Exploiting the Unexploited of Coded Caching for Wireless Content Distribution

Sinong Wang, Xiaohua Tian and Hui Liu
Department of Electronic Engineering, Shanghai Jiao Tong University
{snwang, xtian, huiliu}@sjtu.edu.cn

***Abstract*—Recent studies show that the coded caching technique can facilitate the wireless content distribution by mitigating the wireless traffic volume during the peak-traffic time, where the contents are partially prefetched to the local cache of mobile devices during the off-peak time. The remaining contents are then jointly coded and delivered in multicast, when many content requests are initiated in the peak-traffic time. The requested contents can be recovered from the local-prefetched and multicast data with requesters experiencing less congestions. However, the benefit of the coded caching scheme is still under estimated, where the potential gain by appropriate caching distribution is under exploited. In this paper, we propose a theoretical model to minimize the average wireless traffic volume required in the coded caching, for which the optimized caching distribution is derived with the content popularity distribution taken into account. In order to improve the computational efficiency for determining the appropriate caching distribution, we transform the objective function from the average wireless traffic volume into the average size of un-prefetched contents. We theoretically show the order optimality of the results derived from our model. Numerical results show that the coded caching performance can be further improved with our caching distribution design.**

I. Introduction

The proliferation of mobile devices spurs the dramatic increase in mobile traffic volume over wireless networks. It is predicted that the mobile wireless traffic volume beyond 2020 will be 1000 times higher than that in 2010 [1], where media contents constitute the lion's share. The contradiction of increasing demand of high quality media contents and limited wireless transmission capabilities drives the wireless community to develop techniques in dimensions rather than just expanding the infrastructure to deal with the "mobile data tsunami".

The wireless network such as the cellular network presents a high temporal variability in network traffic volume, where the network resources are extremely scarce in certain time periods and comparatively idle in others. The caching technique is proposed to balance the traffic load over the wireless link: the popular content is partially placed to mobile devices during the off-peak time and the rest of the content data could be delivered to devices upon requests, with requesters experiencing less congestions. Efforts have been devoted to study how to improve the *local caching gain* of such a strategy by smartly design the placement or the delivery mechanism [2]- [5].

However, the local caching gain is hardly enough for today's wireless content distribution. This is because the gain is straightforwardly dependent on the cache size at mobile devices, in particular, the proportion of the total content can be cached at mobile devices' memories. As the media content rapidly grows in quality and size, the local caching gain can be increasingly insignificant. In order to further exploit the potential of memory resources, the *coded caching* scheme is proposed to achieve a *global caching gain*, where the placement and delivery phases are jointly optimized [6], [7]. The major feature distinguishes the coded caching from its uncoded counterpart is that the data delivered over the wireless link are coded data over multiple contents, so that multiple requests for even different contents can be satisfied with a single coded transmission.

It is theoretically proved that the coded caching scheme can reduce the wireless traffic volume more significantly than the uncoded version; however, it is also proved that the required delivery-phase traffic volume over wireless can not be less than a lower bound in the sense of information theory [6], [7]. In order for the coded caching scheme to achieve as near to the theoretical lower bound as possible in the practical engineering, it is vitally important to appropriately choose the design parameters.

Since the mobile users' requests are normally distributed in a non-uniform manner, the content popularity distribution is critical in determining how the memory space of each device should be allocated for contents placement, which is termed as the *caching distribution*. Niesen et al. propose to categorize contents into groups based on the content popularity and perform coded caching presented in [8] on each group separately [9], which may miss some coding opportunities over contents in different groups. Ji et al. propose to only place a few most popular contents in the devices' memories [10]; however, the placement and delivery phases are separately designed in a centralized manner, which may leave some potential of joint optimization unexploited and incur implementation difficulties. The example to be illustrated in the following section of the paper will specifically show that the existing caching distribution scheme designs leave potentials of coded caching scheme unexploited.

In this paper, we present a theoretical model to optimize the caching distribution for coded caching, in order to minimize the average wireless traffic volume in the delivery phase. With the optimization realized in a decentralized manner, the possible control messaging overhead or privacy leakage for coordination among different devices could be avoided.

To improve the computational efficiency for determining the appropriate caching distribution, we propose a relaxed optimization model, which transforms the objective function from the average wireless traffic volume into the average size of non-prefetched contents.

Moreover, we present a theoretical analysis on our proposed primal model and the relaxed one, where the coded caching performance upper bound and lower bound are derived. We show that the existing lower bound of the average wireless traffic volume can be further lowered if the content popularity is taken into account. We also show that the performance of the coded caching scheme adopting the derived optimal caching distribution from our theoretical model can be tightly bounded, where the multiplicative gap to the lower bound is finite. Numerical results show that the coded caching scheme based on the derived caching distribution from our theoretical model outperforms those based on other known caching distributions.

The remainder of the paper is organized as follows. In Section II, we present the service model and design challenges. In Section III, we formulate the coded caching problem by providing the coded caching optimization model and the relaxed model. In Section IV, we present our theoretical analysis on the proposed models. Numerical results and concluding remarks are given in Section VI and VII, respectively. Due to the limitation of the space, we just provide outlines of some theoretical proofs in this paper, and all details can be found in our technical report [11].

II. Service Model and Design Challenges

A. Service Model

1) Network Architecture: The coded caching network architecture studied in this paper is illustrated in Fig. 1 and main notations are listed in Table I. There are N contents that could be requested by K users, where each user has a local cache space of size MF bits. We assume that contents are with various levels of popularity that is measured by the probability the content will be requested but with the same size F bits. The same-size-content assumption is for the convenience of analysis, which however does not hinder the practicability of the coded caching in the real world, because the main body of content objects can be tailored as the same size for coded caching based distribution and the rest in a small quantity can be distributed in the traditional way. Contents are distributed through an error-free shared link to users such as in the long term evolution (LTE) broadcasting system [12], where the error-free can be achieved with error correction scheme or reliable transmission scheme in the upper layer.

The coded caching scheme in such a scenario operates as follows. In the placement phase, part of each content is prefetched in each user's local cache. How much the cache space is allocated for each content depends on the caching distribution $Q = [q_1, q_2, ...q_N]$, where q_i is the proportion of cache space allocated for content i. 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 for the content according to a popularity distribution $P = [p_1, p_2, ..., p_N]$, where p_i is the probability that a user is requesting content i. After collecting and analyzing these requests, the content server transmits the coded data over different contents through the shared wireless link. Each user k recovers the requested content V_{d_k} using both the local cached data and the coded data just received over wireless. Since the delivery phase usually happens in the time when everybody would like to request some contents, the crux of the coded caching is to harness the peak-time traffic in the delivery phase.

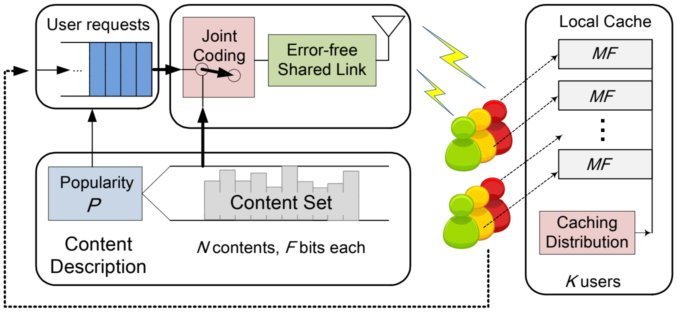


Fig. 1. Network architecture of coded caching.

TABLE I
MAIN NOTATIONS

N	The number of contents to be distributed.
K	The number of users in the system.
F	The size of each content.
V_i	Content i.
Z_k	User k's local cache.
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.
Γ	The codec scheme for the coded caching.
R	$R(P, Q, \Gamma)$, the average traffic volume under popularity distribution P, caching distribution Q and codec scheme Γ.
d_k	The index of the content requested by user k.
$\vec{s}$	$\vec{s} = [\alpha_1, \alpha_2, ...\alpha_N]$, the request situation, where users request α_1 content 1, α_2 content 2,..., and α_N content N.
S	All possible request situations.
P_s	$P_s(\vec{s}, P)$, the probability that situation $\vec{s}$ occurs under the popularity distribution P.
R_s	$R_s(\vec{s}, Q, \Gamma)$, the average traffic volume of situation $\vec{s}$ under caching distribution Q and codec scheme Γ.
v	The Zipf distribution parameter.

2) Coded Caching Operations: The coded caching scheme can be determined by a two-tuple (Q, Γ) [6], [7], where Q is the caching distribution and Γ denotes the codec (encoding-and-decoding) scheme. A coded caching is achievable if the error incurred in the local data recovery process is small enough. Algorithm 1 briefly describes how the coded caching scheme operates, where the codec scheme that has been proved achievable are utilized [6], [7].

The placement phase is uncoded, where each user randomly caches segments of each content. If users' memories are filled independently of each other and just follow the caching distribution in the placement phase, we say the coded caching

Algorithm 1: Decentralized coded caching scheme (Q, Γ) with nonuniform demands

```
Placement Phase
Determine $Q = [q_1, q_2, ..., q_n]$;
for $(k = 0; k < K; k++)$ do
    for $(n = 0; n < N; n++)$ do
        user $k$ randomly prefetches $q_n MF$ 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
        $X_U \leftarrow \oplus_{k \in U} V_{k,U/\{k\}}$ ;
        Multicast the coded data $X_U$ to users in $U$.
```

scheme is *decentralized*, which is the category we will study in the rest of the paper. As shown in Algorithm 1, while each user prefetches the same amount of $q_i MF$ bits for content i, they may cache different segments of the content, thus providing coding opportunities in the delivery phase.

In the delivery phase, $V_{k,U/\{k\}}$ denotes the segments of the content requested by user k but cached in other users' memories, where $U/\{k\}$ denotes the users in the subset U excluding user k. The operator $\oplus$ means the bit-wise XOR operation. This phase requires that the content server traverses all subsets of users and performs XOR operations on all the uncached parts of these users' requested contents, where each element of $V_{k,U/\{k\}}$ is required to be zero padded to the length of the longest element.

B. Design Challenges

In order to minimize the delivery-phase traffic, it is critical to determine which contents and how much of each content should be prefetched to mobile devices in the placement phase. A natural intuition is that the optimal caching distribution Q should follow the content popularity P. However, Example 1 below shows that the influence of P to Q is not so straightforward as intuitively imagined.

Example 1: Suppose that there is a simple system distributing $N = 3$ contents A, B and C with $P = [0.6, 0.3, 0.1]$ to $K = 3$ users. We want to find the average traffic load incurred with different caching distributions: $Q_1 = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$, $Q_2 = [0.5, 0.5, 0]$, $Q_3 = [0.6, 0.3, 0.1]$ and $Q_4 = [0.6, 0.4, 0]$. We use the caching distribution Q_3 to illustrate how to calculate the average delivery-phase traffic volume. In the placement phase, each user caches a subset of $0.6F, 0.3F$ and $0.1F$ bits of contents A, B and C randomly, which means that each bit of contents A, B and C is cached by any user with probability $0.6, 0.3$ and 0.1, respectively.

For content A, the placement phase divides content A into 8 sub-contents: $A = \{A_\varnothing, A_1, A_2, A_3, A_{1,2}, A_{1,3}, A_{2,3}, A_{1,2,3}\}$, where A_U denotes the bits of content A that are prefetched in the cache memories of users in $U \subset \{1, 2, 3\}$. We use $|\cdot|$ to denote the size of the sub-contents, where

$$|A_\varnothing| = (1 - 0.6)^3 F = 0.064F \text{ bits},$$
$$|A_1| = |A_2| = |A_3| = (1 - 0.6)^2 \cdot 0.6F = 0.096F \text{ bits},$$
$$|A_{1,2}| = |A_{1,3}| = |A_{2,3}| = (1 - 0.6) \cdot 0.6^2 F = 0.144F \text{ bits},$$
$$|A_{1,2,3}| = 0.6^3 F = 0.216F \text{ bits}.$$

Based on the same procedure, the segments of content B and C can be determined, respectively.

We now consider the delivery phase and assume that $d_1 = 1, d_2 = 2, d_3 = 3$. As a result, the content server will transmit the following 7 signals as shown in TABLE II according to Algorithm 1. Note that we omit symbols representing set to avoid the lengthy presentation, for example, $V_{1,\{2,3\}}$ is simplified as $V_{1,23}$.

TABLE II
THE CODED DATA AND TRAFFIC INCURRED

X_U	$\oplus_{k \in U} V_{k,U/\{k\}}$	Traffic (F bits)
$V_{1,23} \oplus V_{2,13} \oplus V_{3,12}$	$A_{23} \oplus B_{13} \oplus C_{12}$	0.144
$V_{1,2} \oplus V_{2,1}$	$A_2 \oplus B_1$	0.147
$V_{1,3} \oplus V_{3,1}$	$A_3 \oplus C_1$	0.081
$V_{2,3} \oplus V_{3,2}$	$B_3 \oplus C_2$	0.147
$V_{1,\varnothing}$	$A_\varnothing$	0.064
$V_{2,\varnothing}$	$B_\varnothing$	0.343
$V_{3,\varnothing}$	$C_\varnothing$	0.729

From the multicast coded data, user 1, 2 and 3 can recover all the requested contents A, B and C, with the incurred wireless traffic volume of $1.655F$ bits by summing over those numbers in the third column of TABLE II. This event happens when all contents have been requested, thus the probability of the event is $p_1 \cdot p_2 \cdot p_3 = 0.018$. Since there are 3 users and each user can request a single content, we will have 27 requesting situations in total, where the incurred traffic volume and the probability of each event can be calculated as above. We are able to obtain that the average traffic volume over all 27 cases is $1.27F$ bits with Q_3. Similarly, we can find that the average traffic volumes with Q_1, Q_2 and Q_4 are $1.47F$, $1.35F$ and $1.05F$ bits, respectively.

The caching distribution without considering the content popularity such as Q_1 is obviously the worst; however, configuring the cache distribution directly according to the popularity distribution such as Q_3 does not necessarily yield the best result, which is a quite counter-intuitive finding. The design such as Q_2 is proposed in [10], which only caches some most popular contents with equal probability. Although Q_2 reduces the traffic to some extent, we can still find a smarter distribution such as Q_4 that outperforms Q_2. This simple example reveals that the performance of the coded caching can be further improved with a better caching distribution; however, how to find a theoretical optimal caching distribution is still an open issue.

III. OPTIMIZATION MODEL

In this section, we develop a joint optimization model to find the optimal caching distribution, with influence of all important factors P, Q and Γ taken in to account.

A. Primal Model

$$\textbf{OPT:}\quad \text{Min}\quad R(P,Q,\Gamma)=\sum_{\vec{s}\in S} P_s(\vec{s},P)\cdot R_s(\vec{s},Q,\Gamma) \quad (1)$$

$$\text{s.t.}\quad \sum_{i=1}^{N} q_i = 1, 0\le q_i \le \frac{1}{M}, \quad (2)$$

where (2) is to avoid violating caching capacity constraints and caching redundant segments for each content. $P_s(\vec{s},P)$ is the probability of request situation $\vec{s}$ under the general content popularity distribution P. Consider that α_1 users request content 1, α_2 users request content 2, . . ., and α_N users request content N, the number of all possible cases for such a request situation is $C_K^{\alpha_1}\cdot C_{K-\alpha_1}^{\alpha_2}\cdot C_{K-\alpha_1-\alpha_2}^{\alpha_3}\cdots C_{K-\alpha_1-\cdots-\alpha_{N-1}}^{\alpha_N}$. The probabilities of these cases are the same, thus we have

$$\begin{aligned}P_s(\vec{s},P) &= C_K^{\alpha_1}\cdot C_{K-\alpha_1}^{\alpha_2}\cdot\cdots\cdot C_{K-\alpha_1-\cdots-\alpha_{N-1}}^{\alpha_N}\cdot\prod_{i=1}^{N} p_i^{\alpha_i}\\ &= \frac{K!}{\alpha_1!\cdot\alpha_2!\cdot\cdots\cdot\alpha_N!}\cdot\prod_{i=1}^{N} p_i^{\alpha_i}.\end{aligned}$$

$R_s(\vec{s},Q,\Gamma)$ denotes the traffic volume incurred by the situation $\vec{s}$, which is a function of caching distribution Q, request situation $\vec{s}$ and the specific codec scheme Γ.

Lemma 1: Given the request vector $[d_1,d_2,\ldots,d_K]$ and the coded caching scheme (Q,Γ), the incurred traffic volume is

$$\sum_{i=1}^{K}\sum_{v\subset\{1,2,\ldots,K\},|v|=i}\max_{j\in v}\{(q_{d_j}M)^{i-1}(1-q_{d_j}M)^{K-i+1}\} \quad (3)$$

Proof: For a particular bit in any content i, the probability for the bit to be prefetched by any given user is $p = C_{q_iMF}^1/C_F^1 = q_iM$. For any t-sized subset of K users, the probability that the bit is prefetched by those t users is $(q_iM)^t(1-q_iM)^{K-t}$. Hence, the average number of bits of content i that are cached at those t users is $F(q_iM)^t(1-q_iM)^{K-t}$. Suppose that $|U/\{k\}|=k-1$, the expected size of $U_{k,U/\{k\}}$ is

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

with high probability. The term $o(F)$ is a constant in the order of less than F, which is ignored in the following derivation for a clearer presentation. ■

Lemma 2: Any request vector $[d_1,d_2,\ldots,d_K]$ satisfying $\vec{s}=[\alpha_1,\alpha_2,\ldots,\alpha_N]$ incurs the same traffic volume as shown in (3) under the coded caching scheme (Q,Γ).

The Lemma shows that the traffic volume is only related to the number of requests for each content, and independent of who requests which contents. The strategy for proofing this Lemma is to first prove that the new request vector obtained by exchanging any two requests incurs the same traffic volume as the original vector, and then prove that any two request vectors satisfying the request situation $[\alpha_1,\alpha_2,...,\alpha_N]$ are equivalent after a finite number of such exchanges. The detailed proof can be found in [11]. With Lemma 1 and Lemma 2, we can present the coded caching traffic volume.

Theorem 1: With the given $\vec{s}$, Q and Γ, the traffic volume incurred is

$$R_s(\vec{s},Q,\Gamma)=\sum_{i=1}^{K}\sum_{\substack{v\subset\{1,2,\ldots,K\}\\|v|=i}}\max_{j\in v}\{(q_{d_j}M)^{i-1}(1-q_{d_j}M)^{k-i+1}\}, \quad (4)$$

where $[d_1,d_2,\ldots,d_K]$ is any requested vector that satisfies situation $\vec{s}$.

The problem OPT falls in the category of nonlinear programming, where the objective function is nonlinear. With the standardized methods such as Rosen's gradient projection method [15], the optimal solution Q^* can be obtained. However, the corresponding computational complexity is $O(2^K\cdot C_{K+N-1}^{N-1})$, which increases exponentially with N and K . This is because the computational complexities for obtaining $R_s(\vec{s},Q,\Gamma)$ is $O(2^K)$ for a given $\vec{s}$ 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}.

B. Relaxed Model

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.

$$\textbf{OPT-Relax:}\quad \text{Min}\quad R_a(P,Q,\Gamma)=\sum_{i=1}^{N} p_i(1-q_iM)^K F \quad (5)$$

$$\text{s.t.}\quad \sum_{i=1}^{N} q_i=1, 0\le q_i\le\frac{1}{M} \quad (6)$$

Theorem 2: The optimal caching distribution under the OPT-Relax model is $Q^\dagger=[q_1^\dagger,q_2^\dagger,...,q_i^\dagger,...,q_N^\dagger]$, where

$$q_i^\dagger=\frac{1}{M}\left[1-(N-M)\frac{\left(\frac{1}{p_i}\right)^{\frac{1}{K-1}}}{\sum\limits_{i=1}^{N}\left(\frac{1}{p_i}\right)^{\frac{1}{K-1}}}\right], 1\le i\le N.$$

Proof: We first prove that $R_a(P,Q,\Gamma)$ is a convex function over Q. The Hessian matrix of the objective function $R_a(P,Q,\Gamma)$ is

$$K(K-1)\begin{pmatrix}(1-q_1M)^{K-2} & \cdots & 0\\ \vdots & \ddots & 0\\ 0 & \cdots & (1-q_NM)^{K-2}\end{pmatrix}\succ 0$$

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

$$\mathcal{L}(Q,\sigma,\tau,\lambda) = \sum_{i=1}^{N} p_i(1-q_iM)^K F + \sum_{i=1}^{N}\sigma_i(q_i - \frac{1}{M}) - \sum_{i=1}^{N}\tau_i q_i + \lambda(\sum_{i=1}^{N} q_i - 1).$$

Since the objective function is convex, we can get the Karush-Kuhn-Tucker(KKT) [13] conditions as follows.

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

The optimal solution under the relaxed model denoted as $Q^\dagger$ can then be derived using the general Lagrangian multiplier method [13]. ■

We take the caching distribution $Q^\dagger$ as the approximate caching distribution to Q^* derived from the OPT model. The coded caching scheme based on the approximate caching distribution is then denoted as $(Q^\dagger, \Gamma)$.

IV. Theoretical Analysis

With the theoretical model presented in the previous section, we in fact jointly optimize the performance of the coded caching with the content popularity P, the caching distribution Q and the codec scheme Γ taken into account. A natural question to ask is: what is the fundamental limit of the coded caching scheme based on our design.

Theorem 3: For N contents and K users each with cache memory size of M, where $0 \le M \le N$. If $p_1 \le p_2 \le \cdots \le p_N$ and $q_1 \le q_2 \le \cdots \le q_N$, then

$$\begin{aligned}
R(P,Q,\Gamma) &\le R^{ub}(P,Q,\Gamma)\\
&= F\sum_{i=1}^{N}\mathbb{P}(A_i)\frac{1-q_iM}{q_iM}\left[1-(1-q_iM)^k\right],
\end{aligned}$$

where the equality holds if and only if $p_1 = p_2 = \cdots = p_N$ and $q_1 = q_2 = \cdots = q_N$. We use $\mathbb{P}(A_i)$ to denote the probability that K users request content $i, i+1, \dots, N$, which is

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

Proof: All request situations S are divided into following N cases: A_i : $\alpha_j = 0$, if $j < i$; $\alpha_j > 0$, if $j \ge i$.

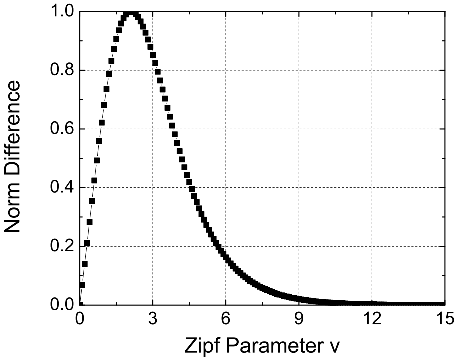


Fig. 2. The normalized difference between $R^{ub}(P_{Zipf}, Q^\dagger, \Gamma)$ and $R(P_{Zipf}, Q^\dagger, \Gamma)$ with respect to the maximal difference, with $N = 3$ and $K = 3$.

In the case A_i, each user only requests one of the contents $i, i+1, \dots, N$ and their corresponding caching distribution satisfies $q_i \le q_{i+1} \le \cdots \le q_N$. Let the caching distributions for contents with indices greater than or equal to i the same as q_i, then the caching distributions for contents $i+1, i+2, \dots, N$ are reduced to q_i. Consequently, the traffic volume in this case is increased to

$$\begin{aligned}
R_s(A_i,Q,\Gamma) &= F\sum_{k=1}^{K}C_K^k(q_iM)^{k-1}(1-q_iM)^{K-k+1}\\
&= F\frac{1-q_iM}{q_iM}\left(1-(1-q_iM)^K\right).
\end{aligned}$$

Hence, the upper bound of total traffic volume is

$$R^{ub}(P,Q,\Gamma) = F\sum_{i=1}^{N}\mathbb{P}(A_i)\frac{1-q_iM}{q_iM}\left[1-(1-q_iM)^k\right].$$

How to derive $\mathbb{P}(A_i)$ can be found in [11]. ■

Theorem 3 gives the upper bound of incurred wireless traffic volume given P, Q and Γ in the general case. In order to give an intuition, we apply the Zipf distribution P_{Zipf} [14] that has been widely adopted in modeling media content popularity, caching distribution $Q^\dagger$ derived from our relaxed model and the Γ in Algorithm 1 to the theorem, and we can find the normalized difference between the upper bound traffic and the yielded traffic with respect to the maximal difference with $N = 3$ and $K = 3$ as shown in Fig. 2. The Zipf parameter v describes how concentrate the content popularity is. With v increases, the popularity concentration level of contents increases, the relaxation error incurred by reducing the caching distributions for contents $i+1, i+2, \dots, N$ to q_i in deriving the upper bound increases thus the difference increases as shown in Fig. 2. After Zipf parameter v achieves to a certain level, the users are just interested in a few most popular contents, which makes the contents with smaller indices unlikely to be requested thus decrease the relaxation error. That is why Fig. 2 has a shape of mountain-top.

Theorem 4: For N contents and K users each with cache size M, where $0 \le M \le N$.

$$R(P,Q,\Gamma) \ge R^{lb}(P) = \sum_{i=1}^{N}\mathbb{P}(B_i)\max_{1\le k\le i,K}\left(k - \frac{k}{\lfloor i/k \rfloor}M\right),$$

where $\mathbb{P}(B_i)$ denotes the probability that K users only requests i kinds of contents, which can be derived by the generating function [11].

Proof: Since the sizes of contents are identical, all request situations are divided into N cases :

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

where $\mathbb{K}$ is the set of integers. Under case i, there are only i contents requested by all users. Consider the cuts separating V_1 and $Z_1, Z_2, \ldots, Z_k$ until $V_{\lfloor i/k \rfloor}$ and $Z_1, Z_2, \ldots, Z_k$. By the cut set bound [16],

$$\lfloor i/k \rfloor R^{lb}(B_i) + kM \geq k \lfloor i/k \rfloor.$$

Optimizing over all possible choices of k, we obtain the lower bound of case i.

$$R^{lb}(B_i) \geq \max_{k \in \{1,\ldots,\min\{i,K\}\}} \left(k - \frac{k}{\lfloor i/k \rfloor} \right).$$

Considering all cases, the average lower bound is

$$\begin{aligned}
R^{lb}(P) &= \sum_{i=1}^{N} \mathbb{P}(B_i) R^{lb}(B_i)\\
&= \sum_{i=1}^{N} \mathbb{P}(B_i) \max_{k \in \{1,\ldots,\min\{i,K\}\}} \left(k - \frac{k}{\lfloor i/k \rfloor} \right).
\end{aligned}$$

■

It is worth noting that the theoretical lower bound given in [6], [7] is oblivious to the content popularity; however, Theorem 3 indicates that the performance lower bound can be further lowered, if the popularity distribution P is taken into account. This is because the case in [6], [7] can be regarded as a special case where all possible request situations are just as in the case B_N, where there is no opportunity to avoid the wireless traffic transmission for certain contents with very low popularity.

With the theoretical upper bound and lower bound derived, we are now interested in how tight the performance of coded caching can be bounded. We want to know if the coded caching scheme based on our design is order optimal:

$$\lim_{K,N \to \infty} \frac{R(P,Q,\Gamma)}{R^{lb}(P)} \leq C. \tag{7}$$

Lemma 3: When $K = \omega(N^v)$ [1] and $v < 1$ or $K = \Theta(N^v)$ and $v < 1$, the wireless traffic incurred by $(Q^\dagger, \Gamma)$ satisfies

$$\lim_{K,N \to \infty} \frac{R^{ub}(P_{Zipf}, Q^\dagger, \Gamma)}{R^{lb}(P_{Zipf})} \leq \frac{M - c}{cM - c^2 - c^2 M} \cdot \frac{1}{1 - e^{v-1}}.$$

Lemma 4: When $K = \omega(N^v)$ and $v > 1$ or $K = \Theta(N^v)$ and $v > 1$, the wireless traffic incurred by $(Q^\dagger, \Gamma)$ satisfies

$$\lim_{K,N \to \infty} \frac{R^{ub}(P_{Zipf}, Q^\dagger, \Gamma)}{R^{lb}(P_{Zipf})} \leq \frac{M - c}{cM - c^2 - c^2 M} \cdot \frac{1}{1 - e^{\frac{1}{-\zeta(v)}}},$$

where c is a constant between 0 and 1 and $\zeta(v)$ is the Riemann function [11].

Lemma 5: When $K = O(N^v)$ and $v > 1$ or $v < 1$, the wireless traffic incurred by $(Q^\dagger, \Gamma)$ satisfies

$$\lim_{K,N \to \infty} \frac{R^{ub}(P_{Zipf}, Q^\dagger, \Gamma)}{R^{lb}(P_{Zipf})} \leq \frac{M - c}{cM - c^2 - c^2 M} \cdot \frac{1}{1 - \lambda'},$$

where λ' is a constant between 0 and 1.

The detailed proofs of the 3 Lemmas above can be found in [11].

Theorem 5. The proposed coded caching scheme (Q^*, Γ) is order optimal and

$$\lim_{K,N \to \infty} \frac{R^{ub}(P_{Zipf}, Q^*, \Gamma)}{R^{lb}(P_{Zipf})} \leq \lim_{K,N \to \infty} \frac{R^{ub}(P_{Zipf}, Q^\dagger, \Gamma)}{R^{lb}(P_{Zipf})} \leq C.$$

Proof: Since the Q^* is the optimal caching distribution to problem OPT, then

$$\lim_{K,N \to \infty} R(P_{Zipf}, Q^*, \Gamma) \leq \lim_{K,N \to \infty} R(P_{Zipf}, Q^\dagger, \Gamma).$$

According to Theorem 3, we have

$$\lim_{K,N \to \infty} R(P_{Zipf}, Q^\dagger, \Gamma) \leq \lim_{K,N \to \infty} R^{ub}(P_{Zipf}, Q^\dagger, \Gamma).$$

Hence,

$$\lim_{K,N \to \infty} \frac{R(P_{Zipf}, Q^*, \Gamma)}{R^{lb}(P_{Zipf})} \leq \lim_{K,N \to \infty} \frac{R^{ub}(P_{Zipf}, Q^\dagger, \Gamma)}{R^{lb}(P_{Zipf})} \leq C,$$

where

$$C = \frac{M - c}{cM - c^2 - c^2 M} \cdot \max\{\frac{1}{1 - e^{v-1}}, \frac{1}{1 - e^{\frac{1}{-\zeta(v)}}}, \frac{1}{1 - \lambda'}\},$$

based on Lemma 3, Lemma 4 and Lemma 5. ■

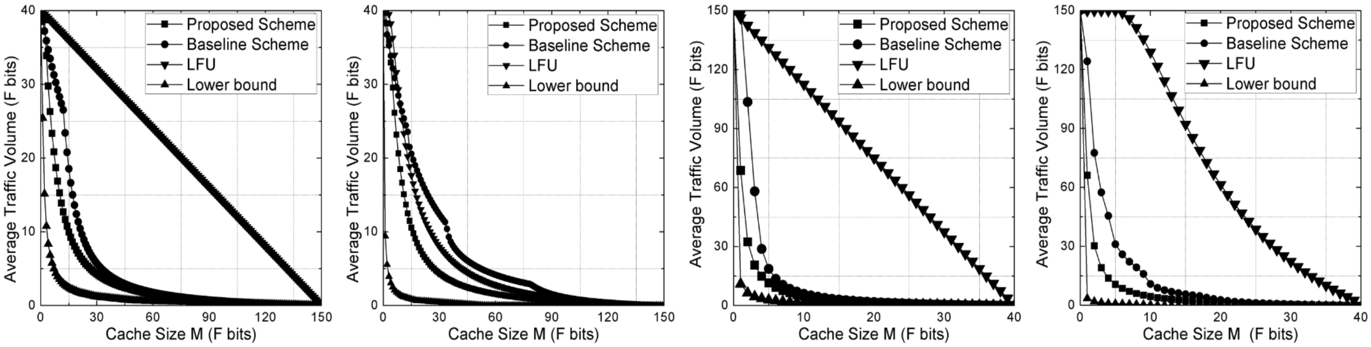
V. Numerical Results

In this section, we perform numerical simulations to examine the performance of the proposed coded caching scheme in practice. We consider two scenarios: $N = 40, K = 150$ and $N = 150, K = 40$. The content popularity is described by Zipf distribution with parameter $v = 0.5$ and 1.5, respectively. We compare the performance of the proposed coded caching scheme with the well-known least frequently used (LFU) caching scheme, and the baseline scheme proposed in [10].

LFU: In such a scheme, each user prefetches the M most popular contents in its local cache, which implies that the caching distribution Q is $q_i = 1/M$, if $1 \leq i \leq M$, and $q_i = 0$, if $M < i \leq N$.

Baseline Scheme: This scheme divides the contents set into two groups, the first of which contains the popular contents while the other contains the unpopular ones. It requires that each user prefetches contents in the first group with the same caching space allocated, and no space is allocated for contents

[1] Notation: given two functions f and g, we say that: 1) $f(n) = \omega(g(n))$ if there exists a constant c and integer N such that $f(n) \geq cg(n)$ for $n > N$. 2) $f(n) = \Theta(g(n))$ if $f(n) = \omega(g(n))$ and $g(n) = \omega(f(n))$. 3) $f(n) = O(g(n))$ if there exists a constant c and integer N such that $f(n) \leq cg(n)$ for $n > N$.



(a) N=150,K=40,v=0.5

(b) N=150,K=40,v=1.5

(c) N=40,K=150,v=0.5

(d) N=40,K=150,v=1.5

Fig. 3. Numerical results.

in the second group: $q_i = 1/\tilde{m}$, if $1 \leq i \leq \tilde{m}$ and $q_i = 0$, if $\tilde{m} < i \leq N$.

As depicted in Fig. 3, we show the worst-case wireless traffic volumes corresponding to the upper bound $R^{ub}(P, Q, \Gamma)$ in the coded caching based on different caching distribution designs, with various cache size M. We normalize the results with respect to F bits. It can be found that increasing the cache size will decrease the wireless traffic volume of the schemes above, and the traffic volumes initially decay quickly with M and then become steady. The curves representing the traffic volume incurred by LFU have two shapes: approximately linear decreasing with M under small v as shown in Fig. 3 (a) and polynomial decreasing with M under large v as shown in Fig. 3 (c).

On the one hand, the small v implies that content popularity distribution tends to be more uniform and each user will request any content arbitrarily; on the other hand, this scheme only prefetches the entire content instead of partial content and leaves no cooperative opportunity for coded multicasting in the delivery phase. Consequently, increasing cache size under the LFU is equivalent to reducing the number of contents need to be delivered over wireless thus equivalent to linearly reducing the traffic volume.

The traffic volume incurred by the baseline scheme is inversely proportional to M, approximate to our scheme under small v; while for large Zipf parameter, the baseline scheme can be very far from optimal and even worse than LFU. We can see that the proposed scheme is able to outperform both the baseline scheme and LFU under four scenarios. In particular, for the second scenario, when $M = 30$, the proposed scheme incurs a traffic volume that is 4.2 times that of the lower bound, while the baseline scheme and the LFU incurs a traffic volume that is 9.8 and 13.5 times that of the lower bound, respectively.

VI. Conclusion

This paper has presented a theoretical model to minimize the average wireless traffic volume required in the coded caching, for which the optimized caching distribution have been derived with the content popularity distribution taken into account. In order to improve the computational efficiency for determining the appropriate caching distribution, we have transformed the objective function from the average wireless traffic rate into the average size of un-prefetched contents. We have theoretically shown the order optimality of the derived results from both the primal model and the relaxed one. Numerical results have shown that the coded caching performance can be further improved with the derived caching distribution.

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. A. Maddah-Ali, U. Niesen, "Fundamental limits of caching," in *Proc. IEEE ISIT*, 2013, pp.1077–1081.

[7] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," Apr. 22, 2013, Online: http://arxiv.org/abs/1304.5856.

[8] M. A. Maddah-Ali, Niesen U, "Decentralized coded caching attains order-optimal memory-rate tradeoff," Jan. 24, 2013, Online: http://arxiv.org/abs/1301.5848.

[9] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands,"Aug. 1, 2013, Online: http://arxiv.org/abs/1308.0178.

[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] S. Wang, X. Tian and H. Liu, "Exploiting the unexploited of coded caching for wireless content distribution: Detailed theoretical proofs," Jul. 5, 2014, Online: http://iwct.sjtu.edu.cn/Personal/xtian/pdfs/IEEE%20ICNC%202015-TechReport.pdf.

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

[13] Z. Luo, W. Yu, "An introduction to convex optimization for communications and signal processing," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1426–1438, 2006.

[14] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, 1999, pp.126–134.

[15] J. B. Rosen, "The gradient projection method for nonlinear programming. Part I. Linear constraints," *Journal of the Society for Industrial & Applied Mathematics*, vol. 8, no.1, pp. 181–217, 1960.

[16] T. M. Cover and J. A. Thomas, Elements of Information Theory. Wiley, 1991.