On the Regret of Coded Caching with Adversarial Requests

Anupam Nayak
IIT Bombay
anupam@ee.iitb.ac.in

Kota Srinivas Reddy
IIT Madras
ksreddy@ee.iitm.ac.in

Nikhil Karamchandani
IIT Bombay
nikhilk@ee.iitb.ac.in

***Abstract*—We study a well-known coded caching problem within the framework of online learning theory. Specifically, our focus is on minimizing the regret of a coded caching problem with adversarial requests. To our knowledge, our study marks the first examination of adversarial regret in coded caching setup with a broadcast channel and coded delivery. To minimize the regret, we introduce a policy based on Follow-The-Perturbed-Leader. Through comparative analysis between our algorithm and a static oracle, we demonstrate that our policy achieves sub-linear regret of $\mathcal{O}(\sqrt{T})$. Furthermore, we also address the issue of switching costs by establishing an upper bound on the expected number of switches in our algorithm under unrestricted switching conditions, and we provide an upper bound on adversarial regret under restricted switching. Additionally, we validate our theoretical insights with numerical results on the Movielens dataset.**

I. Introduction

The unprecedented surge in demand for high-definition content over the internet has resulted in an increased load on the underlying communication networks. This challenge can be effectively mitigated through the widespread adoption of Content Delivery Networks (CDNs). CDNs strategically deploy storage devices or caches across large geographical regions. During off-peak hours, these caches are utilized to proactively pre-fetch popular content [1]. This proactive strategy aims to reduce network traffic during peak hours when users generate the highest volume of requests.

In the realm of caching, traditional policies emphasizing local caching gains have been extensively investigated in the literature, as exemplified by [2] and the associated references. More recently, Maddah-Ali and Niesen [1], [3] delved into cache networks, introducing the concept of 'coded caching' which allows for coding of information while delivering content to users. Their work resulted in policies that not only achieve significant local caching gains but also offer substantial global caching gains.

The field of coded caching has since become a vibrant area of research within information theory, exploring various facets of cache networks, including network topology [4]–[6], content popularity [4], [7]–[12], and security and privacy [13]–[17]. In general, these works propose content placement and delivery schemes and subsequently evaluate their performance against the information-theoretic lower bound, often demonstrating a gap of at most a constant multiplicative factor, independent of the system size.

In our paper, we focus on coded caching in the framework of online learning theory. As previously mentioned, existing works [4], [7]–[12] address content popularity framework and devise policies by assuming known and static content popularity. In practical scenarios, the actual content popularity may be unknown, necessitating an emphasis on online learning policies for coded caching where the actions of the caching policy at each time are based on the history of actions and observations. To judge the performance of an online caching policy, we focus on the notion of *adversarial regret* [18], which considers the worst-case (over all possible request sequences) additive gap between the performance of the policy and an (static) oracle that knows all the requests beforehand.

To the best of our knowledge, online learning policies for coded caching have been previously explored only in [19]–[21]. The setting explored in [19], [20] involved users requesting files from a catalog that evolved (slowly) over time. The aim was to devise a cache update rule while preserving the benefits of coded caching. Alongside differences in the problem formulation, another key difference in our work is the choice of performance metric; we consider the worst-case regret (additive gap), while [19], [20] focus on a multiplicative gap. The setting in [21] is closer to our work, wherein the goal is to minimize the regret in a stochastic setting, assuming that content popularity follows a static probability distribution, which is a priori unknown to the learner.

In contrast, our focus lies in regret minimization within the adversarial setting, where our aim is to design online policies that minimize the maximum regret across all possible request patterns. The adversarial setup holds practical significance as content popularity may not adhere to any specific distribution and may be dynamic rather than static[1]. Additionally, we use the exact expected rate expression, non-linear in policy parameters and observations, unlike [21], which uses an approximate, linear rate expression. Apart from coded delivery, the primary technical challenge arises when incorporating a broadcast channel with multiple users (K). In this scenario, the rate expression becomes more complex, deviating from a straightforward linear counting problem where the total rate is the sum of individual rates across all K channels.

The exploration of caching systems within the online learning framework has recently garnered attention, particularly in the context of a single cache. Motivated by recent advancements in online convex optimization [22]–[24], several online caching policies have been proposed, including Online Gradient Ascent [25]–[27], Online Mirror Descent [28], and Follow the Perturbed Leader (FTPL) [29]–[32]. Notably, these approaches specifically target adversarial requests, showcasing the achievement of an order-optimal regret of $\mathcal{O}(\sqrt{T})$, where T is the time horizon. In a similar vein, our work takes a preliminary step in investigating the popular coded caching framework within the context of online learning.

[1]As an aside, our proposed policy also achieves a constant regret in the stochastic setting similar to [21].

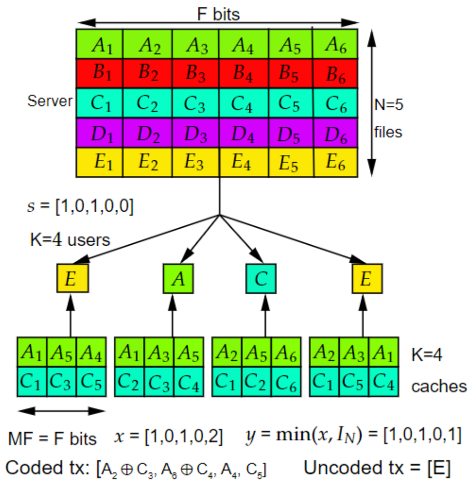

Fig. 1. An illustration of coded caching problem setup. It contains a central server with $N = 5$ files, each of size F bits, and $K = 4$ users, connected to a separate cache of size $MF = F$ bits. Here $s = [1, 0, 1, 0, 0]$ indicates that *stored files* are A and C, and *unstored files* are the remaining. For a request profile $\mathbf{r} = (E, A, C, E)$, it's request pattern is $x = [1, 0, 1, 0, 2]$.

Novelty and Contributions

Our work focuses on the adversarial regret version of the coded caching problem. To the best of our knowledge, our work is the first to study adversarial regret in a coded caching setup with a broadcast channel and coded delivery. Similar to [29]–[32], we employ an FTPL-based algorithm for our coded caching problem and demonstrate its ability to achieve $\mathcal{O}(\sqrt{T})$ regret. A key technical challenge arises from the non-linear nature of the expected rate expression in the coded caching scenario, which complicates the analysis[2]. We resolve this issue by rewriting the rate expression using a careful transformation of the request vector.

II. Problem Formulation

For an integer n, $[n]$ denotes the set $\{1, 2, \ldots, n\}$. Given two N-dimensional vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$, $\langle \mathbf{a}, \mathbf{b} \rangle$ denotes the standard inner product between two real-valued vectors. For a collection of objects $\mathbf{r}_i$ such that $i \in \mathbb{N}$ and an integer t, $(\mathbf{r}_i)_{i=1}^t$ denotes the multi-set $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_t\}$. Consider the system shown in Figure 1, which contains a remote backend server that holds N files, each of size F bits, and is connected to K users via an error-free broadcast medium. Each user seeks files from the server and is connected to a local cache storage of size MF bits. Typically, the cache size M is much smaller compared to the number of files N. Time is divided into slots with horizon $T \in \mathbb{N}$. Each slot contains two phases, namely the placement phase and the delivery phase.

For every $t \in [T]$, in the placement phase, the caches store content related to the N files. Let z_t^k denote the cache content stored at cache/user k at time t. The cache content may change over time and will be a function of the file requests observed in the past as well as the history of cached content. After the placement phase, we get a set of file requests, one from each user, which marks the start of the delivery phase. Let $\mathbf{r}_t = (r_t^1, r_t^2, ..., r_t^K)$ denote the file request profile at time t, where r_t^k denotes the file requested by User k in slot t. Based on the users' request profile $\mathbf{r}_t$ and the cached content profile $\mathbf{z}_t = (z_t^1, z_t^2, ..., z_t^K)$, the central server transmits a message $\mathcal{X}_t$ of size/rate $K_{\mathbf{r}_t}(t)$ such that every user k can recover its requested file r_t^k using the server's transmitted message $\mathcal{X}_t$ and its cached content z_t^k. We aim to design an algorithm π, which determines the cache placement and content delivery schemes in each time slot, such that the server's cumulative transmission rate given by $K^\pi(T) := \sum_{t=1}^T K_{\mathbf{r}_t}(t)$ is minimized.

As is standard in online learning theory [22], we compare the performance of any online algorithm π with that of a static oracle, which knows beforehand the entire request profile from time $t = 1$ to $t = T$ and uses the best static (fixed) placement over the entire time horizon T along with an optimal delivery policy[3] to minimize the cumulative transmission rate. However, the characterization of such an oracle, which includes identifying an optimal placement and delivery policy for every collection of requests, is computationally hard and remains an open problem. Therefore, for analytical tractability and as an initial approach to address the adversarial regret in coded caching problem, we concentrate on a restricted class of policies. Besides having a very natural structure, these policies are also akin to the approach proposed in [7], which is order-wise optimal for the "coded caching under (known) arbitrary popularity distribution" problem with respect to an information-theoretic lower bound.

Next, we describe this class of policies and later provide descriptions of the placement and delivery phases. In each time slot t, based on the history of request patterns $(\mathbf{r}_i)_{i=1}^{t-1}$, the scheme divides the set of files into two categories: *stored files* and *unstored files*. The *stored files* are then cached across the users as per the placement policy in [3]. During the delivery phase, the requests corresponding to the *unstored files* are directly handled by the server using the broadcast channel, while the requests corresponding to the *stored files* are served as per the delivery method proposed in [3]. Note that any scheme in this class is characterized by its choice of *stored files* in each slot. A comprehensive description of the schemes' placement and delivery policies at each time slot is given below:

1) **Placement phase:** Let a binary vector $\mathbb{I}_N = [1, 1, \ldots, 1]$ denotes an N-dimensional vector with each entry being 1. Let a binary vector $\mathbf{s} \in \{0, 1\}^N$ represent a cache configuration, where an entry with a value of 1 represents a *stored file* and an entry with a value of 0 represents an *unstored file*. To avoid underutilization of caches, a cache configuration $\mathbf{s} \in \{0, 1\}^N$ is said to be feasible if it contains more than M entries with a value of 1. Let $\mathcal{S}$ denote the set of all feasible cache configurations. At time t, the algorithm selects a subset of files $\mathbf{s}_t \in \mathcal{S}$ as the cache configuration. Then, each user independently and randomly chooses equal fractions ($\frac{MF}{\langle \mathbf{s}_t, \mathbb{I}_N \rangle}$ bits) of all the files with a "1" entry in $\mathbf{s}_t$ to populate their caches. It's worth noting that the placement policy also mandates that the subset $\mathbf{s}_t$ remains the same across all users. However, the users independently sample bits, resulting in different random fractions of the same set of files being stored in their caches.
2) **Delivery phase:** After the placement phase, users reveal their corresponding requests and the delivery phase starts.

[2]The work in [32] considered bipartite caching networks, for which the rate expression was not linear in terms of the cache configuration. They dealt with it by switching to a 'virtual' action space where the reward function was indeed linear.

[3]For a given user request profile and the content placement, finding the optimal delivery policy is equivalent to solving an *index coding* problem [33].

Let $\mathbf{x}_t = [x_t(1), x_t(2), \ldots, x_t(N)]$ be an N-dimensional vector where $x_t(i)$ denotes the number of users requesting file i at time slot t. Note that $\langle \mathbf{x}_t, \mathbb{I}_N \rangle = K$, the total number of users. Let $\mathbf{y}_t = \min\{\mathbb{I}_N, \mathbf{x}_t\}$, a pointwise minimum of $\mathbf{x}_t$ with a vector of all ones, denotes whether a file was requested by at least one user. Once the users make the requests, the delivery at time t takes place using the following two steps.

Uncoded Transmission: The requests corresponding to the "unstored files" at time t (i.e., the requests for the files with a "0" entry in $\mathbf{s}_t$) are served directly by the server through uncoded transmission. Since we have a broadcast medium, we only need one transmission for these files even if multiple users request them. Thus, the length of the uncoded transmission is given by the inner product of $\mathbf{y}_t$ with the set of files not cached $(\mathbb{I}_N - \mathbf{s}_t) = \langle (\mathbb{I}_N - \mathbf{s}_t), \mathbf{y}_t \rangle$.[4]

Coded Transmission: The remaining requests (i.e., the requests for the files with a "1" entry in $\mathbf{s}_t$) are served jointly by the cache contents and a coded message from the server. The coded message design follows the decentralized coded caching [3] given the subset of the files to be cached is $\mathbf{s}_t$ at time t. Let U_1 denote the set of users that request files from the cached set $\mathbf{s}_t$, $|U_1| = \langle \mathbf{x}_t, \mathbf{s}_t \rangle$. For every subset $u \in U_1$ with $|u| \neq 0$, transmit $\bigoplus_{k \in u} V_{k,u\setminus\{k\}}$. Here, $V_{k,u\setminus\{k\}}$ denotes all the bits that are requested by user $k \in u$, are present in the cache of all users in $u \setminus \{k\}$, and that are not stored in the caches of any other user in $U_1 \setminus u$. Note that using the above coded message and the content stored in its caches, every user can recover their corresponding requested file, see [7] for more details about the decoding. Therefore, the expected length of the coded transmission is equal to the expected sum of lengths of all these messages $\bigoplus_{k \in u} V_{k,u\setminus\{k\}}$ for all $u \subseteq U_1$, where the length of a message $\bigoplus_{k \in u} V_{k,u\setminus\{k\}}$ is $|\bigoplus_{k \in u} V_{k,u\setminus\{k\}}| = \max_{k \in u} |V_{k,u\setminus\{k\}}|$.

The following proposition gives the message size for a given cache configuration $\mathbf{s}_t$ and the request pattern $\mathbf{x}_t$.

Proposition 1. *For a coded caching problem with the given cache configuration $\mathbf{s}_t$ and the request vector $\mathbf{x}_t$, the above-discussed placement and delivery policies give a transmission rate of expected length $K(\mathbf{s}_t, \mathbf{x}_t)$ given by*

$$K(\mathbf{s}_t, \mathbf{x}_t) = \underbrace{\langle (\mathbb{I}_N - \mathbf{s}_t), \mathbf{y}_t \rangle}_{\text{Uncoded transmission}} + \underbrace{\left(\frac{\langle \mathbf{s}_t, \mathbb{I}_N \rangle}{M} - 1 \right) \left(1 - \left(1 - \frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N \rangle} \right)^{\langle \mathbf{x}_t, \mathbf{s}_t \rangle} \right)}_{\text{Coded transmission}}, \quad (1)$$

where $\mathbf{y}_t = \min\{\mathbb{I}_N, \mathbf{x}_t\}$.

Proof. The expected coded transmission length is derived in the appendix of the full paper [34]. □

Note that $K(\mathbf{s}_t, \mathbf{x}_t)$, given in (1), depends on the requests only through $\mathbf{x}_t$. So the knowledge of exact request profile $\mathbf{r}_t = (r_t^1, r_t^2, \ldots, r_t^K)$ is not necessary to compute $K(\mathbf{s}_t, \mathbf{x}_t)$. Thus, knowing the request pattern $\mathbf{x}_t$ is sufficient to choose the cache configuration $(\mathbf{s}_t)$ for the policy at time t. Equation (1) can be rewritten as follows:

$$K(\mathbf{s}_t, \mathbf{x}_t) = \underbrace{\left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_t) - \mathbf{y}_t \right\rangle}_{T_0} + h(\mathbf{x}_t), \quad (2)$$

where $f(\mathbf{x}_t, \mathbf{s}_t) = \frac{1}{M} \left(1 - \left(1 - \frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N \rangle} \right)^{\langle \mathbf{x}_t, \mathbf{s}_t \rangle} \right) \mathbb{I}_N$ and $h(\mathbf{x}_t) = \left(1 - \frac{M}{N} \right) \langle \mathbf{y}_t, \mathbb{I}_N \rangle$. Note that in (2), for given a request pattern $\mathbf{x}_t$, $h(\mathbf{x}_t)$ is a constant and T_0 is the only part that can be minimized using an appropriate choice of $\mathbf{s}_t$. In order to minimize the cumulative transmission rate, every algorithm chooses a cache configuration $\mathbf{s}_t \in \mathcal{S}$, based on the history of request patterns $(\mathbf{x}_i)_{i=1}^{t-1}$, in each time slot t. After that, placement and delivery phases occur according to the above description. For a given request pattern $(\mathbf{x}_t)_{t=1}^T$, if an algorithm π's cache configuration is $(\mathbf{s}_t)_{t=1}^T$, then its cumulative expected transmission rate is given by

$$K^\pi(T) = \sum_{t=1}^{T} \mathbb{E}[K(\mathbf{s}_t, \mathbf{x}_t)], \quad (3)$$

where the expectation is with respect to the randomness generated by the algorithm and the storage of random bits involved in the placement phase of the policy. To evaluate an algorithm's performance, we compare its cumulative expected transmission rate given in (3) with the performance of a static oracle that knows the entire request pattern from time $t = 1$ to $t = T$ and uses a fixed cache configuration over the entire time horizon T, chosen to minimize the cumulative transmission size. Therefore, for a given request pattern $(\mathbf{x}_t)_{t=1}^T$, the cumulative oracle rate is given by

$$K^{\text{o}}(T) = \min_{\mathbf{s} \in \mathcal{S}} \sum_{t=1}^{T} \mathbb{E}[K(\mathbf{s}, \mathbf{x}_t)], \quad (4)$$

where $\mathcal{S}$ is the collection of all feasible cache configurations. Thus, for a request pattern $(\mathbf{x}_t)_{t=1}^T$, the regret of an online algorithm π is given by $R_\pi((\mathbf{x}_t)_{t=1}^T, T) := K^\pi(T) - K^{\text{o}}(T)$. In adversarial regret minimization problems, no assumptions are made about the requests. Therefore, an algorithm's adversarial regret is calculated for the worst case, and is given by

$$R_\pi(T) = \sup_{\mathbf{x}_t \in X, \forall t} R_\pi((\mathbf{x}_t)_{t=1}^T, T), \quad (5)$$

where X represents the set of all feasible $\mathbf{x}_t$ $(\mathbf{x}|x(i) \geq 0, \langle \mathbf{x}, \mathbb{I}_N \rangle = K)$, i.e., all feasible request patterns.

Our aim is to design an algorithm π that achieves the minimum worst-case regret $R_\pi(T)$ with respect to the static oracle. With this aim, in the rest of our paper, we propose an algorithm (Algorithm 1) and show that our algorithm achieves $\mathcal{O}(\sqrt{T})$ adversarial regret.

An algorithm that fetches a large number of files into the caches each time to change the cache configuration is not ideal, as fetching files into the caches causes latency and consumes bandwidth. Thus, we also consider the switching cost issue. Let $C_\pi(T)$ denote the expected number of cache configuration switches until time T for an Algorithm π, then $C_\pi(T)$ is given by

$$C_\pi(T) = \mathbb{E}\left[\sum_{t=1}^{T-1} \mathbb{I}(\mathbf{s}_{t+1} \neq \mathbf{s}_t) \right].$$

[4] Note that the actual transmission length is $\langle (\mathbb{I}_N - \mathbf{s}_t), \mathbf{y}_t \rangle |F|$. But, we will be dealing with a normalized size.

We address the issue of switching costs using two results. Firstly, we provide an upper bound on the expected number of switches incurred by our algorithm for the unrestricted switching case. We also provide an upper bound on the regret incurred by our proposed algorithm when the cache configuration is allowed to switch only in a set of predefined arbitrary L restricted slots given by $\mathcal{T} = \{t_i : i \in [L], t_0 = 0, t_L = T; t_i \in \mathbb{N}, 0 \leq t_{i-1} < t_i \leq T\}$. Note that if $\mathcal{T} = [T]$, then the restricted case is equivalent to the unrestricted case.

The rest of the paper is organized as follows. We discuss our online subset selection algorithm in Section III. Section IV contains the main results describing the performance of our algorithm. We include our numerical findings in Section V and conclusions and future directions in Section VI. Due to lack of space, the proofs of our results are presented in the Appendix of the full paper [34].

III. Our Proposed Algorithm

Algorithm 1 Algorithm for Coded Caching Problem with Adversarial Requests

```
1: Input: M, N, T, s_0 = [1, 1 ··· 1], 𝒯, η_t = α√t   ∀t ∈ 𝒯
2: Sample γ ~ 𝒩(0, I_{N×N})
3: for t ≤ T do
4:     Placement phase:
5:     Derive x_{t−1} from r_{t−1}
6:     y_{t−1} ← min{x_{t−1}, 𝕀_N}
7:     Y_t ← Y_{t−1} + y_{t−1}
8:     if t ∈ 𝒯 then
9:         Ȳ_t = Y_t − η_t γ
10:        Subset selection for placement:
11:        s_t ← argmin_{s∈𝒮} ⟨(s − (M/N) 𝕀_N), Σ_{i=1}^{t−1} f(x_i, s) − Ȳ_t⟩
12:        Perform cache placement with s_t as per the placement phase of Section II
13:    end if
14:    Delivery phase:
15:    Receive r_t
16:    Perform content delivery as per the delivery phase of Section II
17: end for
```

In this section, we present our online algorithm, which achieves $\mathcal{O}(\sqrt{T})$ regret. Our algorithm is a variant of the standard FTPL algorithm commonly used in online learning settings, and its pseudocode is provided in Algorithm 1. Recall that our algorithm has to identify the subset of files $\mathbf{s}_t$ to cache in each time slot $t \in \mathcal{T}$, guided by the history of request patterns $(\mathbf{r}_i)_{i=1}^{t-1}$. A detailed description of our algorithm is given below: As we mentioned earlier, time is divided into slots. During the placement phase,

- We first set $\mathbf{y}_{t-1} = \min\{\mathbf{x}_{t-1}, \mathbb{I}_N\}$ and $\mathbf{Y}_t = \mathbf{Y}_{t-1} + \mathbf{y}_{t-1}$, based on the request pattern $\mathbf{x}_{t-1}$ received in the previous slot $t-1$. (See lines 5 and 6 in Algorithm 1).
- Next, if $t \notin \mathcal{T}$, we directly start the delivery phase. Otherwise, we update our cache content first before starting the delivery phase. The cumulative rate until slot t that would have been incurred by the cache configuration $\mathbf{s}$ is given by $\left\langle \left(\mathbf{s} - \frac{M}{N}\mathbb{I}_N\right), \sum_{i=1}^{t-1} f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_t \right\rangle + \sum_{i=1}^{t-1} h(\mathbf{x}_i)$. For $t \in \mathcal{T}$, when we are allowed to change the cache configuration, the subset $\mathbf{s}_t$ is then determined as the one that minimizes the cumulative rate until t with the perturbed vector $\overline{\mathbf{Y}}_t = \mathbf{Y}_t - \eta_t\gamma$ as input.
- After the subset selection, cache placement occurs according to the placement phase described in Section II. (See lines 7-12 in Algorithm 1).

This is followed by the delivery phase as per Section II. Note that the $h(\mathbf{x}_i)$ terms in the cumulative rate expression are ignored during subset selection (line 11), since they do not depend on the policy parameters $\mathbf{s}$. In a single-user caching scenario, FTPL tracks the frequency of each file's requests and selects the top M files after introducing a Gaussian perturbation to the cumulative request numbers. Extensively studied in prior works [30]–[32], the FTPL policy has proven to achieve $\mathcal{O}(\sqrt{T})$ regret in adversarial settings. In a single-user/uncoded setting, the rate expression is linear in both cache configuration $\mathbf{s}_t$ and cumulative request numbers. Consequently, the cache configuration choice simply involves selecting the set of M files with the maximum cumulative requests after perturbation. However, in the coded caching setting, the expected rate expression becomes a non-linear function of the request pattern $\mathbf{x}_t$ and the cache configuration $\mathbf{s}_t$. Introducing perturbation directly to the request pattern $\mathbf{x}_t$ complicates the regret analysis significantly.

Nevertheless, it is noteworthy that the expected cumulative rate that would be incurred for any cache configuration $\mathbf{s}$ until time t given by $\sum_{i=1}^{t-1} K(\mathbf{s}, \mathbf{x}_i) = \left\langle \left(\mathbf{s} - \frac{M}{N}\mathbb{I}_N\right), \sum_{i=1}^{t-1} f(\mathbf{s}_i, \mathbf{x}) - \mathbf{Y}_t \right\rangle + \sum_{i=1}^{t-1} h(\mathbf{x}_i)$, is linear in $\mathbf{Y}_t$. Here, $\mathbf{Y}_t = \sum_{i=1}^{t-1} \mathbf{y}_i$ is a function of the requests. Our Algorithm relies on perturbing $\mathbf{Y}_t$ for subset selection at $t \in \mathcal{T}$. The performance of our proposed policy under various scenarios is evaluated in Section IV. Our results in Section IV will demonstrate that this will yield performance guarantees similar to those achieved by FTPL in a single-user uncoded caching setting that directly perturbs the request numbers.

IV. Main Results

Now, we discuss the performance of our algorithm under the adversarial requests setting. We evaluate the performance of our algorithm in two different scenarios. The first scenario is unrestricted switching, where there are no restrictions on switching cache configurations; i.e., $\mathcal{T} = \{1, 2 \cdots T\}$, we are allowed to switch in every time slot. The second scenario is restricted switching, where we can only switch in a set of pre-specified time slots.

1) Unrestricted Switching: Let $|\mathcal{S}|$ be the cardinality of the set of feasible cache configurations. The following result gives an upper bound on the adversarial regret of Algorithm 1 with no restrictions on switching.

Theorem 1. *For a coded caching problem with N files, K users/caches, and cache size MF bits, let $R_{UR}(T)$ be the adversarial regret of Algorithm 1 under unrestricted switching. Then,*

$$R_{UR}(T) \leq c_1 + c_2\sqrt{T} + \frac{K^2(\sqrt{2} + 3|\mathcal{S}|)}{2\sqrt{\pi}\alpha} \sum_{t=1}^{T} \frac{1}{\sqrt{t}} = \mathcal{O}(\sqrt{T})$$

here c_1 and c_2 are constants independent of the problem parameters.

Remark 1. *Note that unlike the single-cache/uncoded caching settings [30]–[32], the expected rate expression in the coded caching setting is a non-linear function of the requests (both request profile r_t and request pattern x_t) as well as the cache configuration s_t (See Proposition 1). However, similar to single-cache settings, Algorithm 1 achieves $\mathcal{O}(\sqrt{T})$ regret for the coded caching problem.*

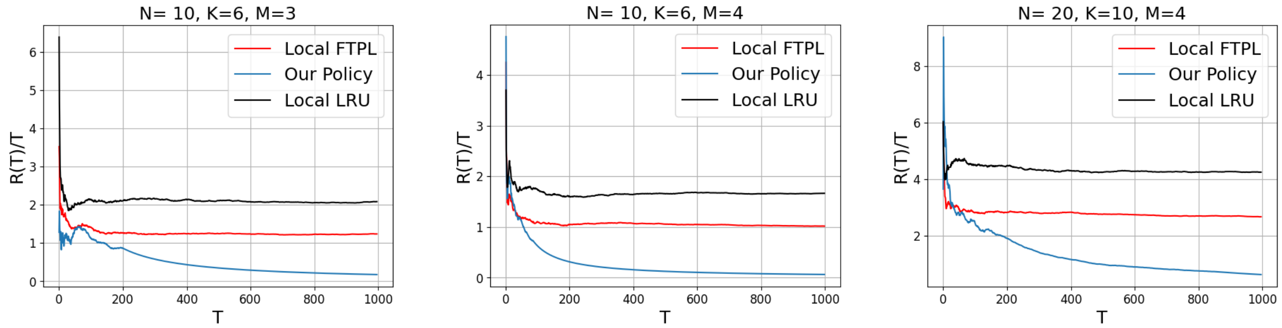

Fig. 2. We compare the performance of our proposed policy in Algorithm 1 against the benchmark policies *Local caching with FTPL* and *Local LRU*. The plots display the average regret per time step $R(T)/T$ against horizon T. The first figure compares the performance of benchmark policies against our policy for $N = 10$ files, $K = 6$ users, and cache size $M = 3$. In the next figure, we increase the cache size to $M = 4$. The last figure is plotted for $N = 20$, $K = 10$, and $M = 4$. In all cases, our policy outperforms the benchmarks policies and achieves a sublinear regret with respect to the oracle since $R(T)/T$ approaches 0 as T increases, which matches our theoretical findings. The observed trends align with our expectations regarding changes in N, K, and M.

Theorem 2, presents an upper bound on the expected number of switches incurred by Algorithm 1 under unrestricted switching.

Theorem 2. *For a coded caching problem with N files, K users/caches, and cache size MF bits, let $C_{UR}(T)$ denote the expected number of switches in cache configuration until time T for Algorithm 1 with unrestricted switching and $|\mathcal{S}|$ be the cardinality of the set of feasible cache configurations. Then*

$$C_{UR}(T) \leq \frac{3K(|\mathcal{S}|-1)}{2\sqrt{\pi}\alpha}\sum_{t=1}^{T}\frac{1}{\sqrt{t}} = \mathcal{O}(\sqrt{T})$$

Remark 2. *From Theorems 1 and 2, we conclude that Algorithm 1 with unrestricted switching achieves $\mathcal{O}(\sqrt{T})$ adversarial regret while using $\mathcal{O}(\sqrt{T})$ expected cache configuration switches.*

2) Restricted Switching: : In this scenario, the cache content can only be changed in restricted time slots given by the set $\mathcal{T}$; i.e., we are not allowed to change cache configuration outside these time slots. Let $l_k \triangleq t_k - t_{k-1}$ define the time gap between the $(k-1)^{th}$ and k^{th} switching slot, and $L \triangleq |\mathcal{T}|$ define the maximum number of allowed switches. Recall that by convention, $t_0 = 0$ and $t_L = T$, and we have $\sum_{k=1}^{L} l_k = T$. The following result gives an upper bound on the adversarial regret of Algorithm 1 when cache updates are restricted to time slots in the set $\mathcal{T}$.

Theorem 3. *For a coded caching problem with N files, K users/caches, and cache size MF bits, let $R_R^{\mathcal{T}}(T)$ be the adversarial regret of Algorithm 1 under restricted switching with switching slots $\mathcal{T}$. Then,*

$$R_R^{\mathcal{T}}(T) \leq R_{UR}(T) + \sum_{k=1}^{L}\frac{3r_{\max}^2(|\mathcal{S}|-1)l_k(l_k-1)}{4\alpha\sqrt{\pi}\sqrt{\sum_{i=1}^{k-1} l_i + 1}}.$$

Remark 3. *If $\mathcal{T} = [T]$, then we get back the unrestricted switching setting. In this case, the upper bound collapses to $R_{UR}(T)$, and the second term in RHS becomes zero. Also, for a fixed intermittent switching period $l_k = l\ \forall k$, i.e., we have $\lfloor T/l \rfloor + 1$ total switching slots, and the second term in the upper bound grows as $\mathcal{O}(\sqrt{T})$.*

V. Numerical Experiments

In this section, we compare the performance of our policy against the following benchmark policies.

1) **Local caching with FTPL**: During the placement phase, Local caching with FTPL runs the FTPL algorithm [30] independently at each user on the requests until time $t-1$ to select the files to be cached at time t in their respective local caches. In essence, the policy stores the M most popular files locally based on the requests until time $t-1$ at each user after Gaussian perturbation. Thus, the user's cache will be representative of their preferences.

2) **Local LRU**: During the placement phase, stores the last M ("Least Recently Used") requested files by user k in its cache at any point t.

For both of these policies, during the delivery phase, requests for files that are present in their respective user caches are served directly by the cache, while the remaining requests are served by the server through the broadcast channel. Furthermore, only one transmission is needed if multiple requests for the same file are made.

In Figure 2, we compare our proposed policy in Algorithm 1 against the benchmark policies for different values of M, K, N and observe that our proposed policy outperforms the benchmark policies and achieves a sublinear regret with respect to the oracle since $R(T)/T$ approaches 0 as T increases, which matches our theoretical findings. All the simulations were performed on the Movielens 1M dataset [35] containing $\sim$ 1 million ratings from 6040 users on 3706 movies. We restricted our analysis to a subset of files with a significant number of requests (>1000) and randomly selected N files from this subset. To ensure an adequate number of requests per virtual user from the restricted file set, we combined multiple users into "virtual" users.

VI. Conclusions and Future Work

Our work focuses on the widely studied coded caching problem using the lens of online learning under adversarial settings. We propose an algorithm and show that our policy achieves $\mathcal{O}(\sqrt{T})$ adversarial regret. We also consider the issue of switching cost by providing an upper bound on the expected number of switches of our algorithm under unrestricted switching and giving an upper bound on the adversarial regret with restricted switching.

There are several avenues for future work, starting from deriving universal lower bounds on Regret. Other interesting problems involve online learning with expert advice and online learning with a fixed switching budget within the context of coded caching under adversarial requests.

REFERENCES

[1] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on information theory*, vol. 60, no. 5, 2014.

[2] D. Wessels, *Web caching*. " O'Reilly Media, Inc.", 2001.

[3] M.-A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Transactions On Networking*, vol. 23, no. 4, pp. 1029–1040, 2014.

[4] J. Hachem, N. Karamchandani, and S. N. Diggavi, "Coded caching for multi-level popularity and access," *IEEE Transactions on Information Theory*, vol. 63, no. 5, pp. 3108–3141, 2017.

[5] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, "Hierarchical coded caching," *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3212–3229, 2016.

[6] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless d2d networks," *IEEE Transactions on Information Theory*, vol. 62, no. 2, pp. 849–869, 2015.

[7] J. Zhang, X. Lin, and X. Wang, "Coded caching under arbitrary popularity distributions," *IEEE Transactions on Information Theory*, vol. 64, no. 1, pp. 349–366, 2017.

[8] Y. Deng and M. Dong, "Fundamental structure of optimal cache placement for coded caching with nonuniform demands," *IEEE Transactions on Information Theory*, vol. 68, no. 10, pp. 6528–6547, 2022.

[9] J. Hachem, N. Karamchandani, and S. N. Diggavi, "Effect of number of users in multi-level coded caching," in *2015 IEEE international symposium on information theory (ISIT)*. IEEE, 2015.

[10] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," *IEEE Transactions on Information Theory*, vol. 63, no. 2, pp. 1146–1158, 2017.

[11] S. Sahraei, P. Quinton, and M. Gastpar, "The optimal memory-rate trade-off for the non-uniform centralized caching problem with two files under uncoded placement," *IEEE Transactions on Information Theory*, vol. 65, no. 12, pp. 7756–7770, 2019.

[12] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order-optimal rate of caching and coded multicasting with random demands," *IEEE Transactions on Information Theory*, vol. 63, no. 6, 2017.

[13] C. Gurjarpadhye, J. Ravi, S. Kamath, B. K. Dey, and N. Karamchandani, "Fundamental limits of demand-private coded caching," *IEEE Transactions on Information Theory*, vol. 68, no. 6, 2022.

[14] Q. Yan and D. Tuninetti, "Fundamental limits of caching for demand privacy against colluding users," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 1, pp. 192–207, 2021.

[15] V. Ravindrakumar, P. Panda, N. Karamchandani, and V. M. Prabhakaran, "Private coded caching," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 685–694, 2017.

[16] K. Wan and G. Caire, "On coded caching with private demands," *IEEE Transactions on Information Theory*, vol. 67, no. 1, pp. 358–372, 2020.

[17] A. A. Zewail and A. Yener, "Device-to-device secure coded caching," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1513–1524, 2019.

[18] T. Lattimore and C. Szepesvari, "Bandit algorithms," 2017. [Online]. Available: https://tor-lattimore.com/downloads/book/book.pdf

[19] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 836–845, 2015.

[20] E. Peter and B. S. Rajan, "Decentralized and online coded caching with shared caches: Fundamental limits with uncoded prefetching," *arXiv preprint arXiv:2101.09572*, 2021.

[21] A. Nayak, S. F. Shah, and N. Karamchandani, "On the regret of online coded caching," in *2024 National Conference on Communications (NCC)*, 2024, pp. 1–6.

[22] A. Cohen and T. Hazan, "Following the perturbed leader for online structured learning," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1034–1042.

[23] J. Abernethy, C. Lee, A. Sinha, and A. Tewari, "Online linear optimization via smoothing," in *Conference on Learning Theory*. PMLR, 2014.

[24] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proceedings of the 20th international conference on machine learning (icml-03)*, 2003, pp. 928–936.

[25] G. S. Paschos, A. Destounis, L. Vigneri, and G. Iosifidis, "Learning to cache with no regrets," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 235–243.

[26] G. S. Paschos, A. Destounis, and G. Iosifidis, "Learning to cooperate in d2d caching networks," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2019, pp. 1–5.

[27] G. Paschos, A. Destounis, and G. Iosifidis, "Online convex optimization for caching networks," vol. 28, no. 2, pp. 625–638, 2020.

[28] T. S. Salem, G. Neglia, and S. Ioannidis, "No-regret caching via online mirror descent," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[29] F. Z. Faizal, P. Singh, N. Karamchandani, and S. Moharir, "Regret-optimal online caching for adversarial and stochastic arrivals," in *Performance Evaluation Methodologies and Tools*. Springer Nature Switzerland, 2023, pp. 147–163.

[30] S. Mukhopadhyay and A. Sinha, "Online caching with optimal switching regret," in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 1546–1551.

[31] R. Bhattacharjee, S. Banerjee, and A. Sinha, "Fundamental limits on the regret of online network-caching," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2020.

[32] D. Paria and A. Sinha, "Leadcache: Regret-optimal caching in networks," *Advances in Neural Information Processing Systems*, 2021.

[33] K. S. Reddy and N. Karamchandani, "Structured index coding problem and multi-access coded caching," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 4, pp. 1266–1281, 2021.

[34] A. Nayak, K. S. Reddy, and N. Karamchandani, "On the regret of coded caching with adversarial requests," *https://github.com/hunchings/CodedCaching-fullpaper*, 2024.

[35] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, dec 2015.

VII. Appendix

A. Rate expression derivation

Coded Transmission: Recall $V_{k,u\setminus\{k\}}$ denotes all the bits that are requested by user $k \in u$, are present in the cache of all users in u, and that are not stored in the caches of any other user in $U_1 \setminus u$. The expected rate expression is equal to the sum of lengths of all these $\bigoplus_{k\in u} V_{k,u\setminus\{k\}}$ for all $u \subseteq U_1$ and $|\bigoplus_{k\in u} V_{k,u\setminus\{k\}}| = \max_{k\in u} |V_{k,u\setminus\{k\}}|$. Since the users independently pick $\frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N\rangle}$ fraction of each file being stored. The probability of a particular bit being in the cache of a particular user is $\frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N\rangle}$ for the stored files. We assume that the file size $|F|$ is large. we have

$$\max_{k\in u} |V_{k,u\setminus\{k\}}| = |F| \left(\frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N\rangle}\right)^{|u|-1} \left(1 - \frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N\rangle}\right)^{\langle \mathbf{x}_t, \mathbf{s}_t\rangle - |u|+1} + o(|F|) \tag{6}$$

$$\approx |F| \left(\frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N\rangle}\right)^{|u|-1} \left(1 - \frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N\rangle}\right)^{\langle \mathbf{x}_t, \mathbf{s}_t\rangle - |u|+1} \tag{7}$$

$$\tag{8}$$

Now for each $|u| \in \{1, 2 \cdots \langle \mathbf{x}_t, \mathbf{s}_t\rangle\}$ we have $\binom{\langle \mathbf{x}_t, \mathbf{s}_t\rangle}{|u|}$ subsets u and thus the coded rate will be

$$= |F| \sum_{|u|=1}^{\langle \mathbf{x}_t, \mathbf{s}_t\rangle} \binom{\langle \mathbf{x}_t, \mathbf{s}_t\rangle}{|u|} \left(\frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N\rangle}\right)^{|u|-1} \left(1 - \frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N\rangle}\right)^{\langle \mathbf{x}_t, \mathbf{s}_t\rangle - |u|+1} \tag{9}$$

$$= |F| \left(\frac{\langle \mathbf{s}_t, \mathbb{I}_N\rangle}{M} - 1\right) \left(1 - \left(1 - \frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N\rangle}\right)^{\langle \mathbf{x}_t, \mathbf{s}_t\rangle}\right) \tag{10}$$

Thus, the normalized expected length of the coded transmission at time t given the subset of files stored is $\mathbf{s}_t$ for a request $\mathbf{x}_t$ is given by

$$R_C = \left(\frac{\langle \mathbf{s}_t, \mathbb{I}_N\rangle}{M} - 1\right) \left(1 - \left(1 - \frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N\rangle}\right)^{\langle \mathbf{x}_t, \mathbf{s}_t\rangle}\right) \tag{11}$$

Uncoded Transmission: Each file $\notin \mathbf{s}$ that is requested by at least one of the users is directly broadcast by the server and the normalized rate (normalized by file size) of the uncoded transmission for a request vector $\mathbf{x}$ is given by

$$R_{UC} = \langle (\mathbb{I}_N - \mathbf{s}_t), \min\{\mathbf{x}, \mathbb{I}_N\}\rangle = \langle (\mathbb{I}_N - \mathbf{s}_t), \mathbf{y}_t\rangle + \tag{12}$$

Finally the expected rate $R = R_C + R_{UC}$

B. Regret upper bound for Adversarial requests

Theorem 1
For a coded caching problem with N files, K users/caches, and cache size MF bits, let $R^A_{(\eta,\mathrm{UR})}(T)$ be the adversarial regret of Algorithm 1 under unrestricted switching with a learning rate $\alpha\sqrt{t}$. Then,

$$R^A_{(\eta,\mathrm{UR})}(T) \le \frac{3r_{\max} r^C_{\max}(|\mathcal{S}|-1)}{2\sqrt{\pi}\alpha} \sum_{t=1}^{T} \frac{1}{\sqrt{t}} + \alpha\sqrt{T}\mathcal{G}_{\max}(\gamma) + \frac{\max\left\{\frac{M}{N}, \left(1-\frac{M}{N}\right)\right\} K^2}{\sqrt{2\pi}\alpha} \sum_{t=1}^{T} \frac{1}{\sqrt{t}} + \eta_1 \mathcal{G}(\gamma) = \mathcal{O}(\sqrt{T}) \tag{13}$$

Here $\mathcal{G}_{\max}(\gamma)$ is the maximum Gaussian width given by

$$\mathcal{G}_{\max}(\gamma) = \max_{a\in\mathcal{A}} \mathbb{E}_\gamma[\langle a, \gamma\rangle] = -\min_{a\in\mathcal{A}} \mathbb{E}_\gamma[\langle a, \gamma\rangle]$$

with

$$\mathcal{A} = \left\{a = [a_1, a_2, ..., a_N]^T | a_i \in \{0,1\}; \langle a, \mathbb{I}_N\rangle \ge M\right\}$$

and $\gamma \sim \mathcal{N}(0, I_{N\times N})$. $\mathcal{G}_{\max}(\gamma)$ can be upper bounded as $\max_{M\le m\le N} m\sqrt{\log(\frac{N}{m})}$. Here, the constants $r_{\max}$ and $r^C_{\max}$ are the maximum rate of the server transmission (coded + uncoded) and coded transmission over the set of all request pattern and cache configuration pairs, respectively. Note that both $r_{\max}$ and $r^C_{\max}$ can be upper bounded by K since the maximum number of unique requests in a slot is K.

Proof. The normalized expected rate expression for a subset (denoted by $\mathbf{s}_t$) of all files such that $\mathbf{s}_t \in \mathcal{S}$ is cached at time t according to the placement and delivery scheme described in section II is given by

$$\left(\frac{\langle \mathbf{s}_t, \mathbb{I}_N\rangle}{M} - 1\right) \left(1 - \left(1 - \frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N\rangle}\right)^{\langle \mathbf{x}_t, \mathbf{s}_t\rangle}\right) + \langle (\mathbb{I}_N - \mathbf{s}_t), y_t\rangle \tag{14}$$

This can be rewritten as

$$\left\langle \left(\mathbf{s}_t - \frac{M}{N}\mathbb{I}_N\right), \left(f(\mathbf{x}_t, \mathbf{s}_t) - \mathbf{y}_t\right)\right\rangle + h(\mathbf{x}_t) \tag{15}$$

where $f(\mathbf{x}_t,\mathbf{s}_t)=\frac{1}{M}\left(1-\left(1-\frac{M}{\langle\mathbf{s}_t,\mathbb{I}\rangle}\right)^{\langle\mathbf{x}_t,\mathbf{s}_t\rangle}\right)\mathbb{I}_N$ and
$h(\mathbf{x}_t)=\left(1-\frac{M}{N}\right)\langle\mathbf{y}_t,\mathbb{I}_N\rangle$. Now, from the definition of regret for the subset selection algorithm 1, we have

$$R^{A}_{(\eta,\mathrm{UR})}(T)=\underbrace{\sum_{t=1}^{T}\left(\mathbb{E}_\gamma\left[\left(\frac{\langle\mathbf{s}_t,\mathbb{I}_N\rangle}{M}-1\right)\left(1-\left(1-\frac{M}{\langle\mathbf{s}_t,\mathbb{I}_N\rangle}\right)^{\langle\mathbf{x}_t,\mathbf{s}_t\rangle}\right)+\langle\mathbf{y}_t,(\mathbb{I}_N-\mathbf{s}_t)\rangle\right]\right)}_{\text{Policy Cumulative Rate}}-$$
$$\underbrace{\min_{\mathbf{s}\in\mathcal{S}}\sum_{t=1}^{T}\left[\left(\frac{\langle\mathbf{s},\mathbb{I}_N\rangle}{M}-1\right)\left(1-\left(1-\frac{M}{\langle s,\mathbb{I}_N\rangle}\right)^{\langle\mathbf{x}_t,s\rangle}\right)+\langle\mathbf{y}_t,(\mathbb{I}_N-\mathbf{s})\rangle\right]}_{\text{Oracle Cumulative Rate}} \quad (16)$$

which can be rewritten as

$$R^{A}_{(\eta,\mathrm{UR})}(T)=\sum_{t=1}^{T}\mathbb{E}_\gamma\left[\left\langle\left(\mathbf{s}_t-\frac{M}{N}\mathbb{I}_N\right),(f(\mathbf{x}_t,\mathbf{s}_t)-\mathbf{y}_t)\right\rangle+h(\mathbf{x}_t)\right]-$$
$$\min_{s\in\mathcal{S}}\sum_{t=1}^{T}\left[\left\langle\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right),(f(\mathbf{x}_t,\mathbf{s})-y_t)\right\rangle+h(\mathbf{x}_t)\right] \quad (17)$$

$$=\sum_{t=1}^{T}\mathbb{E}_\gamma\left[\left\langle\left(\mathbf{s}_t-\frac{M}{N}\mathbb{I}_N\right),(f(\mathbf{x}_t,\mathbf{s}_t)-y_t)\right\rangle\right]-\min_{s\in\mathcal{S}}\left\langle\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right),\sum_{t=1}^{T}(f(\mathbf{x}_t,\mathbf{s})-y_t)\right\rangle \quad (18)$$

Equation 18 comes from canceling $h(\mathbf{x}_t)$ terms on both sides, which are independent of s. Let $\mathbf{X}_t$ be a $N\times(t-1)$ vector $[\mathbf{x}_1,\mathbf{x}_2\cdots\mathbf{x}_{t-1}]$ of observations until time t and $\mathbf{Y}_t=\sum_{i=1}^{t-1}\mathbf{y}_t$. Define a potential function ϕ_t for all instants t.

$$\phi_t(X,Y)=\mathbb{E}_\gamma\left[\min_{\mathbf{s}\in\mathcal{S}}\left\langle\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right),\sum_{i=1}^{t-1}f(\mathbf{x}_i,\mathbf{s})-\mathbf{Y}+\eta_t\gamma\right\rangle\right] \quad (19)$$

Note that the dimension of X input to ϕ_t is $N\times(t-1)$ and Y input is N. The function maps them to a scalar. We have the partial derivative (as shown in [23], [22])

$$\nabla_{\mathbf{Y}}\phi_t(\mathbf{X}_t,\mathbf{Y}_t)=-\mathbb{E}_\gamma\left[\mathbf{s}_t-\frac{M}{N}\mathbb{I}_N\right] \quad (20)$$

Thus

$$\mathbb{E}_\gamma\left[\left\langle\left(\mathbf{s}_t-\frac{M}{N}\mathbb{I}_N\right),(f(\mathbf{x}_t,\mathbf{s}_t)-\mathbf{y}_t)\right\rangle\right] \quad (21)$$
$$=\mathbb{E}_\gamma\left[\left\langle\left(\mathbf{s}_t-\frac{M}{N}\mathbb{I}_N\right),f(\mathbf{x}_t,\mathbf{s}_t)-(\mathbf{Y}_{t+1}-\mathbf{Y}_t)\right\rangle\right] \quad (22)$$
$$=\mathbb{E}_\gamma\left[\left\langle\left(\mathbf{s}_t-\frac{M}{N}\mathbb{I}_N\right),f(\mathbf{x}_t,\mathbf{s}_t)\right\rangle\right]-\mathbb{E}_\gamma\left[\left\langle\left(\mathbf{s}_t-\frac{M}{N}\mathbb{I}_N\right),(\mathbf{Y}_{t+1}-\mathbf{Y}_t)\right\rangle\right] \quad (23)$$

Consider the term

$$-\mathbb{E}_\gamma\left[\left\langle\left(\mathbf{s}_t-\frac{M}{N}\mathbb{I}_N\right),(\mathbf{Y}_{t+1}-\mathbf{Y}_t)\right\rangle\right] \quad (24)$$
$$=\langle\nabla_{\mathbf{Y}}\phi_t(\mathbf{X}_t,\mathbf{Y}_t),(\mathbf{Y}_{t+1}-\mathbf{Y}_t)\rangle \quad (25)$$
$$=\phi_t(\mathbf{X}_t,\mathbf{Y}_{t+1})-\phi_t(\mathbf{X}_t,\mathbf{Y}_t)-\frac{1}{2}\left\langle\mathbf{y}_t,\nabla^2_{\mathbf{Y}}\phi_t(\mathbf{X}_t,\tilde{\mathbf{Y}}_t)\mathbf{y}_t\right\rangle \quad (26)$$

The above steps come from using the Taylor series expansion and $\tilde{\mathbf{Y}}_t=\mathbf{Y}_t+\theta\mathbf{y}_t$ for some $\theta\in[0,1]$. Thus we have

$$\sum_{t=1}^{T}\mathbb{E}_\gamma\left[\left\langle\left(\mathbf{s}_t-\frac{M}{N}\mathbb{I}_N\right),(f(\mathbf{x}_t,\mathbf{s}_t)-\mathbf{y}_t)\right\rangle\right] \quad (27)$$
$$=\sum_{t=1}^{T}\mathbb{E}_\gamma\left[\left\langle\left(\mathbf{s}_t-\frac{M}{N}\mathbb{I}_N\right),f(\mathbf{x}_t,\mathbf{s}_t)\right\rangle\right]-\sum_{t=1}^{T}\mathbb{E}_\gamma\left[\left\langle\left(\mathbf{s}_t-\frac{M}{N}\mathbb{I}_N\right),(\mathbf{Y}_{t+1}-\mathbf{Y}_t)\right\rangle\right] \quad (28)$$
$$=\sum_{t=1}^{T}\mathbb{E}_\gamma\left[\left\langle\left(\mathbf{s}_t-\frac{M}{N}\mathbb{I}_N\right),f(\mathbf{x}_t,\mathbf{s}_t)\right\rangle\right]+\underbrace{\sum_{t=1}^{T}[\phi_t(\mathbf{X}_t,\mathbf{Y}_{t+1})-\phi_t(\mathbf{X}_t,\mathbf{Y}_t)]}_{T_1}-\underbrace{\sum_{t=1}^{T}\frac{1}{2}\left\langle\mathbf{y}_t,\nabla^2_{\mathbf{Y}}\phi_t(\mathbf{X}_t,\tilde{\mathbf{Y}}_t)\mathbf{y}_t\right\rangle}_{T_2} \quad (29)$$

We simplify T_1 as follows

$$\sum_{t=1}^{T}\left[\phi_t(\mathbf{X}_t,\mathbf{Y}_{t+1})-\phi_t(\mathbf{X}_t,\mathbf{Y}_t)\right] \tag{30}$$

$$=\phi_T(\mathbf{X}_T,\mathbf{Y}_{T+1})+\underbrace{\sum_{t=1}^{T-1}\left[\phi_t(\mathbf{X}_t,\mathbf{Y}_{t+1})-\phi_{t+1}(\mathbf{X}_{t+1},\mathbf{Y}_{t+1})\right]}_{T_3}-\phi_1(\mathbf{X}_1,\mathbf{Y}_1) \tag{31}$$

Consider T_3. We have

$$\phi_t(\mathbf{X}_t,\mathbf{Y}_{t+1})-\phi_{t+1}(\mathbf{X}_{t+1},\mathbf{Y}_{t+1})=\mathbb{E}_\gamma\left[\min_{\mathbf{s}\in\mathcal{S}}\left\langle\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right),\sum_{i=1}^{t-1}f(\mathbf{x}_i,\mathbf{s})-\mathbf{Y}_{t+1}+\eta_t\gamma\right\rangle\right.$$
$$\left.-\min_{\mathbf{s}\in\mathcal{S}}\left\langle\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right),\sum_{i=1}^{t}f(\mathbf{x}_i,\mathbf{s})-\mathbf{Y}_{t+1}+\eta_{t+1}\gamma\right\rangle\right. \tag{32}$$

$$=\mathbb{E}_\gamma\left[\min_{\mathbf{s}\in\mathcal{S}}\left\langle\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right),\sum_{i=1}^{t-1}f(\mathbf{x}_i,\mathbf{s})-\mathbf{Y}_{t+1}+\eta_t\gamma\right\rangle\right.$$
$$\left.-\left\langle\left(\mathbf{s}_{t+1}-\frac{M}{N}\mathbb{I}_N\right),\sum_{i=1}^{t}f(\mathbf{x}_i,\mathbf{s}_{t+1})-\mathbf{Y}_{t+1}+\eta_{t+1}\gamma\right\rangle\right. \tag{33}$$

$$\leq-\mathbb{E}_\gamma\left[\min_{\mathbf{s}\in\mathcal{S}}\left\langle\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right),(\eta_{t+1}-\eta_t)\gamma\right\rangle\right]-\mathbb{E}_\gamma\left[\left\langle\left(\mathbf{s}_{t+1}-\frac{M}{N}\mathbb{I}_N\right),f(\mathbf{x}_t,\mathbf{s}_{t+1})\right\rangle\right] \tag{34}$$

$$=-\mathbb{E}_\gamma\left[\min_{\mathbf{s}\in\mathcal{S}}\langle\mathbf{s},(\eta_{t+1}-\eta_t)\gamma\rangle\right]-\mathbb{E}_\gamma\left[\left\langle\left(\mathbf{s}_{t+1}-\frac{M}{N}\mathbb{I}_N\right),f(\mathbf{x}_t,\mathbf{s}_{t+1})\right\rangle\right] \tag{35}$$

$$\leq|\eta_{t+1}-\eta_t|\mathcal{G}(\gamma)-\mathbb{E}_\gamma\left[\left\langle\left(\mathbf{s}_{t+1}-\frac{M}{N}\mathbb{I}_N\right),f(\mathbf{x}_t,\mathbf{s}_{t+1})\right\rangle\right] \tag{36}$$

Equation 33 follows as the minimizer in the second term is $\mathbf{s}_{t+1}$. Equation 35 comes from the fact $\mathbb{E}[\gamma]=0$, and equation 34 follows from the steps below. Here $\mathcal{G}(\gamma)$ is the maximum Gaussian width, which can be upper bounded by $\max_{M\leq m\leq N} m\sqrt{\log(\frac{N}{m})}$ as shown in [22].

$$-\mathbb{E}_\gamma\left[\left\langle\left(\mathbf{s}_{t+1}-\frac{M}{N}\mathbb{I}_N\right),\sum_{i=1}^{t}f(\mathbf{x}_i,\mathbf{s}_{t+1})-\mathbf{Y}_{t+1}+\eta_{t+1}\gamma\right\rangle\right] \tag{37}$$

$$=-\mathbb{E}_\gamma\left[\left\langle\left(\mathbf{s}_{t+1}-\frac{M}{N}\mathbb{I}_N\right),\sum_{i=1}^{t-1}f(\mathbf{x}_i,\mathbf{s}_{t+1})-\mathbf{Y}_{t+1}+\eta_{t+1}\gamma\right\rangle\right]-\mathbb{E}_\gamma\left[\left(\mathbf{s}_{t+1}-\frac{M}{N}\mathbb{I}_N\right)f(\mathbf{x}_t,\mathbf{s}_{t+1})\right] \tag{38}$$

$$\leq-\mathbb{E}_\gamma\left[\min_{\mathbf{s}\in\mathcal{S}}\left\langle\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right),\sum_{i=1}^{t-1}f(\mathbf{x}_i,\mathbf{s})-\mathbf{Y}_{t+1}+\eta_{t+1}\gamma\right\rangle\right]-\mathbb{E}_\gamma\left[\left(\mathbf{s}_{t+1}-\frac{M}{N}\mathbb{I}_N\right)f(\mathbf{x}_t,\mathbf{s}_{t+1})\right] \tag{39}$$

$$\leq-\mathbb{E}_\gamma\left[\min_{\mathbf{s}\in\mathcal{S}}\left\langle\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right),\sum_{i=1}^{t-1}f(\mathbf{x}_i,\mathbf{s})-\mathbf{Y}_{t+1}+\eta_t\gamma\right\rangle\right]-\mathbb{E}_\gamma\left[\left(\mathbf{s}_{t+1}-\frac{M}{N}\mathbb{I}_N\right)f(\mathbf{x}_t,\mathbf{s}_{t+1})\right]-$$
$$\mathbb{E}_\gamma\left[\min_{\mathbf{s}\in\mathcal{S}}\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right)(\eta_{t+1}-\eta_t)\gamma\right] \tag{40}$$

40 here follows from the fact $\min_x(f_1(x))+\min_x(f_2(x))\leq\min_x(f_1(x)+f_2(x))$. Also, we have

$$\phi_T(\mathbf{X}_T,Y_{T+1})=\mathbb{E}_\gamma\left[\min_{\mathbf{s}\in\mathcal{S}}\left\langle\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right),\sum_{i=1}^{T-1}f(\mathbf{x}_i,\mathbf{s})-\mathbf{Y}_{T+1}+\eta_T\gamma\right\rangle\right] \tag{41}$$

$$\leq\mathbb{E}_\gamma\left[\min_{\mathbf{s}\in\mathcal{S}}\left\langle\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right),\sum_{i=1}^{T}f(\mathbf{x}_i,\mathbf{s})-\mathbf{Y}_{T+1}+\eta_T\gamma\right\rangle\underbrace{-\min_{\mathbf{s}\in\mathcal{S}}\left\langle\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right),f(\mathbf{x}_T,\mathbf{s})\right\rangle}_{\leq 0}\right] \tag{42}$$

$$\leq\mathbb{E}_\gamma\left[\min_{\mathbf{s}\in\mathcal{S}}\left\langle\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right),\sum_{i=1}^{T}f(\mathbf{x}_i,\mathbf{s})-\mathbf{Y}_{T+1}+\eta_T\gamma\right\rangle\right] \tag{43}$$

$$\leq \min_{\mathbf{s}\in\mathcal{S}} \mathbb{E}_\gamma \left[\left\langle \left(\mathbf{s} - \frac{M}{N}\mathbb{I}_N \right), \sum_{i=1}^{T} f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_{T+1} + \eta_T \gamma \right\rangle \right] \tag{44}$$

$$= \min_{\mathbf{s}\in\mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N}\mathbb{I}_N \right), \sum_{i=1}^{T} f(\mathbf{x}_i, \mathbf{s}) - Y_{T+1} \right\rangle \tag{45}$$

42 again comes from $\min_x(f_1(x)) + \min_x(f_2(x)) \leq \min_x(f_1(x) + f_2(x))$ and 44 follows from Jenson's inequality. Now from 45 and 36 we have

$$T_1 = \phi_T(\mathbf{X}_T, Y_{T+1}) + \underbrace{\sum_{t=1}^{T-1} \left[\phi_t(\mathbf{X}_t, Y_{t+1}) - \phi_{t+1}(\mathbf{X}_{t+1}, Y_{t+1}) \right]}_{T_3} - \phi_1(\mathbf{X}_1, Y_1) \tag{46}$$

$$\leq \phi_T(\mathbf{X}_T, Y_{T+1}) + \underbrace{\sum_{t=1}^{T-1} |\eta_{t+1} - \eta_t| \mathcal{G}(\gamma) - \mathbb{E}_\gamma \left[\left(\mathbf{s}_{t+1} - \frac{M}{N}\mathbb{I}_N \right) f(\mathbf{x}_t, \mathbf{s}_{t+1}) \right]}_{\geq T_3} - \phi_1(\mathbf{X}_1, Y_1) \tag{47}$$

$$\tag{48}$$

$$\leq \min_{\mathbf{s}\in\mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N}\mathbb{I}_N \right), \sum_{i=1}^{T} f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_{T+1} \right\rangle + |\eta_T - \eta_1| \mathcal{G}(\gamma) - \sum_{t=1}^{T-1} \mathbb{E}_\gamma \left[\left(\mathbf{s}_{t+1} - \frac{M}{N}\mathbb{I}_N \right) f(\mathbf{x}_t, \mathbf{s}_{t+1}) \right] \underbrace{-\phi_1(\mathbf{X}_1, Y_1)}_{\eta_1 \gamma} \tag{49}$$

$$\leq \min_{\mathbf{s}\in\mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N}\mathbb{I}_N \right), \sum_{i=1}^{T} f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_{T+1} \right\rangle + \eta_T \mathcal{G}(\gamma) - \sum_{t=1}^{T-1} \mathbb{E}_\gamma \left[\left(\mathbf{s}_{t+1} - \frac{M}{N}\mathbb{I}_N \right) f(\mathbf{x}_t, \mathbf{s}_{t+1}) \right] + \eta_1 \mathcal{G}(\gamma) \tag{50}$$

Lastly, T_2 can be bounded in the same way as

$$-\langle \mathbf{y}_t, \nabla^2 \phi_t(\mathbf{X}_t, \tilde{\mathbf{Y}}_t) \mathbf{y}_t \rangle \leq \left| \sum_{1=1}^{N} \sum_{j=1}^{N} \mathbf{y}_t^i \mathbf{y}_t^j |\nabla^2 \phi_t(\mathbf{X}_t, \tilde{Y}_t)|_{ij} \right| \tag{51}$$

As explained in Lemma 7 of [23] and done in section 7.3 of [31]

$$\nabla^2 \phi_t(\mathbf{X}_t, \tilde{\mathbf{Y}}_t)_{ij} = \frac{1}{\eta_t} \mathbb{E}_\gamma [\nabla \widehat{\phi}_t(\mathbf{X}_t, \tilde{\mathbf{Y}}_t - \eta\gamma)_i \gamma_j] \tag{52}$$

Where for $\mathbf{X}$ a $N \times (t-1)$ dimension vector and $\mathbf{Y}$ a N-dimensional vector we have

$$\widehat{\phi}_t(\mathbf{X}, \mathbf{Y}) = \min_{\mathbf{s}\in\mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N}\mathbb{I}_N \right), \sum_{i=1}^{t-1} f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y} \right\rangle \tag{53}$$

Thus $|\nabla^2 \phi_t(\mathbf{X}_t, \tilde{\mathbf{Y}}_t)|_{ij} \leq \frac{1}{\eta_t} \mathbb{E}_\gamma [|\nabla \widehat{\phi}_t(\mathbf{X}_t, \tilde{\mathbf{Y}}_t - \eta\gamma)_i||\gamma_j|] \leq \frac{1}{\eta_t} \mathbb{E}_\gamma \left[\left| \left(\mathbf{s}(\mathbf{X}_t, \tilde{\mathbf{Y}}_t - \eta_t\gamma) - \frac{M}{N}\mathbb{I}_N \right)_i \right| |\gamma_j| \right] \leq \frac{(1-\frac{M}{N})}{\eta_t} \mathbb{E}_\gamma[|\gamma_j|] = \frac{(1-\frac{M}{N})}{\eta_t} \sqrt{\frac{2}{\pi}}$. The last step comes from the fact that the max absolute value of an entry in $\left(\mathbf{s}(\mathbf{X}_t, \mathbf{Y}_t - \eta_t\gamma) - \frac{M}{N}\mathbb{I}_N\right)$ is $\left(1 - \frac{M}{N}\right)$ for $2M \leq N$. Here $\mathbf{s}(\mathbf{X}_t, \mathbf{Y}_t + \eta_t\gamma)_i$ is the i^{th} entry of the vector with $\mathbf{s}$ chosen under inputs $\mathbf{X}_t$ and $Y_t - \eta\gamma$

$$-\langle \mathbf{y}_t, \nabla^2 \phi_t(\mathbf{x}_t, \tilde{\mathbf{Y}}_t) y_t \rangle \leq \frac{(1-\frac{M}{N})}{\eta_t} \sqrt{\frac{2}{\pi}} \left| \sum_{1=1}^{N} \sum_{j=1}^{N} \mathbf{y}_t^i \mathbf{y}_t^j \right| \leq \frac{K^2 \left(1-\frac{M}{N}\right)}{\eta_t} \sqrt{\frac{2}{\pi}} \tag{54}$$

since

$$\left| \sum_{1=1}^{N} \sum_{j=1}^{N} y^i y^j \right| \leq K^2 \tag{55}$$

Thus $T_2 \leq \frac{K^2\left(1-\frac{M}{N}\right)}{\sqrt{2\pi}} \sum_{t=1}^{T} \frac{1}{\eta_t}$. This follows because we have at most K distinct requests at any time t. Thus, from 18, 29, 49, 54 and the above expressions, we have the regret for a horizon T bounded by

$$\leq \underbrace{\sum_{t=1}^{T} \mathbb{E}_\gamma \left[\left\langle \left(\mathbf{s}_t - \frac{M}{N}\mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_t) \right\rangle - \left\langle \left(\mathbf{s}_{t+1} - \frac{M}{N}\mathbb{I}_N \right) f(\mathbf{x}_t, \mathbf{s}_{t+1}) \right\rangle \right]}_{T_5} + \eta_T \mathcal{G}(\gamma) + \eta_1 \mathcal{G}(\gamma)$$

$$+\min_{\mathbf{s}\in\mathcal{S}}\left\langle\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right),\sum_{t=1}^{T}\left(f(\mathbf{x}_t,\mathbf{s})-\mathbf{y}_t\right)\right\rangle+\frac{K^2\left(1-\frac{M}{N}\right)}{\sqrt{2\pi}}\sum_{t=1}^{T}\frac{1}{\eta_t}-\underbrace{\min_{\mathbf{s}\in\mathcal{S}}\left\langle\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right),\sum_{t=1}^{T}\left(f(\mathbf{x}_t,\mathbf{s})-\mathbf{y}_t\right)\right\rangle}_{\text{Oracle rate}} \tag{56}$$

$$=\underbrace{\sum_{t=1}^{T}\mathbb{E}_\gamma\left[\left\langle\left(\mathbf{s}_t-\frac{M}{N}\mathbb{I}_N\right),f(\mathbf{x}_t,\mathbf{s}_t)\right\rangle-\left\langle\left(\mathbf{s}_{t+1}-\frac{M}{N}\mathbb{I}_N\right)f(\mathbf{x}_t,\mathbf{s}_{t+1})\right\rangle\right]}_{T_5}+\eta_T\mathcal{G}(\gamma)+\frac{K^2\left(1-\frac{M}{N}\right)}{\sqrt{2\pi}}\sum_{t=1}^{T}\frac{1}{\eta_t}+\eta_1\mathcal{G}(\gamma) \tag{57}$$

$$\leq\frac{3r_{\max}r^C_{\max}(|\mathcal{S}|-1)}{2\sqrt{\pi}\alpha}\sum_{t=1}^{T}\frac{1}{\sqrt{t}}+\eta_T\mathcal{G}(\gamma)+\frac{K^2\left(1-\frac{M}{N}\right)}{\sqrt{2\pi}}\sum_{t=1}^{T}\frac{1}{\eta_t}+\eta_1\mathcal{G}(\gamma)+\eta_1\mathcal{G}(\gamma) \tag{58}$$

Equation 58 follows from the bound on the expected number of switches in theorem 2 and the fact that the terms inside the T_5 summation are zero when $\mathbf{s}_t=\mathbf{s}_{t+1}$. The term $\left\langle\left(\mathbf{s}_t-\frac{M}{N}\mathbb{I}_N\right),f(\mathbf{x}_t,\mathbf{s}_t)\right\rangle-\left\langle\left(\mathbf{s}_{t+1}-\frac{M}{N}\mathbb{I}_N\right)f(\mathbf{x}_t,\mathbf{s}_{t+1})\right\rangle$ is essentially the difference in expected coded rate at time t and $t+1$ and can be upper bounded by max coded rate $r^C_{\max}$. Thus, we upper bound T_5 by the product of the expected number of switches and $r^C_{\max}$ i.e,

$$T_5=\sum_{t=1}^{T}\mathbb{E}_\gamma\left[\left\langle\left(\mathbf{s}_t-\frac{M}{N}\mathbb{I}_N\right),f(\mathbf{x}_t,\mathbf{s}_t)\right\rangle-\left\langle\left(\mathbf{s}_{t+1}-\frac{M}{N}\mathbb{I}_N\right)f(\mathbf{x}_t,\mathbf{s}_{t+1})\right\rangle\right] \tag{59}$$

$$=\sum_{t=1}^{T}\mathbb{E}_\gamma\left[\left(\left\langle\left(\mathbf{s}_t-\frac{M}{N}\mathbb{I}_N\right),f(\mathbf{x}_t,\mathbf{s}_t)\right\rangle-\left\langle\left(\mathbf{s}_{t+1}-\frac{M}{N}\mathbb{I}_N\right)f(\mathbf{x}_t,\mathbf{s}_{t+1})\right\rangle\right)\mathbb{I}(\mathbf{s}_{t+1}\neq\mathbf{s}_t)\right] \tag{60}$$

$$\leq\sum_{t=1}^{T}\mathbb{E}_\gamma\left[r^C_{\max}\mathbb{I}(\mathbf{s}_{t+1}\neq\mathbf{s}_t)\right]=r^C_{\max}\sum_{t=1}^{T}\mathbb{E}_\gamma\left[\mathbb{I}(\mathbf{s}_{t+1}\neq\mathbf{s}_t)\right] \tag{61}$$

Note that $\left(1-\frac{M}{N}\right)$ should be replaced by $\left(\frac{M}{N}\right)$ for the case $2M>N$ in equation 58 □

C. Switching Cost lemma: Adversarial requests

Lemma 1 The probability of switching cache configuration for our policy (Algorithm 1) in the step t given the history up to time $t-1$ and perturbation vector γ, $(\mathbf{x}_1\cdots\mathbf{x}_{t-1},\gamma)$ in the adversarial setting for $s\neq\mathbf{s}_t$ is given by

$$\mathbb{P}\left((\mathbf{s}_{t+1}=\mathbf{s})|(\mathbf{x}_1\cdots\mathbf{x}_{t-1},\gamma)\right)\leq\frac{3r_{\max}}{2\alpha\sqrt{\pi}\sqrt{t+1}} \tag{62}$$

Here, the constants $r_{\max}$ is the maximum rate of the server transmission (coded + uncoded) over the set of all request pattern and cache configuration pairs. Note that $\mathbf{s}_t$ is deterministic for the algorithm given $(\mathbf{x}_1\cdots\mathbf{x}_{t-1},\gamma)$

Proof. Define

$$R_t(\mathbf{s},\mathbf{X},\mathbf{Y})=\left\langle\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right),\sum_{i=1}^{t-1}f(\mathbf{x}_i,\mathbf{s})-\mathbf{Y}\right\rangle \tag{63}$$

Note that the dimension of $\mathbf{X}$ input to R_t is $N\times(t-1)$ and that of $\mathbf{s}$ and $\mathbf{Y}$ input is N. The function maps them to a scalar. For $\mathbf{s}\neq\mathbf{s}_t$. For a given $\mathbf{s}_t$ at time t, a switch happens can happen when for some $\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}_t$ we have $\{R_t(\mathbf{s}_t,\mathbf{X}_t,\mathbf{Y}_t-\eta_t\gamma)\leq R_t(\mathbf{s},\mathbf{X}_t,\mathbf{Y}_t-\eta_t\gamma)\}\ \cap\ \{R_{t+1}(\mathbf{s}_t,\mathbf{X}_{t+1},\mathbf{Y}_{t+1}-\eta_{t+1}\gamma)\geq R_{t+1}(\mathbf{s},\mathbf{X}_{t+1},\mathbf{Y}_{t+1}-\eta_{t+1}\gamma)\}$. Let $\mathbb{P}^{\mathcal{H}_t}(.)$ denote the function $\mathbb{P}\left((.|(\mathbf{x}_1\cdots\mathbf{x}_{t-1},\gamma)\right)$. Thus we have $\mathbb{P}\left((\mathbf{s}_{t+1}=\mathbf{s})|(\mathbf{x}_1\cdots\mathbf{x}_{t-1},\gamma)\right)$

$$\leq\mathbb{P}^{\mathcal{H}_t}_\gamma(\{R_t(\mathbf{s}_t,\mathbf{X}_t,\mathbf{Y}_t-\eta_t\gamma)\leq R_t(\mathbf{s},\mathbf{X}_t,\mathbf{Y}_t-\eta_t\gamma)\}\cap\{R_{t+1}(\mathbf{s}_t,\mathbf{X}_{t+1},\mathbf{Y}_{t+1}-\eta_{t+1}\gamma)\geq R_{t+1}(\mathbf{s},\mathbf{X}_{t+1},\mathbf{Y}_{t+1}-\eta_{t+1}\gamma)\}) \tag{64}$$

$$=\mathbb{P}^{\mathcal{H}_t}_\gamma\left(\left\{R_t(\mathbf{s}_t,\mathbf{X}_t,\mathbf{Y}_t)+\left\langle\left(\mathbf{s}_t-\frac{M}{N}\mathbb{I}_N\right),\eta_t\gamma\right\rangle\leq R_t(\mathbf{s},\mathbf{X}_t,\mathbf{Y}_t)+\left\langle\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right),\eta_t\gamma\right\rangle\right\}\cap\left\{R_{t+1}(\mathbf{s}_t,\mathbf{X}_{t+1},Y_{t+1})+\left\langle\left(\mathbf{s}_t-\frac{M}{N}\mathbb{I}_N\right),\eta_{t+1}\gamma\right\rangle\geq R_{t+1}(\mathbf{s},\mathbf{X}_{t+1},\mathbf{Y}_{t+1})+\left\langle\left(\mathbf{s}-\frac{M}{N}\mathbb{I}_N\right),\eta_{t+1}\gamma\right\rangle\right\}\right) \tag{65}$$

$$=\mathbb{P}^{\mathcal{H}_t}_\gamma\left(\frac{R_t(\mathbf{s}_t,\mathbf{X}_t,\mathbf{Y}_t)-R_t(\mathbf{s},\mathbf{X}_t,\mathbf{Y}_t)}{\eta_t}\leq\langle(\mathbf{s}-\mathbf{s}_t),\gamma\rangle\leq\frac{R_{t+1}(\mathbf{s}_t,\mathbf{X}_{t+1},\mathbf{Y}_{t+1})-R_{t+1}(\mathbf{s},\mathbf{X}_{t+1},\mathbf{Y}_{t+1})}{\eta_{t+1}}\right) \tag{66}$$

$$\leq \mathbb{P}_{\gamma}^{\mathcal{H}_t}\left(\frac{R_t(\mathbf{s}_t,\mathbf{X}_t,\mathbf{Y}_t)-R_t(\mathbf{s},\mathbf{X}_t,\mathbf{Y}_t)}{\eta_t} \leq \langle(\mathbf{s}-\mathbf{s}_t),\gamma\rangle \leq \frac{R_t(\mathbf{s}_t,\mathbf{X}_t,\mathbf{Y}_t)+r_{\max}-R_t(\mathbf{s},\mathbf{X}_t,\mathbf{Y}_t)}{\eta_{t+1}}\right) \tag{67}$$

$$\leq \frac{1}{\sqrt{2\pi}\sigma_{(\mathbf{s},\mathbf{s}_t)}}\left(\frac{r_{\max}}{\eta_{t+1}}+\left(\frac{1}{\eta_t}-\frac{1}{\eta_{t+1}}\right)\left(R_t(\mathbf{s}_{t+1},\mathbf{X}_t,Y_t)-R_t(\mathbf{s}_t,\mathbf{X}_t,Y_t)\right)\right) \tag{68}$$

$$\leq \frac{1}{\sqrt{2\pi}\alpha}\left(\frac{r_{\max}}{\sqrt{t+1}}+\left(\frac{1}{\sqrt{t}}-\frac{1}{\sqrt{t+1}}\right)tr_{\max}\right)=\frac{1}{\sqrt{2\pi}\alpha}\left(\frac{r_{\max}}{\sqrt{t+1}}+\left(\frac{\sqrt{t}}{\sqrt{t+1}(\sqrt{t+1}+\sqrt{t})}\right)r_{\max}\right) \tag{69}$$

$$\leq \frac{3r_{\max}}{2\alpha\sqrt{\pi}\sqrt{t+1}} \tag{70}$$

where the max rate incurred in one time step is denoted by $r_{\max}$. Equation 67 can be obtained by expressing $R_t(\mathbf{s},\mathbf{X},\mathbf{Y})$ terms as a sum of $K(\mathbf{s},\mathbf{x})$ terms. $R_{t+1}(\mathbf{s}_t,\mathbf{x}_{t+1},\mathbf{Y}_{t+1}) - R_{t+1}(\mathbf{s},\mathbf{x}_{t+1},\mathbf{Y}_{t+1}) = (\sum_{i=1}^{t} K(\mathbf{s}_t,\mathbf{x}_t) - \sum_{i=1}^{t} h(\mathbf{x}_i)) - (\sum_{i=1}^{t} K(\mathbf{s}_t,\mathbf{x}_t) - \sum_{i=1}^{t} h(\mathbf{x}_i))$. where $h(\mathbf{x}_t) = \left(1-\frac{M}{N}\right)\langle \mathbf{y}_t, \mathbb{I}_N\rangle$. Also we have $\sum_{i=1}^{t} K(\mathbf{s}_t,\mathbf{x}_t) \leq \sum_{i=1}^{t-1} K(\mathbf{s}_t,\mathbf{x}_t) + r_{\max}$ and $\sum_{i=1}^{t} K(s,\mathbf{x}_t) \geq \sum_{i=1}^{t-1} K(\mathbf{s}_t,\mathbf{x}_t))$ as the rate at time t is positive.

Equation 68 comes from the fact that the Gaussian PDF is upper bounded by $\frac{1}{\sqrt{2\pi}\sigma}$. Equation 69 comes from $\langle(\mathbf{s}-\mathbf{s}_t),\gamma\rangle$ being a sum of unit variance independent Gaussian random variables. Let $\sigma_{(s,\mathbf{s}_t)}$ be the standard deviation for a given $(\mathbf{s}_t,\mathbf{s})$ pair. The last step comes from choosing $\eta_t = \alpha\sqrt{t}$, and $\sigma_{(\mathbf{s},\mathbf{s}_t)} = 1$ which happens when $(\mathbf{s}_t,\mathbf{s})$ differ at exactly one file. In this section, we bound the number of switches that occur up to time T. □

D. Proof of Theorem 2

Theorem 2 The expected number of switches in cache configuration until time T for Algorithm 1 with unrestricted switching and input parameters $\eta = (\eta_t = \alpha\sqrt{t})_{t\in\mathcal{T}}$ is given by

$$C^{A}_{(\eta,\mathrm{UR})}(T) \leq \frac{3r_{\max}(|\mathcal{S}|-1)}{2\sqrt{\pi}\alpha}\sum_{t=1}^{T}\frac{1}{\sqrt{t}} = \mathcal{O}(\sqrt{T})$$

where $|\mathcal{S}|$ is the cardinality of the set of feasible cache configurations.

Proof. The term in the above expression can be bounded as

$$C^{A}_{(\eta,\mathrm{UR})}(T) = \mathbb{E}\left[\sum_{t=1}^{T-1}\mathbb{I}(\mathbf{s}_{t+1}\neq\mathbf{s}_t)\right] \tag{71}$$

$$= \sum_{t=1}^{T-1}\mathbb{E}\left[\mathbb{E}\left[\mathbb{I}(\mathbf{s}_{t+1}\neq\mathbf{s}_t)|(\mathbf{x}_1\cdots\mathbf{x}_{t-1},\gamma)\right]\right] \tag{72}$$

$$= \sum_{t=1}^{T-1}\mathbb{E}\left[\mathbb{P}\left((\mathbf{s}_{t+1}\neq\mathbf{s}_t)|(\mathbf{x}_1\cdots\mathbf{x}_{t-1},\gamma)\right)\right] \tag{73}$$

$$= \sum_{t=1}^{T-1}\sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}_t}\mathbb{E}\left[\mathbb{P}\left((\mathbf{s}_{t+1}=\mathbf{s})|(\mathbf{x}_1\cdots\mathbf{x}_{t-1},\gamma)\right)\right] \tag{74}$$

$$\leq \sum_{t=1}^{T-1}\sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}_t}\mathbb{E}\left[\frac{3r_{\max}}{2\alpha\sqrt{\pi}\sqrt{t+1}}\right] \tag{75}$$

$$= \frac{3r_{\max}(|\mathcal{S}|-1)}{2\sqrt{\pi}\alpha}\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \tag{76}$$

Here, equation 74 comes from the fact that for the policy (Algorithm 1) $\mathbf{s}_t$ can be determined given past observations $\mathbf{x}_t$ and the perturbation vector γ. Equation 75 comes from the lemma VII-C. □

E. Restricted Switching: Adversarial requests

In this scenario, the cache updates are allowed only at certain time steps. These time steps are indicated by inter-switch periods l_i and $\sum_{i=1}^{L} l_i = T$. The k^{th} cache update is allowed during the $\sum_{i=1}^{k-1} l_i$.

Theorem 3:

For an adversarial coded caching problem with N files, K users/caches, and cache size MF bits, let $R^{\mathcal{T}}_{(\eta,\mathrm{R})}(T)$ be the adversarial regret of Algorithm 1 under restricted switching with switching slots $\mathcal{T}$ and input parameters $\eta = (\eta_t = \alpha\sqrt{t})_{t\in\mathcal{T}}$. Then,

$$R^{\mathcal{T}}_{(\eta,\mathrm{R})}(T) \leq R_{(\eta,\mathrm{UR})}(T) + \sum_{k=1}^{L}\frac{3r^2_{\max}(|\mathcal{S}|-1)l_k(l_k-1)}{4\alpha\sqrt{\pi}\sqrt{\sum_{i=1}^{k-1}l_i+1}}.$$

Proof. The proof again follows from the earlier switching cost analysis, suggesting switches do not happen frequently. Note that the subset selection algorithm here is the same as the unrestricted case and will take the same actions as the unrestricted FTPL whenever it is allowed to take action/change its cache. The difference in the rate incurred in these two cases solely comes from the cache switches that take place in between these inter-switch periods. For the restricted switching case, we will enjoy the same rate starting from $\sum_{i=1}^{k-1} l_i$ as the FTPL for the unrestricted case until there is a switch at say time t in the unrestricted case, where $\sum_{i=1}^{k-1} l_i < t < \sum_{i=1}^{k} l_i$. Once a switch happens at instant t, there is no guarantee that the regret incurred from time t to $\sum_{i=1}^{k} l_i - 1$ is the same as that of the original policy.

Let R_k be the total additional regret(compared to the original FTPL algorithm) incurred during the k^{th} inter-switch period and $R^{\mathcal{T}}_{(\eta,\mathrm{R})}(T) = R^{\mathcal{T}}_{(\eta,\mathrm{UR})}(T) + \sum_{k=1}^{L} R_k$. If the switch happens for the first time after $\sum_{i=1}^{k-1} l_i$ at t we have the regret in R_k being upper bounded by $\left(\sum_{i=1}^{k} l_i - t\right) r_{\max}$. Using this, we have

$$R_k \leq \sum_{t=\sum_{i=1}^{k-1} l_i+1}^{\sum_{i=1}^{k} l_i-1} \mathbb{E}\left[\mathbb{I}(\mathrm{Switch}(t)\right] \left(\sum_{i=1}^{k} l_i - t\right) r_{\max} \tag{77}$$

$$\leq \sum_{t=\sum_{i=1}^{k-1} l_i+1}^{\sum_{i=1}^{k} l_i-1} \mathbb{E}\left[\mathbb{E}\left[\mathbb{I}(\mathbf{s}_t \neq \mathbf{s}_{t-1})|(\mathbf{x}_1 \cdots \mathbf{x}_{t-1}, \gamma)\right)\right] \left(\sum_{i=1}^{k} l_i - t\right) r_{\max} \tag{78}$$

$$\leq \sum_{t=\sum_{i=1}^{k-1} l_i+1}^{\sum_{i=1}^{k} l_i-1} \mathbb{E}\left[\sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}_{t-1}} \mathbb{P}\left((\mathbf{s}_t = \mathbf{s})|(\mathbf{x}_1 \cdots \mathbf{x}_{t-2}, \gamma)\right)\right] \left(\sum_{i=1}^{k} l_i - t\right) r_{\max} \tag{79}$$

$$\leq \sum_{t=\sum_{i=1}^{k-1} l_i+1}^{\sum_{i=1}^{k} l_i-1} \sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}_{t-1}} \frac{3r_{\max}}{2\alpha\sqrt{\pi}\sqrt{t}} \left(\sum_{i=1}^{k} l_i - t\right) r_{\max} \tag{80}$$

$$\leq \frac{3r_{\max}^2(|\mathcal{S}|-1)}{2\alpha\sqrt{\pi}\sqrt{\sum_{i=1}^{k-1} l_i+1}} \sum_{t=\sum_{i=1}^{k-1} l_i+1}^{\sum_{i=1}^{k} l_i-1} \left(\sum_{i=1}^{k} l_i - t\right) \tag{81}$$

$$\leq \frac{3r_{\max}^2(|\mathcal{S}|-1)}{2\alpha\sqrt{\pi}\sqrt{\sum_{i=1}^{k-1} l_i+1}} \frac{l_k(l_k-1)}{2} \tag{82}$$

Equation 79 come from lemma VII-C and equation 81 comes from $\sum_{i=1}^{k-1} l_i < t$ .Adding all these R_k terms gives the above result. □

F. Stochastic request setting:

In this setting each user i possesses an undisclosed underlying preference distribution across N files, denoted by $p_i = [p_i(1), p_i(2) \cdots p_i(N)]$. This distribution signifies the probability of user i requesting a specific file. This distribution remains the same across time slots and is known to the oracle. As a result of requests being generated from this distribution, $\mathbf{x}_t$ is a random vector $\forall t$. Let $\mathcal{X}$ be the distribution of this random vector over the set of all feasible $\mathbf{x}_t$ $(\{\mathbf{x}_t | \langle \mathbf{x}_t, \mathbb{I}_N \rangle = K, \mathbf{x}_t(i) \geq 0\rangle\})$ which can be obtained from the distribution of p_i, $\forall i$. As a result, the distribution $\mathcal{X}$ is known to the oracle. The expected total rate incurred by a policy π is denoted by $\mathcal{K}^{\pi}_{S}(T)$. The regret incurred by the policy π for the stochastic case after T time steps is given by

$$R^S_{\pi}(T) = \mathcal{K}^{\pi}(T) - T.K^S_o = \sum_{t=1}^{T} \mathbb{E}[K_{\pi}(t) - K^S_o]. \tag{83}$$

Where K^S_o is the expected rate incurred by the static stochastic oracle and is given by

$$K^S_o = \min_{\mathbf{s}\in\mathcal{S}} K(\mathbf{s}) \tag{84}$$

where

$$K(\mathbf{s}) = \mathbb{E}_{\mathbf{x}_1\sim\mathcal{X}}\left[K(\mathbf{s}, \mathbf{x}_1)\right]$$

Note that π here denotes the algorithm that chooses the subset $\mathbf{s}_t$ of files to be cached.

G. Results for Stochastic setting

Our first result outlines the regret incurred by the policy described in Algorithm 1 w.r.t the stochastic oracle defined in equation (83). The regret incurred in the stochastic case is upper-bounded by a constant, which is consistent with the results obtained in most online learning settings.

Theorem 4. *Let $\Delta_{\mathbf{s}} = K(\mathbf{s}) - K^S_o$. The regret incurred by the proposed online policy (Algorithm 1) and the placement delivery mechanism described above with a learning rate $\eta_t = \alpha\sqrt{t}$ is upper-bounded by*

$$R^S_{\eta}(T) \leq \sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*} \frac{64}{\Delta_{\mathbf{s}}} \left((r^C_{\max})^2 + K^2 + \beta\right) \tag{85}$$

Here $r^C_{\max}$ is the maximum possible length of a coded transmission over the set of all possible requests $\mathbf{x}_t$ and cache configurations $\mathbf{s}_t$, and $\mathbf{s}^* = \arg\min_{s\in\mathcal{S}} K(s)$ is the cache configuration used by the oracle. $\beta = \alpha^2 \max\left\{\frac{M^2}{N}, \frac{(N-M)^2}{N}\right\}$

Proof. The proof for this theorem relies on the idea that, given the users have a stationary preference distribution, the policy can learn the distribution of $f(\mathbf{x}_t, \mathbf{s})$ for all $\mathbf{s}$ and $\mathbf{y}_t$ over time after observing an adequate number of request samples. Consequently, it deviates from the Oracle policy with low probability, which is described formally via the lemma 1.

Let $K_t^{\eta_t}(\mathbf{s}) = \left\langle \left(\mathbf{s} - \frac{M}{N}\mathbb{I}\right), \sum_{i=1}^t f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_{t+1} + \eta_t\gamma \right\rangle$. Also, let $\mathcal{B}^t(\mathbf{s})$ be the event that $K_t^{\eta_t}(\mathbf{s}) \leq K_t^{\eta_t}(\mathbf{s}^*)$ (an event which will result in a choice of a wrong cache configuration). Note that the event that the policy chooses a cache configuration different from the oracle at time t is a subset of the event $\bigcup_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*} \mathcal{B}^t(\mathbf{s})$. Then we have

Lemma 1. *The probability of the event $\mathcal{B}^t(\mathbf{s})$ is upper bounded by*

$$\mathbb{P}(\mathcal{B}^t(\mathbf{s})) \leq 2\left(e^{-\frac{t\Delta_{\mathbf{s}}^2}{32(r_{\max}^C)^2}} + e^{-\frac{t\Delta_{\mathbf{s}}^2}{32(K)^2}} + e^{-\frac{t\Delta_{\mathbf{s}}^2}{32\beta}}\right) \tag{86}$$

Proof. We can upper bound the probability of this event as follows

$$\mathbb{P}(\mathcal{B}^t(\mathbf{s})) = \mathbb{P}(K_t^{\eta_t}(\mathbf{s}) \leq K_t^{\eta_t}(\mathbf{s}^*)) \tag{87}$$

$$= \mathbb{P}\left(\underbrace{\left\langle \left(\mathbf{s} - \frac{M}{N}\mathbb{I}_N\right), \sum_{i=1}^t f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_{t+1} + \eta_t\gamma \right\rangle}_{T_a} \leq \underbrace{\left\langle \left(\mathbf{s}^* - \frac{M}{N}\mathbb{I}_N\right), \sum_{i=1}^t f(\mathbf{x}_i, \mathbf{s}^*) - Y_{t+1} + \eta_t\gamma \right\rangle}_{T_b}\right) \tag{88}$$

$$\leq \mathbb{P}\left(\frac{T_b - tK(\mathbf{s}^*)}{t} \geq \frac{\Delta_{\mathbf{s}}}{2}\right) + \mathbb{P}\left(\frac{tK(\mathbf{s}) - T_a}{t} \geq \frac{\Delta_{\mathbf{s}}}{2}\right) \tag{89}$$

Note that in equation 88, the expected value of $\mathbb{E}[T_b] = tK(\mathbf{s}^*)$ and the expected value of $\mathbb{E}[T_a] = tK(\mathbf{s})$. Let $T_{aC} = \left\langle \left(\mathbf{s}^* - \frac{M}{N}\mathbb{I}_N\right), \sum_{i=1}^t f(\mathbf{x}_i, \mathbf{s}^*) \right\rangle$, $T_{aU} = \left\langle \left(\mathbf{s}^* - \frac{M}{N}\mathbb{I}_N\right), \mathbf{Y}_{t+1} \right\rangle$ and $T_{aG} = \left\langle \left(\mathbf{s}^* - \frac{M}{N}\mathbb{I}_N\right), \eta_t\gamma \right\rangle$, also $T_{aR} = T_{aC} + T_{aU}$, $T_a = T_{aC} + T_{aU} + T_{aG}$. Similarly, we define T_{bC}, T_{bU}, T_{bR} and T_{bG}. Now we have

$$\mathbb{P}\left(\frac{T_b - tK(\mathbf{s}^*)}{t} \geq \frac{\Delta_{\mathbf{s}}}{2}\right) \tag{90}$$

$$\leq \mathbb{P}\left(\frac{T_{bR} - t\mathbb{E}[T_{bR}]}{t} \geq \frac{\Delta_{\mathbf{s}}}{4}\right) + \mathbb{P}\left(\frac{T_{bG} - t\mathbb{E}[T_{bG}]}{t} \geq \frac{\Delta_{\mathbf{s}}}{4}\right) \tag{91}$$

$$\leq \mathbb{P}\left(\frac{T_{bC} - t\mathbb{E}[T_{bC}]}{t} \geq \frac{\Delta_{\mathbf{s}}}{8}\right) + \mathbb{P}\left(\frac{T_{bU} - t\mathbb{E}[T_{bU}]}{t} \geq \frac{\Delta_{\mathbf{s}}}{8}\right) + \mathbb{P}\left(T_{bG} \geq t\frac{\Delta_{\mathbf{s}}}{4}\right) \tag{92}$$

$$\leq e^{-\frac{t\Delta_{\mathbf{s}}^2}{32(r_{\max}^C)^2}} + e^{-\frac{t\Delta_{\mathbf{s}}^2}{32(K)^2}} + \mathbb{P}\left(T_{bG} \geq t\frac{\Delta_{\mathbf{s}}}{4}\right) \tag{93}$$

$$\leq e^{-\frac{t\Delta_{\mathbf{s}}^2}{32(r_{\max}^C)^2}} + e^{-\frac{t\Delta_{\mathbf{s}}^2}{32(K)^2}} + e^{-\frac{t\Delta_{\mathbf{s}}^2}{32\beta}} \tag{94}$$

Equation 93 comes from Hoeffding inequality and the fact $T_{bC} \in [0, r_{\max}^C]$ (Its the coded rate incurred when the cache configuration is $\mathbf{s}^*$) and $T_{bU} \in \left[-\frac{M}{N}K, \left(1 - \frac{M}{N}\right)K\right]$. Equation 94 comes from the upper deviation inequality for a Gaussian random variable $X \sim \mathcal{N}(\mu, \sigma^2)$ which says $\mathbb{P}[\mathbf{X} \geq \mu + t] \leq \exp(-t^2/2\sigma^2)$ and the max variance of T_{bG} is $\max\left\{\frac{M^2}{N}, \frac{(N-M)^2}{N}\right\}\eta_t^2 = \beta t$. ($\eta_t = \alpha\sqrt{t}$), . Similarly, one can upperbound the second term in equation 89 to get the desired result. □

Let $\Delta_{\mathbf{s}} = K(\mathbf{s}) - K_o^S$. Let $K_t^{\eta_t}(\mathbf{s}) = \left\langle \left(s - \frac{M}{N}\mathbb{I}_N\right), \sum_{i=1}^t f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_{t+1} + \eta_t\gamma \right\rangle$. We have

$$R_\eta^S(T) = \sum_{t=1}^T \mathbb{E}[K_\pi(t) - K_o^S(t)] = \sum_{t=1}^T \sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*} \mathbb{E}[(K_{\mathbf{s}}(t) - K_o^S(t))\mathbb{I}(\mathbf{s}_t = s)] \tag{95}$$

$$= \sum_{t=1}^T \sum_{s\in\mathcal{S}\setminus s^*} \Delta_{\mathbf{s}_t}\mathbb{P}(\mathbf{s}_t = \mathbf{s}) \tag{96}$$

$$\leq \sum_{t=1}^T \sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*} \Delta_{\mathbf{s}}\mathbb{P}(K_t^{\eta_t}(\mathbf{s}) \leq K_t^{\eta_t}(\mathbf{s}^*)) = \sum_{t=1}^T \sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*} \Delta_{\mathbf{s}}\mathbb{P}(\mathcal{B}^t(\mathbf{s})) \tag{97}$$

In equation 95 $K_{\mathbf{s}}(t) = \left\langle \left(\mathbf{s} - \frac{M}{N}\mathbb{I}_N\right), f(\mathbf{x}_t, \mathbf{s}) - \mathbf{y}_t \right\rangle + h(\mathbf{x}_t)$ and $K_o(t) = \left\langle \left(\mathbf{s}^* - \frac{M}{N}\mathbb{I}_N\right), f(\mathbf{x}_t, \mathbf{s}^*) - \mathbf{y}_t \right\rangle + h(\mathbf{x}_t)$. Note that the random variables $K_{\mathbf{s}}(t) - K_o^S(t)$ (expected value $\Delta_{\mathbf{s}}$) and $\mathbb{I}(\mathbf{s}_t = \mathbf{s})$ (expected value $\mathbb{P}(\mathbf{s}_t = \mathbf{s})$) are independent. Also, $\mathcal{B}^t(\mathbf{s})$ is the event that $K_t^{\eta_t}(\mathbf{s}) \leq K_t^{\eta_t}(\mathbf{s}^*)$ (an event which results in a choice of a wrong cache configuration). From the lemma below, we have

$$\sum_{t=1}^T \sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*} \Delta_{\mathbf{s}}\mathbb{P}(\mathcal{B}^t(\mathbf{s})) \tag{98}$$

$$\leq \sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*}\sum_{t=1}^{T} 2\Delta_{\mathbf{s}}\left(e^{-\frac{t\Delta_{\mathbf{s}}^2}{32(r^C_{\max})^2}}+e^{-\frac{t\Delta_{\mathbf{s}}^2}{32(K)^2}}+e^{-\frac{t\Delta_{\mathbf{s}}^2}{32\beta}}\right) \tag{99}$$

$$\leq \sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*}\sum_{t=1}^{\infty} 2\Delta_{\mathbf{s}}\left(e^{-\frac{t\Delta_{\mathbf{s}}^2}{32(r^C_{\max})^2}}+e^{-\frac{t\Delta_{\mathbf{s}}^2}{32(K)^2}}+e^{-\frac{t\Delta_{\mathbf{s}}^2}{32\beta}}\right) \tag{100}$$

$$\leq \sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*} 2\Delta_{\mathbf{s}}\left(\frac{\exp(-\frac{\Delta_{\mathbf{s}}^2}{32(r^C_{\max})^2})}{1-\exp(-\frac{\Delta_{\mathbf{s}}^2}{32(r^C_{\max})^2})}+\frac{\exp(-\frac{\Delta_{\mathbf{s}}^2}{32(K)^2})}{1-\exp(-\frac{\Delta_{\mathbf{s}}^2}{32(K)^2})}+\frac{\exp(-\frac{\Delta_{\mathbf{s}}^2}{32\beta})}{1-\exp(-\frac{\Delta_{\mathbf{s}}^2}{32\beta})}\right) \tag{101}$$

$$\leq \sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*} \frac{64}{\Delta_{\mathbf{s}}}\left((r^C_{\max})^2+K^2+\beta\right) \tag{102}$$

Last step follows since $e^{-x}/(1-e^{-x}) \leq \frac{1}{x}$ □

H. Switching Cost: Stochastic requests

Theorem 5. *The number of switches up to time T for the Algorithm 1 for the stochastic setting defined in equation 83 can be upper bounded as*

$$\sum_{t=1}^{T-1}\mathbb{E}[\mathbb{I}(\mathbf{s}_{t+1}\neq\mathbf{s}_t] \leq \sum_{s\in\mathcal{S}\setminus\mathbf{s}^*}\frac{64}{\Delta_{\mathbf{s}}^2}\left((r^C_{\max})^2+K^2+\beta\right) \tag{103}$$

Proof. The idea of bounding the number of switches in cache configuration until time t here is again based on the idea that eventually, with high probability, the policy will choose the same cache configurations as the oracle. Let the sequence of cache configurations cache upto time t be $\mathbf{s}_{\text{Seq}} = (\mathbf{s}_1, \mathbf{s}_2\cdots\mathbf{s}_T)$. Now consider the sequence $\mathbf{s}^M_{\text{Seq}} = (\mathbf{s}_1, \mathbf{s}^*, \mathbf{s}_2, \mathbf{s}^*, \cdots\mathbf{s}_T, \mathbf{s}^*)$. Now observe that the number of switches in the sequence $\mathbf{s}_{\text{Seq}}$ will be less than the number of switches in the sequence $\mathbf{s}^M_{\text{Seq}}$. One can further observe that the number of switches is the sequence $\mathbf{s}^M_{\text{Seq}}$ is less than $2\sum_{t=1}^{T}\mathbb{I}(\mathbf{s}_t\neq\mathbf{s}^*)$. Thus we have

$$\begin{aligned}
\sum_{t=1}^{T-1}\mathbb{E}[\mathbb{I}(\mathbf{s}_{t+1}\neq\mathbf{s}_t] \leq& \sum_{t=1}^{T-1}\left(\mathbb{E}[\mathbb{I}(\mathbf{s}_t\neq\mathbf{s}^*)]+\mathbb{E}[\mathbb{I}(\mathbf{s}^*\neq\mathbf{s}_{t+1})]\right) && (104)\\
\leq& 2\sum_{t=1}^{T-1}\mathbb{E}[\mathbb{I}(\mathbf{s}_t\neq\mathbf{s}^*)] && (105)\\
=& 2\sum_{t=1}^{T}\mathbb{P}(\mathbf{s}_t\neq\mathbf{s}^*) && (106)\\
\leq& 2\sum_{t=1}^{T}\mathbb{P}\left(\bigcup_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*}\mathcal{B}^t(\mathbf{s})\right) && (107)\\
\leq& 2\sum_{t=1}^{T}\sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*}\mathbb{P}\left(\mathcal{B}^t(\mathbf{s})\right) && (108)\\
\leq& 2\sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*}\sum_{t=1}^{\infty}\left(e^{-\frac{t\Delta_{s}^2}{32(r^C_{\max})^2}}+e^{-\frac{t\Delta_{\mathbf{s}}^2}{32(K)^2}}+e^{-\frac{t\Delta_{\mathbf{s}}^2}{32\beta}}\right) && (109)\\
\leq& 2\sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*}\left(\frac{\exp(-\frac{\Delta_{\mathbf{s}}^2}{32(r^C_{\max})^2})}{1-\exp(-\frac{\Delta_{\mathbf{s}}^2}{32(r^C_{\max})^2})}+\frac{\exp(-\frac{\Delta_{\mathbf{s}}^2}{32(K)^2})}{1-\exp(-\frac{\Delta_{\mathbf{s}}^2}{32(K)^2})}+\frac{\exp(-\frac{\Delta_{\mathbf{s}}^2}{32\beta})}{1-\exp(-\frac{\Delta_{\mathbf{s}}^2}{32\beta})}\right) && (110)\\
\leq& \sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*}\frac{64}{\Delta_{\mathbf{s}}^2}\left((r^C_{\max})^2+K^2+\beta\right) && (111)
\end{aligned}$$

Equation 107 comes from the fact that the event under which the policy chooses a cache configuration different from the oracle at time t as ($\{\mathbf{s}_t\neq\mathbf{s}^*\}$) is a subset of the event $\bigcup_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*}\mathcal{B}^t(\mathbf{s})$. Equation 109 comes from the lemma 1. Last step follows since $e^{-x}/(1-e^{-x}) \leq \frac{1}{x}$ □

I. Restricted Switching: Stochastic requests

Switching the cache configuration is only allowed in slots t_i for $i\in\{1,2\cdots L\}$ where $t_k=\sum_{i=1}^{k} l_i$ where l_i are the inter-switch periods. The policy (Algorithm 1) is used here whenever we have a switching slot. i.e., The cache placement step is done only during the switching slot, and the delivery step is performed in every slot. By convention, we allow switching after the last slot i.e., $\sum_{k=1}^{L} l_k = t_L = T$

Theorem 6. *Let $R^S_{(\eta,R)}(T)$ be the regret incurred by the policy (algorithm 1) in the stochastic restricted switching case with a horizon T. Then, the regret incurred by this policy in the restricted switching scenario with a learning rate $\alpha\sqrt{t}$ is upper-bounded as*

$$R^S_{(\eta,R)}(T) \leq r_{\max} l_1 + \sum_{k=1}^{L} \sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*} 2l_k\Delta_{\mathbf{s}} \left(e^{-\frac{t_k\Delta_{\mathbf{s}}^2}{32(r^C_{\max})^2}} + e^{-\frac{t_k\Delta_{\mathbf{s}}^2}{32(K)^2}} + e^{-\frac{l_k\Delta_{\mathbf{s}}^2}{32\beta}} \right) \tag{112}$$

Proof. If the algorithm chooses a cache configuration different from the oracle at time t_k then it incurs non-zero regret w.r.t. the oracle from time t_k to time $t_{k+1}-1$. Let R^S_k be the regret incurred by the oracle between t_k and $t_{k+1}-1$. Let $\mathbf{s} \neq \mathbf{s}^*$ be the cache configuration chosen at time t_i. Then we have

$$R^S_K = \sum_{t=t_k}^{t_{k+1}-1} \mathbb{E}[K_\pi(t) - K_o(t)] \tag{113}$$

$$= \sum_{t=t_k}^{t_{k+1}-1} \sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*} \mathbb{E}[(K_{\mathbf{s}}(t) - K_o(t))\mathbb{I}(\mathbf{s}_t = \mathbf{s})] \tag{114}$$

$$= l_k \sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*} \mathbb{E}[(K_{\mathbf{s}_{t_k}} - K_o(t_k))\mathbb{I}(\mathbf{s}_{t_k} = \mathbf{s})] \tag{115}$$

$$= l_k \sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*} \Delta_{\mathbf{s}_{t_k}} \mathbb{P}(\mathbf{s}_{t_k} = \mathbf{s}) \tag{116}$$

$$\leq l_k \sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*} \Delta_{\mathbf{s}} \mathbb{P}\left(K^{\eta_{t_k}}_{t_k}(s) \leq K^{\eta_{t_k}}_{t_k}(\mathbf{s}^*)\right) = l_k \sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*} \Delta_{\mathbf{s}} \mathbb{P}(\mathcal{B}^{t_k}(\mathbf{s})) \tag{117}$$

$$\leq \sum_{\mathbf{s}\in\mathcal{S}\setminus\mathbf{s}^*} 2l_k\Delta_{\mathbf{s}} \left(e^{-\frac{t_k\Delta_{\mathbf{s}}^2}{32(r^C_{\max})^2}} + e^{-\frac{t_k\Delta_{\mathbf{s}}^2}{32(K)^2}} + e^{-\frac{t_k\Delta_{\mathbf{s}}^2}{32\beta}} \right) \tag{118}$$

Equation 115 the cache configuration remains fixed for these slots. Equation 116 comes from the independence of the random variables within the expectation. The equation 118 comes from lemma 1. Upper bounding the regret before the first switching slot by $l_1 r_{\max}$, we have the required result. □