17), (19) does not occur is $O\left(\frac{UK}{t^3}\right)$ $\forall t$ s.t. $\frac{w(t)}{\log t} \geq \frac{2}{\epsilon_u}$ and $v_u(t) + v'_u(t) \leq t/2$, and therefore we have the required result. □

Using the preceding upper bound on the regenerative cycle-length, we derive tighter bounds on the queue-length and the regenerative cycle-length in Lemmas 14 and 15 respectively. The following lemma is a useful intermediate result.

**Lemma 13.** *For any $u \in [U]$ and $t_2$ s.t. $1 \leq t_2 \leq t$,*

$$\mathbb{P}\left[ \max_{1 \leq s \leq t_2} \left\{ \sum_{l=t_2-s+1}^{t_2} A_u(l) - R_{uk_u^*}(l) \right\} \geq \frac{2 \log t}{\epsilon_u} \right] \leq \frac{1}{t^3}.$$

*Proof.* Let $X_s = \sum_{l=t_2-s+1}^{t_2} A_u(l) - R_{uk_u^*}(l)$. Since $X_s$ is the sum of $s$ i.i.d. random variables with mean $\epsilon_u$ and is bounded within $[-1, 1]$, Hoeffding's inequality gives

$$\begin{aligned} \mathbb{P}\left[ X_s \geq \frac{2 \log t}{\epsilon_u} \right] &= \mathbb{P}\left[ X_s - \mathbb{E} X_s \geq \epsilon_u s + \frac{2 \log t}{\epsilon_u} \right] \\ &\leq \exp\left( -\frac{2\left(\epsilon_u s + \frac{2 \log t}{\epsilon_u}\right)^2}{4s} \right) \\ &\leq \exp\left(-4 \log t\right), \end{aligned}$$

where the last inequality follows from the fact that $(a + b)^2 > 4ab$ for any $a, b \geq 0$. Using union bound over all $1 \leq s \leq t_2$ gives the required result. □

**Lemma 14.** *Let $v'_u(t) = \frac{6K}{\epsilon_u}w(t)$ and $v_u$ be an arbitrary function. Then,*

$$\mathbb{P}\left[Q_u(t - v_u(t)) > \left(\frac{2}{\epsilon_u} + 5\right)\log t + 30K\frac{v'_u(t)\log^2 t}{t}\right] = O\left(\frac{UK}{t^3}\right)$$

*$\forall t$ s.t. $\frac{w(t)}{\log t} \geq \frac{2}{\epsilon_u}$,$t \geq \exp\left(6/\Delta^2\right)$ and $v_u(t) + v'_u(t) \leq t/2$.*

*Proof.* Let $r(t) = t - v_u(t)$. Now, consider the events

$$B_u(r(t)) \leq v'_u(t), \tag{20}$$

$$\sum_{l=r(t)-s+1}^{r(t)} A_u(l) - R_{uk^*_u}(l) \leq \frac{2\log t}{\epsilon_u}\ 1 \leq s \leq v'_u(t), \tag{21}$$

$$\sum_{l=r(t)-v'_u(t)+1}^{r(t)} \mathsf{E}(l) + \sum_{k\neq k^*_u} \mathsf{I}_{uk}(l) \leq 5\log t + 5K\left(\log^3\left(r(t)\right) - \log^3\left(r(t) - v'_u(t)\right)\right). \tag{22}$$

Given the above events, we have

$$\begin{aligned}
Q_u(r(t)) &= \sum_{l=r(t)-B_u(r(t))+1}^{r(t)} A_u(l) - S(l) \\
&\leq \sum_{l=r(t)-B_u(r(t))+1}^{r(t)} A_u(l) - R_{uk^*_u}(l) + \mathsf{E}(l) + \sum_{k\neq k^*} \mathsf{I}_{uk}(l) \\
&\leq \left(\frac{2}{\epsilon_u} + 5\right)\log t + 5K\left(\log^3\left(r(t)\right) - \log^3\left(r(t) - v'_u(t)\right)\right) \\
&\leq \left(\frac{2}{\epsilon_u} + 5\right)\log t + 15K\frac{v'_u(t)\log^2 t}{(r(t) - v'_u(t))} \\
&\leq \left(\frac{2}{\epsilon_u} + 5\right)\log t + 30K\frac{v'_u(t)\log^2 t}{t},
\end{aligned}$$

where the last inequality is true if $v_u(t) + v'_u(t) \leq t/2$. From Lemmas 12, 13, 9 and 8, probability of each the events (20)-(22) is $1 - O\left(\frac{UK}{t^3}\right)$ and therefore, we have the required result. □

**Lemma 15.** *Let $v'_u(t) = \frac{6K}{\epsilon_u}w(t)$ and $v_u(t) = \frac{24\log t}{\epsilon_u^2} + \frac{60K}{\epsilon_u}\frac{v'_u(t)\log^2 t}{t}$. Then,*

$$\mathbb{P}\left[B_u(t) > v_u(t)\right] = O\left(\frac{UK}{t^3}\right)$$

*$\forall t$ s.t. $\frac{w(t)}{\log t} \geq \frac{2}{\epsilon_u}$,$t \geq \exp\left(6/\Delta^2\right)$ and $v_u(t) + v'_u(t) \leq t/2$.*

*Proof.* Let $r(t) = t - v_u(t)$. As in Lemma 12, consider the events

$$Q_u(r(t)) \leq \left(\frac{2}{\epsilon_u} + 5\right)\log t + 30K\frac{v'_u(t)\log^2 t}{t}, \tag{23}$$

$$\sum_{l=r(t)+1}^{t} A_u(l) - R_{uk_u^*}(l) \le -\frac{\epsilon_u}{2} v_u(t), \tag{24}$$

$$\sum_{l=r(t)+1}^{t} \mathsf{E}(l) + \sum_{k \neq k_u^*} \mathsf{I}_{uk}(l) \le 5 \log t + 5K \left(\log^3 t - \log^3 (r(t))\right). \tag{25}$$

The definition of $v_u(t)$ and events (23)-(25) imply that

$$\begin{aligned} Q_u(r(t)) + \sum_{l=r(t)+1}^{t} A_u(l) &\le \sum_{l=r(t)+1}^{t} R_{uk_u^*}(l) - \sum_{l=r(t)+1}^{t} \mathsf{E}(l) + \sum_{k \neq k_u^*} \mathsf{I}_{uk}(l) \\ &\le \sum_{l=r(t)+1}^{t} S_u(l), \end{aligned}$$

which further implies that $Q(l) = 0$ for some $l \in [r(t)+1, t]$ and therefore $B_u(t) \le v_u(t)$. We can again show that each of the events (23)-(25) occurs with high probability. Particularly, by Lemmas 8, 9 and 14, the probability that any one of the events (23), (25) does not occur is $O\left(\frac{UK}{t^3}\right)$ $\forall t$ s.t. $\frac{w(t)}{\log t} \ge \frac{2}{\epsilon_u}$ and $v_u(t) + v_u'(t) \le t/2$. We can bound the probability of event (24) in the same way as event (21) in Lemma 12 to show that it occurs with probability at least $\frac{1}{t^3}$. Combining all these gives us the required high probability result. □

*Proof of Theorem 6.* The proof is based on two main ideas: one is that the regenerative cycle length is not very large, and the other is that the algorithm has correctly identified the optimal matching in late stages. We combine Lemmas 9 and 15 to bound the regret at any time $t$ s.t. $\frac{w(t)}{\log t} \ge \frac{2}{\epsilon_u}$ and $v_u(t) + v_u'(t) \le t/2$:

$$\begin{aligned} \Psi_u(t) &= \mathbb{E}\left[Q_u(t) - Q_u^*(t)\right] \\ &\le \mathbb{E}\left[Q_u(t) - Q_u^*(t) \middle| B_u(t) \le v_u(t)\right] \mathbb{P}\left[B_u(t) \le v_u(t)\right] \\ &\quad + \mathbb{E}\left[Q_u(t) - Q_u^*(t) \middle| B_u(t) > v_u(t)\right] \mathbb{P}\left[B_u(t) > v_u(t)\right] \\ &\le \mathbb{E}\left[\sum_{l=t-v_u(t)+1}^{t} \mathsf{E}(l) + \sum_{k \neq k_u^*} \mathsf{I}_{uk}(l)\right] + t\mathbb{P}\left[B_u(t) > v_u(t)\right] \qquad (26) \\ &\le K\left(\log^3(t) - \log^3(t - v_u(t))\right) + t\mathbb{P}\left[\sum_{l=t-v_u(t)+1}^{t} \sum_{k \neq k_u^*} \mathsf{I}_{uk}(l) > 0\right] + t\mathbb{P}\left[B_u(t) > v_u(t)\right] \qquad (27) \\ &\le 3K \log^2 t \log\left(1 + \frac{v_u(t)}{t - v_u(t)}\right) + O\left(\frac{UK}{t^2}\right) \\ &= O\left(K \frac{v_u(t) \log^2 t}{t - v_u(t)}\right) + O\left(\frac{U}{tw(t)}\right) \\ &= O\left(K \frac{v_u(t) \log^2 t}{t}\right), \end{aligned}$$

where (26) follows from Lemma 10, and the last two terms in inequality (27) are bounded using Lemmas 9 and 15. □

*Proof of Corollary 7.* We first note the following:

(i) $\frac{t}{w(t)} \geq \frac{24K}{\epsilon_u}$ implies that $v'_u(t) \leq \frac{t}{4}$ ,

(ii) $\frac{t}{w(t)} \geq 15K^2 \log t$ implies that $\frac{24}{\epsilon_u^2} \log t \geq \frac{60K}{\epsilon_u} \frac{v'_u(t) \log^2 t}{t}$, and therefore $v_u(t) \leq \frac{48}{\epsilon_u^2} \log t$

(iii) $\frac{t}{\log t} \geq \frac{198}{\epsilon_u^2}$ implies that $v_u(t) \leq \frac{t}{4}$.

These inequalities when applied to Theorem 6 give the required result. □

### 8.2 Lower Bounds for a Class of Policies

As mentioned earlier, we prove asymptotic and early stage lower bounds for a class of policies called the $\alpha$-consistent class (Definition 1). As before we will be proving our results for a more general case where there are $U$ queues and $K$ servers. Theorems 1 and 4 are special cases of the analogous theorems stated below, under the unique optimal matching assumption.

**Theorem 16.** *For any problem instance $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ with a unique optimal matching, and any $\alpha$-consistent policy, the regret $\boldsymbol{\Psi}(t)$ satisfies*

*(a)*

$$\frac{1}{U} \sum_{u \in [U]} \Psi_u(t) \geq \left( \frac{\lambda_{min}}{8} D(\boldsymbol{\mu})(1-\alpha)(K-1) \right) \frac{1}{t},$$

*(b) and for any $u \in [U]$,*

$$\Psi_u(t) \geq \left( \frac{\lambda_{min}}{8} D(\boldsymbol{\mu})(1-\alpha) \max\{U-1, 2(K-U)\} \right) \frac{1}{t}$$

*for infinitely many $t$, where*

$$D(\boldsymbol{\mu}) = \frac{\Delta}{\mathrm{KL}\left(\mu_{min}, \frac{\mu_{max}+1}{2}\right)}. \tag{28}$$

**Theorem 17.** *Given any problem instance $(\boldsymbol{\lambda}, \boldsymbol{\mu})$, and for any $\alpha$-consistent policy and $\gamma > \frac{1}{1-\alpha}$, the regret $\boldsymbol{\Psi}(t)$ satisfies*

*(a)*

$$\frac{1}{U} \sum_{u \in [U]} \Psi_u(t) \geq \frac{D(\boldsymbol{\mu})}{4} (K-1) \frac{\log t}{\log \log t},$$

*for* $t \in \left[ \max\{C_4 K^\gamma, \tau\}, (K-1)\frac{D(\boldsymbol{\mu})}{4\epsilon} \right]$, *and*

*(b) for any $u \in [U]$,*

$$\Psi_u(t) \geq \frac{D(\boldsymbol{\mu})}{4} \max\{U-1, 2(K-U)\} \frac{\log t}{\log \log t}$$

*for* $t \in \left[ \max\{C_4 K^\gamma, \tau\}, (K-1)\frac{D(\boldsymbol{\mu})}{2\epsilon_u} \right]$ , *where $D(\boldsymbol{\mu})$ is given by equation 28, $\bar{\epsilon} = \frac{1}{U}\sum_{u\in[U]}\epsilon_u$, and $\tau$ and $C_4$ are constants that depend on $\alpha$, $\gamma$ and the policy.*

In order to prove Theorems 16 and 17, we use techniques from existing work in the MAB literature along with some new lower bounding ideas specific to queueing systems. Specifically, we use lower bounds for $\alpha$-consistent policies on the expected number of times a sub-optimal server is scheduled. This lower bound, shown (in Lemma 19) specifically for the problem of scheduling a unique optimal matching, is similar in style to the traditional bandit lower bound by Lai et al. [7] but holds in the non-asymptotic setting. Also, as opposed the traditional change of measure proof technique used in [7], the proof (similar to the more recent ones [21, 22, 19]) uses results from hypothesis testing (Lemma 18).

**Lemma 18** ([23]). *Consider two probability measures $P$ and $Q$, both absolutely continuous with respect to a given measure. Then for any event $\mathcal{A}$ we have:*

$$P(\mathcal{A}) + Q(\mathcal{A}^c) \geq \frac{1}{2}\exp\{-\min(\mathrm{KL}(P||Q), \mathrm{KL}(Q||P))\}.$$

*Proof.* Let $p = P(\mathcal{A})$ and $q = Q(\mathcal{A}^c)$ . From standard properties of KL divergence we have that,

$$\mathrm{KL}(P||Q) \geq \mathrm{KL}(p,q)$$

Therefore, it is sufficient to prove that

$$p + q \geq \frac{1}{2}\exp\left(-p\log\frac{p}{1-q} - (1-p)\log\frac{1-p}{q}\right) = \frac{1}{2}\left(\frac{1-q}{p}\right)^p\left(\frac{q}{1-p}\right)^{1-p}.$$

Now,

$$\begin{aligned}
\left(\frac{1-q}{p}\right)^p\left(\frac{q}{1-p}\right)^{1-p} &= \left(\sqrt{\frac{1-q}{p}}\right)^{2p}\left(\sqrt{\frac{q}{1-p}}\right)^{2(1-p)} \\
&\leq \left(\frac{1}{2}\left(2p\cdot\sqrt{\frac{1-q}{p}} + 2(1-p)\cdot\sqrt{\frac{q}{1-p}}\right)\right)^2 \\
&= \left(\sqrt{p(1-q)} + \sqrt{q(1-p)}\right)^2 \\
&\leq 2(p(1-q) + q(1-p)) \\
&< 2(p+q)
\end{aligned}$$

as required. □

**Lemma 19.** *For any problem instance $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ and any $\alpha$-consistent policy, there exist constants $\tau$ and $C$ s.t. for any $u \in [U]$, $k \neq k_u^*$ and $t > \tau$,*

$$\mathbb{E}\left[T_{uk}(t)\right] + \sum_{u'\neq u}\mathbb{1}\left\{k_{u'}^* = k\right\}\mathbb{E}\left[T_{u'k_u^*}(t)\right] \geq \frac{1}{\mathrm{KL}\left(\mu_{min}, \frac{\mu_{max}+1}{2}\right)}\left((1-\alpha)\log t - \log(4KC)\right).$$

*Proof.* Without loss of generality, let the optimal servers for the $U$ queues be denoted by the first $U$ indices. In other words, a server $k > U$ is not an optimal server for any queue, i.e., for any $u' \in [U]$, $K \geq k > U$, $\mathbb{1}\left\{k_{u'}^* = k\right\} = 0$. Also, let $\beta = \frac{\mu_{max}+1}{2}$.

We will first consider the case $k \leq U$. For a fixed user $u$ and server $k \leq U$, let $u'$ be the queue that has $k$ as the best server, i.e., $k_{u'}^* = k$. Now consider the two problem instances $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ and $(\boldsymbol{\lambda}, \hat{\boldsymbol{\mu}})$, where $\hat{\boldsymbol{\mu}}$ is the same as $\boldsymbol{\mu}$ except for the two entries corresponding to indices $(u,k),(u',k_u^*)$ replaced by $\beta$. Therefore, for the problem instance $(\boldsymbol{\lambda},\hat{\boldsymbol{\mu}})$, the best servers are swapped for queues $u$ and $u'$ and remain the same for all the other queues. Let $\mathbb{P}_{\boldsymbol{\mu}}^t$ and $\mathbb{P}_{\hat{\boldsymbol{\mu}}}^t$ be the distributions corresponding to the arrivals, chosen servers and rates obtained in the first $t$ plays for the two instances under a fixed $\alpha$-consistent policy. Recall that $T_{uk}(t)=\sum_{s=1}^t \mathbb{1}\{\kappa_u(s)=k\}\ \forall u\in[U],k\in[K]$. Define the event $\mathcal{A}=\{T_{uk}(t)>t/2\}$. By the definition of $\alpha$-consistency there exists a fixed integer $\tau$ and a fixed constant $C$ such that for all $t>\tau$ we have,

$$\mathbb{E}_{\boldsymbol{\mu}}^t\left[\sum_{s=1}^t \mathbb{1}\{\kappa_u(s)=k\}\right]\le Ct^\alpha$$
$$\mathbb{E}_{\hat{\boldsymbol{\mu}}}^t\left[\sum_{s=1}^t \mathbb{1}\{\kappa_u(s)=k'\}\right]\le Ct^\alpha\ ,\forall k'\neq k.$$

A simple application of Markov's inequality yields

$$\mathbb{P}_{\boldsymbol{\mu}}^t(\mathcal{A})\le\frac{2C}{t^{1-\alpha}}$$
$$\mathbb{P}_{\hat{\boldsymbol{\mu}}}^t(\mathcal{A}^c)\le\frac{2C(K-1)}{t^{1-\alpha}}.$$

We can now use Lemma 18 to conclude that

$$\mathrm{KL}(\mathbb{P}_{\boldsymbol{\mu}}^t||\mathbb{P}_{\hat{\boldsymbol{\mu}}}^t)\ge(1-\alpha)\log t-\log(4KC). \tag{29}$$

It is, therefore, sufficient to show that

$$\mathrm{KL}\left(\mathbb{P}_{\boldsymbol{\mu}}^t||\mathbb{P}_{\hat{\boldsymbol{\mu}}}^t\right)=\mathrm{KL}\left(\mu_{uk},\beta\right)\mathbb{E}_{\boldsymbol{\mu}}^t[T_{uk}(t)]+\mathrm{KL}\left(\mu_{u'k_u^*},\beta\right)\mathbb{E}_{\boldsymbol{\mu}}^t[T_{u'k_u^*}(t)].$$

For the sake of brevity we write the scheduling sequence in the first $t$ time-slots $\{\boldsymbol{\kappa}(1),\boldsymbol{\kappa}(2),...,\boldsymbol{\kappa}(t)\}$ as $\boldsymbol{\kappa}^{(t)}$, and similarly we define $\mathbf{A}^{(t)}$ as the number of arrivals to the queue and $\mathbf{S}^{(t)}$ as the service offered by the scheduled servers in the first $t$ time-slots. Let $\mathbf{Z}^{(t)}=(\boldsymbol{\kappa}^{(t)},\mathbf{A}^{(t)},\mathbf{S}^{(t)})$. The KL-divergence term can now be written as

$$\mathrm{KL}(\mathbb{P}_{\boldsymbol{\mu}}^t||\mathbb{P}_{\hat{\boldsymbol{\mu}}}^t)=\mathrm{KL}(\mathbb{P}_{\boldsymbol{\mu}}^t(\mathbf{Z}^{(t)})||\mathbb{P}_{\hat{\boldsymbol{\mu}}}^t(\mathbf{Z}^{(t)})).$$

We can apply the chain rule of divergence to conclude that

$$\begin{aligned}\mathrm{KL}(\mathbb{P}_{\boldsymbol{\mu}}^t(\mathbf{Z}^{(t)})||\mathbb{P}_{\hat{\boldsymbol{\mu}}}^t(\mathbf{Z}^{(t)}))&=\mathrm{KL}(\mathbb{P}_{\boldsymbol{\mu}}^t(\mathbf{Z}^{(t-1)})||\mathbb{P}_{\hat{\boldsymbol{\mu}}}^t(\mathbf{Z}^{(t-1)}))\\&\quad+\mathrm{KL}(\mathbb{P}_{\boldsymbol{\mu}}^t(\boldsymbol{\kappa}(t)\mid\mathbf{Z}^{(t-1)})||\mathbb{P}_{\hat{\boldsymbol{\mu}}}^t(\boldsymbol{\kappa}(t)\mid\mathbf{Z}^{(t-1)}))\\&\quad+\mathbb{E}_{\boldsymbol{\mu}}^t\left[\mathbb{1}\{\kappa_u(t)=k\}\mathrm{KL}\left(\mu_{uk},\beta\right)+\mathbb{1}\{\kappa_{u'}(t)=k_u^*\}\mathrm{KL}\left(\mu_{u'k_u^*},\beta\right)\right].\end{aligned}$$

We can apply this iteratively to obtain

$$\begin{aligned}\mathrm{KL}(\mathbb{P}_{\boldsymbol{\mu}}^t||\mathbb{P}_{\hat{\boldsymbol{\mu}}}^t)&=\sum_{s=1}^t\mathbb{E}_{\boldsymbol{\mu}}^t\left[\mathbb{1}\{\kappa_u(s)=k\}\mathrm{KL}\left(\mu_{uk},\beta\right)\right]\\&\quad+\sum_{s=1}^t\mathbb{E}_{\boldsymbol{\mu}}^t\left[\mathbb{1}\{\kappa_{u'}(s)=k_u^*\}\mathrm{KL}\left(\mu_{u'k_u^*},\beta\right)\right]\\&\quad+\sum_{l=1}^t\mathrm{KL}(\mathbb{P}_{\boldsymbol{\mu}}^t(\boldsymbol{\kappa}(l)\mid\mathbf{Z}^{(l-1)})||\mathbb{P}_{\hat{\boldsymbol{\mu}}}^t(\boldsymbol{\kappa}(l)\mid\mathbf{Z}^{(l-1)}))\end{aligned}\tag{30}$$

Note that the second summation in (30) is zero, as over a sample path the policy pulls the same servers irrespective of the parameters. Therefore, we obtain

$$\mathrm{KL}(\mathbb{P}^t_{\boldsymbol{\mu}}||\mathbb{P}^t_{\hat{\boldsymbol{\mu}}}) = \mathrm{KL}\left(\mu_{uk}, \beta\right) \mathbb{E}^t_{\boldsymbol{\mu}}[T_{uk}(t)] + \mathrm{KL}\left(\mu_{u'k^*_u}, \beta\right) \mathbb{E}^t_{\boldsymbol{\mu}}[T_{u'k^*_u}(t)],$$

which can be substituted in (29) to obtain the required result for $K \leq U$.

Now, consider the case $k > U$, where $\sum_{u \in U} \mathbb{1}\left\{k^*_u = k\right\} = 0$. We again compare the two problem instances $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ and $(\boldsymbol{\lambda}, \hat{\boldsymbol{\mu}})$, where $\hat{\boldsymbol{\mu}}$ is the same as $\boldsymbol{\mu}$ except for the entry corresponding to index $(u, k)$ replaced by $\beta$. Therefore, for the problem instance $(\boldsymbol{\lambda}, \hat{\boldsymbol{\mu}})$, the best server for user $u$ is server $k$ while the best servers for all other queues remain the same. We can again use the same technique as before to obtain

$$\mathrm{KL}(\mathbb{P}^t_{\boldsymbol{\mu}}||\mathbb{P}^t_{\hat{\boldsymbol{\mu}}}) = \mathrm{KL}\left(\mu_{uk}, \beta\right) \mathbb{E}^t_{\boldsymbol{\mu}}[T_{uk}(t)],$$

which, along with (29), gives the required result for $K > U$. □

As a corollary of the above result, we now derive lower bound on the total expected number of sub-optimal schedules summed across all queues. In addition, we also show, for each individual queue, a lower bound for those servers which are sub-optimal for all the queues. As in the proof of Lemma 19, we assume without loss of generality that the first $U$ indices denote the optimal servers for the $U$ queues.

**Corollary 20.** *For any problem instance $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ and any $\alpha$-consistent policy, there exist constants $\tau$ and $C$ s.t. for any $t > \tau$,*

*(a)*

$$2\Delta \sum_{u \in [U]} \sum_{k \neq k^*_u} \mathbb{E}\left[T_{uk}(t)\right] \geq U(K-1) D(\boldsymbol{\mu}) \left((1-\alpha)\log t - \log(4KC)\right),$$

*(b) for any $u \in [U]$,*

$$2\Delta \sum_{k \neq k^*_u} \mathbb{E}\left[T_{uk}(t)\right] \geq (U-1) D(\boldsymbol{\mu}) \left((1-\alpha)\log t - \log(4KC)\right),$$

*(c) and for any $u \in [U]$,*

$$\Delta \sum_{k > U} \mathbb{E}\left[T_{uk}(t)\right] \geq (K-U) D(\boldsymbol{\mu}) \left((1-\alpha)\log t - \log(4KC)\right),$$

*where $D(\boldsymbol{\mu})$ is given by (28).*

*Proof.* To prove part (a), we observe that a unique optimal server for each queue in the system implies that

$$\begin{aligned}\sum_{u \in [U]} \sum_{k \neq k^*_u} \mathbb{E}\left[T_{uk}(t)\right] &\geq \sum_{u \in [U]} \sum_{u' \neq u} \mathbb{E}\left[T_{uk^*_{u'}}(t)\right] \\ &= \sum_{u \in [U]} \sum_{k \neq k^*_u} \sum_{u' \neq u} \mathbb{1}\left\{k^*_{u'} = k\right\} \mathbb{E}\left[T_{u'k^*_u}(t)\right].\end{aligned}$$

Now, from Lemma 19, there exist constants $C$ and $\tau$ such that for $t > \tau$,

$$\begin{aligned}2\sum_{u\in[U]}\sum_{k\neq k_u^*}\mathbb{E}\left[T_{uk}(t)\right] &\geq \sum_{u\in[U]}\sum_{k\neq k_u^*}\left(\mathbb{E}\left[T_{uk}(t)\right]+\sum_{u'\neq u}\mathbb{1}\left\{k_{u'}^*=k\right\}\mathbb{E}\left[T_{u'k_u^*}(t)\right]\right)\\ &\geq \frac{U(K-1)}{\mathrm{KL}\left(\mu_{min},\frac{\mu_{max}+1}{2}\right)}\left((1-\alpha)\log t-\log(4KC)\right).\end{aligned}$$

Using the definition of $D(\boldsymbol{\mu})$ in the above inequality gives part (a) of the corollary.

To prove part (b), we can assume without loss of generality that a perfect matching is scheduled in every time-slot. Using this, and the fact that any server is assigned to at most one queue in every time-slot, for any $u \in [U]$, we have

$$T_{uk_u^*}(t)+\sum_{k\neq k_u^*}T_{uk}(t)=t\geq T_{uk_u^*}(t)+\sum_{u'\neq u}T_{u'k_u^*}(t),$$

which gives us

$$\sum_{k\neq k_u^*}T_{uk}(t)\geq \max\left\{\sum_{u'\neq u}T_{uk_{u'}^*}(t),\sum_{u'\neq u}T_{u'k_u^*}(t)\right\}. \tag{31}$$

From Lemma 19 we have, for any $u' \neq u$ and for $t > \tau$,

$$\mathbb{E}\left[T_{uk_{u'}^*}(t)\right]+\mathbb{E}\left[T_{u'k_u^*}(t)\right]\geq \frac{1}{\mathrm{KL}\left(\mu_{min},\frac{\mu_{max}+1}{2}\right)}\left((1-\alpha)\log t-\log(4KC)\right),$$

which gives

$$\sum_{u'\neq u}\mathbb{E}\left[T_{uk_{u'}^*}(t)\right]+\mathbb{E}\left[T_{u'k_u^*}(t)\right]\geq \frac{U-1}{\mathrm{KL}\left(\mu_{min},\frac{\mu_{max}+1}{2}\right)}\left((1-\alpha)\log t-\log(4KC)\right).$$

Combining the above with (31), we have for $t > \tau$

$$\begin{aligned}\sum_{k\neq k_u^*}\mathbb{E}\left[T_{uk}(t)\right] &\geq \max\left\{\sum_{u'\neq u}\mathbb{E}\left[T_{uk_{u'}^*}(t)\right],\sum_{u'\neq u}\mathbb{E}\left[T_{u'k_u^*}(t)\right]\right\}\\ &\geq \frac{U-1}{2\mathrm{KL}\left(\mu_{min},\frac{\mu_{max}+1}{2}\right)}\left((1-\alpha)\log t-\log(4KC)\right).\end{aligned}$$

To prove part (c), we use the fact that $\mathbb{1}\left\{k_{u'}^*=k\right\}=0$ for any $u' \in [U]$, $K \geq k > U$. Therefore, for $t > \tau$, we have

$$\begin{aligned}\sum_{k>U}\mathbb{E}\left[T_{uk}(t)\right] &= \sum_{k>U}\left(\mathbb{E}\left[T_{uk}(t)\right]+\sum_{u'\neq u}\mathbb{1}\left\{k_{u'}^*=k\right\}\mathbb{E}\left[T_{u'k_u^*}(t)\right]\right)\\ &\geq \frac{K-U}{\mathrm{KL}\left(\mu_{min},\frac{\mu_{max}+1}{2}\right)}\left((1-\alpha)\log t-\log(4KC)\right),\end{aligned}$$

which gives the required result. □

### 8.2.1 Late Stage: Proof of Theorem 16

The following lemma, which gives a lower bound on the queue-regret in terms of probability of sub-optimal schedule in a single time-slot, is the key result used in the proof of Theorem 16. The proof for this lemma is based on the idea that the growth in regret in a single-time slot can be lower bounded in terms of the probability of sub-optimal schedule in that time-slot.

**Lemma 21.** *For any problem instance characterized by $(\boldsymbol{\lambda}, \boldsymbol{\mu})$, and for any scheduling policy, and user $u \in [U]$,*

$$\Psi_u(t) \geq \lambda_u \sum_{k \neq k_u^*} \Delta_{uk} \mathbb{P}\left[\mathbb{1}\{\kappa_u(t) = k\} = 1\right].$$

*Proof.* For the given queueing system, consider an alternate coupled queueing system such that

1. the two systems start with the same initial condition,
2. the arrival process for both the systems is the same, and
3. the service process for the alternate system is independent of the arrival process and i.i.d. across time-slots. For each queue in the alternate system, the service offered by different servers at any time-slot could possibly be dependent on each other but has the same marginal distribution as that in the original system and is independent of the service offered to other queues.

We first show that, under any scheduling policy, the regret for the alternate system has the same distribution as that for the original system. Note that the evolution of the queues is a function of the process $(\mathbf{Z}(l))_{l \geq 1} := (\mathbf{A}(l), \boldsymbol{\kappa}(l), \mathbf{S}(l))_{l \geq 1}$. To prove that this process has the same distribution in both the systems, we use induction on the size of the finite-dimensional distribution of the process. In other words, we show that the distribution of the vector $(\mathbf{Z}(l))_{l=1}^{t}$ is the same for the two systems for all $t$ by induction on $t$.

Suppose that the hypothesis is true for $t-1$. Now consider the conditional distribution of $\mathbf{Z}(t)$ given $(\mathbf{Z}(l))_{l=1}^{t-1}$. Given $(\mathbf{Z}(l))_{l=1}^{t-1}$, the distribution of $(\mathbf{A}(t), \boldsymbol{\kappa}(t))$ is identical for the two systems for any scheduling policy since the two systems have the same arrival process. Also, given $\left((\mathbf{Z}(l))_{l=1}^{t-1}, \mathbf{A}(t), \boldsymbol{\kappa}(t)\right)$, the distribution of $\mathbf{S}(t)$ depends only on the marginal distribution of the scheduled servers given by $\boldsymbol{\kappa}(t)$ which is again the same for the two systems. Therefore, $(\mathbf{Z}(l))_{l=1}^{t}$ has the same distribution in the two systems. Since the statement is true for $t = 1$, it is true for all $t$.

Thus, to lower bound the queue-regret for any queue $u \in [U]$ in the original system, it is sufficient to lower bound the corresponding queue-regret of an alternate queueing system constructed as follows: let $\{U(t)\}_{t \geq 1}$ be i.i.d. random variables distributed uniformly in $(0, 1)$. For the alternate system, let the service process for queue $u$ and server $k$ be given by $R_{uk}(t) = \mathbb{1}\{U(t) \leq \mu_{uk}\}$. Since $\mathbb{E}[R_{uk}(t)] = \mu_{uk}$, the marginals of the service offered by each of the servers is the same as the original system. In addition, the initial condition, the arrival process and the service process for all other queues in the alternate system are identical to those in the original system.

We now lower bound the queue-regret for queue $u$ in the alternate system. Note that, since $\mu_u^* > \mu_{uk}\ \forall k \neq k_u^*$, we have $R_{uk_u^*}(t) \geq R_{uk}(t)\ \forall k \neq k_u^*, \forall t$. This implies that $Q_u^*(t) \leq Q_u(t)\ \forall t$. Now, for any given $t$, using the fact that $Q_u^*(t-1) \leq Q_u(t-1)$, it is easy to see that

$$Q_u(t) - Q_u^*(t) \geq \mathbb{1}\{A_u(t) = 1\} \left( R_{k_u^*}(t) - \sum_{k=1}^{K} \mathbb{1}\{\kappa_u(t) = k\} R_{uk}(t) \right).$$

Therefore,

$$\begin{aligned}\mathbb{E}\left[Q_u(t) - Q_u^*(t)\right] &\geq \mathbb{E}\left[\mathbb{1}\{A_u(t) = 1\}\left(R_{k_u^*}(t) - \sum_{k=1}^{K}\mathbb{1}\{\kappa_u(t) = k\}R_{uk}(t)\right)\right] \\ &= \lambda_u \sum_{k \neq k_u^*} \mathbb{P}\left[\mathbb{1}\{\kappa_u(t) = k\} = 1\right]\mathbb{P}\left[\mu_{uk} < U(t) \leq \mu_u^*\right] \\ &= \lambda_u \sum_{k \neq k_u^*} \Delta_{uk}\mathbb{P}\left[\mathbb{1}\{\kappa_u(t) = k\} = 1\right].\end{aligned}$$

□

We now use Lemma 21 in conjunction with the lower bound for the expected number of sub-optimal schedules for an $\alpha$-consistent policy (Corollary 20) to prove Theorem 16.

*Proof of Theorem 16.* From Lemma 21 we have,

$$\begin{aligned}\Psi_u(t) &\geq \lambda_u \sum_{k \neq k_u^*} \Delta_{uk}\mathbb{P}\left[\mathbb{1}\{\kappa_u(t) = k\} = 1\right] \\ &\geq \lambda_{min}\Delta \sum_{k \neq k_u^*} \mathbb{P}\left[\mathbb{1}\{\kappa_u(t) = k\} = 1\right]. \end{aligned} \tag{32}$$

Therefore,

$$\sum_{s=1}^{t}\sum_{u \in [U]} \Psi_u(s) \geq \lambda_{min}\Delta \sum_{u \in [U]}\sum_{k \neq k_u^*} \mathbb{E}\left[T_{uk}(t)\right].$$

We now claim that

$$\sum_{u \in [U]} \Psi_u(t) \geq \frac{U(K-1)}{8t}\lambda_{min}D(\boldsymbol{\mu})(1-\alpha) \tag{33}$$

for infinitely many $t$. This follows from part (a) of Corollary 20 and the following fact:

**Fact 1.** *For any bounded sequence $\{a_n\}$, if there exist constants $C$ and $n_0$ such that $\sum_{m=1}^{n} a_m \geq C \log n \;\forall n \geq n_0$, then $a_n \geq \frac{C}{2n}$ infinitely often.*

Similarly, for any $u \in U$, it follows from parts (b) and (c) of Corollary 20 that

$$\Psi_u(t) \geq \frac{\max\{U-1, 2(K-U)\}}{8t}\lambda_{min}D(\boldsymbol{\mu})(1-\alpha) \tag{34}$$

for infinitely many $t$. □

#### 8.2.2 Early Stage: Proof of Theorem 17

In order to prove Theorem 17, we first derive, in the following lemma, a lower bound on the queue-regret in terms of the expected number of sub-optimal schedules.

**Lemma 22.** *For any system with parameters $(\boldsymbol{\lambda}, \boldsymbol{\mu})$, any policy, and any user $u \in [U]$, the regret is lower bounded by*

$$\Psi_u(t) \geq \sum_{k \neq k^*} \Delta_{uk}\mathbb{E}\left[T_{uk}(t)\right] - \epsilon_u t.$$

*Proof.* Since $Q_u(0) \sim \pi_{\lambda_u, \mu_u^*}$, we have,

$$\begin{aligned}
\Psi_u(t) &= \mathbb{E}\left[Q_u(t) - Q_u^*(t)\right] \\
&= \mathbb{E}\left[Q_u(t) - Q_u(0)\right] \\
&\geq \mathbb{E}\left[\sum_{l=1}^{t} A_u(l) - S_u(l)\right] \\
&= \lambda_u t - \sum_{k=1}^{K} \mathbb{E}\left[T_{uk}(t)\right] \mu_{uk} \\
&= \lambda_u t - \left(t - \sum_{k \neq k_u^*} \mathbb{E}\left[T_{uk}(t)\right]\right) \mu *_u - \sum_{k \neq k_u^*} \mathbb{E}\left[T_{uk}(t)\right] \mu_{uk} \\
&= \sum_{k \neq k_u^*} \Delta_{uk} \mathbb{E}\left[T_{uk}(t)\right] - \epsilon_u t.
\end{aligned}$$

□

We now use this lower bound along with the lower bound on the expected number of sub-optimal schedules for $\alpha$-consistent policies (Corollary 20).

*Proof of Theorem 17.* To prove part (a) of the theorem, we use Lemma 22 and part (a) of corollary 20 as follows: For any $\gamma > \frac{1}{1-\alpha}$, there exist constants $C_4$ and $\tau$ such that for all $t \in [\max\{C_4 K^\gamma, \tau\}, (K-1)\frac{D(\boldsymbol{\mu})}{4\bar{\epsilon}}]$,

$$\begin{aligned}
\frac{1}{U} \sum_{u \in [U]} \Psi_u(t) &\geq \frac{\Delta}{U} \sum_{u \in [U]} \left(\sum_{k \neq k_u^*} \mathbb{E}\left[T_k(t)\right] - \epsilon_u t\right) \\
&\geq (K-1)\frac{D(\boldsymbol{\mu})}{2}\left((1-\alpha)\log t - \log(KC_4)\right) - \bar{\epsilon} t \\
&\geq (K-1)\frac{D(\boldsymbol{\mu})}{2}\frac{\log t}{\log\log t} - \bar{\epsilon} t \\
&\geq (K-1)\frac{D(\boldsymbol{\mu})}{4}\frac{\log t}{\log\log t},
\end{aligned}$$

where the last two inequalities follow since $t \geq C_4 K^\gamma$ and $t \leq (K-1)\frac{D(\boldsymbol{\mu})}{4\bar{\epsilon}}$.

Part (b) of the theorem can be similarly shown using parts (b) and (c) of corollary 20. □

**Additional Discussion:** As mentioned in Section 7, we note that (unstructured) Thompson sampling [20] is an intriguing candidate for future study.

In Figure 3, we benchmark the performance of Q-ThS against unstructured versions of UCB-1, Thompson Sampling and also a structured version of UCB (Q-UCB) analogous to Q-ThS. Note that there are two variants of Q-ThS displayed: the first has exploration probability $3K\log^2 t/t$, as suggested by the theory; the second has a tuned constant, with an exploration probability of $0.4K\log^2 t/t$.

It can be observed that in the early stage the unstructured algorithms perform better which is an artifact of the extra exploration required by Q-UCB and Q-ThS. In the late stage we observe that Q-UCB gives marginally better performance than UCB-1, however Thompson sampling has the best performance in both stages. This opens up additional research questions, discussed in

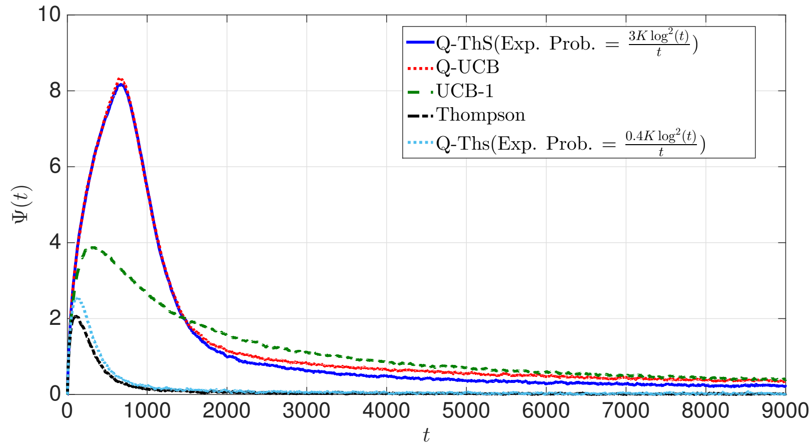



Figure 3: Comparison of queue-regret performance of Q-ThS, Q-UCB, UCB-1 amd Thompson Sampling in a 5 server system with $\epsilon_u = 0.15$ and $\Delta = 0.17$. Two variants of Q-ThS are presented, with different exploration probabilities; note that $3K \log^2 t/t$ is the exploration probability suggested by theoretical analysis (which is necessarily conservative). Tuning the constant significantly improves performance of Q-ThS relative to Thompson sampling.

Section 7. Q-ThS is dominated as well, but can be made to nearly match Thompson sampling by tuning the exploration probability (cf. the discussion above).

Nevertheless, it appears that Thompson sampling dominates UCB-1, Q-UCB, and the theoretically analyzed version of Q-ThS, at least over the finite time intervals considered. In some sense this is not surprising; empirically, similar observations in standard bandit problems [24, 25] are what have led to a surge of interest in Thompson sampling in the first place. Given these numerical experiments, it is important to quantify whether theoretical regret bounds can be established for Thompson sampling (e.g., in the spirit of the analysis in [26, 6, 27]).

## References

[1] J. C. Gittins, "Bandit processes and dynamic allocation indices," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 148–177, 1979.

[2] A. Mahajan and D. Teneketzis, "Multi-armed bandit problems," in *Foundations and Applications of Sensor Management*. Springer, 2008, pp. 121–151.

[3] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[5] A. Garivier and O. Cappé, "The kl-ucb algorithm for bounded stochastic bandits and beyond," *arXiv preprint arXiv:1102.2490*, 2011.

[6] S. Agrawal and N. Goyal, "Analysis of thompson sampling for the multi-armed bandit problem," *arXiv preprint arXiv:1111.1797*, 2011.

[7] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in applied mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[8] J.-Y. Audibert and S. Bubeck, "Best arm identification in multi-armed bandits," in *COLT-23th Conference on Learning Theory-2010*, 2010, pp. 13–p.

[9] W. Whitt, "Heavy traffic limit theorems for queues: a survey," in *Mathematical Methods in Queueing Theory*. Springer, 1974, pp. 307–350.

[10] H. Kushner, *Heavy traffic analysis of controlled queueing and communication networks*. Springer Science & Business Media, 2013, vol. 47.

[11] J. Niño-Mora, "Dynamic priority allocation via restless bandit marginal productivity indices," *Top*, vol. 15, no. 2, pp. 161–198, 2007.

[12] P. Jacko, "Restless bandits approach to the job scheduling problem and its extensions," *Modern trends in controlled stochastic processes: theory and applications*, pp. 248–267, 2010.

[13] D. Cox and W. Smith, "Queues," *Wiley*, 1961.

[14] C. Buyukkoc, P. Varaiya, and J. Walrand, "The c$\mu$ rule revisited," *Advances in applied probability*, vol. 17, no. 1, pp. 237–238, 1985.

[15] J. A. Van Mieghem, "Dynamic scheduling with convex delay costs: The generalized c— mu rule," *The Annals of Applied Probability*, pp. 809–833, 1995.

[16] J. Niño-Mora, "Marginal productivity index policies for scheduling a multiclass delay-/loss-sensitive queue," *Queueing Systems*, vol. 54, no. 4, pp. 281–312, 2006.

[17] C. Lott and D. Teneketzis, "On the optimality of an index rule in multichannel allocation for single-hop mobile networks with multiple service classes," *Probability in the Engineering and Informational Sciences*, vol. 14, pp. 259–297, 2000.

[18] A. Salomon, J.-Y. Audiber, and I. El Alaoui, "Lower bounds and selectivity of weak-consistent policies in stochastic multi-armed bandit problem," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 187–207, 2013.

[19] R. Combes, C. Jiang, and R. Srikant, "Bandits with budgets: Regret lower bounds and optimal algorithms," in *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*. ACM, 2015, pp. 245–257.

[20] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, pp. 285–294, 1933.

[21] S. Bubeck, V. Perchet, and P. Rigollet, "Bounded regret in stochastic multi-armed bandits," *arXiv preprint arXiv:1302.1611*, 2013.

[22] V. Perchet, P. Rigollet, S. Chassang, and E. Snowberg, "Batched bandit problems," *arXiv preprint arXiv:1505.00369*, 2015.

[23] A. B. Tsybakov, *Introduction to nonparametric estimation*. Springer Science & Business Media, 2008.

[24] O. Chapelle and L. Li, "An empirical evaluation of thompson sampling," in *Advances in neural information processing systems*, 2011, pp. 2249–2257.

[25] S. L. Scott, "A modern bayesian look at the multi-armed bandit," *Appl. Stoch. Models in Business and Industry*, vol. 26, no. 6, pp. 639–658, 2010.

[26] E. Kaufmann, N. Korda, and R. Munos, "Thompson sampling: An asymptotically optimal finite-time analysis," in *Algorithmic Learning Theory*. Springer, 2012, pp. 199–213.

[27] D. Russo and B. Van Roy, "Learning to optimize via posterior sampling," *Mathematics of Operations Research*, vol. 39, no. 4, pp. 1221–1243, 2014.