Traffic-delay Trade-off in Wireless Caching Network

***Abstract*—Recent studies show that the multicast and cache can facilitate the wireless content distribution by mitigating the wireless traffic rate during the peak-traffic time, where the contents are prefetched to the local cache of mobile devices during the off-peak time. The remaining contents are then delivered in the multicast fashion such that multiple requests can be satisfied by one transmission. These techniques are based on the assumption that many requested contents are initiated in the peak-traffic time. Instead, the user requests are asynchronous and these techniques actually implement the trade-off between the traffic and delay. To illustrate the main idea, we investigate two classes of schemes under two scenarios in the single bottleneck network. Explicitly, the first scheme strives for minimum complexity by resorting to the simple cache procedure and pure multicast procedure; the second scheme is based on the famous coded cache scheme that consists of a cooperative cache procedure and coded multicast procedure. Under the first scenario, the content popularity distribution is uniform and there exists trival analysis of such trade-off, while under the second scenario, the content popularity distribution is nonuniform and the corresponding analysis becomes much more intractable since it is sensitive to the content popularity distribution. Compared to the constructed lower bounds, the proposed schemes are both shown to achieve the optimal trade-off between traffic and delay. As far as we know, this is the first work to address this problem.**

I. Introduction

The wireless data traffic volume beyond 2020 is predicted to be 1000 times higher than that in 2010 [1], which will place tremendous pressure in modern infrastructure such as backhaul and base station. This explosive tendency is driven mainly by the proliferation of smart user terminals (UT) and various multimedia request.

One promising way is to bring the content closer to user terminals via caching in order to reduce data traffic volume to accomodate such an explosive increase. The wireless network presents a high temporal variability in network traffic volume since the network resources are extremely scarce in certain time periods and comparatively idle in others. Then some caching techniques are proposed to balance the traffic load over the wireless link, in a word, popular contents are partially prefetched into user terminals during the off-peak time so the number of contents needed to be delivered in the peak-traffic time can be reduced, with requesters experiencing less congestions [2]– [5].

Another main point to alleviate such pressure is to multicast the contents to group of users. This method is based on the observation that the multimedia contents are not consumed with the same frequency, at least statistically [6]. In fact, among the huge quantity of requests, the users access a small portion of most popular contents and always send the same request. The base station can multicast a content to group of users who have the same request. Then the multiple requests can be satisfied by only one transmission, thus reduce the traffic over the wireless. However, this technique requires the base station to collect amount of requests from users, which leads to a large waiting time or delay for pre-arrival users. Thus, it should be exploited under some delay constraint.

Based on above two techniques, a novel scheme called coded cache is recently proposed in the single bottleneck network [7], which exploits the cooperation of different user caches and the coded multicast to significantly reduce the network traffic. In fact, users' requests are asynchronous and this scheme will yield a large delay for users who first send requests and unavailable in the practical scenario. Instead, we should consider the delay in such network and analyze the trade-off between traffic and delay. For example, in one extreme case, if we restrict the max tolerant delay is zero, the base station will lose ability to gather user request and only unicast to each user, which produces the largest traffic. In another extreme case, if the max tolerant delay is large enough to collect all users' requests in a cellular network, the base station can take a smart multicast scheme considering the characteristics of each user's local cache and their requests, thus the traffic can be minimized.

There are large amount of works investigating the trade-off between delay and throughput in Device-to-Device(D2D) network and ingenious results have been presented [12]– [14]. The way in which delay scales for traffic optimal schemes in caching network, however, has not been well-studied. Indeed, it is unclear what "delay" and "traffic" precisely means, especially in such setting. One of our work is the definition of "delay" and "traffic", which is both meaningful and theoretical tractable. As far as we know, this is the first work that addresses the problem and our main contributions are two-fold:

- We point out the traffic of the single bottleneck network can be improved via caching and multicasting, while increasing user's delay.
- We investigate the trade-off between user's delay and traffic in such network under two kinds of user demands and construct the information theoretical lower bound of the traffic under given delay. Explicitly, we present two kinds of schemes with progressively increasing complexity and analyze their delay-traffic trade-off under the given conditions. Based on the constructed lower bound, we prove both schemes are order optimal and only exhibits multiplicative gap.

The remainder of the paper is organized as follows. The preliminary is spread in Section II. Next, we present the service model and main results. In Section IV and V, we

investigate the trade-off for two schemes under two scenarios. In Section VI, we present a more complicated scheme and analyze its trade-off. Numerical results are given in Section VII. Finally, we summarize this work and present the future work in the last section.

II. Preliminary

In this section, we introduce the recently proposed scheme, called coded cache, which exploits the cooperation of different user caches and the coded multicast to significantly reduce the network traffic. Besides that, we briefly introduce the user request model to facilitate the delay analysis in the sequel.

A. Coded Cache

In the coded caching scheme, contents are divided into segments and partially prefetched during the off-peak-traffic time, and content retrieval requests issued during the peak-traffic time are responded by multicasting coded data over different contents. Moreover, an efficient caching scheme is proposed in [8], in which the content placement is performed in a decentralized manner by developing a caching algorithm that creates simultaneous coded-multicasting opportunities without coordination in the placement phase. While in [7], [8], it is assumed that the contents popularity is uniformly distributed. When considering the nonuniformity of content popularity distribution, some more coded caching schemes are developed. Knowing the popularity distribution of the contents in the base station, they classify contents into several groups and execute the coded caching for each group independently [9]. In a special case, that is, given that content popularity follows a Zipf distribution, they introduce a Least Recently Sent (LRS) scheme to characterize the minimum average number of transmissions to satisfy all user demands in the information theoretic sense to significantly improve multicast efficiency [10]. We make use of the coded caching scheme from [8] in this paper. Therefore, we now briefly overview this scheme.

Algorithm 1: Decentralized coded caching scheme with uniform demands

Placement Phase
for $(k = 0; k < K; k++)$ **do**
  for $(n = 0; n < N; n++)$ **do**
    user k randomly prefetches MF/N bits of content n ;
Delivery Phase
for $(k = K, k > 0; k--)$ **do**
  for *choose k users from K users to form a subset U* **do**
    server sends $\oplus_{k\in U} V_{k,U/\{k\}}$ to users in U.

The set in [8] assumes that the popularity of each content is uniform. It considers a similar architecture that has N contents and K users. Each user has a local cache of size MF bits with $M \leq N$. The codec scheme under such scenario operates as follows. In the placement phase, each user randomly caches the same amount MF/N bits of each content. Then in the later delivery phase, all users send their requests and the server transmits multiple coded signals $\oplus_{k\in U} V_{k,U/\{k\}}$[1] over the shared link to users to retrieve their requested contents. This scheme achieves a peak traffic rate of

$$K \cdot (1 - \frac{M}{N}) \cdot \frac{N}{KM} \cdot (1 - (1 - M/N)^K)F, \qquad (1)$$

which is shown to be within a constant multiplicative gap of the information theoretical lower bound. Illustration of this scheme is provided in Example 1.

Example 1 (Codec scheme in [8]). Suppose that there is a simple system distributing $N = 2$ contents A and B to $K = 2$ users, each with the cache size MF bits. In the placement phase, each user randomly caches $MF/2$ bits of content A and B independently. Let us focus on content A. The operations of placement phase partitions content A into four subcontents, $A = (A_\emptyset, A_1, A_2, A_{1,2})$, where $U \subset \{1, 2\}$, A_U denotes the bits of content A that are prefetched in the memories of users in U. For example, A_1 represents the bits of A only available in first user's memory. We adopt $|\cdot|$ as the operation of size, thus, $|A_\emptyset| = (1 - M/2)^2 F$ bits, $|A_1| = |A_2| = (M/2)(1 - M/2)F$ bits, $|A_{1,2}| = (M/2)^2 F$ bits. The same analysis holds for content B.

In the delivery phase, we assume that user 1 and user 2 request content A and B, respectively. User 1 has accessed to subcontent A_1 and $A_{1,2}$ in its local cache and lacks $A_\emptyset$ and A_2. Similarly, user 2 has already accessed to B_2 and $B_{1,2}$, and lacks $B_\emptyset$ and B_1. In traditional uncoded caching scheme, the server is required to unicast $A_\emptyset$ and A_2 to user 1 and unicast $B_\emptyset$ and B_1 to user 2. The total rate is

$$2(M/2)(1 - M/2)F + 2(M/2)^2 F = MF \text{ bits},$$

Under the coded caching manner, the server can satisfy the requests by transmitting $A_\emptyset$, $B_\emptyset$ and $A_2 \oplus B_1$ over the shared link, where $\oplus$ denotes the bit-wise XOR operation. The rate over the shared link is

$$(M/2)(1-M/2)F+2(M/2)^2 F = R(M, 2, 2)F < MF \text{ bits},$$

For other possible user requests, this rate is also achievable.

From **Example 1**, it can be found the gain of uncoded cache scheme comes from the isolated cache memory, called the *local cache gain*, captured by a factor $(1 - M/N)$ in (1), which linearly decreases with the cache size MF. Instead, the main gain of codec scheme comes from the multicasting opportunities created by a jointly designed placement and delivery phase, which can be captured by a factor called *global cache gain*, $N/K/M \cdot (1 - (1 - M/N)^K)$ in (1). This gain is inverse proportional to the cache size M and much more significant in aspects of reducing the traffic volume.

B. User Request model

Requests from users arrive at the server according to a random process. Typically we model the arrival process as

[1] $V_{k,U/\{k\}}$ denotes the bits of the content d_k requested by user k cached exclusively at users in U

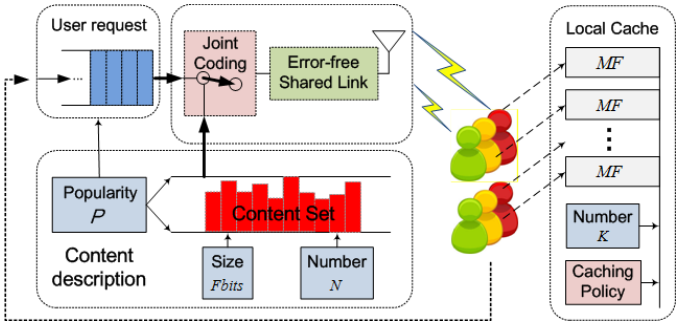

Fig. 1: System model

Poisson process. It is a viable model when the calls from a large population of independent users. Under the right circumstances, the number of requests from users arriving during a fixed time interval is a random number with a Poisson distribution.

In probability theory and statistics, the Poisson distribution is a discrete probability distribution that expresses the probability of a given number of events occurring in a fixed interval of time and/or space if these events occur with a known average rate and independently of the time since the last event [15].

A Poisson process is an example of an arrival process. Its also a commonly used model for random, mutually independent message arrivals, and the interarrival times provide the most convenient description since the interarrival times are defined to be i.i.d.

A discrete random variable X is said to have a poisson distribution if the random variable X only takes non-negative integer $0, 1, 2, ...$, and the probability mass function of X is given by:

$$\mathbb{P}(X = k) = \frac{(\lambda T)^k e^{-\lambda T}}{k!} \tag{2}$$

where λ is a constant parameter, independent of the time t, and independent of arrivals in earlier intervals. λ is called the arrival rate and λ. T is the time period.

III. System Model and Main Results

A. System Model

We consider a single bottleneck network where a base station serves some users, each user has a local cache of size MF bits. There are N contents in the base station, each has the same size of F bits. The contents requested by the different users from the base station are independent of each other and obeys a content popularity distribution $P = [p_1, p_2, \ldots, p_N]$, where p_i is the probability that a user requests content i. The main notations are listed in the Table I.

In practical scenario, the channel state of each user is fluctuated and different among them, thus the number of bits carried in each channel might be different. There are many resource allocation methods to counteract this fading effect, such as the power control and bandwidth or time slot allocation. This line of work, however, is not pursued here and we only rely on the scheme how to cache and how to make a delivery in the application level. Thus, we assume the

TABLE I: Main notations

N	The number of contents to be distributed.
F	The size of each content.
V_i	Content i.
P	The content popularity distribution, $P = [p_1, p_2, ..., p_N]$, where p_i is the popularity of content i measured by the probability content i is requested.
Q	The caching distribution, $Q = [q_1, q_2, ...q_N]$, where q_i is the proportion of cache space allocated for content i.
M	The number of contents can be prefetched by each user.
λ	The arrival rate of requests.
Γ_u^e	The pure multicasting scheme under uniform user demands
Γ_c^e	The coding multicasting scheme under uniform user demands
Γ_u^d	The pure multicasting scheme under nonuniform user demands
Γ_c^d	The coding multicasting scheme under nonuniform user demands
Γ_{tc}	The coding multicasting scheme with inter-transmission-coding under uniform user demands
$R^e(D, \Gamma)$	The traffic under uniform user demands, which is dependent on N, M, λ , delay D and the scheme Γ. Note this is a brief notation.
$R^d(D, \Gamma)$	The traffic under nonuniform user demands, which is dependent on N, M, λ, popularity distribution P, caching distribution Q, delay D and the scheme Γ. Note this is a brief notation.

contents are distributed through an error-free shared link to users such as in the long term evolution (LTE) broadcasting system [16], where the error-free can be achieved with error correction scheme or reliable transmission scheme in the upper layer.

The system operates as follows. In the placement phase, part of contents are prefetched in each user's local cache. As the placement phase happens in the off-peak time such as the late night, the wireless traffic incurred in this phase is tolerable. In the delivery phase, any user k sends its request d_k in the time t_k and follows a poission arrival model (2) with rate λ. The base station collects amount of requests in its buffer without exceeding the max tolerant delay for each user, then it takes a multicast fashion to transmit specific data through the shared wireless link to those users. Each user recover its requested contents using both the local cached data and the data just received over wireless.

Based on above system model, we formulate the following traffic-delay trade-off analysis problem:

Definition 1. ***(Traffic-delay trade-off Problem)*** *For given delay constraint, cache size, the number of contents and the arrival rate of user requests, how to harness the peak-time traffic in the delivery phase?*

Intuitively, increasing such delay will reduce the traffic, since the base station can collect more user requests and operate smartly. However, different schemes will produce a different trade-off between traffic and delay. Therefore, this problem requires not only designing the specific scheme including prefetching and multicasting, but also a trade-off analysis under such scheme, i.e., how the traffic scales as delay? This leads to the following definitions for traffic and delay that will be adopted in the sequel.

Definition 2. ***(Delay)*** *The delay D of our model is defined as*

the average maximum delay between the instant representing the start of a request of a user, and the instant representing the start of transmission of a packet that can be tolerated for this user.

Definition 3. ***(Traffic)*** *The traffic of a scheme is defined as the average traffic volume to satisfy one request produced in the shared link, with the unit of F bits.*

A brief comment on the notion of delay adopted in our work is now in order. This definition of D refers to the queuing delay in the network layer and does not consider the other kinds of delay such as processing delay, propagation delay and transmission delay. Since the queuing delay in our system dominates the main part, and as argued in the sequel, this delay analysis offers a lower bound on the total delay. Besides, this definition of D means that, in each transmission of the base station, the user's experienced delay can be larger or less than D. The only constraint is that the long-term average delay is less than D. Note this notion can be regarded as the ergodic trade-off analysis instead of the outage trade-off analysis that any transmission should obey this delay constraint.

B. Main Results and Outline

We investigate the trade-off between traffic and delay by analyzing the explicit schemes proposed in such network. Two scenarios are considered: the first is that contents has the uniform content popularity distribution and there exists some trival schemes that are order optimal; the second is that the contents has the nonuniform content popularity distribution and the design of such order optimal scheme is a non-trival procedure, including how to optimally prefetch the contents in accord with the content popularity distribution,

Under uniform user demands, that means, $p_1 = p_2 = \cdots = p_N$. We show a possible lower bound of this system is $R = \Theta\left(\frac{1}{D}\right)$. Moreover, we proposes two classes of schemes with progressively increasing complexity to approximate this lower bound. Both of them scale as the same law of $R = \Theta\left(\frac{1}{D}\right)$ while producing different multiplicative gap. Furthermore, the following technical assumption is imposed.

We let

$$D = \max\left\{\omega\left(\left(1-\left(1-\frac{1}{N-M}\right)^{1-\frac{M}{N}}\right)^{-1}\right), \omega\left(\frac{N}{M}\right)\right\}. \tag{3}$$

This technical assumption is made to ensure (as shown in the sequel) that the average traffic is not donminated by the scaling behaviour of D. Note that for constant N, M and K, this assumption is always established.

1) The first scheme strives for minimum complexity by resorting to a simple caching rule that all users store the same M contents, along with non-coding multicast and yields a scaling law of
$$R = \frac{N-M}{\lambda}\Theta\left(\frac{1}{D}\right). \tag{4}$$
2) The second, and more complex, scheme is based on the famous coded cache technique, benefiting from the cooperation between the cache phase and joint coding delivery phase, yielding the scaling law of
$$R = \frac{N-M}{\lambda M}\Theta\left(\frac{1}{D}\right). \tag{5}$$
This scheme shows a constant gain M over the first scheme.

Under nonuniform user demands, that means, some contents are hot and always being required while others are desolate and rarely being required. In our analysis, we assume $p_1 \le p_2 \le \cdots \le p_N$. Furthermore, the following technical assumption is imposed. We let

$$D = \max\left\{\max_{1\le i\le N}\left\{\frac{1}{p_i}\right\}, \max_{M+1\le i\le N}\left\{\left(1-(1-p_i')^{1-S_M}\right)^{-1}\right\}\right\}, \tag{6}$$

where $p_i' = p_i\left(\sum_{j=M+1}^{N} p_j\right)^{-1}$ and $S_M = \sum_{i=M+1}^{N} p_i$. This technical assumption is also made to ensure (as shown in the sequel) that the average throughput is not donminated by the scaling behaviour of D.

According to the lower bound constructed under the uniform user demands, we show a possible lower bound of nonuniform demands is also $R = \Theta\left(\frac{1}{D}\right)$ under condition (6).

1) The first scheme requires each user caches the M most popular contents and takes non-coding multicast yielding a same scaling law of (4).
2) The second scheme is much more complicated. The cache space of each user is allocated based on a specific caching distribution $Q = [q_1, q_2, \ldots, q_N]$, where q_i denotes the proportion of content i occupying each user's cache. Then, it takes a same coded multicast method of coded cache technique. After struggling to determine an optimal caching distribution, we show that the scaling law of such scheme can be at least
$$R = \frac{(N-1)\left(1-\left(\prod_{i=1}^{N} p_i\right)^{\frac{1}{N}}\right)}{\lambda} O\left(\frac{1}{D}\right). \tag{7}$$

IV. Uniform User Demand Scenario

In this section, we consider the uniform user demand scenario where the content has the same popularity that $p_1 = p_2 = \cdots = p_N$. Under this scenario, we can take the fair cache space allocation strategy, i.e., each user cache the same M contents or each user's local cache is divided equally to N parts and stores different parts of these N contents.

A. Scheme with Pure Multicast

In this class of scheme Γ_u^e, we take a simple cache rule that each user prefetches the same M contents in the placement phase. Since the contents have a uniform popularity distribution, we choose from first M contents. Then, in the delivery phase, the server divides the users into G groups according to their requests, where the users in the same group has the same request. The server multiacasts the requested contents to each group, respectively. This procedure does not consider the coding opportunities among different requests and refers to pure multicast.

Theorem 1. *Under the condition (3), the average traffic produced by scheme Γ_u scales as*

$$R^e(D,\Gamma_u^e)=\frac{N-M}{\lambda}\Theta\left(\frac{1}{D}\right). \tag{8}$$

Theorem 1 provides the trade off produced by scheme Γ_u^e. It can be found that the traffic is inverse proportional to the D. Besides that, it produce a cache gain that is linear to M.

B. Scheme with coding Multicast

This scheme is based on the decentralized coded cache scheme in [8] with considering the delay constraint. The main procedure is same to the scheme Γ_c^e in Section II. We now focus on the trade off analysis of this scheme.

Theorem 2. *Under the condition (3), the average traffic produced by scheme Γ_c^e scales as*

$$R^e(D,\Gamma_c^e)=\frac{N-M}{\lambda M}\Theta\left(\frac{1}{D}\right). \tag{9}$$

It can be found that the scheme Γ_c produces a same order delay gain compared to scheme Γ_u, while it provides a larger constant gain of $1/M$. This constant gain comes from the cooperative of the distributed cache and coding multicast.

C. Lower Bound

Having presented two schemes under this scenario and showing their trade-off between traffic and delay, we proceed with a lower bound on it, which is independent of any practical schemes.

Theorem 3. *For given N contents, delay D. The user requests arrive at a rate of λ and each user has a cache of size $0\leq M\leq N$,*

$$R_{lb}^e(D)=\Omega\left(\frac{1}{D}\right). \tag{10}$$

Based on the constructed lower bound, we can find that above two schemes implement the same order as the lower bound and provides the different constant gain. In general, a key insight from the procedure of above proof is that, if the traffic under N contents and k requests is bounded by the number of contents N instead of the number of requests k, it always show an trade-off $\Theta\left(\frac{1}{D}\right)$. For example, if we take the traditional unicast scheme, the traffic under N contents and k requests is $k(1-M/N)$ and shows a trade-off that $R=1-M/N$ that is independent of delay D.

V. Nonuniform User Demand Scenario

When the contents have different popularity such as $p_1\leq p_2\leq\cdots p_N$, it is inappropriate to take above fair cache rule. A tendentious cache rule is needed, i.e., allocate more space to the more popular content.

A. Scheme with Pure Multicast

This scheme does not consider the coding opportunities in the delivery phase and is same as delivery procedure of the scheme Γ_u^e, thus the heterogeneity of the contents cached in each user's local cache has no influence on the delivery phase. It only considers the the the number of different contents they request. It is enough to adopt a cache rule that each user prefetches the M most popular contents in the placement phase.

Theorem 4. *Under the condition (6), the traffic produced by scheme Γ_u^d scales as*

$$R^d(D,\Gamma_u^d)=\frac{N-M}{\lambda}\Theta\left(\frac{1}{D}\right). \tag{11}$$

Theorem 4 shows that the trade-off of pure multicast scheme in nonuniform case equals to the trade-off in the uniform case under specific condition.

B. Scheme with coding Multicast

In the scheme Γ_c^e, it takes a fair cache allocation strategy that each user randomly prefetches MF/N bits of each content. When the content popularity distribution is nonuniform, some contents are requested more frequently while others are desolate. Intuitively, we'd better allocate more cache space to more popular contents. We take a metric called caching distribution $Q=[q_1,q_2,\ldots,q_N]$ to describe such tendentious cache allocation strategy, where q_i denotes the proportion that content i occupying user's local cache. In the placement phase, the contents are prefetched based on Q. Then, it operates a same coded multicast in the delivery phase.

Note that different caching distribution will produce a different traffic and might drive a different trade-off between delay and traffic. For comparison between the different schemes, we determine an optimal caching distribution to minimize the traffic and analyze the trade-off behaviour under such optimal caching distribution.

OPT:

$$\text{Min}\quad R^d(D,\Gamma_c^d)=\sum_{k=0}^{\infty}\frac{(\lambda D)^{k-1}e^{-\lambda D}}{k!}\overline{R^d}(k,Q,\Gamma_c^d) \tag{12}$$

$$\text{s.t.}\quad \sum_{i=1}^{N}q_i=1, 1\leq q_i\leq\frac{1}{M}, \tag{13}$$

where the constraint is to avoid violating caching capacity constraints and caching duplicated packets for each user.

$\overline{R^d}(k,Q,\Gamma_c^d)$ is the average traffic under scheme Γ_c^d, k requests and caching distribution Q, according to definition,

$$\overline{R^d}(k,Q,\Gamma_c^d)=\sum_{U_i\in U,|U_i|=k}\mathbb{P}\left(U_i\in U\big||U_i|=k\right)R_Q(U_i,\Gamma_c^d). \tag{14}$$

The following lemma shows another way to determine $\overline{R^d}(k,Q,\Gamma_c^d)$.

Lemma 1. *Under the request situation $\vec{s}=(\alpha_1,\alpha_2,\ldots,\alpha_N)$ and the given content popularity distribution P, where α_i*

denotes the number of users requesting content i and satisfies $\alpha_1 + \alpha_2 + \ldots + \alpha_N = k$,

$$\overline{R^d}(k, Q, \Gamma_c^d) = \sum_{\vec{s}} \mathbb{P}(\vec{s}) \cdot R_{\vec{s}}(Q, \Gamma_c^d), \tag{15}$$

where $R_{\vec{s}}(Q, \Gamma_c^d)$ denotes the traffic rate under request situation $\vec{s}$, caching distribution Q and specific scheme Γ_c^d, and calculated by

$$\sum_{i=1}^{k} \sum_{v \subset [k], |v|=i} \max_{j \in v} \{(q_{d_j} M)^{i-1} (1 - q_{d_j} M)^{k-i+1}\}. \tag{16}$$

It shows the traffic is only related to the number of requests for each content, and independent of who requests which content. The proof of Lemma 1 is based on a symmetric analysis and can be seen in the Appendix **??**.

$\mathbb{P}(\vec{s})$ is the the probability of request situation $\vec{s}$ occurring. Consider that α_1 users request content 1, α_2 users request content 2, α_3 users request content 3, ..., and α_N users request content N, the number of all possible cases for such an event is $C_K^{\alpha_1} \cdot C_{K-\alpha_1}^{\alpha_2} \cdot C_{K-\alpha_1-\alpha_2}^{\alpha_3} \cdots C_{K-\alpha_1-\cdots-\alpha_{N-1}}^{\alpha_N}$, and the probability of these cases are identical, thus,

$$\mathbb{P}\left(U_i \in U \middle| |U_i| = k\right) = \frac{k!}{\alpha_1! \cdot \alpha_2! \cdot \cdots \alpha_N!} \cdot \prod_{i=1}^{N} p_i^{\alpha_i}. \tag{17}$$

Then the optimal caching distribution is given by

$$Q^* = \arg\min_{Q} R^d(D, \Gamma_c^d). \tag{18}$$

We now consider the computation complexity of the optimization problem OPT. The computation complexity of calculating $R_{\vec{s}}(Q, \Gamma_c^d)$ is $O(2^k)$ and the number of all possible $\vec{s}$ is equivalent to the number of solutions of equation: $\sum_{i=1}^{N} \alpha_i = k$, which is C_{k+N-1}^{N-1}. Thus, the computation complexity of obtaining optimal caching distribution Q^* under $R^d(D, \Gamma_c^d)$ is at least $O(2^k \cdot C_{k+N-1}^{N-1})$. It can be found that the complexity increases exponentially as N, k and obtaining the optimal caching distribution is unavailable when the number of users and contents become large. Nevertheless, it provides a feasible approach to compute the optimal solution as a benchmark.

Since the computational complexity of the problem OPT is high, it is natural to ask if there is any equivalence, which can obtain a solution that is approximate to the optimal one but incurs low computational complexity. We note that the traffic over wireless is used to deliver the part of content that has not been cached on the mobile devices. In an extreme case that every content has been cached locally, there is no need of the wireless traffic in the delivery phase; therefore, minimizing the average size of uprefetched contents can reduce the traffic volume. The corresponding problem formulation is as follows.

OPT-Relax:

$$\text{Min} \quad \sum_{k=0}^{\infty} \frac{(\lambda D)^{k-1} e^{-\lambda D}}{k!} V(k, Q, \Gamma_c^d) \tag{19}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} q_i = 1, 1 \leq q_i \leq \frac{1}{M}, \tag{20}$$

where $V(k, Q, \Gamma_c^d)$ is the average size of uprefetched contents under k users, and can be calculated by

$$V(k, Q, \Gamma_c^d) = \sum_{i=1}^{N} p_i (1 - q_i M)^k. \tag{21}$$

Then, the objective function is transformed into

$$\sum_{i=1}^{N} p_i e^{\lambda D M q_i}. \tag{22}$$

Theorem 5. *The optimal caching distribution $Q^\dagger$ under the **OPT-Relax** model is*

$$q_i^\dagger = \frac{1}{NM} \left[1 + \frac{1}{\lambda D} \ln \left(\frac{p_i^N}{\prod_{j=1}^{N} p_j} \right) \right], 1 \leq i \leq N. \tag{23}$$

This theorem provides a relaxed solution to the original model. It seems like a kind of luffing method: for those contents with higher popularity, up in the average line $\frac{1}{NM}$; for those contents with lower popularity, down in the average line $\frac{1}{NM}$. The proof is based on a convex analysis and general Lagrangian multiplier method, and can be seen in the Appendix **??**.

We take the caching distribution $Q^\dagger$ as the approximate caching distribution for the **OPT** model and further show the trade-off between traffic and delay under this caching distribution.

Theorem 6. *Under the condition (6), the traffic produced by scheme Γ_c^d scales as*

$$R = \frac{(N-1)\left(1 - \left(\prod_{i=1}^{N} p_i\right)^{\frac{1}{N}}\right)}{\lambda} O\left(\frac{1}{D}\right). \tag{24}$$

Theorem 6 shows that the traffic produced by scheme Γ_c^d is at most $O\left(\frac{1}{D}\right)$.

C. Lower Bound

The information theoretical lower bound is independent of any delivery scheme and caching distribution. Theorem 7 presents a possible lower bound based on average cut-set bound argument.

Theorem 7. *For given N contents, content popularity distribution, delay D and condition (6). The user requests arrive at a rate of λ and each user has a cache of size $0 \leq M \leq N$,*

$$R_{lb}^d(D) = \Omega\left(\frac{1}{D}\right). \tag{25}$$

VI. Numerical Results

In this section, we validate the theoretical analysis by providing numerical results on the network performance and compare them to the analytical results. We develop a simulation platform based on the system model on Matlab to emulate behaviour of such network. In this platform, we consider a circular cell with no inter-cell interference from other cells,

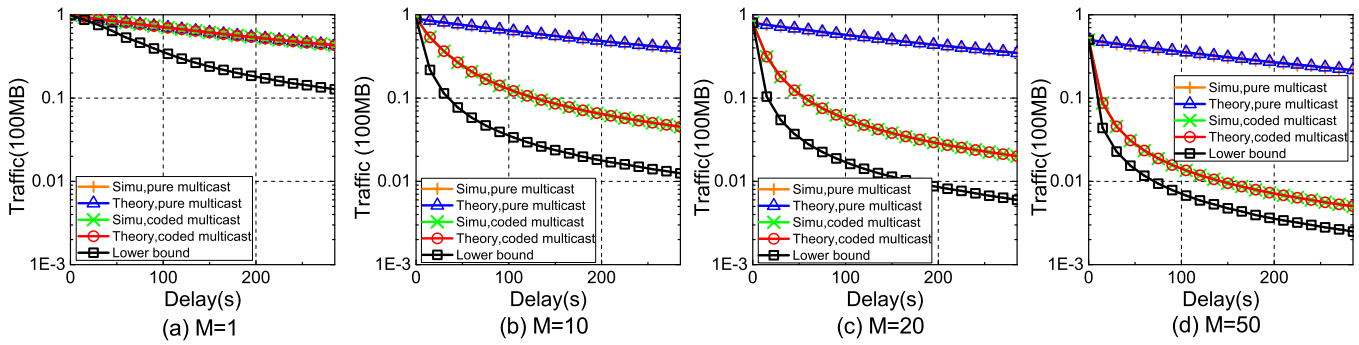

Fig. 2: The theoretical and simulation trade-off produced by different schemes under uniform user demands.

which has a radius of 600m and users are randomly and independently distributed in the cell. The content is of size 100MB, which is reasonable by assuming a screen size 1280×720 and 150 seconds playing time. Based on the platform, intensive simulations are performed, from which we obtain statistics about trade-off between traffic and delay. We first evaluate the accuracy of the derived analytical expressions. The impact of content popularity distribution and network parameters is then discussed as following.

A. Uniform user demands

The theoretical results is based on the accurate description (29) and (32) instead of the scaling law, since the scaling law can be attained straightforward by these equations. The lower bound is based on equation (37). As shown in Fig. 2, the theoretical results are in excellent agreement with the simulation results and they all show a inverse proportional fashion respect to the delay. The coded cache scheme outperforms the pure multicast scheme under small delay, while shows a constant gap under large delay.

We further investigate how the cache size affects above schemes. From Fig. (2)(a) to Fig. (2)(d), we operate these two schemes under different cache size $M = \{1, 10, 20, 50\}$ and observe that, when the cache size increases, the gain of coded cache scheme increases much faster than the pure muticast scheme and approximates to the theoretical lower bound. The main reason is that the coded cache scheme exploits the cooperation of distributed cache and introduces a *global cache gain* or a *coded gain*, which is higher order than the traditional pure multicast scheme [8].

A fundamental problem in this setting is how much traffic saving can be obtained via sacrificing the delay. For example, when the cache size is small, as shown in Fig. (**??**)(b), we can see that when the delay increases from 1 to 10, the traffic is reduced about 3.4% and 30% under pure multicast scheme and coded cache scheme, respectively. When the cache size is large, as shown in Fig. (2)(b), we can see that when the delay increases from 100 to 200, the traffic is reduced about 30% and 60% under pure multicast scheme and coded cache scheme, respectively. Note this traffic reduce effect is strengthen when the cache size increases.

B. Nonuniform user demands

We adopt the Zipf distribution to describe the nonuniform user demands.

$$p_i = \frac{\frac{1}{i^v}}{\sum_{j=1}^{N} \frac{1}{j^v}}, 1 \le i \le N. \tag{26}$$

The Zipf parameter v describe the skewness of the distribution that the distribution shows large skewness under large v and vice versa. In most cases, the Zipf parameter $v = 0.6$ and we fix it in the following simulation. The theoretical results of pure multicast scheme is also based on the accurate description (42) and the theoretical results of coded cache scheme is based on its upperbound (61). Note that the different cache distribution will affect the trade-off of the coded cache scheme. We first adopt the caching distribution $Q^\dagger$ for comparison. Then we investigate how the cache distribution affect such trade-off.

As shown in Fig. 3, we compare above two schemes under different cache size $M = \{1, 2, 5, 10\}$. The theoretical results of pure multicast scheme is in excellent agreement with its simulation results, since we take a accurate description of trade-off. While the theoretical results of coded multicast scheme is much larger than the simulation results, since the theoretical results are based on the upper bound. The simulation results of coded multicast scheme approximates to the theoretical lower bound and shows a significant gain of decreasing traffic when delay increases. Compared to the uniform case, the traffic decreasing gain is more obvious in the nonuniform cases. As shown in Fig. (3)(b), we can see that when the delay increases from 1 to 10, the traffic is reduced about 10% and 50% under pure multicast scheme and coded cache scheme, respectively. Besides that, when the cache size increases, the pure multicast scheme shows little gain while the coded multicast scheme shows a significant gain of decreasing the traffic.

Further, we investigate how the cache distribution affect such trade-off for the coded multicast scheme. Here we compare the $Q^\dagger$ with the well-known Least Recently Used (LRU) caching scheme and the baseline scheme in [11].

LFU: In such a scheme, each user prefetches the M most popular contents in its local cache.

Baseline Scheme: This scheme divides the contents set into two groups and fairly prefetches the contents of the first group.

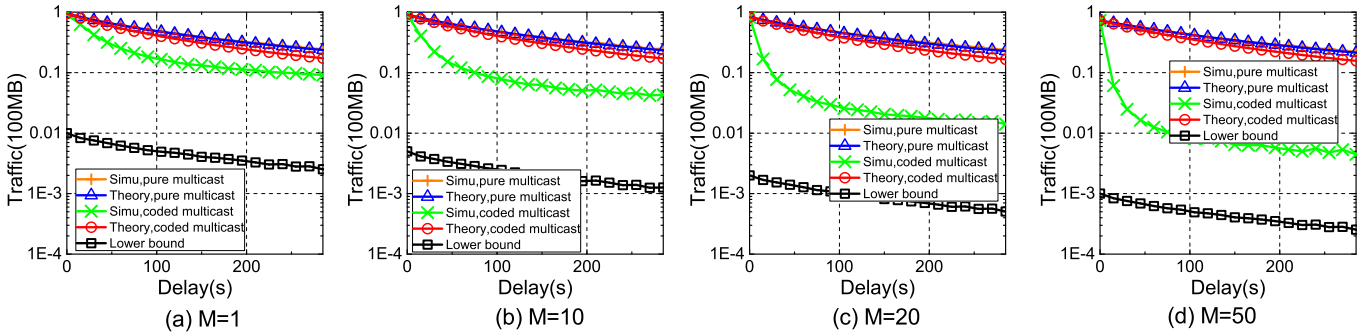

Fig. 3: The theoretical and simulation trade-off produced by different schemes when $v = 0.6$.

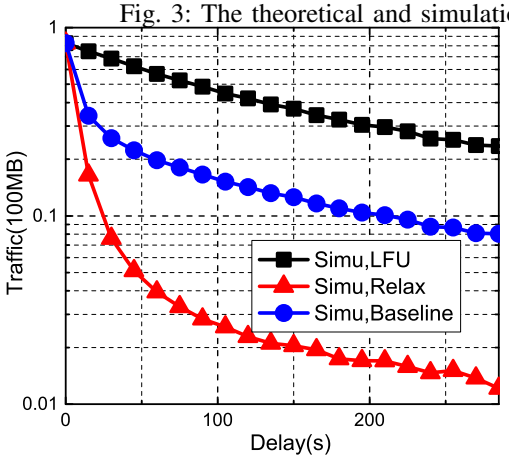

Fig. 4: The theoretical and simulation trade-off produced by different schemes when $v = 0.6$.

Fig. 4 shows the traffic versus the delay under different caching distribution. It can be found that $Q^{\dagger}$ produces the lowest traffic and the baseline scheme is laid between the $Q^{\dagger}$ and LFU. The trade-off produced by $Q^{\dagger}$ shows apparent inverse proportional manner while another two caching distributions produces the partial inverse proportional manner. The main reason behind this trend is that LFU and baseline scheme only prefetches part of contents, which decreases the coding opportunities among different requests, thus diminishes such global cache gain in the transmission.

VII. Conclusion and Future Work

This paper has addressed the *traffic-delay trade-off* problem in the single bottleneck network and has presented two schemes to implement it, including pure multicast and code multicast scheme. We theoretically show the trade-off produced by both schemes under two kinds of scenarios: uniform user demands and nonuniform user demands. For the second scheme, the optimized caching distribution have been derived with the content popularity distribution taken into account. Based on the constructed lower bound, both schemes show the order optimality with respect to the delay. Numerical results have shown that the theoretical analysis are in excellent agreement with the simulation results.

Our future work is focusing on the more efficient scheme to implement such trade-off. For example, we assume the content is transmitted instantaneously. Indeed, there exists transmission delay for each content that can be exploited to further reduce the traffic.

References

[1] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update 2012-2017," *Whiter paper*, 2013.

[2] N. Golrezaei, A. F. Molisch, A. G. Dimakis, G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Communications Magazine*, vol. 51, no. 4, pp.142–149, 2013

[3] J. Llorca, A. M. Tulino, K. Guan, J. Esteban, M. Varvello, N. Choi, and D. Kilper, "Network-coded caching-aided multicast for efficient content delivery," in *Proc. IEEE ICC*, 2013, pp.3557–3562.

[4] J. Llorca and A. M. Tulino, "The content distribution problem and its complexity classification," *Alcatel-Lucent technical report*, 2013.

[5] S. Gitzenis, G. S. Paschos, L. Tassiulas, "Asymptotic laws for joint content replication and delivery in wireless networks," *IEEE Transactions on Information Theory*, vol. 59, no. 5, pp. 2760–2776, 2013

[6] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: Analyzing the worlds largest user generated content video system," in *Proc. ACM SIGCOMM*, 2007, pp. 1–14.

[7] M. A. Maddah-Ali, U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, no.5, pp. 2856–2867, 2014.

[8] M. A. Maddah-Ali, Niesen U, "Decentralized coded caching attains order-optimal memory-rate trade-off," *IEEE/ACM Transactions on Networking*, vol. PP, no.1, pp. 1, 2014.

[9] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," in *Proc. IEEE INFOCOM*, 2014, pp. 221–226.

[10] M. Ji, A. M. Tulino, J. Llorca, G. Caire, "Order optimal coded caching-aided multicast under Zipf demand distributions," Feb. 19, 2014, Online: http://arxiv.org/abs/1402.4576.

[11] M. Ji, G. Caire and A. F. Molisch, "Order optimal coded caching-aided multicast under Zipf demand distributions," in *Proc. IEEE Information Theory Workshop*, 2013, pp. 1–5.

[12] A. E. Gamal, J. Mammen, B. Prabhakar, S. Devavrat, "Throughput-Delay Trade-off in Wireless Networks," in *Proc. IEEE INFOCOM*, 2004, pp. 464–475.

[13] M. J. Neely, E. Modiano, "Capacity and delay trade-offs for ad hoc mobile networks," in *IEEE Transactions on Information Theory*, vol. 52, no.6, pp. 1917–1937, 2005.

[14] S. Toumpisa, A. J. Goldsmith, "Large wireless networks under fading, mobility, and delay constraints," in *Proc. IEEE INFOCOM*, 2004, pp. 609–619.

[15] R. D. Yates, D. J. Goodman, "Probability and stochastic processes," *John Wiley and Sons*, 1999.

[16] EXPWAY, Online: http://www.expway.com/.

[17] D. D. Sleator and R. E. Tarjan, "Amortized efficiency of list update and paging rules," *ACM Communications*, vol. 28, no.2, pp. 202–208, 1985.

[18] M. Chrobak and J. Noga, "LRU is better than FIFO," in *Proc. ACM SODA*, 1999, pp. 78–81.

[19] R. Pedarsani, M. A. Maddah-Ali and U. Niesen "Online coded caching,"Nov. 14, 2013, Online: http://arxiv.org/abs/1311.3646.

[20] G. S. Ladde and D. D. Siljak, "Convergence and stability of distributed stochastic iterative processes," *IEEE Transactions on Automatic Control*, vol. 35, no. 6, pp. 665–672, 1990.

[21] G. S. Ladde and M. Sambandham, "Random difference inequalities," *North-Holland Mathematics Studies*, vol. 110, pp. 231–240, 1985.

[22] A. E. Raftery "A model for high-order Markov chains," *Journal of the Royal Statistical Society*, vol. 47, no. 3, pp. 528–539, 1985.

[23] G. Szabo and B. A. Huberman, "redicting the popularity of online content," *ACM Communications*, vol. 53, no. 8, pp. 80–88, 2010.

[24] M. Cha, H. Kwak, P. Rodriguez, Y-Y. Ahn and S. Moon, "I tube, You Tube, Everybody Tubes: Analyzing the Worlds Largest UserGenerated Content Video System," in *Proc. ACM SIGCOMM conference on Internet measurement*, 2007, pp. 1–14.
[25] X. Cheng, J. Liu and C. Dale, "Understanding the characteristics of internet short video sharing: A YouTube-based measurement study," *IEEE Transactions on Multimedia*, vol. 15, no. 5, pp. 1184–1194, 2013.
[26] V. Chandrasekhar, J. G. Andrews, and A. Gatherer, "Femtocell networks: a survey," *IEEE Communications Magzine*, vol. 46, no. 9, pp. 59–67, 2008.
[27] http://traces.cs.umass.edu/index.php/Network/Network.

APPENDIX A

A. Proof of Theorem 1

Assume there exists a user request sequence set $S = \{t_1, t_2, \ldots, t_L\}$ and $I = \{k_1, k_2, \ldots, k_L\}$with $L \to \infty$, where t_i is the arrival time of ith request and k_i is the user id of ith request. We define a partitioning $U = \{U_1, U_2, \ldots, U_h\}$ of set I that satisfies $E_{U_i}\left[\max_{k\in U_i} t_k - \min_{k\in U_i} t_k\right] \leq D$. According to the definition of the D, the set U_i is the user group that can be satisfied in one transmission without violating delay constraint, hence,

$$R^e(D, \Gamma_u) = \lim_{L\to\infty} \frac{1}{L} \sum_{i=1}^{h} R_u(M, U_i, \Gamma_u). \tag{27}$$

Since the user request behaviour can be regarded as a a wide-sense stationary process, thus we take the statistical average to represent above long sample of the process,

$$\begin{aligned} R^e(D, \Gamma_u^e) &= \frac{1}{\lambda D} E\left[R_u(U_i, \Gamma_u^e)\right] \\ &= \frac{1}{\lambda D} \sum_{k, U_i} R_u(U_i, \Gamma_u^e) \mathbb{P}\left(|U_i| = k, U_i \in U\right) \\ &= \frac{1}{\lambda D} \sum_{k, U_i} R_u(U_i, \Gamma_u^e) \mathbb{P}\left(|U_i| = k\right) \mathbb{P}\left(U_i \in U \middle| |U_i| = k\right). \end{aligned}$$

Note that $\mathbb{P}\left(|U_i| = k\right) = \frac{(\lambda D)^k e^{-\lambda D}}{k!}, \mathbb{P}\left(U_i \in U \middle| |U_i| = k\right) = \frac{1}{N^k}$, hence,

$$\begin{aligned} R^e(D, \Gamma_u^e) &= \sum_{k=0}^{\infty} \frac{(\lambda D)^{k-1} e^{-\lambda D}}{k!} \sum_{|U_i|=k} \frac{R_u(U_i, \Gamma_u^e)}{N^k} \\ &= \sum_{k=0}^{\infty} \frac{(\lambda D)^{k-1} e^{-\lambda D}}{k!} \overline{R^e}(k, \Gamma_u). \end{aligned}$$

The $\overline{R^e}(k, \Gamma_u^e)$ denotes the average traffic under scheme Γ_u^e and k requests, which can be calculated by the following way.

Since each user prefetches first M contents, the average number of users should be transmitted by the share link is actually $k(1 - M/N)$. We define $(N - M)$ binary random variable $X_i = 0, 1, M + 1 \leq i \leq N$, which equals to 1 if and only if the content i is requested by at least one user. Then,

$$\begin{aligned} \overline{R^e}(k, \Gamma_u) &= E\left[\sum_{i=M+1}^{N} X_i\right] = \sum_{i=M+1}^{N} E\left[X_i\right] \\ &= (N - M)\left(1 - \left(1 - \frac{1}{N - M}\right)^{k\left(1 - \frac{M}{N}\right)}\right). \end{aligned} \tag{28}$$

Thus,

$$R^e(D, \Gamma_u) = \frac{N - M}{\lambda D}\left\{1 - e^{\lambda D\left[\left(1 - \frac{1}{N-M}\right)^{1 - \frac{M}{N}} - 1\right]}\right\}. \tag{29}$$

Under the condition (3), we can get

$$R^e(D, \Gamma_u) = \frac{N - M}{\lambda} \Theta\left(\frac{1}{D}\right). \tag{30}$$

Hence it is clear that the assumption on D in (3) ensures that the average traffic is not dominated by the scaling behaviour of D.

B. Proof of Theorem 2

We take a same procedure in the proof of Theorem 1, we can get

$$R^e(D, \Gamma_c^e) = \sum_{k=0}^{\infty} \frac{(\lambda D)^{k-1} e^{-\lambda D}}{k!} \overline{R}(k, \Gamma_c^e). \tag{31}$$

Based on the results of [8], we can get

$$\overline{R^e}(k, \Gamma_c^e) = \frac{N - M}{M}\left[1 - \left(1 - \frac{M}{N}\right)^k\right]. \tag{32}$$

Hence,

$$R^e(D, \Gamma_c^e) = \frac{N - M}{\lambda M} \Theta\left(\frac{1}{D}\right). \tag{33}$$

C. Proof of Theorem 3

We consider a same procedure in the proof of Theorem 1, then,

$$R_{lb}^e(D) = \sum_{k=0}^{\infty} \frac{(\lambda D)^{k-1} e^{-\lambda D}}{k!} \overline{R_{lb}^e}(k). \tag{34}$$

$\overline{R_{lb}^e}(k)$ denotes the lower bound of average traffic under k requests. Based on the results of [7], we can get

$$\overline{R_{lb}^e}(k) \geq \max_{s\in\{1,\ldots,\min\{N,k\}\}} \left(s - \frac{s}{\lfloor N/s \rfloor} M\right). \tag{35}$$

This results is based on a cut-set bound argument. Further, we can get

$$\overline{R_{lb}^e}(k) \geq \max_{s\in\{1,\ldots,\min\{N,k\}\}} s\left(1 - \frac{sM}{N - s}\right). \tag{36}$$

We choose $s = cM/N$ that $s \in \{1, \ldots, \min\{N, k\}\}$ and then,

$$\overline{R_{lb}^e}(k) \geq \frac{cM - c^2 - c^2 M}{M - c} \cdot \frac{N}{M}. \tag{37}$$

Then,

$$R_{lb}^e(D) \geq \frac{cM - c^2 - c^2 M}{M - c} \cdot \frac{N}{M} \cdot \frac{1}{\lambda D}. \tag{38}$$

Hence,

$$R_{lb}^e(D) = \Omega\left(\frac{1}{D}\right). \tag{39}$$

APPENDIX B

A. The Proof of Theorem 4

We consider a same procedure in the proof of Theorem 1, then,

$$R^d(D,\Gamma_u^d)=\sum_{k=0}^{\infty}\frac{(\lambda D)^{k-1}e^{-\lambda D}}{k!}\overline{R}(k,\Gamma_u^d). \tag{40}$$

Assume that $p_1 \le p_2 \le \cdots p_N$, the average number of users should be transmitted by the shared link is actually $k\left(1-\sum_{i=M+1}^{N}p_i\right)$. We define $N-M$ binary random variable $X_i=0,1, 1\le i\le M$, which equals to 1 if and only if the content i is requested by at least one user. Then,

$$\begin{aligned}\overline{R^d}(k,\Gamma_u^d)&=E\left[\sum_{i=1}^{M}X_i\right]=\sum_{i=1}^{M}E\left[X_i\right]\\&=(N-M)\left(1-\frac{1}{N-M}\sum_{i=1}^{M}(1-p_i')^{1-S_M}\right),\end{aligned} \tag{41}$$

where $p_i'=p_i/\left(1-S_M\right)$ and $S_M=\sum_{i=M+1}^{N}p_i$, then,

$$R^d(D,\Gamma_u^d)=\frac{N-M}{\lambda D}\left\{1-\sum\frac{e^{\lambda D\left[\left(1-p_i'\right)^{1-S_M}-1\right]}}{N-M}\right\}. \tag{42}$$

Under the condition (6), we can get

$$R^d(D,\Gamma_u^d)=\frac{N-M}{\lambda M}\Theta\left(\frac{1}{D}\right). \tag{43}$$

B. The Proof of Lemma 1

We take two steps to prove this lemma: we first derive the $R_{\vec{s}}(Q,\Gamma_c^d)$ by a specific request vector that satisfying $\vec{s}=(\alpha_1,\alpha_2,\ldots,\alpha_N)$; we then prove that any request vector satisfying $\vec{s}$ produce the same traffic.

Firstly, consider a request vector $(d_1,d_2,\ldots,d_k)$. Consider a particular bit in one content, termed as content i. Since the prefetching is uniform, this bit has probability $p=C_{q_iMF}^1/C_F^1=q_iM$ of being prefetched in the cache of any fixed user. For any fixed subset of t out of k users, the probability that this bit is prefetched at exactly those t users is

$$(q_iM)^t(1-q_iM)^{k-t}. \tag{44}$$

Hence, the average number of bits of content i that are cached at exactly those t users is

$$F(q_iM)^t(1-q_iM)^{k-t}. \tag{45}$$

Since $|U/\{k\}|=s-1$, the expected size of $U_{k,S/\{k\}}$ is

$$F(q_iM)^{s-1}(1-q_iM)^{k-s+1}\pm o(F),$$

with high probability. For simplicity, the $o(F)$ term is ignored in the following derivation. Thus, the traffic $R_{\vec{s}}(Q,\Gamma_c^d)$ is

$$\sum_{i=1}^{k}\sum_{v\subset[k],|v|=i}\max_{j\in v}\{(q_{d_j}M)^{i-1}(1-q_{d_j}M)^{k-i+1}\}. \tag{46}$$

Secondly, we prove that any request vector satisfying $\vec{s}$ will produce the same traffic volume. We consider two requested vectors:

$$U_i=(d_1,\ldots,d_m,\ldots,d_n,\ldots,d_k),$$

$$U_i^{'}=(d_1,,\ldots,d_n^*=d_m,\ldots,d_m^*=d_n,\ldots,d_k).$$

Remark that these two requested vectors satisfying request situation: $\vec{s}=(\alpha_1,\alpha_2,\ldots,\alpha_N)$ and the difference between them is that request m and request n exchanges their requested contents. Factually, the different request vectors satisfying the same request situation $\vec{s}$, can be converted to each other via finite exchange. Then, we will show that, under these two requested vectors, the traffic is equal.

The equation (47) refers to the traffic under $U_i=(d_1,\ldots,d_m,\ldots,d_n,\ldots,d_k)$ and the equation (48) refers to the traffic under $U_i'=(d_1,\ldots,d_m^*,\ldots,d_n^*,\ldots,d_k)$.

Consider $d_m^*=d_n$ and $d_n^*=d_m$, we can get

$$\begin{aligned}&R_m(\vec{s},Q)=R_n^*(\vec{s},Q),\\&R_n(\vec{s},Q)=R_m^*(\vec{s},Q),\\&R_{m,n}^*(\vec{s},Q)=R_{n,m}(\vec{s},Q)=R_{m,n}^*(\vec{s},Q),\\&R_{\emptyset}(\vec{s},Q)=R_{\emptyset}^*(\vec{s},Q).\end{aligned}$$

Then, $R_{\vec{s}}(Q,\Gamma_c^d)=R_{\vec{s}}^*(Q,\Gamma_c^d)$.

Hence, the request situation can be divided by the $\vec{s}$, and the average traffic is

$$\overline{R^d}(k,Q,\Gamma_c^d)=\sum_{\vec{s}}\mathbb{P}(\vec{s})\cdot R_{\vec{s}}(Q,\Gamma_c^d). \tag{49}$$

C. The Proof of Theorem 5

We first prove that $\sum_{i=1}^{N}p_ie^{\lambda DMq_i}$ is a convex function over Q. The Hessian matrix is

$$(\lambda DM)^2\begin{pmatrix}p_1e^{\lambda DMq_1}&\cdots&0\\\vdots&\ddots&0\\0&\cdots&p_Ne^{\lambda DMq_N}\end{pmatrix}\succ 0.$$

It is strictly positive definite over Q. Then we define a generalized Lagrangian function

$$\begin{aligned}\mathcal{L}(Q,\sigma,\tau,\kappa)=&\sum_{i=1}^{N}p_ie^{\lambda DMq_i}+\sum_{i=1}^{N}\sigma_i(q_i-\frac{1}{M})\\&-\sum_{i=1}^{N}\tau_iq_i+\kappa(\sum_{i=1}^{N}q_i-1).\end{aligned}$$

Since the objective function is convex, we can get the optimal solution $Q^{\dagger}$ by the **Karush-Kuhn-Tucker(KKT)** con-

$$R_{\vec{s}}(Q,\Gamma_c^d)=\sum_{i=1}^{K}R_m(\vec{s},Q)+R_n(\vec{s},Q)+R_{m,n}(\vec{s},Q)+R_{\emptyset}(\vec{s},Q), \tag{47}$$

where

$$\begin{aligned}
R_m(\vec{s},Q)&=\sum_{v\subset[k],|v|=i,n\in v,m\notin v}\max\{(q_{d_m}M)^{i-1}(1-q_{d_m}M)^{k-i+1},\max_{j\in v/m}\{(q_{d_j}M)^{i-1}(1-q_{d_j}M)^{k-i+1}\}\},\\
R_n(\vec{s},Q)&=\sum_{v\subset[k],|v|=i,n\notin v,m\in v}\max\{(q_{d_n}M)^{i-1}(1-q_{d_n}M)^{k-i+1},\max_{j\in v/n}\{(q_{d_j}M)^{i-1}(1-q_{d_j}M)^{k-i+1}\}\},\\
R_{m,n}(\vec{s},Q)&=\sum_{v\subset[k],|v|=i,n\notin v,m\in v}\max\left\{\begin{array}{c}(q_{d_m}M)^{i-1}(1-q_{d_m}M)^{k-i+1},(q_{d_n}M)^{i-1}(1-q_{d_n}M)^{k-i+1},\\ \max\limits_{j\in v/n}\{(q_{d_j}M)^{i-1}(1-q_{d_j}M)^{k-i+1}\}\end{array}\right\},\\
R_{\emptyset}(\vec{s},Q)&=\sum_{v\subset[k],|v|=i}\max_{j\in v/\{m,n\}}\{(q_{d_j}M)^{i-1}(1-q_{d_j}M)^{k-i+1}\}.
\end{aligned}$$

$$R^*_{\vec{s}}(Q,\Gamma_c^d)=\sum_{i=1}^{K}R^*_m(\vec{s},Q)+R^*_n(\vec{s},Q)+R^*_{m,n}(\vec{s},Q)+R^*_{\emptyset}(\vec{s},Q) \tag{48}$$

where

$$\begin{aligned}
R^*_m(\vec{s},Q)&=\sum_{v\subset[K],|v|=i,n\in v,m\notin v}\max\{(q_{d_m}M)^{i-1}(1-q_{d_m}M)^{K-i+1},\max_{j\in v/m}\{(q_{d_j}M)^{i-1}(1-q_{d_j}M)^{K-i+1}\}\},\\
R^*_n(\vec{s},Q)&=\sum_{v\subset[K],|v|=i,n\notin v,m\in v}\max\{(q_{d_n}M)^{i-1}(1-q_{d_n}M)^{K-i+1},\max_{j\in v/n}\{(q_{d_j}M)^{i-1}(1-q_{d_j}M)^{K-i+1}\}\},\\
R^*_{m,n}(\vec{s},Q)&=\sum_{v\subset[K],|v|=i,n\notin v,m\in v}\max\left\{\begin{array}{c}(q_{d_m}M)^{i-1}(1-q_{d_m}M)^{K-i+1},(q_{d_n}M)^{i-1}(1-q_{d_n}M)^{K-i+1},\\ \max\limits_{j\in v/n}\{(q_{d_j}M)^{i-1}(1-q_{d_j}M)^{K-i+1}\}\end{array}\right\},\\
R^*_{\emptyset}(\vec{s},Q)&=\sum_{v\subset[K],|v|=i}\max_{j\in v/\{m,n\}}\{(q_{d_j}M)^{i-1}(1-q_{d_j}M)^{K-i+1}\}.
\end{aligned}$$

ditions as follows.

$$\begin{aligned}
\frac{\partial}{\partial q_i^\dagger}\mathcal{L}(Q^\dagger,\sigma^*,\tau^*,\kappa^*)=0,i=1,\dots,N,\\
\frac{\partial}{\partial\lambda^*}\mathcal{L}(Q^\dagger,\sigma^*,\tau^*,\kappa^*)=0,\\
\sigma_i^*(q_i^\dagger-\frac{1}{M})=0,i=1,\dots,N,\\
q_i^\dagger\le\frac{1}{M},i=1,\dots,N,\\
\sigma_i^*\ge0,i=1,\dots,N,\\
\tau_i^*q_i^\dagger=0,i=1,\dots,N,\\
q_i^\dagger\ge0,i=1,\dots,N,\\
\tau_i^*\le0,i=1,\dots,N,
\end{aligned}$$

$Q^\dagger$ can be derived by the constraints above.

D. The Proof of Theorem 6

Due to the difficulty to get a accurate traffic by (12), we first upper bound it,

$$\overline{R^d}(k,Q,\Gamma_c^d)\le\sum_{i=1}^{N}\mathbb{P}(A_i)\frac{1-q_iM}{q_iM}\left[1-(1-q_iM)^k\right], \tag{50}$$

equal if only if $p_1=p_2=\cdots=p_N$ and $q_1=q_2=\cdots=q_N$. Where $\mathbb{P}(A_i)$ represents the probability that K users request content $i,i+1,\dots,N$, and can be calculated as

$$\mathbb{P}(A_i)=(1-\sum_{j=1}^{i-1}p_j)^k\cdot[1-(1-\frac{p_i}{1-\sum_{j=1}^{i-1}p_j})^K]. \tag{51}$$

We prove this inequality by the following scaling method.

All request situation is divided into following N cases: $A_i:\alpha_j=0,j<i,\alpha_j>0,j\ge i$. In the case A_i, each user only request one of contents $i,i+1,\dots,N$ and their corresponding caching distribution satisfies $q_i\le q_{i+1}\le\cdots\le q_N$. Let $q_i,q_{i+1},\cdots,q_N\leftarrow q_i$. Then, the caching distribution of content $i+1,i+2,\dots,N$ are reduced to q_i. Thus, the traffic rate under this case is

$$\sum_{s=1}^{K}C_K^s(q_iM)^{s-1}(1-q_iM)^{k-s+1} \tag{52}$$

$$=\frac{1-q_iM}{q_iM}\left(1-(1-q_iM)^k\right). \tag{53}$$

The probability of case A_i is calculated based on multipli-

cation formula. Then,

$$\overline{R^d}(k,Q,\Gamma_c^d) \le \sum_{i=1}^{N} \mathbb{P}(A_i)\frac{1-q_iM}{q_iM}\left[1-(1-q_iM)^k\right]. \tag{54}$$

Thus,

$$R^d(D,\Gamma_c^d) = \sum_{k=0}^{\infty}\frac{(\lambda D)^{k-1}e^{-\lambda D}}{k!}\overline{R}(k,Q,\Gamma_c^d) \tag{55}$$

$$\le \sum_{k=0}^{\infty}\frac{(\lambda D)^{k-1}e^{-\lambda D}}{k!}\sum_{i=1}^{N}\mathbb{P}(A_i)\frac{1-q_iM}{q_iM}\left[1-(1-q_iM)^k\right] \tag{56}$$

$$= \frac{1}{\lambda D}\sum_{i=1}^{N}\mathbb{P}(A_i)\frac{1-q_iM}{q_iM}\left(1-e^{-\lambda D M q_i}\right). \tag{57}$$

Considering the relaxed solution $Q^\dagger$, D is large enough and condition (6), we can get

$$R^d(D,\Gamma_c^d) \le \frac{N-1}{\lambda D}\sum_{i=1}^{N}\mathbb{P}(A_i)\left[1-\frac{\left(\prod_{j=1}^{N}p_j\right)^{\frac{1}{N}}}{p_i}\right] \tag{58}$$

$$\le \frac{N-1}{\lambda D}\left[1-\left(\prod_{j=1}^{N}p_j\right)^{\frac{1}{N}}\sum_{j=1}^{N}\frac{\mathbb{P}(A_i)}{p_i}\right]. \tag{59}$$

Considering the affect that

$$\sum_{i=1}^{N}\frac{\mathbb{P}(A_i)}{p_i} \ge \frac{\sum_{i=1}^{N}\mathbb{P}(A_i)}{\sum_{i=1}^{N}p_i} \ge 1, \tag{60}$$

Hence,

$$R = \frac{(N-1)\left(1-\left(\prod_{i=1}^{N}p_i\right)^{\frac{1}{N}}\right)}{\lambda}O\left(\frac{1}{D}\right). \tag{61}$$

E. The Proof of Theorem 7

We consider a same procedure in the proof of Theorem 7, then,

$$R_{lb}^d(D) = \sum_{k=0}^{\infty}\frac{(\lambda D)^{k-1}e^{-\lambda D}}{k!}\overline{R_{lb}}(k,P). \tag{62}$$

$\overline{R_{lb}^d}(k,P)$ denotes the lower bound of average traffic under k requests and content popularity distribution P.

Since the size of each content is identical, all request situation is divided into N cases:

$$\begin{aligned}
&B_1 : \alpha_i > 0, i = k_1, \alpha_i = 0, i \ne k_1;\\
&B_2 : \alpha_i > 0, i = k_1, k_2, \alpha_i = 0, i \ne k_1, k_2;\\
&\cdots\cdots;\\
&B_N : \alpha_i > 0, i \in \mathbb{K}.
\end{aligned}$$

Under the case i, there are only i contents are requested by all users. Based on the results in (36), we can get

$$\overline{R_{lb}^d}(k,P) \ge \frac{cM-c^2-c^2M}{M-c}\sum_{i=1}^{N}\mathbb{P}(B_i)\cdot\frac{i}{M} \tag{63}$$

$$= \frac{cM-c^2-c^2M}{M(M-c)}\sum_{i=1}^{N}\left(1-(1-p_i)^k\right). \tag{64}$$

Hence,

$$R_{lb}^d(D) = \sum_{k=0}^{\infty}\frac{(\lambda D)^{k-1}e^{-\lambda D}}{k!}\overline{R_{lb}^d}(k,P). \tag{65}$$

$$\ge \frac{cM-c^2-c^2M}{M(M-c)}\cdot\frac{1}{\lambda D}\cdot\left(N-\sum_{i=1}^{N}e^{-\lambda D p_i}\right) \tag{66}$$

$$= O\left(\frac{1}{D}\right). \tag{67}$$