

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 the well-known coded caching problem in an online learning framework, wherein requests arrive sequentially, and an online policy can update the cache contents based on the history of requests seen thus far. We introduce a caching policy based on the Follow-The-Perturbed-Leader principle and show that for any time horizon T and any request sequence, it achieves a sub-linear regret of $\mathcal{O}(\sqrt{T})$ with respect to an oracle that knows the request sequence beforehand. Our study marks the first examination of adversarial regret in the coded caching setup. Furthermore, we also address the issue of switching cost by establishing an upper bound on the expected number of cache updates made by our algorithm under unrestricted switching and also provide an upper bound on the regret under restricted switching when cache updates can only happen in a pre-specified subset of timeslots. Finally, we validate our theoretical insights with numerical results using a real-world 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 static¹. 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.

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

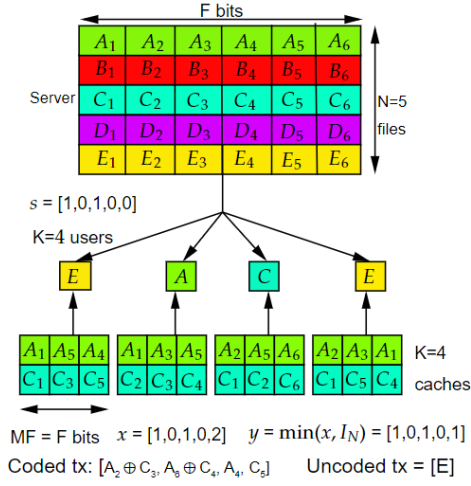


Fig. 1. An illustration of coded caching problem setup. It contains a central server with $N = 5$ files, each of size F bits, and $K = 4$ users, connected to a separate cache of size $MF = F$ bits. Here $s = [1, 0, 1, 0, 0]$ indicates that stored files are A and C , and unstored files are the remaining. For a request profile $\mathbf{r} = (E, A, C, E)$, its 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 analysis². We resolve this issue by rewriting the rate expression using a careful transformation of the request vector.

II. PROBLEM FORMULATION

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 $\mathbf{r}_t = (r_t^1, r_t^2, \dots, r_t^K)$ 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 $\mathbf{z}_t = (z_t^1, z_t^2, \dots, z_t^K)$, 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 .

²The 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.

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 policy³ 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 first briefly describe this class of policies and then give details of the placement and delivery phases of any such policy later. 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 according to 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 according to the delivery method proposed in [3]. Note that any scheme in this class is characterized by its choice of *stored files* in each slot. The formal and full description of the schemes' placement and delivery policies at each time slot is given below:

- 1) **Placement phase:** Let a binary vector $\mathbb{I}_N = [1, 1, \dots, 1]$ 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 the set \mathcal{S} denotes the collection 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 $(\frac{MF}{|\mathbf{s}_t| \cdot \mathbb{I}_N})$ 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. Let $\mathbf{x}_t = [x_t(1), x_t(2), \dots, x_t(N)]$ be an N -dimensional vector where $x_t(i)$ denotes the number of users request-

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

ing 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 $|\bigoplus_{k \in u} V_{k, u \setminus \{k\}}| = \max_{k \in u} |V_{k, u \setminus \{k\}}|$.

Note that by using the above coded and uncoded transmissions, every user can recover their requested files, See [3, Theorem 1 proof] for more details.

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. Derivation of expected coded transmission length provided in Appendix of the full paper [34]. \square

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 $\mathbf{r}_t = (r_t^1, r_t^2, \dots, r_t^K)$ is not necessary to compute $K(\mathbf{s}_t, \mathbf{x}_t)$.

⁴Note 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.

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 which knows the entire request pattern profile 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 oracle performance is given by

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

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

$$R_\pi((\mathbf{x}_t)_{t=1}^T, T) := K^\pi(T) - K^o(T). \quad (5)$$

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

$$R_\pi(T) = \sup_{\mathbf{x}_t \in X, \forall t} R_\pi((\mathbf{x}_t)_{t=1}^T, T), \quad (6)$$

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 $C_\pi(T)$ denotes the expected number of cache

configuration switches until time T for an Algorithm π , then $C_\pi(T)$ is given by

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

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. In Section III, we discuss our online subset selection algorithm. 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 spatial limitations, the proofs of our theorems and proposition are provided in the Appendix of the complete version of our paper [34].

III. OUR PROPOSED ALGORITHM

Algorithm 1 Algorithm for Coded Caching Problem with Adversarial Requests

```

1: Input:  $M, N, T, s_0 = [1, 1 \dots 1], \mathcal{T}, \eta_t = \alpha\sqrt{t} \quad \forall t \in \mathcal{T}$ 
2: Sample  $\gamma \sim \mathcal{N}(0, I_{N \times N})$ 
3: for  $t \leq T$  do
4:   Placement phase:
5:   Derive  $\mathbf{x}_{t-1}$  from  $\mathbf{r}_{t-1}$ 
6:    $\mathbf{y}_{t-1} \leftarrow \min\{\mathbf{x}_{t-1}, \mathbb{I}_N\}$ 
7:    $\mathbf{Y}_t \leftarrow \mathbf{Y}_{t-1} + \mathbf{y}_{t-1}$ 
8:   if  $t \in \mathcal{T}$  then
9:      $\bar{\mathbf{Y}}_t = \mathbf{Y}_t - \eta_t \gamma$ 
10:    Subset selection for placement:
11:     $\mathbf{s}_t \leftarrow \arg \min_{\mathbf{s} \in \mathcal{S}} \left\langle \left( \mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^{t-1} f(\mathbf{x}_i, \mathbf{s}) - \bar{\mathbf{Y}}_t \right\rangle$ 
12:    Perform cache placement with  $\mathbf{s}_t$  according to the placement phase of Section II
13:  end if
14:  Delivery phase:
15:  Receive  $\mathbf{r}_t$ 
16:  Perform content delivery according to 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 find the vectors \mathbf{x}_{t-1} , $\mathbf{y}_{t-1} = \min\{\mathbf{x}_{t-1}, \mathbb{I}_N\}$ and $\mathbf{Y}_t = \mathbf{Y}_{t-1} + \mathbf{y}_{t-1}$, based on the request profile \mathbf{r}_{t-1} received in the previous slot $t-1$. (See lines 5 - 7 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 $\bar{\mathbf{Y}}_t = \mathbf{Y}_t - \eta_t \gamma$ as input.

- After the subset selection, cache placement occurs according to the placement phase described in Section II. (See lines 8-13 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 [29], [30], the FTPL policy has proven to achieve $\mathcal{O}(\sqrt{T})$ regret in adversarial settings. In a single-user 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 is given by $\sum_{i=1}^{t-1} K(\mathbf{s}, \mathbf{x}_i) = \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^{t-1} f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_t \right\rangle + \sum_{i=1}^{t-1} h(\mathbf{x}_i)$, and it 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.

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., $\mathcal{T} = \{1, 2 \dots T\}$, 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 with unrestricted switching. Then,*

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

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

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

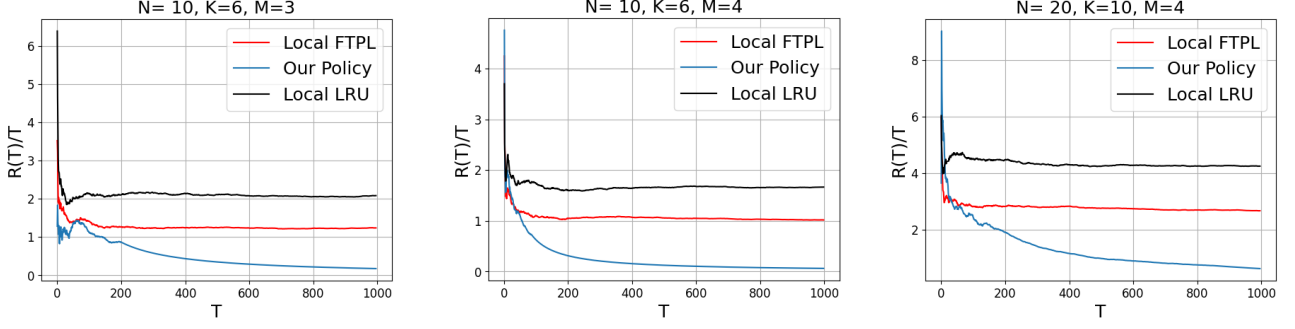


Fig. 2. We compare the performance of our proposed policy in Algorithm 1 against the benchmark policies *Local* caching with FTPL and *Local* LRU under different system parameters. The plots display the average regret per time step $R(T)/T$ against horizon T . The first figure compares the performance 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 $M = 4$. For all the above system parameters, our 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.

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. 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 1. From Theorems 1 and 2, we conclude that Algorithm 1 with unrestricted switching achieves $\mathcal{O}(\sqrt{T})$ adversarial regret by just 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 2. 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 $\mathcal{O}(l\sqrt{T})$.

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 ~ 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 potential avenues for future work, starting from deriving universal lower bounds on Regret. Another interesting problem is proposing algorithms for the case where switching budgets are fixed. Unlike our restricted switching case with given switching slots, fixed switching budgets would require the policy to decide when to switch the cache configuration within budget constraints. One more interesting problem is addressing the issue of online learning with expert advice within the context of coded caching.

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 problem with adversarial requests. [Online]. Available: <https://github.com/hunchings/CodedCaching-fullpaper>
- [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. Proof of Proposition 1

Proof. Recall from Section II that we perform both coded and uncoded transmissions during the delivery phase. Let $R_{UC}(\mathbf{s}_t, \mathbf{x}_t)$ and $R_C(\mathbf{s}_t, \mathbf{x}_t)$ denote the expected rate of the uncoded and coded transmissions associated with a cache configuration \mathbf{s}_t and the request pattern \mathbf{x}_t , respectively. Therefore, $K(\mathbf{s}_t, \mathbf{x}_t) = R_{UC}(\mathbf{s}_t, \mathbf{x}_t) + R_C(\mathbf{s}_t, \mathbf{x}_t)$. One can calculate $R_{UC}(\mathbf{s}_t, \mathbf{x}_t)$ and $R_C(\mathbf{s}_t, \mathbf{x}_t)$ separately as follows:

R_{UC} calculation: If an *unstored file* is requested by at least one of the users, then it is directly broadcast by the server. Therefore, the expected uncoded transmission rate $R_{UC}(\mathbf{s}_t, \mathbf{x}_t)$ is given by

$$R_{UC}(\mathbf{s}_t, \mathbf{x}_t) = \langle (\mathbb{I}_N - \mathbf{s}_t), \min\{\mathbf{x}_t, \mathbb{I}_N\} \rangle = \langle (\mathbb{I}_N - \mathbf{s}_t), \mathbf{y}_t \rangle. \quad (7)$$

R_C calculation: Let U_1 denote the set of users that request files from the cached set \mathbf{s}_t , then $|U_1| = \langle \mathbf{x}_t, \mathbf{s}_t \rangle$. Recall from Section II that, during the coded transmission, 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 $U_1 \setminus u$. Therefore, the expected length of the coded transmission $R_C(\mathbf{s}_t, \mathbf{x}_t)$ 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 $|\bigoplus_{k \in u} V_{k, u \setminus \{k\}}| = \max_{k \in u} |V_{k, u \setminus \{k\}}|$.

Since the users independently select $\frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N \rangle}$ fraction of each *stored file*, the probability of a particular bit of a *stored file* being in the cache of a particular user is $\frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N \rangle}$. Therefore, for large enough F , we have

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

Now for each $|u| \in \{1, 2, \dots, \langle \mathbf{x}_t, \mathbf{s}_t \rangle\}$, we have $\binom{\langle \mathbf{x}_t, \mathbf{s}_t \rangle}{|u|}$ subsets of size $|u|$. Therefore, the coded transmission message size $R_C(\mathbf{s}_t, \mathbf{x}_t) \cdot F$ is

$$\begin{aligned} R_C(\mathbf{s}_t, \mathbf{x}_t) \cdot F &= F \sum_{|u|=1}^{\langle \mathbf{x}_t, \mathbf{s}_t \rangle} \binom{\langle \mathbf{x}_t, \mathbf{s}_t \rangle}{|u|} \left(\frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N \rangle} \right)^{|u|-1} \left(1 - \frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N \rangle} \right)^{\langle \mathbf{x}_t, \mathbf{s}_t \rangle - |u| + 1} \\ &= F \left(\frac{\langle \mathbf{s}_t, \mathbb{I}_N \rangle}{M} - 1 \right) \left(1 - \left(1 - \frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N \rangle} \right)^{\langle \mathbf{x}_t, \mathbf{s}_t \rangle} \right). \end{aligned} \quad (8)$$

From (7) and (8), we have

$$K(\mathbf{s}_t, \mathbf{x}_t) = R_{UC}(\mathbf{s}_t, \mathbf{x}_t) + R_C(\mathbf{s}_t, \mathbf{x}_t) = \langle (\mathbb{I}_N - \mathbf{s}_t), \mathbf{y}_t \rangle + \left(\frac{\langle \mathbf{s}_t, \mathbb{I}_N \rangle}{M} - 1 \right) \left(1 - \left(1 - \frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N \rangle} \right)^{\langle \mathbf{x}_t, \mathbf{s}_t \rangle} \right) \quad (9)$$

□

B. Proof of Theorem 1

We use the following lemmas to prove Theorem 1.

Lemma 1. For a coded caching problem with the given cache configuration \mathbf{s}_t and the request vector \mathbf{x}_t , the placement and delivery policies discussed in Section II transmits a message of expected size $K(\mathbf{s}_t, \mathbf{x}_t)$ given by

$$K(\mathbf{s}_t, \mathbf{x}_t) = \left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_t) - \mathbf{y}_t \right\rangle + h(\mathbf{x}_t), \quad (10)$$

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 $h(\mathbf{x}_t) = \left(1 - \frac{M}{N} \right) \langle \mathbf{y}_t, \mathbb{I}_N \rangle$.

Proof. Using the fact that $\langle \mathbb{I}_N, \mathbb{I}_N \rangle = N$, we have

$$\begin{aligned} R_C(\mathbf{s}_t, \mathbf{x}_t) &= \left(\frac{\langle \mathbf{s}_t, \mathbb{I}_N \rangle}{M} - 1 \right) \left(1 - \left(1 - \frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N \rangle} \right)^{\langle \mathbf{x}_t, \mathbf{s}_t \rangle} \right) \\ &= \left\langle \mathbf{s}_t, \frac{1}{M} \left(1 - \left(1 - \frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N \rangle} \right)^{\langle \mathbf{x}_t, \mathbf{s}_t \rangle} \right) \mathbb{I}_N \right\rangle - \left\langle \frac{M}{N} \mathbb{I}_N, \frac{1}{M} \left(1 - \left(1 - \frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N \rangle} \right)^{\langle \mathbf{x}_t, \mathbf{s}_t \rangle} \right) \mathbb{I}_N \right\rangle \\ &= \left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_t) \right\rangle \end{aligned}$$

and

$$R_{UC}(\mathbf{s}_t, \mathbf{x}_t) = \langle (\mathbb{I}_N - \mathbf{s}_t), \mathbf{y}_t \rangle = \left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), -\mathbf{y}_t \right\rangle + \left\langle \left(1 - \frac{M}{N} \right) \mathbb{I}_N, \mathbf{y}_t \right\rangle.$$

Therefore,

$$K(\mathbf{s}_t, \mathbf{x}_t) = R_C(\mathbf{s}_t, \mathbf{x}_t) + R_{UC}(\mathbf{s}_t, \mathbf{x}_t) = \left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_t) - \mathbf{y}_t \right\rangle + h(\mathbf{x}_t).$$

□

Let the constant r_{\max}^C denotes the maximum size of the coded transmission message over the set of all request pattern and cache configuration pairs, i.e.,

$$r_{\max}^C = \max_{\substack{\mathbf{x} \in \mathcal{X} \\ \mathbf{s} \in \mathcal{S}}} R_C(\mathbf{s}, \mathbf{x}),$$

and the constant r_{\max} denotes the maximum size of the server's transmission message over the set of all request pattern and cache configuration pairs, i.e.,

$$r_{\max} = \max_{\substack{\mathbf{x} \in \mathcal{X} \\ \mathbf{s} \in \mathcal{S}}} K(\mathbf{s}, \mathbf{x}).$$

Note that both r_{\max} and r_{\max}^C can be upper bounded by K since the maximum number of unique requests in a slot is K . Let $\mathcal{G}_{\max}(\gamma)$ be the Gaussian width of the set \mathcal{S} a non-negative constant defined as

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

where, $\gamma = [\gamma_1, \gamma_2, \dots, \gamma_N]^T \sim \mathcal{N}(0, I_{N \times N})$. Note that $\mathcal{G}_{\max}(\gamma)$ can be easily upper bounded as $\mathcal{G}_{\max}(\gamma) = \mathbb{E}_{\gamma} \left[\max_{a \in \mathcal{S}} \langle a, \gamma \rangle \right] \leq \mathbb{E}_{\gamma} \left[\max_{a \in \{0,1\}^N} \langle a, \gamma \rangle \right] = \sum_{i=1}^N \gamma_i \mathbb{I}(\gamma_i \geq 0) = \frac{N}{\sqrt{2\pi}}$.

Lemma 2. For a coded caching problem with N files, K users/caches, and cache size MF bits, let $R_{UR}(T)$ be the regret under adversarial requests incurred by Algorithm 1 under unrestricted switching with $\eta_t = \alpha\sqrt{t}$. Then,

$$R_{UR}(T) \leq \left(\frac{3r_{\max}r_{\max}^C(|\mathcal{S}| - 1)}{2\sqrt{\pi}\alpha} + \frac{\max\left\{\frac{M}{N}, \left(1 - \frac{M}{N}\right)\right\} K^2}{\sqrt{2\pi}\alpha} \right) \sum_{t=1}^T \frac{1}{\sqrt{t}} + \alpha\sqrt{T}\mathcal{G}_{\max}(\gamma) + \eta_1\mathcal{G}_{\max}(\gamma) + r_{\max}^C$$

Proof. From Proposition 1, we know that, for a given cache configuration \mathbf{s}_t and a request pattern \mathbf{x}_t , the expected transmission size $K(\mathbf{s}_t, \mathbf{x}_t)$ is

$$K(\mathbf{s}_t, \mathbf{x}_t) = R_{UC} + R_{UC} = \langle (\mathbb{I}_N - \mathbf{s}_t), \mathbf{y}_t \rangle + \left(\frac{\langle \mathbf{s}_t, \mathbb{I}_N \rangle}{M} - 1 \right) \left(1 - \left(1 - \frac{M}{\langle \mathbf{s}_t, \mathbb{I}_N \rangle} \right)^{\langle \mathbf{x}_t, \mathbf{s}_t \rangle} \right).$$

From Lemma 1, $K(\mathbf{s}_t, \mathbf{x}_t)$ can be rewritten as

$$K(\mathbf{s}_t, \mathbf{x}_t) = \left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), (f(\mathbf{x}_t, \mathbf{s}_t) - \mathbf{y}_t) \right\rangle + h(\mathbf{x}_t), \quad (11)$$

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 $h(\mathbf{x}_t) = \left(1 - \frac{M}{N} \right) \langle \mathbf{y}_t, \mathbb{I}_N \rangle$.

For a request pattern $(\mathbf{x}_t)_{t=1}^T$, let $R_{UR}((\mathbf{x}_t)_{t=1}^T, T)$ denote the regret incurred by Algorithm 1. From equations (5), we have

$$R_{UR}((\mathbf{x}_t)_{t=1}^T, T) = K^\pi(T) - K^o(T) \quad (12)$$

$$\begin{aligned} &= \sum_{t=1}^T \mathbb{E}[K(\mathbf{s}_t, \mathbf{x}_t)] - \min_{\mathbf{s} \in \mathcal{S}} \sum_{t=1}^T \mathbb{E}[K(\mathbf{s}, \mathbf{x}_t)] \quad (\text{from equations (3) and (4)}) \\ &= \underbrace{\sum_{t=1}^T \mathbb{E}_{\gamma} \left[\left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), (f(\mathbf{x}_t, \mathbf{s}_t) - \mathbf{y}_t) \right\rangle + h(\mathbf{x}_t) \right]}_{\text{Policy Cumulative Rate}} - \underbrace{\min_{\mathbf{s} \in \mathcal{S}} \sum_{t=1}^T \left[\left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), (f(\mathbf{x}_t, \mathbf{s}) - \mathbf{y}_t) \right\rangle + h(\mathbf{x}_t) \right]}_{\text{Oracle Cumulative Rate}} \\ &= \sum_{t=1}^T \mathbb{E}_{\gamma} \left[\left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), (f(\mathbf{x}_t, \mathbf{s}_t) - \mathbf{y}_t) \right\rangle \right] - \min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{t=1}^T (f(\mathbf{x}_t, \mathbf{s}) - \mathbf{y}_t) \right\rangle \quad (13) \end{aligned}$$

The last equality comes from canceling $h(\mathbf{x}_t)$ terms on both sides, which are independent of the cache configurations. Then we define the potential function at time t taking two independent inputs \mathbf{W} and \mathbf{Z} as follows

$$\phi_t(\mathbf{W}, \mathbf{Z}) = \mathbb{E}_{\gamma} \left[\min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^{t-1} f(\mathbf{w}_i, \mathbf{s}) - \mathbf{Z} + \eta_t \gamma \right\rangle \right] \quad (14)$$

Note that the inputs must be of the form $\mathbf{W} = (\mathbf{w}_i)_{i=1}^{t-1}$ where $\mathbf{w}_i \in \mathbb{R}^N$ and $\mathbf{Z} \in \mathbb{R}^N$. Now let $X_t = (\mathbf{x}_i)_{i=1}^{t-1}$, all request patterns observed until time t and $\mathbf{Y}_t = \sum_{i=1}^{t-1} \mathbf{y}_t$. We have the partial derivative (similar to [23] Equation 5, [22])

$$\nabla_{\mathbf{Y}} \phi_t(\mathbf{X}_t, \mathbf{Y}_t) = -\mathbb{E}_{\gamma} \left[\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right] \quad (15)$$

Therefore, the term $\mathbb{E}_{\gamma} [\langle (\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N), (f(\mathbf{x}_t, \mathbf{s}_t) - \mathbf{y}_t) \rangle]$ in (13) can be written as

$$\begin{aligned} \mathbb{E}_{\gamma} \left[\left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), (f(\mathbf{x}_t, \mathbf{s}_t) - \mathbf{y}_t) \right\rangle \right] &= \mathbb{E}_{\gamma} \left[\left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_t) - (\mathbf{Y}_{t+1} - \mathbf{Y}_t) \right\rangle \right] \\ &= \mathbb{E}_{\gamma} \left[\left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_t) \right\rangle \right] - \mathbb{E}_{\gamma} \left[\left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), (\mathbf{Y}_{t+1} - \mathbf{Y}_t) \right\rangle \right]. \end{aligned} \quad (16)$$

Using the Taylor series expansion and for some $\tilde{\mathbf{Y}}_t = \mathbf{Y}_t + \theta \mathbf{y}_t$ for some $\theta \in [0, 1]$, we have

$$\begin{aligned} -\mathbb{E}_{\gamma} \left[\left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), (\mathbf{Y}_{t+1} - \mathbf{Y}_t) \right\rangle \right] &= \langle \nabla_{\mathbf{Y}} \phi_t(\mathbf{X}_t, \mathbf{Y}_t), (\mathbf{Y}_{t+1} - \mathbf{Y}_t) \rangle \\ &= \phi_t(\mathbf{X}_t, \mathbf{Y}_{t+1}) - \phi_t(\mathbf{X}_t, \mathbf{Y}_t) - \frac{1}{2} \langle \mathbf{y}_t, \nabla_{\mathbf{Y}}^2 \phi_t(\mathbf{X}_t, \tilde{\mathbf{Y}}_t) \mathbf{y}_t \rangle \end{aligned} \quad (17)$$

Therefore,

$$\begin{aligned} &\sum_{t=1}^T \mathbb{E}_{\gamma} \left[\left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), (f(\mathbf{x}_t, \mathbf{s}_t) - \mathbf{y}_t) \right\rangle \right] \\ &= \sum_{t=1}^T \mathbb{E}_{\gamma} \left[\left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_t) \right\rangle \right] - \sum_{t=1}^T \mathbb{E}_{\gamma} \left[\left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), (\mathbf{Y}_{t+1} - \mathbf{Y}_t) \right\rangle \right] \\ &= \sum_{t=1}^T \mathbb{E}_{\gamma} \left[\left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_t) \right\rangle \right] + \underbrace{\sum_{t=1}^T [\phi_t(\mathbf{X}_t, \mathbf{Y}_{t+1}) - \phi_t(\mathbf{X}_t, \mathbf{Y}_t)]}_{T_1} - \underbrace{\sum_{t=1}^T \frac{1}{2} \langle \mathbf{y}_t, \nabla_{\mathbf{Y}}^2 \phi_t(\mathbf{X}_t, \tilde{\mathbf{Y}}_t) \mathbf{y}_t \rangle}_{T_2}, \end{aligned} \quad (18)$$

In the above equation, the last equality comes from equation (17). Now, we can simplify the term T_1 as follows

$$T_1 := \sum_{t=1}^T [\phi_t(\mathbf{X}_t, \mathbf{Y}_{t+1}) - \phi_t(\mathbf{X}_t, \mathbf{Y}_t)] = \phi_T(\mathbf{X}_T, \mathbf{Y}_{T+1}) + \underbrace{\sum_{t=1}^{T-1} [\phi_t(\mathbf{X}_t, \mathbf{Y}_{t+1}) - \phi_{t+1}(\mathbf{X}_{t+1}, \mathbf{Y}_{t+1})]}_{T_3} - \phi_1(\mathbf{X}_1, \mathbf{Y}_1). \quad (19)$$

Consider the term $\phi_T(\mathbf{X}_T, \mathbf{Y}_{T+1})$. We have

$$\begin{aligned} \phi_T(\mathbf{X}_T, \mathbf{Y}_{T+1}) &= \mathbb{E}_{\gamma} \left[\min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^{T-1} f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_{T+1} + \eta_T \gamma \right\rangle \right] \\ &\leq \mathbb{E}_{\gamma} \left[\min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^T f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_{T+1} + \eta_T \gamma \right\rangle - \underbrace{\min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_T, \mathbf{s}) \right\rangle}_{\leq 0} \right] \end{aligned} \quad (20)$$

$$\begin{aligned} &\leq \mathbb{E}_{\gamma} \left[\min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^T f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_{T+1} + \eta_T \gamma \right\rangle \right] \\ &\leq \min_{\mathbf{s} \in \mathcal{S}} \mathbb{E}_{\gamma} \left[\left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^T f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_{T+1} + \eta_T \gamma \right\rangle \right] \end{aligned} \quad (21)$$

$$= \min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^T f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_{T+1} \right\rangle \quad (22)$$

$$= \min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^T (f(\mathbf{x}_i, \mathbf{s}) - \mathbf{y}_i) \right\rangle \quad (23)$$

Here, equation (20) comes from $\min_x(f_1(x)) + \min_x(f_2(x)) \leq \min_x(f_1(x) + f_2(x))$ and the second term in equation (20) is less than or equal to zero, since the term $\langle (\mathbf{s} - \frac{M}{N} \mathbb{I}_N), f(\mathbf{x}_T, \mathbf{s}) \rangle$ is the coded message size $R_C(\mathbf{s}, x_T)$ and is greater than or equal to 0. Equation (21) follows from Jensen's inequality and equation (23) comes from the fact that $\mathbb{E}_{\gamma}[\gamma] = 0$.

Now, consider the term T_3 in equation(19). We have

$$\begin{aligned} \phi_t(\mathbf{X}_t, \mathbf{Y}_{t+1}) - \phi_{t+1}(\mathbf{X}_{t+1}, \mathbf{Y}_{t+1}) &= \mathbb{E}_\gamma \left[\min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^{t-1} f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_{t+1} + \eta_t \gamma \right\rangle \right. \\ &\quad \left. - \min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^t f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_{t+1} + \eta_{t+1} \gamma \right\rangle \right] \\ &= \mathbb{E}_\gamma \left[\min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^{t-1} f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_{t+1} + \eta_t \gamma \right\rangle \right. \\ &\quad \left. - \left\langle \left(\mathbf{s}_{t+1} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^t f(\mathbf{x}_i, \mathbf{s}_{t+1}) - \mathbf{Y}_{t+1} + \eta_{t+1} \gamma \right\rangle \right] \quad (24) \end{aligned}$$

$$\begin{aligned} &\leq -\mathbb{E}_\gamma \left[\min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), (\eta_{t+1} - \eta_t) \gamma \right\rangle \right] \\ &\quad - \mathbb{E}_\gamma \left[\left\langle \left(\mathbf{s}_{t+1} - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_{t+1}) \right\rangle \right] \quad (25) \end{aligned}$$

$$= -\mathbb{E}_\gamma \left[\min_{\mathbf{s} \in \mathcal{S}} \langle \mathbf{s}, (\eta_{t+1} - \eta_t) \gamma \rangle \right] - \mathbb{E}_\gamma \left[\left\langle \left(\mathbf{s}_{t+1} - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_{t+1}) \right\rangle \right] \quad (26)$$

$$\leq |\eta_{t+1} - \eta_t| \mathcal{G}_{\max}(\gamma) - \mathbb{E}_\gamma \left[\left\langle \left(\mathbf{s}_{t+1} - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_{t+1}) \right\rangle \right] \quad (27)$$

Here,

- Equation (24) comes from the fact that, according to Algorithm 1,

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

- Equation (25) comes from the following steps:

$$\begin{aligned} &-\mathbb{E}_\gamma \left[\left\langle \left(\mathbf{s}_{t+1} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^t f(\mathbf{x}_i, \mathbf{s}_{t+1}) - \mathbf{Y}_{t+1} + \eta_{t+1} \gamma \right\rangle \right] \\ &= -\mathbb{E}_\gamma \left[\left\langle \left(\mathbf{s}_{t+1} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^{t-1} f(\mathbf{x}_i, \mathbf{s}_{t+1}) - \mathbf{Y}_{t+1} + \eta_{t+1} \gamma \right\rangle \right] - \mathbb{E}_\gamma \left[\left\langle \left(\mathbf{s}_{t+1} - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_{t+1}) \right\rangle \right] \\ &\leq -\mathbb{E}_\gamma \left[\min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^{t-1} f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_{t+1} + \eta_{t+1} \gamma \right\rangle \right] - \mathbb{E}_\gamma \left[\left\langle \left(\mathbf{s}_{t+1} - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_{t+1}) \right\rangle \right] \\ &\leq -\mathbb{E}_\gamma \left[\min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^{t-1} f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_{t+1} + \eta_t \gamma \right\rangle \right] - \mathbb{E}_\gamma \left[\left\langle \left(\mathbf{s}_{t+1} - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_{t+1}) \right\rangle \right] - \\ &\quad \mathbb{E}_\gamma \left[\min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), (\eta_{t+1} - \eta_t) \gamma \right\rangle \right] \quad (28) \end{aligned}$$

- Here, Equation (28) follows from the fact $\min_x(f_1(x)) + \min_x(f_2(x)) \leq \min_x(f_1(x) + f_2(x))$.
- Equation (26) comes from the fact that $\mathbb{E}[\gamma] = 0$.

By substituting (23) and (27) in (19), we get

$$\begin{aligned} T_1 &= \phi_T(\mathbf{X}_T, Y_{T+1}) + \underbrace{\sum_{t=1}^{T-1} [\phi_t(\mathbf{X}_t, Y_{t+1}) - \phi_{t+1}(\mathbf{X}_{t+1}, Y_{t+1})]}_{T_3} - \phi_1(\mathbf{X}_1, Y_1) \\ &\leq \phi_T(\mathbf{X}_T, Y_{T+1}) + \underbrace{\sum_{t=1}^{T-1} \left[|\eta_{t+1} - \eta_t| \mathcal{G}_{\max}(\gamma) - \mathbb{E}_\gamma \left[\left\langle \left(\mathbf{s}_{t+1} - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_{t+1}) \right\rangle \right] \right]}_{\geq T_3} - \phi_1(\mathbf{X}_1, Y_1) \\ &\leq \min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^T f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_{T+1} \right\rangle + |\eta_T - \eta_1| \mathcal{G}_{\max}(\gamma) - \\ &\quad \sum_{t=1}^{T-1} \mathbb{E}_\gamma \left[\left\langle \left(\mathbf{s}_{t+1} - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_{t+1}) \right\rangle \right] \underbrace{- \phi_1(\mathbf{X}_1, Y_1)}_{\eta_1 \mathcal{G}_{\max}(\gamma)} \quad (29) \end{aligned}$$

$$\leq \min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^T f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_{T+1} \right\rangle + \eta_T \mathcal{G}_{\max}(\gamma) - \sum_{t=1}^{T-1} \mathbb{E}_\gamma \left[\left\langle \mathbf{s}_{t+1} - \frac{M}{N} \mathbb{I}_N, f(\mathbf{x}_t, \mathbf{s}_{t+1}) \right\rangle \right] + \eta_1 \mathcal{G}_{\max}(\gamma) \quad (30)$$

Note that $\mathbf{X}_1 = \mathbf{Y}_1 = \hat{\mathbf{0}}$ thus in equation 29 the term $-\phi_1(\mathbf{X}_1, \mathbf{Y}_1) = -\mathbb{E}_\gamma \left[\min_{\mathbf{s} \in \mathcal{S}} \langle \mathbf{s} - \frac{M}{N} \mathbb{I}_N, \eta_1 \gamma \rangle \right] = -\mathbb{E}_\gamma \left[\min_{\mathbf{s} \in \mathcal{S}} \langle \mathbf{s}, \eta_1 \gamma \rangle \right] = \eta_1 \mathcal{G}_{\max}(\gamma)$.

For an N -dimensional vector $\mathbf{a} \in \mathbb{R}$ let $(\mathbf{a})_i \in \mathbb{R}$ denote its i^{th} entry. Lastly, T_2 can be bounded as

$$-\langle \mathbf{y}_t, \nabla^2 \phi_t(\mathbf{X}_t, \tilde{\mathbf{Y}}_t) \mathbf{y}_t \rangle \leq \left| \sum_{i=1}^N \sum_{j=1}^N (\mathbf{y}_t)_i (\mathbf{y}_t)_j |\nabla^2 \phi_t(\mathbf{X}_t, \tilde{\mathbf{Y}}_t)|_{ij} \right| \quad (31)$$

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

$$\nabla^2 \phi_t(\mathbf{X}_t, \tilde{\mathbf{Y}}_t)_{ij} = \frac{1}{\eta_t} \mathbb{E}_\gamma [\nabla \hat{\phi}_t(\mathbf{X}_t, \tilde{\mathbf{Y}}_t - \eta_t \gamma)_i \gamma_j] \quad (32)$$

where

$$\hat{\phi}_t(\mathbf{X}, \mathbf{Y}) = \min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^{t-1} f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y} \right\rangle \quad (33)$$

Thus $|\nabla^2 \phi_t(\mathbf{X}_t, \tilde{\mathbf{Y}}_t)|_{ij} \leq \frac{1}{\eta_t} \mathbb{E}_\gamma [|\nabla \hat{\phi}_t(\mathbf{X}_t, \tilde{\mathbf{Y}}_t - \eta_t \gamma)_i| |\gamma_j|] \leq \frac{1}{\eta_t} \mathbb{E}_\gamma \left[\left| \left(\mathbf{s}(\mathbf{X}_t, \tilde{\mathbf{Y}}_t - \eta_t \gamma) - \frac{M}{N} \mathbb{I}_N \right)_i \right| |\gamma_j| \right] \leq \frac{\max\{\frac{M}{N}, (1 - \frac{M}{N})\}}{\eta_t} \mathbb{E}_\gamma [|\gamma_j|] = \frac{\max\{\frac{M}{N}, (1 - \frac{M}{N})\}}{\eta_t} \sqrt{\frac{2}{\pi}}$. The last step comes from the fact that the max absolute value of an entry in $(\mathbf{s}(\mathbf{X}_t, \tilde{\mathbf{Y}}_t - \eta_t \gamma) - \frac{M}{N} \mathbb{I}_N)$ is $(1 - \frac{M}{N})$ for $2M \leq N$ and $\frac{M}{N}$ for $2M > N$ since $\mathbf{s} \in \mathcal{S} \subseteq \{0, 1\}^N$. Here $\mathbf{s}(\mathbf{X}_t, \tilde{\mathbf{Y}}_t - \eta_t \gamma)_i$ is the i^{th} entry of the vector with \mathbf{s} chosen under inputs \mathbf{X}_t and $\tilde{\mathbf{Y}}_t - \eta_t \gamma$ to minimize $\hat{\phi}_t(\mathbf{X}, \mathbf{Y})$

$$-\langle \mathbf{y}_t, \nabla^2 \phi_t(\mathbf{x}_t, \tilde{\mathbf{Y}}_t) \mathbf{y}_t \rangle \leq \frac{\max\{\frac{M}{N}, (1 - \frac{M}{N})\}}{\eta_t} \sqrt{\frac{2}{\pi}} \left| \sum_{i=1}^N \sum_{j=1}^N (\mathbf{y}_t)_i (\mathbf{y}_t)_j \right| \leq \frac{K^2 \max\{\frac{M}{N}, (1 - \frac{M}{N})\}}{\eta_t} \sqrt{\frac{2}{\pi}} \quad (34)$$

since

$$\left| \sum_{i=1}^N \sum_{j=1}^N (\mathbf{y}_t)_i (\mathbf{y}_t)_j \right| \leq K^2 \quad (35)$$

This follows because we have at most K distinct requests at any time t . Thus

$$T_2 \leq \frac{K^2 \max\{\frac{M}{N}, (1 - \frac{M}{N})\}}{\sqrt{2\pi}} \sum_{t=1}^T \frac{1}{\eta_t}. \quad (36)$$

From equations (13), (18), (30) and (36), we have

$$\begin{aligned} R_{\text{UR}}((\mathbf{x}_t)_{t=1}^T, T) &\leq \underbrace{\sum_{t=1}^{T-1} \mathbb{E}_\gamma \left[\left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_t) \right\rangle - \left\langle \left(\mathbf{s}_{t+1} - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_{t+1}) \right\rangle \right]}_{T_5} + \\ &\quad \mathbb{E}_\gamma \left[\left\langle \left(\mathbf{s}_T - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_T, \mathbf{s}_T) \right\rangle \right] + \eta_T \mathcal{G}_{\max}(\gamma) + \min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{t=1}^T (f(\mathbf{x}_t, \mathbf{s}) - \mathbf{y}_t) \right\rangle \\ &\quad + \frac{K^2 \max\{\frac{M}{N}, (1 - \frac{M}{N})\}}{\sqrt{2\pi}} \sum_{t=1}^T \frac{1}{\eta_t} + \eta_1 \mathcal{G}_{\max}(\gamma) - \min_{\mathbf{s} \in \mathcal{S}} \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{t=1}^T (f(\mathbf{x}_t, \mathbf{s}) - \mathbf{y}_t) \right\rangle \quad (37) \\ &\leq r_{\max}^C + \underbrace{\sum_{t=1}^{T-1} \mathbb{E}_\gamma \left[\left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_t) \right\rangle - \left\langle \left(\mathbf{s}_{t+1} - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_{t+1}) \right\rangle \right]}_{T_5} \\ &\quad + \eta_T \mathcal{G}_{\max}(\gamma) + \frac{K^2 \max\{\frac{M}{N}, (1 - \frac{M}{N})\}}{\sqrt{2\pi}} \sum_{t=1}^T \frac{1}{\eta_t} + \eta_1 \mathcal{G}_{\max}(\gamma) \quad (38) \end{aligned}$$

Note that we bound $\mathbb{E}_\gamma [\langle (\mathbf{s}_T - \frac{M}{N} \mathbb{I}_N), f(\mathbf{x}_T, \mathbf{s}_T) \rangle]$ by r_{\max}^C to obtain equation 38.

Now consider the term T_5 . We have

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

$$= \sum_{t=1}^{T-1} \mathbb{E}_\gamma \left[\left(\left\langle \left(\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_t) \right\rangle - \left\langle \left(\mathbf{s}_{t+1} - \frac{M}{N} \mathbb{I}_N \right), f(\mathbf{x}_t, \mathbf{s}_{t+1}) \right\rangle \right) \mathbb{I}(\mathbf{s}_{t+1} \neq \mathbf{s}_t) \right] \quad (39)$$

$$\leq \sum_{t=1}^{T-1} \mathbb{E}_\gamma [r_{\max}^C \mathbb{I}(\mathbf{s}_{t+1} \neq \mathbf{s}_t)] \quad (40)$$

$$= r_{\max}^C \sum_{t=1}^{T-1} \mathbb{E}_\gamma [\mathbb{I}(\mathbf{s}_{t+1} \neq \mathbf{s}_t)]$$

$$= \frac{3r_{\max}r_{\max}^C(|\mathcal{S}| - 1)}{2\sqrt{\pi}\alpha} \sum_{t=1}^T \frac{1}{\sqrt{t}} \quad (41)$$

Here,

- Equation (39) comes from the fact that the terms inside the T_5 summation are zero when $\mathbf{s}_t = \mathbf{s}_{t+1}$.
- Equation (40) comes from the fact that the term $\langle (\mathbf{s}_t - \frac{M}{N} \mathbb{I}_N), f(\mathbf{x}_t, \mathbf{s}_t) \rangle - \langle (\mathbf{s}_{t+1} - \frac{M}{N} \mathbb{I}_N), f(\mathbf{x}_t, \mathbf{s}_{t+1}) \rangle$ is essentially the difference in expected coded rates $R_C(\mathbf{s}_t, \mathbf{x}_t)$ and $R_C(\mathbf{s}_{t+1}, \mathbf{x}_t)$ which can be upper bounded by max coded rate r_{\max}^C .
- Equation (41) comes from lemma 4.

By substituting (41) in (38), we get

$$R_{\text{UR}}((\mathbf{x}_t)_{t=1}^T, T) \leq \frac{3r_{\max}r_{\max}^C(|\mathcal{S}| - 1)}{2\sqrt{\pi}\alpha} \sum_{t=1}^T \frac{1}{\sqrt{t}} + \eta_T \mathcal{G}_{\max}(\gamma) + \frac{K^2 \max\{\frac{M}{N}, (1 - \frac{M}{N})\}}{\sqrt{2\pi}} \sum_{t=1}^T \frac{1}{\eta_t} + \eta_1 \mathcal{G}_{\max}(\gamma) + r_{\max}^C \quad (42)$$

Since this is true for every sequence of request patterns $(\mathbf{x}_t)_{t=1}^T$ we have

$$R_{\text{UR}}(T) \leq \frac{3r_{\max}r_{\max}^C(|\mathcal{S}| - 1)}{2\sqrt{\pi}\alpha} \sum_{t=1}^T \frac{1}{\sqrt{t}} + \eta_T \mathcal{G}_{\max}(\gamma) + \frac{K^2 \max\{\frac{M}{N}, (1 - \frac{M}{N})\}}{\sqrt{2\pi}} \sum_{t=1}^T \frac{1}{\eta_t} + \eta_1 \mathcal{G}_{\max}(\gamma) + r_{\max}^C \quad (43)$$

Theorem 1 follows by replacing $\max\{\frac{M}{N}, (1 - \frac{M}{N})\} \leq 1$, $r_{\max} \leq K$ and $r_{\max}^C \leq K$ with their respective upper bounds. \square

C. Proof of Theorem 2

We use the following lemma to prove Theorem 2.

Lemma 3. For a given history of request vectors $(\mathbf{x}_i)_{i=1}^{t-1}$ up to time $t - 1$ and a perturbation vector γ , the probability of switching cache configuration for Algorithm 1 in the step t for $\mathbf{s} \neq \mathbf{s}_t$ is given by

$$\mathbb{P}((\mathbf{s}_{t+1} = \mathbf{s}) | (\mathbf{x}_1 \cdots \mathbf{x}_{t-1}, \gamma)) \leq \frac{3r_{\max}}{2\alpha\sqrt{\pi}\sqrt{t+1}} \quad (44)$$

Proof. Let $\mathbf{X}_t = (\mathbf{x}_i)_{i=1}^{t-1}$ and $\mathbf{Y}_t = \sum_{i=1}^{t-1} \min\{\mathbf{x}_i, \mathbb{I}_N\}$. For a given \mathbf{X}_t and \mathbf{Y}_t , define $R_t(\mathbf{s}, \mathbf{X}_t, \mathbf{Y}_t)$ as

$$R_t(\mathbf{s}, \mathbf{X}_t, \mathbf{Y}_t) = \left\langle \left(\mathbf{s} - \frac{M}{N} \mathbb{I}_N \right), \sum_{i=1}^{t-1} f(\mathbf{x}_i, \mathbf{s}) - \mathbf{Y}_t \right\rangle. \quad (45)$$

This is essentially the cumulative rate incurred by a cache configuration \mathbf{s} until time $t - 1$ minus the $h(\mathbf{x}_i)$ terms. Let \mathbf{s}_t and \mathbf{s}_{t+1} denote the cache configurations of Algorithm 1 in time slots t and $t + 1$ respectively. Consider a cache configuration $\mathbf{s} \neq \mathbf{s}_t$. If $\mathbf{s}_{t+1} = \mathbf{s}$, then we have the event that

$$\{R_t(\mathbf{s}_t, \mathbf{X}_t, \mathbf{Y}_t - \eta_t \gamma) \leq R_t(\mathbf{s}, \mathbf{X}_t, \mathbf{Y}_t - \eta_t \gamma)\} \cap \{R_{t+1}(\mathbf{s}_t, \mathbf{X}_{t+1}, \mathbf{Y}_{t+1} - \eta_{t+1} \gamma) \geq R_{t+1}(\mathbf{s}, \mathbf{X}_{t+1}, \mathbf{Y}_{t+1} - \eta_{t+1} \gamma)\}.$$

Let $\mathbb{P}_\gamma^{\mathcal{H}_t}(\cdot)$ denote the function $\mathbb{P}_\gamma(\cdot | (\mathbf{x}_1 \cdots \mathbf{x}_{t-1}, \gamma))$. Then, we have

$$\begin{aligned} \mathbb{P}_\gamma^{\mathcal{H}_t}(\mathbf{s}_{t+1} = \mathbf{s}) &\leq \mathbb{P}_\gamma^{\mathcal{H}_t}(\{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 \\ &\quad \{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)\}) \\ &= \mathbb{P}_\gamma^{\mathcal{H}_t} \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 \right. \\ &\quad \left. \left\{ R_{t+1}(\mathbf{s}_t, \mathbf{X}_{t+1}, \mathbf{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) \\ &= \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+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) \\ &\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) \quad (46) \\ &\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) (R_t(\mathbf{s}_{t+1}, \mathbf{X}_t, \mathbf{Y}_t) - R_t(\mathbf{s}_t, \mathbf{X}_t, \mathbf{Y}_t)) \right) \quad (47) \end{aligned}$$

$$\leq \frac{1}{\sqrt{2\pi\alpha}} \left(\frac{r_{\max}}{\sqrt{t+1}} + \left(\frac{1}{\sqrt{t}} - \frac{1}{\sqrt{t+1}} \right) t r_{\max} \right) = \frac{1}{\sqrt{2\pi\alpha}} \left(\frac{r_{\max}}{\sqrt{t+1}} + \left(\frac{\sqrt{t}}{\sqrt{t+1}(\sqrt{t+1} + \sqrt{t})} \right) r_{\max} \right) \quad (48)$$

$$\leq \frac{3r_{\max}}{2\alpha\sqrt{\pi}\sqrt{t+1}} \quad (49)$$

Equation (46) can be obtained by expressing $R_t(\mathbf{s}, \mathbf{X}, \mathbf{Y})$ terms as a sum of $K(\mathbf{s}, \mathbf{x})$ terms. $R_{t+1}(\mathbf{s}_t, \mathbf{x}_{t+1}, \mathbf{Y}_{t+1}) - R_{t+1}(\mathbf{s}, \mathbf{x}_{t+1}, \mathbf{Y}_{t+1}) = (\sum_{i=1}^t K(\mathbf{s}_t, \mathbf{x}_t) - \sum_{i=1}^t h(\mathbf{x}_i)) - (\sum_{i=1}^t K(\mathbf{s}, \mathbf{x}_t) - \sum_{i=1}^t h(\mathbf{x}_i))$. where $h(\mathbf{x}_t) = (1 - \frac{M}{N}) \langle \mathbf{y}_t, \mathbb{I}_N \rangle$. The right side of the inequality in equation 46 follows as $\sum_{i=1}^t K(\mathbf{s}_t, \mathbf{x}_t) \leq \sum_{i=1}^{t-1} K(\mathbf{s}_t, \mathbf{x}_t) + r_{\max}$ and $\sum_{i=1}^t K(\mathbf{s}, \mathbf{x}_t) \geq \sum_{i=1}^{t-1} K(\mathbf{s}, \mathbf{x}_t)$ as the rate at time t is positive.

Equation 47 comes from the fact that the Gaussian PDF is upper bounded by $\frac{1}{\sqrt{2\pi}\sigma}$. Equation 48 comes from $\langle (\mathbf{s} - \mathbf{s}_t), \gamma \rangle$ being a sum of unit variance independent Gaussian random variables. Let $\sigma_{(\mathbf{s}, \mathbf{s}_t)}$ 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 $\min_{\mathbf{s} \neq \mathbf{s}_t} \sigma_{(\mathbf{s}, \mathbf{s}_t)} = 1$ which happens when $(\mathbf{s}_t, \mathbf{s})$ differ at exactly one file. \square

The proof of Theorem 2 can be obtained by replacing r_{\max} by K ($r_{\max} \leq K$) in the following lemma 4

Lemma 4. 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. Then

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

Proof. Recall that $C_{UR}(T)$ denotes the expected number of switches in cache configuration until time T for Algorithm 1 with unrestricted switching. Then, we have

$$C_{UR}(T) = \mathbb{E} \left[\sum_{t=1}^{T-1} \mathbb{I}(\mathbf{s}_{t+1} \neq \mathbf{s}_t) \right] \quad (50)$$

$$= \sum_{t=1}^{T-1} \mathbb{E} [\mathbb{E} [\mathbb{I}(\mathbf{s}_{t+1} \neq \mathbf{s}_t) | (\mathbf{x}_1 \cdots \mathbf{x}_{t-1}, \gamma)]] \quad (51)$$

$$= \sum_{t=1}^{T-1} \mathbb{E} [\mathbb{P}((\mathbf{s}_{t+1} \neq \mathbf{s}_t) | (\mathbf{x}_1 \cdots \mathbf{x}_{t-1}, \gamma))] \quad (52)$$

$$= \sum_{t=1}^{T-1} \sum_{\mathbf{s} \in \mathcal{S} \setminus \mathbf{s}_t} \mathbb{E} [\mathbb{P}((\mathbf{s}_{t+1} = \mathbf{s}) | (\mathbf{x}_1 \cdots \mathbf{x}_{t-1}, \gamma))] \quad (53)$$

$$\leq \sum_{t=1}^{T-1} \sum_{\mathbf{s} \in \mathcal{S} \setminus \mathbf{s}_t} \mathbb{E} \left[\frac{3r_{\max}}{2\alpha\sqrt{\pi}\sqrt{t+1}} \right] \quad (54)$$

$$= \frac{3r_{\max}(|\mathcal{S}| - 1)}{2\sqrt{\pi}\alpha} \sum_{t=1}^T \frac{1}{\sqrt{t}} \quad (55)$$

Here, equation 53 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 54 comes from the lemma 3. \square

D. Proof of Theorem 3

Proof. Theorem 2 (proof) suggests that cache configurations do not change frequently. Note that Algorithm 1 is the same for both the unrestricted switching scenario and the restricted switching scenario. However, in the unrestricted switching scenario, the cache configuration is allowed to change in every time slot $t \in [T]$, whereas in the restricted switching scenario, the cache configuration is allowed to change only in restricted time slots given by the set \mathcal{T} . Therefore, the difference in the transmission sizes incurred in these two cases solely comes from the cache switches that take place in between these inter-switch periods.

Recall that $l_k \triangleq t_k - t_{k-1}$ denotes the time gap between the $(k-1)^{th}$ and k^{th} switching slots, and $L \triangleq |\mathcal{T}|$ denotes the maximum number of allowed switches. For the restricted switching scenario, we enjoy the same transmission size starting from $\sum_{i=1}^{k-1} l_i$ as the transmission size for the unrestricted scenario until there is a switch at some time t for the unrestricted scenario, where $\sum_{i=1}^{k-1} l_i < t < \sum_{i=1}^k l_i$. Once a switch happens at instant $\sum_{i=1}^{k-1} l_i < t < \sum_{i=1}^k l_i$, there is no guarantee that the regret incurred for the restricted switching scenario from time t to $\sum_{i=1}^k l_i - 1$ is the same as that of the unrestricted switching scenario.

Let R_k be the total additional regret (compared to the unrestricted switching scenario) incurred during the k^{th} inter-switch period. Then, $R_{(\eta, R)}^{\mathcal{T}}(T) = R_{(\eta, UR)}^{\mathcal{T}}(T) + \sum_{k=1}^L R_k$. If the switch happens for the first time after $\sum_{i=1}^{k-1} l_i$ at time-slot t , we have the regret in R_k being upper bounded by $(\sum_{i=1}^k l_i - t) r_{\max}$. Therefore, we have

$$\begin{aligned}
R_k &\leq \sum_{t=\sum_{i=1}^{k-1} l_i+1}^{\sum_{i=1}^k l_i-1} \mathbb{E} [\mathbb{I}(\mathbf{s}_t \neq \mathbf{s}_{t-1})] \left(\sum_{i=1}^k l_i - t \right) r_{\max} \\
&= \sum_{t=\sum_{i=1}^{k-1} l_i+1}^{\sum_{i=1}^k l_i-1} \mathbb{E} [\mathbb{E} [\mathbb{I}(\mathbf{s}_t \neq \mathbf{s}_{t-1}) | (\mathbf{x}_1 \cdots \mathbf{x}_{t-1}, \gamma)]] \left(\sum_{i=1}^k l_i - t \right) r_{\max} \\
&= \sum_{t=\sum_{i=1}^{k-1} l_i+1}^{\sum_{i=1}^k l_i-1} \mathbb{E} \left[\sum_{\mathbf{s} \in \mathcal{S} \setminus \mathbf{s}_{t-1}} \mathbb{P}((\mathbf{s}_t = \mathbf{s}) | (\mathbf{x}_1 \cdots \mathbf{x}_{t-2}, \gamma)) \right] \left(\sum_{i=1}^k l_i - t \right) r_{\max} \tag{56}
\end{aligned}$$

$$\leq \sum_{t=\sum_{i=1}^{k-1} l_i+1}^{\sum_{i=1}^k l_i-1} \sum_{\mathbf{s} \in \mathcal{S} \setminus \mathbf{s}_{t-1}} \frac{3r_{\max}}{2\alpha\sqrt{\pi}\sqrt{t}} \left(\sum_{i=1}^k l_i - t \right) r_{\max} \tag{57}$$

$$\leq \frac{3r_{\max}^2(|\mathcal{S}| - 1)}{2\alpha\sqrt{\pi}\sqrt{\sum_{i=1}^{k-1} l_i + 1}} \sum_{t=\sum_{i=1}^{k-1} l_i+1}^{\sum_{i=1}^k l_i-1} \left(\sum_{i=1}^k l_i - t \right) \tag{58}$$

$$\leq \frac{3r_{\max}^2(|\mathcal{S}| - 1)}{2\alpha\sqrt{\pi}\sqrt{\sum_{i=1}^{k-1} l_i + 1}} \frac{l_k(l_k - 1)}{2} \tag{59}$$

Equation 56 come from lemma 3 and equation 58 comes from $\sum_{i=1}^{k-1} l_i < t$. Adding all these R_k terms gives the above result. \square