Coded Caching in Tree-Like Network

Yuan Sun

Shanghai Jiaotong University

May 17, 2017

Abstract

Caching of popular content during off-peak hours is a strategy to reduce network loads during peak hours. Recent work has shown significant benefits of designing such caching strategies not only to locally deliver the part of the content, but also to provide coded multicasting opportunities even among users with different demands. Exploiting both of these gains was shown to be approximately optimal for caching systems with a single layer of caches. Motivated by practical scenarios, we consider, in this paper, a tree-like network with transfer stations. We apply the coded caching into this scheme and study the different performance in centralized setting and decentralized setting. In centralized setting, the transmission rate to the transfer station is calculated and compared with uncoded caching scheme, with deeper level of the system, the average gain per user is reduced. In decentralized setting, at every transfer station, dropping unneeded packets, it can still use the decentralized coded caching to delivery to the next transfer station.

I. Introduction

The demand for video streaming services, such as those offered by YouTube and Netfilx, is growing rapidly. However, conflicted with the increasing demand, the limited wireless delivery capacities drive the communication society to develop a new technique to mitigate the peak traffic. Local caching is a promising technique introduced to meet this requirement. The traffic in network, especially in content delivery networks, shows strong temporary variability, where the resources are extremely scarce in peak hours and underutilized in off-peak hour. Therefore, caching is a necessary and useful technique to balance the traffic load over the network. In the idle period, the content of the caches will be updated, and when user request the file that is cached locally, then the request will be satisfied without transmission with the base station. The gain of this effort is named as local cache gain.

In 2014, Maddah-Ali has proposed a new method, called coded caching, to further exploit the potential of the distributed network, with significantly releasing the peak traffic of networks. Generally speaking, the network operation is in two different phase: placement phase and delivery phase. In placement phase, each file in the base-station is divided into several sub-files and cached in every user. In delivery phase, the network is usually congested and some sub-files are delivered from the server to the users to reconstruct the requested files with the content of the cache. In this way, the possibility is large enough that all requests will be satisfied with a reduced transmission rate.

Most of the proposed setting only consists of a single layer between the server and the end users. The server communicates all the users directly with all the caches via a shared link, and for this basic network scenario, coded caching is shown there to be optimal within a constant factor. However, in practice, many caching systems consist of not only one but multiple layers of users equipped with a cache, usually arranged in a tree-like hierarchy with the origin server at the root node and the users as the leaves and some transfer station. Each transfer station communicates with its children users in the next layer and deliveries needed files to the next transfer station, and the objective is to minimize the transmission rates in the

Algorithm Coded Caching

procedure PLACEMENT($W_1,\dots,W_N$)
  $t \leftarrow MK/N$
  $\mathfrak{T} \leftarrow \{\mathcal{T} \subset [K] : |\mathcal{T}| = t\}$
  for $n \in [N]$ **do**
    split W_n into $\langle W_{n,\mathcal{T}} : \mathcal{T} \in \mathfrak{T}\rangle$ of equal size
  end for
  for $k \in [K]$ **do**
    $Z_k \leftarrow \{W_{n,\mathcal{T}} : n \in [N], \mathcal{T} \in \mathfrak{T}, k \in \mathcal{T}\}$
  end for
end procedure

procedure DELIVERY($W_1,\dots,W_N, d_1,\dots,d_K$)
  $t \leftarrow MK/N$
  $\mathfrak{S} \leftarrow \{\mathcal{S} \subset [K] : |\mathcal{S}| = t+1\}$
  $X_{(d_1,\dots,d_K)} \leftarrow (\oplus_{k\in\mathcal{S}} W_{d_k,\mathcal{S}\setminus\{k\}} : \mathcal{S} \in \mathfrak{S})$
end procedure

Algorithm Decentralized Coded Caching

1: **procedure** PLACEMENT
2: **for** $k \in [K], n \in [N]$ **do**
3: user k independently caches a subset of $\frac{MF}{N}$ bits of file n, chosen uniformly at random
4: **end for**
5: **end procedure**

6: **procedure** DELIVERY $(d_1,\dots,d_K)$
7: **for** $s = K, K-1,\dots,1$ **do**
8: **for** $\mathcal{S} \subset [K] : |\mathcal{S}| = s$ **do**
9: server sends $\oplus_{k\in\mathcal{S}} V_{k,\mathcal{S}\setminus\{k\}}$
10: **end for**
11: **end for**
12: **end procedure**

13: **procedure** DELIVERY'$(d_1,\dots,d_K)$
14: **for** $n \in [N]$ **do**
15: server sends enough random linear combinations of bits in file n for all users requesting it to decode
16: **end for**
17: **end procedure**

Figure 1: *centralized and decentralized coded caching*

various layers. There are several key questions when analyzing such hierarchical caching system. A first question is how to extend coded caching approach to this setting. And a second one is to analyze the gain obtained from this setting.

The remainder of the paper is organized as follows. The problem setting will be described in section2. The section3 presents our main results and discuss the obtained gain. The proof of the results is discussed in section4 and section5 discuss some follow-up results and directions for the future research.

II. Problem Setting

To simplify, first we consider a tree-like delivery network as illustrated in Fig.2. The system consists of a sever as the root of the tree hosting a collection of N=5 files each of size F bits, donated with A, B, C, D, E, and connected with a user as the transfer station. Next to the transfer station, there are four layers of users, separately containing K1=1, K2=1, K3=1 and K4=2 users. Each user is equipped with a cache of size MF=2F bits. Thus there are total 5 users of the system.

In coded caching scheme, each file is split into 10 non-over-lapping sub-files of equal size, for instance, the file A is split into A12, A13, A14, A15, A23, A24, A25, A34, A35, A45 and so do the other files. In the placement phase, the caching strategy is Z1=(A12, A13, A14, A15, B12, B13, B14, B15, C12, C13, C14, C15, D12, D13, D14, D15, E12, E13, E14, E15)and it is satisfied for the cache memory constrain. If

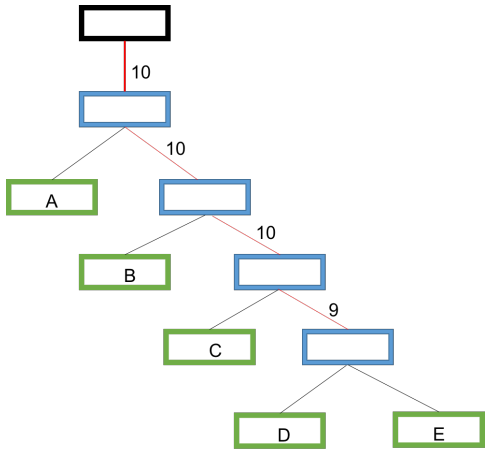

Figure 2: *The tree-like structure of coded caching scheme with N=5, K=5, M=2 and four layers.The black block means the server holding whole files; blue blocks mean transfer station without caches; green blocks mean users, each equipped with a cache.*

the users request all the files illustrated in the figure.1, firstly there are 10 files packets in total delivered from the server to the first transfer station. Then six of them are passed to the user1 and all of them are transferred to the next station. So does the second and the third layer. As regard of the forth layer, the transfer station received 9 packets, which is reduced one packet for the tree structure, and each user received 6 packets to construct their requested file. During the whole process, the total transmission rate can be calculated as 69/10, compared with the uncoded caching scheme of 114/10, reduced by 9/2.

III. Main Results

i. Centralized Setting

From above example, it is concluded that the transmission rate can be reduced due to the tree-like system. In terms of the gain obtained from the whole system, it is interesting that for the principle of coded caching, there are actually no gains in the channel between the transfer station and the users. It is gained from the channels between the transfer station. So

the key point is to figure out the transmission rate of these channels. As we can see form the Fig.1, with increasing the level, more users are separated from the remained delivery phase. When the number of these users are larger than or equal to t=KM/N, the file packets that are only about these usersâĂŹ request can be missed during the transmission to the next layer. Hence, we can get the theorem1.

Theorem 1 *In centralized coded caching scheme, there are N files in server, with K users mappedÂăinto different layers, each equipped with a cache memory of M. Regards to one specific layer, there are total x users above this layer, so the transmission rate to the transfer station is*

$$\begin{aligned} R(M) &= \frac{\binom{K}{t+1} - \binom{K-x}{t+1}}{\binom{K}{t}} \\ &= K(1-\frac{M}{N})\frac{1}{1+\frac{KM}{N}}(1-\prod_{K}^{i=K-t}(1-\frac{X}{i})) \end{aligned}$$

Next we consider the uniform gain of each user. The next theorem describes the performance of the tree-like system compared with the uncoded caching scheme.

Theorem 2 *Compared with uncoded caching scheme, the gain obtained from coded caching scheme is*

$$Gain = x(1-\frac{M}{N}) - \frac{\binom{K}{t+1} - \binom{K-x}{t+1}}{\binom{K}{t}}$$

and with deeper level of the system, the average gain per user is reduced.

Now we illustrate the proof of theorem2. Use Δ dominating the average gain per user and it can be expressed as

$$\Delta = (1-\frac{M}{N}) - \frac{1}{x} \left\{ \frac{K-t}{t-1} - \frac{(K-x)!}{(K-x-t-1)!}\frac{(K-t)!}{K!(t+1)} \right\}$$

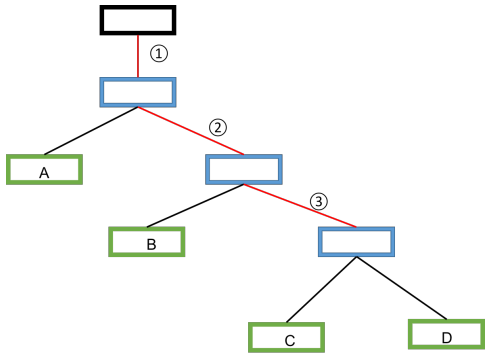

Figure 3: *The tree-like structure of coded caching scheme in decentralized setting with N=4, K=4, M=1 and three layers.The black block means the server holding whole files; blue blocks mean transfer station without caches; green blocks mean users, each equipped with a cache.*

and the function of x is increment.Hence, with deeper of the layer, the value of Δ is decreasing, which means that the average gain per user is reduced with less users remained in the delivery phase.

ii. Decentralized Setting

In centralized setting, the number of users are pre-fetched and in terms of the two phase there is a strong connection between users, which is not practical in mobility network. Hence, another scheme of coded caching is considered that creates simultaneous coded-multicasting opportunity without coordination in the placement phase, named as decentralized setting. We still use a simple example to describe the question. Illustrated in Fig.3, the system consists of 4 users, each requesting a different file and equipped with a cache memory of 1F bits. For the channel1, the packets are transferred are

$$\begin{array}{llll} A_{234}\oplus B_{134}\oplus C_{124}\oplus D_{123} & & & \\ A_{23}\oplus B_{13}\oplus C_{12} & A_{24}\oplus B_{14}\oplus D_{12} & B_{34}\oplus C_{24}\oplus D_{23} & A_{34}\oplus C_{14}\oplus D_{13} \\ A_2\oplus B_1 & A_3\oplus C_1 & A_4\oplus D_1 & B_3\oplus C_2 \quad B_4\oplus D_2 \quad C_4\oplus D_3 \\ A_\phi & B_\phi & C_\phi & D_\phi \end{array}$$

For the channel2, the packets are transferred are

Table 1: *The packets only correlated to the L users*

Packet	Number	Size
|S|=0	$\binom{L}{1}$	$(1-\frac{M}{N})^{K+L}$
|S|=1	$\binom{L}{2}$	$\frac{M}{N}(1-\frac{M}{N})^{K+L-1}$
|S|=2	$\binom{L}{3}$	$(\frac{M}{N})^2(1-\frac{M}{N})^{K+L-1}$
...	...	...
|S|=L-1	$\binom{L}{L}$	$(\frac{M}{N})^{L-1}(1-\frac{M}{N})^{K+1}$

$$
\begin{aligned}
&B_{34}\oplus C_{24}\oplus D_{23} = A_{234}\oplus B_{134}\oplus C_{124}\oplus D_{123} \quad \&\& \quad B_{34}\oplus C_{24}\oplus D_{23}\\
&B_3\oplus C_2 \quad B_4\oplus D_2 \quad C_4\oplus D_3 = A_{23}\oplus B_{13}\oplus C_{12} \ \&\& \ A_{24}\oplus B_{14}\oplus D_{12}\\
&\qquad\qquad\qquad\qquad\qquad\qquad A_{34}\oplus C_{14}\oplus D_{13} \ \&\& \ B_3\oplus C_2\\
&\qquad\qquad\qquad\qquad\qquad\qquad B_4\oplus D_2 \quad \&\& \quad C_4\oplus D_3\\
&B_\phi \quad C_\phi \quad D_\phi = B_\phi \quad C_\phi \quad D_\phi \quad \&\& \quad A_2\oplus B_1 \quad A_3\oplus C_1 \quad A_4\oplus D_1
\end{aligned}
$$

it can be found that the packet of A_ϕ is missed during the transmission to the next transfer station. For the channel3, we can get the same result. The packets are transferred are

$$
\begin{aligned}
&C_4\oplus D_3 = C_4\oplus D_3 \ \&\& \ A_{34}\oplus C_{14}\oplus D_{13} \ \&\& \ B_{34}\oplus C_{24}\oplus D_{23}\\
&\qquad \&\& \ A_{234}\oplus B_{134}\oplus C_{124}\oplus D_{123}\\
&C_\phi \quad D_\phi = C_\phi \quad D_\phi \ \&\& \ A_3\oplus C_1 \ \&\& \ A_4\oplus D_1 \ \&\& \ B_3\oplus C_2\\
&\qquad B_4\oplus D_2 \ \&\& \ A_{23}\oplus B_{13}\oplus C_{12} \ \&\& \ A_{24}\oplus B_{14}\oplus D_{12}
\end{aligned}
$$

and the packets $A_\phi, B_\phi, A_2 \oplus B_1$ are dropped. From above example and according to the properties of the decentralized coded caching, it is concluded that

Theorem 3 *In decentralized setting, at every transfer station, dropping unneeded packets, it can still use the decentralized coded caching to delivery to the next transfer station. Specifically, every channel's transmission rate between transfer stations is*

$$R(M) = x(1-\frac{M}{N})\frac{N}{xM}(1-(1-\frac{M}{N})^x)$$

Next we prove the correctness of theorem3. Assuming that in terms of one specific layer, there are K+L users above this layer and there are K users remained, the packets transferred form the above layer is possessed by dropping the unneeded packets. Thus the transmission rate is the one of the above layer subtracts the dropped packets. According to the algorithm of the decentralized setting, the dropped packets are listed in the table1.Therefore

$$R_K = R_{K+L} - \Delta R$$

$$
\begin{aligned}
\Delta R = &\binom{L}{1}(1-\frac{M}{N})^{K+L} + \binom{L}{2}\frac{M}{N}(1-\frac{M}{N})^{K+L-1}\\
&+\binom{L}{3}(\frac{M}{N})^2(1-\frac{M}{N})^{K+L-2} + ...\\
&+\binom{L}{L}(\frac{M}{N})^{L-1}(1-\frac{M}{N})^{K+1}
\end{aligned}
$$

ΔR actually is just possessing the dilivery phase to L users, and what different is that the size of packets is larger $(1-\frac{M}{N})^K$. Therefore,

$$
\begin{aligned}
&R_L = R_{K+L} - \Delta\\
&= (K+L)(1-\frac{M}{N})\frac{N}{(K+L)M}(1-(1-\frac{M}{N})^{K+L}) -\\
&L(1-\frac{M}{N})\frac{N}{LM}(1-(1-\frac{M}{N})^L)(1-\frac{M}{N})^K\\
&= (1-\frac{M}{N})\frac{N}{M}[1-(1-\frac{M}{N})^K]
\end{aligned}
$$

Hence, Theorem3 has been proved.

IV. Discussion

The discussion in this paper focuses on tree networks consisting of a single shared link. However, whether it is sufficient for order-optimality is another question worth pursuing. To be relevant in practice, the results here need to be extended to more general networks. An interesting extension of the coded caching approach proposed in this paper to device-to-device networks without a central server is considered. Adapting coded caching to D2D network and analyzing the performance of the scheme, such as lower bound and upper bound, is also worth researching. Finally, designing computationally efficient coded caching schemes which are scalable to large systems is of great interest.