Cache Content Placement Using Triangular Network Coding

Pouya Ostovari*, Abdallah Khreishah†, and Jie Wu*
*Department of Computer & Information Sciences, Temple University, Philadelphia, PA 19122
†Department of Electrical & Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102

***Abstract*—Video is one of the main causes of the dramatic increase in data traffic over cellular networks. Caching is an effective mechanism that decreases the download rate from base stations and, as a result, the load on the base station, by storing the most popular files or videos on the caches and providing them to the users. The problem of efficient content placement on the caches is known as an NP-complete problem. In this paper, we study the role of network coding by increasing the amount of available data to the users through the cache nodes. We propose a network coding-based content placement method, and we compare it to the best uncoded content placement and the best triangular network coding strategies. Our method not only increases the amount of available data to the users, but also results in a fair distribution of data.**

***Index Terms*—Linear network coding, triangular network coding, caching, content placement, wireless networks.**

I. Introduction

Video is one of the main causes of the dramatic increase in data traffic over cellular networks. This growth of data traffic between the base station, e.g. 4G, and the users will lead to a slowdown of the cellular systems. Live streaming uses a relatively tiny fraction of the overall video traffic. In contrast, most video traffic is generated by encoded and pre-stored videos on large video on-demand servers, such as YouTube and Netflix, and a large portion of the video on-demand traffic corresponds to a few popular files [1]. For example, 20-30% [2], [3] of the web traffic on the Internet is for YouTube and Netflix.

Based on this observation, the concept of caching helpers has been introduced [1], [4]–[8]. The authors in [1] propose a system architecture to achieve a high throughput. The main idea is that caching can decrease the required amount of communication between the base station and the users if there is enough content reuse going on and if many users are requesting the same few files. Therefore, the files are always available locally to the users who are requesting them.

The cache nodes (helpers) are assumed to have a large storage capacity and high-bandwidth wireless communication capabilities. They can cache popular files and serve users' requests. The caches work in conjunction with the base station, which provides the users with files that cannot be obtained from the caches (Figure 2). It is obvious that the base station will be able to serve more users if we can decrease the load on the base station. For this purpose, it it critical to use a mechanism to increase the number of available files through the cache nodes to the users.

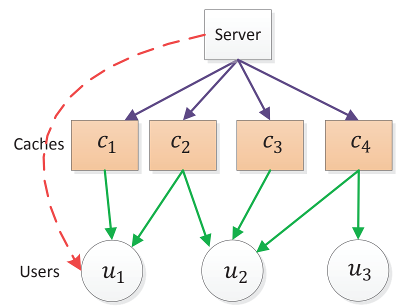


Fig. 1. Setting: there are m users and n caches in the network.

The main challenge in the wireless distributed caching problem is determining which files should be stored on each cache. In the case where each user only has access to one cache, the most popular file should be stored on each cache. However, when each user has access to multiple caches, the placement of files on the caches becomes nontrivial. The authors in [1] show that this problem is NP-complete.

Network coding [9]–[15] can be used to increase the efficiency of the content placement on cache nodes. Linear network coding is introduced in [10], and it is shown in [16] that linear network coding achieves the capacity for the single multicast session problem. In this case, each node generates and sends linear combinations of the received packets over a finite field. A received packet is called innovative if its coefficient vector increases the rank of the matrix formed by the received coefficient vectors. When a node receives an innovative packet, the node stores this packet and the corresponding coefficients vector in its packet buffer and coefficients buffer, respectively. Each relay node continues this process. Assume that k original packets are linearly coded together. After receiving k linearly independent coded packets, the destination node will be able to decode the coded packets and retrieve all of the original packets. The decoding process is done by using Gaussian Elimination to solve a system of linear equations. In [17], it is shown that randomly selecting the coefficients of the coded packets achieves the capacity asymptotically with respect to the finite field size.

Consider Figure 2 (a), in which there are three users u_1, u_2, and u_3. Each user is connected to two caches, and each cache is adjacent to two users. Assume that we want to provide the users with access to two files, p_1 and p_2. Also, assume that

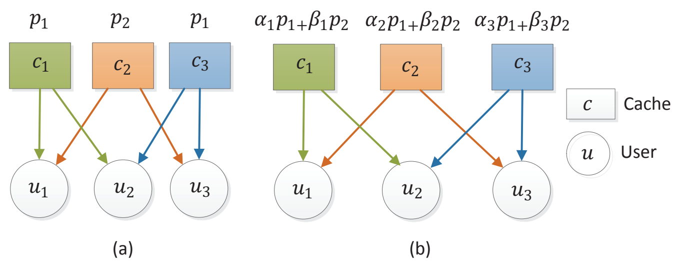


Fig. 2. Motivation: the benefit of using linear network coding for content placement.

the size of the caches and the files are the same. Without using network coding, the efficient way to place the files is to place one of them on two caches and place the other one on the last cache, as shown in Figure 2 (a). In this case, users u_1 and u_3 have access to both of the files, but user u_2 has access to just file p_1. In Figure 2 (b), three random linear combinations of the files are stored on the caches. Here, α and β are random coefficients. In this case, each user has access to two random linearly coded files, so they are able to decode the coded files and retrieve both p_1 and p_2.

In [4], [18], the authors assume that the request of each user is known, and they propose a joint bandwidth allocation and content placement approach. In this paper, we consider a different problem. We address the problem of layered video placement on caches using network coding. In this problem, the video layers have priority, and the i-th video layer will not be useful unless all of the layers that have a higher priority (smaller indices) are available to the user. We propose a content placement approach that achieves fairness in providing the layers to the users, while increasing the number of available video layers to the users.

The remainder of the paper is organized as follows. In Section II, we introduce the setting and problem definition. We propose our solution to the problem in Section III. Simulation results will be presented in Section IV. Section V concludes the paper.

II. Setting

We consider a single cell, equipped with a base station (such as 4G), which serves m users, $u_1, .., u_m$ with the help of n independent caches, $c_1, .., c_n$. The cache nodes are placed in fixed positions in the cell, and each cache node can cover a subset of the users based on their locations. There are h video layers, $p_1, .., p_h$ on a base station where $h < n$. The capacity of the caches is equal to the size of the video layers, and if a layer is not available on any adjacent caches to a user, the user will contact the central server to download the layer. However, the cost of downloading from the server is more than downloading from the cache nodes.

The key point is that, if there is enough content reuse, such that many users are requesting the same video layers, caching can replace the communication between the users and the base station. We assume that in order to use the layer p_i, all of the layers p_j $\forall j, 0 \le j \le i$ should be available to the user. We represent the availability of the video layers to the users with matrix $Z = z_{ij}$. If the layers p_1 to p_i are available to user u_j, then $z_{ij} = 1$; otherwise, $z_{ij} = 0$. The set of symbols used in this paper is summarized in Table I. Our goal is to minimize the number of contacts to the central server for layers retrieval. In other words, we want to maximize the total number of video layers that are available to the users. Therefore, our objective becomes:

$$\max \sum_{i=1}^{h} \sum_{j=1}^{n} z_{ij} \qquad (1)$$

TABLE I
The set of symbols used in this paper.

Notation	Definition
m	The number of users
n	The number of caches
h	The number of video layers
u_i	The i-th user
c_j	The j-th cache
p_k	The k-th layer
d_i	The degree of the ith user
f_i	The number of filled out adjacent caches to the ith user
r_i	The rank of the coded video layers on the filled out adjacent caches to the ith user
s_i	The number of triangular codes of degree i that are available to a user
f	The fairness
f'	The unfairness
v	The average number of available video layers to the users
q_i	The number of available video layers to the ith user

The solution is straightforward when the cache nodes are not dense and each user is connected to a single cache. In this case, each cache should store the first video layer (with the smallest index), since other layers are not useful without accessing to this layer. However, if the cache nodes deployment is dense enough, each user will have more than one cache node that is a part of a distributed caching system. In this case, the content placement problem becomes more complicated because of the correlation among the cache nodes. It is shown in [1] that the uncoded distributed caching problem is NP-complete.

III. Content Placement Approach Using Network Coding

In the problem of content placement using network coding, each video layer can be coded with any other layers. Therefore, there are $2^h - 1$ possibilities for coding the video layers, where h is the number of layers on the server (the empty set should not be considered). On the other hand, we have n caches in the network; in total, there are $(2^h - 1)^n$ different possible placements. To calculate the number of video layers that are available to the users in each placement, we need to calculate the layers that can be retrieved by each user. Note that we cannot use the Gaussian Elimination directly on the coded video layers that are available to a user nor compute the rank of the available layers. This is because by using Gaussian Elimination, or the overall rank of the coded video layers, we can just find out if all of the original layers are decodable or

$$\begin{cases} p_1 \\ p_2 \\ p_3 \end{cases} \quad \begin{cases} p_1,\ p_2,\ p_3 \\ p_1+p_2,\ \ p_1+p_3,\ p_2+p_3 \\ p_1+p_2+p_3 \end{cases} \quad \begin{cases} p_1 \\ p_1+p_2 \\ p_1+p_2+p_3 \end{cases}$$

(a) (b) (c)

Fig. 3. (a) Original video layers. (b) General form of random linear network coding. (c) Triangular network coding.

Algorithm 1 Checking the number of decodable packets in triangular coding

for i=1 to m **do**
  decodability=True
  for k=0 to i-1 **do**
    if $\sum_{j=i-k}^{i} s_j \leq k+1$ **then**
      decodability=False
    if decodability=True **then**
      number of decodable layers=i

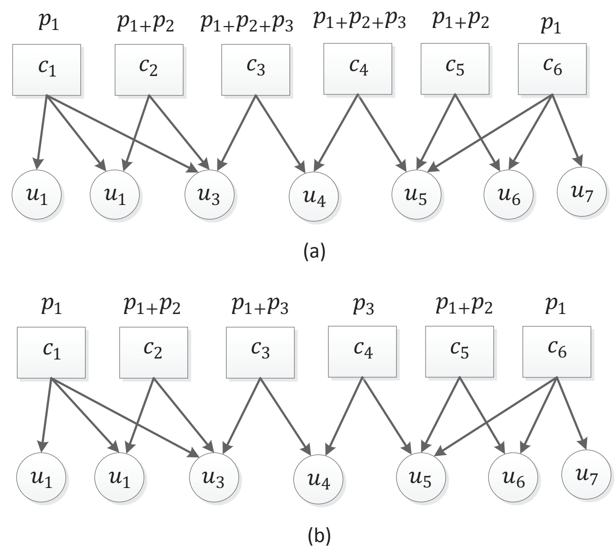


Fig. 4. A comparison of triangular network coding and random linear coding. Coefficients are not shown for simplicity.

not. Therefore, we need to calculate the rank of each subset of the coded packets that are available to a user or use the Gaussian Elimination to decode the coded video layers. As a result, the total complexity becomes $(2^h-1)^n m \sum_{i=1}^{m}(2^{d_i}-1)$.

One approach to decrease the complexity of the problem is to use triangular network coding [19], which is a kind of random linear network coding. In triangular network coding, the encoded video layers are in the form of $\sum_{j=1}^{k} \alpha_j p_j$, where $1 \leq k \leq h$ and α_j are random coefficients. In other words, each coded layer is a random linear combination of the first k original layers. Therefore, there are just n possibilities for coding n original layers. Figures 3 (b) and (c) show the possible coded layers of the original layers in Figure 3 (a) using the general form of network coding and triangular coding, respectively. As shown in Figures 3 (b), there are $7 = 2^3-1$ choices for coding 3 original packets in the case of the general form of random linear network coding, but just 3 possibilities for the case of triangular network coding. The coefficients are not shown in the figures for simplicity. For example, p_1+p_2 means $\alpha_1 p_1 + \alpha_2 p_2$, where α_1 and α_2 are two random coefficients.

We prefer using triangular network coding over the general form of network coding for three reasons. First, it limits the coding space of the coding problem. Second, based on our setting, the i-th video layer is not useful without accessing to the layers with a smaller index. This assumption has the same nature as the coding rule in triangular network coding. Third, in addition to the lesser complexity of triangular coding, compared with linear coding, we do not need to use Gaussian Elimination to check the number of decodable video layers at each user. This checking can be done by a simple algorithm in order of $O(h^2)$ [20]. Assume that s_k represents the number of coded packets in the form of $\sum_{j=1}^{k} \alpha_j p_j$ that are available to a user. In this scheme, the user can decode the first i layers if $\sum_{j=i-k}^{i} s_j \geq k+1, \forall k \in [0, i-1]$. The condition implies that the rank of the coefficients matrix of the available encoded video layers from coding any of the first i layers is at least i. Therefore, Gaussian Elimination will be able to decode the first i layers. For example, assume that three packets, $\alpha_1 p_1$, $\alpha_2 p_1 + \alpha_3 p_2$, and $\alpha_4 p_1 + \alpha_5 p_2 + \alpha_6 p_3$, are available to a user. In this example, $s_3 \geq 1$, $s_2 + s_3 \geq 2$, and $s_1 + s_2 + s_3 \geq 3$, so all three layers can be decoded by the user.

Lemma 1: the complexity of the content placement using triangular network coding is $h^{n+2}m$.

Proof: In this case, there are just h different coding possibilities since $1 \leq k \leq h$. Therefore, the total number of different placements will be h^n. As mentioned before, the number of decodable original video layers at each user can be calculated in order on $O(h^2)$. Therefore, the total complexity of checking all possible triangular coding placements becomes $h^{n+2}m$. ■

Although the complexity of triangular network coding is much less than random linear coding, it might not achieve efficiency compared to linear coding. Consider Figures 4 (a) and (b), where the optimal placement using triangular and random linear coding are shown in Figures 4 (a) and (b), respectively. In these figures, coefficients are not shown for simplicity. For example, p_1+p_2 means $\alpha_1 p_1 + \alpha_2 p_2$, where α_1 and α_2 are random coefficients. In the case of triangular network coding, user u_4 will not be able to decode any packet. However, when we use random linear coding, this user can decode video layers p_1 and p_3. Note that, based on our setting, the layer p_3 is not useful for user u_4 since the user does not have access to the layer p_2.

A. Network Coding-based Content Placement Heuristic

Although the complexity of finding the optimal placement using a triangular network coding is much less than random linear network coding, it is still high. For this reason, we propose a network coding-based content placement heuristic (NCCP) in this section. Based on the proposed setting, we know that the video layer p_i is not useful to a user unless the layers p_j $\forall j, 0 < j < i-1$ are available to the user. Based on this observation, we give more priority to the first layer, and we try to provide it to all of the users. Then, if it is possible, we provide the second layer to the users. In more detail, we try to provide the second video layer to as many users as possible. This process is repeated until we cannot provide the content p_i to any user for a given $1 < i \leq h$. This method can also provide a fair solution to the content placement problem. This is because the algorithm tries to provide each layer that has a greater priority (smaller index) to as many users as possible, and then the algorithm will start to provide the next layer, which is less popular to the users. We can observe that this solution has a greedy nature, and thus we can use a greedy algorithm to implement this solution.

In each step, our greedy algorithm selects one of the users and fills up its adjacent cache with random linearly coded video layers. Because of the greedy nature of our algorithm, the generated coded layer will always be a triangular code. The user selection rules are as follows:

- Rule 1: the user with the minimum degree is selected. The degree of user u_i, represented as d_i, is defined as the number of its adjacent caches.
- Rule 2: in the case where two users have the same degree, the user with a larger number of filled-up cache nodes is selected.
- Rule 3: if two users have the same degree and the same number of filled caches, the user whose adjacent caches have less cumulative ranks is selected.

We give more priority to Rule 1 compared to Rule 2 and Rule 3. Also, the priority of Rule 2 is more that that of the Rule 3. The reason for selecting user u_i with the minimum degree is that there are less content placement choices for adjacent cache nodes to that user. If we do not start with such a user u_i, it is possible for us to fill the adjacent cache nodes to user u_i with an undecodable packet. For example, in Figure 5, user u_1 has just one adjacent cache. Consequently, we can just provide video layer p_1 to this user. By selecting user u_1 and filing its adjacent cache c_1, we can reduce the number of choices for cache c_2, which is adjacent to user u_2. Then, user u_3 is selected, and its adjacent cache will be filled with p_1. In this example, if we start from user u_2 and fill its adjacent cache nodes with random linear combinations of p_1 and p_2, users u_1 and u_3 will not be able to decode the coded video layers on cache nodes c_1 and c_3, respectively. Note that network coding does not have any benefit in this example, and we use this simple example just to describe our user selection policy.

After selecting the appropriate user based on Rules 1, 2, and 3, the user's neighboring cache nodes should be filled up. Let d_i be the degree of user u_i. We represent the number of user u_i's filled-up adjacent cache nodes and their cumulative rank as v_i and r_i, respectively. Assume that, in the current

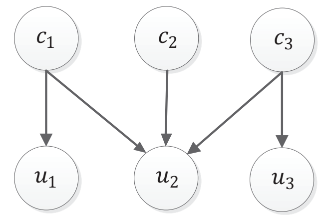


Fig. 5. User selection rules.

Algorithm 2 Content placement

while There is an empty cache **do**
  Select an unprocessed node u_i based on Rules 1, 2, and 3
  $g = d_i - v_i + r_i$
  for each empty adjacent cache c_k to user u_i **do**
    Fill up cache c_k with $\sum_{j=1}^{g} \alpha_j p_j$
  Tag node u_i as a processed node

iteration, user u_i has been selected based on the rules. The degree of this user is d_i, so this user cannot receive more than the first d_i video layers. On the other hand, v_i of the caches are already filled, and their communicative rank is r_i. As a result, the maximum number of decodable layers to user u_i is $d_i - v_i + r_i$. Therefore, the algorithm fills up all of the empty adjacent cache nodes to user u_i with a random linear combination of the $d_i - v_i + r_i$ first video layers. Then, the algorithm selects another unprocessed user by utilizing the user selection rules, and the algorithm uses the same policy to fill the cache nodes that are adjacent to the new user. The proposed greedy algorithm is described in Algorithm 2.

Assume that user u_i is selected, and all of the d_i adjacent caches are empty. The algorithm fills up all of them with $\sum_{j=1}^{d_i} \alpha_j p_j$. In this case, the user will have access to d_i linearly independent coded video layers. Thus, the user will be able to decode all of the encoded layers. Now, assume that the degree of user u_j is 3, and two of the adjacent caches are filed with $\alpha_1 p_1$ and $\alpha_2 p_1$. In this case, the rank of the filled caches is 1. Therefore, it is not possible to provide 3 layers to the user, and if we fill up the remaining unfilled cache with a linear combination of p_1, p_2, and p_3, the user will not be able to decode the coded packets. As a result, based on our algorithm, we code the first $g = 3-2+1$ layers together, and assign $\alpha_3 p_1 + \alpha_4 p_2$ to the remaining cache.

Consider Figure 6, in which user u_1 has a minimum degree of 2. As a result, in the first step, the content placement algorithm selects user u_1 and places a random linear combination of the first two layers on the cache nodes adjacent to the user (caches c_1 and c_2). After the first step, the user nodes u_2, u_3, and u_4 have a degree equal to three. However, user u_2 has two adjacent cache nodes that are filled. Consequesntly, in the second round, the algorithm selects user u_2 and fills up the remaining cache that is adjacent to the user, which is cache c_3. In this case, $d_2 = 3$, $f_2 = 2$, and $r_2 = 2$. Therefore,

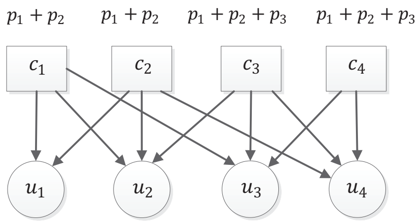
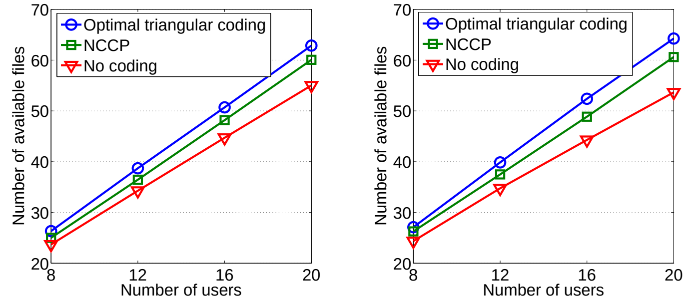


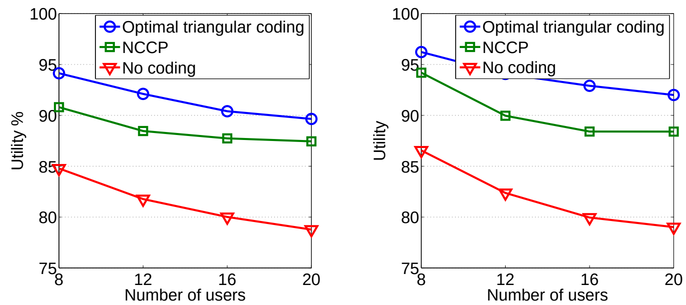
Fig. 6. Content placement algorithm description. Coefficients are not shown for simplicity.



(a)

(b)

Fig. 7. Total number of video layers that are available to the users. (a) n=5, h=4. (b) n=7, h=4.



(a)

(b)

Fig. 8. Utility of the content placement methods (the number of video layers that are available to a user divided by its degree). (a) n=5, h=4. (b) n=7, h=4.

we fill cache c_3 with a random linear combination of the $3 - 2 + 2 = 3$ first packets. In the last round, the degree, the number of filled-up caches that are adjacent to users u_3 and u_4, and their ranks are equal; thus, we select one of the users randomly. Assume that we choose user u_3. The algorithm fills cache c_4 with a random coded layer of the first three layers. Note that, in Figure 6, we do not show the coefficients of the coded packets for simplicity.

IV. Simulation

In this section, we evaluate and compare the proposed network coding-based cache placement method (NCCP) with the best uncoded and triangular network coding placement strategies. For this purpose, we implemented a simulator in the MATLAB environment. In order to find the best uncoded placement, we try all of the possible original video placements on the caches, and we find the placement that provides the largest amount of available video layers to the users. We perform the same approach to find the best placement using the triangular network coding.

A. Setting

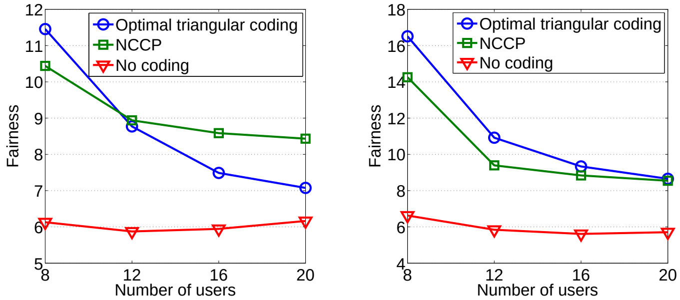
We perform our simulations on random topologies with different numbers of users and caches. For each setting, we run the simulation for 1000 randomly generated topologies (random adjacency of the users with the cache nodes). The plots in this paper are based on the average outputs of the simulation runs. The following metrics are compared in the simulations:

1) Amount of available video layers to the users: the total number of available layers to the users through the cache nodes. Note that, based on the proposed setting, a layer will not be useful to a user unless all of the layers with a smaller index are provided to the user first.
2) Average utility: the utility of each user is defined as the number of available layers to that user through its neighboring caches (the cumulative rank of the neighboring cache nodes) divided by the degree of the user.
3) Fairness: we define unfairness as the average difference between the number of available layers to each user and the average number of available layers to the users. We represent the number of layers available to the i-th user, unfairness, and fairness as q_i, f', and f, respectively. Based on the definition, we have $f' = \frac{\sum_{i=1}^{m} |q_i - e|}{m}$, where $e = \frac{\sum_{i=1}^{m} q_i}{m}$ is the average number of layers that are available to the users. Fairness is defined as $f = \frac{1}{f'}$.

B. Results

In the first experiment, we evaluate the total number of video layers that are available to the users. In Figure 7 (a), the number of cache nodes and layers are 5 and 4, respectively. This figure shows that the optimal triangular network coding can provide the largest number of layers to the users, which is about 14% more than that of the best no-coding placement strategy. Also, the NCCP approach can provide the users with about 9% more video layers than the best no-coding placement strategy. This difference might not be huge, but it should be noted that finding the best no-coding placement is NP-complete, and the complexity of finding such an optimal non-coding placement is $O(k^n \sum_{i=1}^{n} d_i)$. As we increase the number of users, the effect of network coding increases, which leads to a larger difference between the coding approaches and the no-coding method.

In Figure 7 (b), we change the number of cache nodes to 7, and we repeat the previous experiment. In this figure, the total number of available packets in the optimal triangular network coding method is about 17% more than that of the best no-coding placement. Also, the gain of the NCCP approach is up to 12% more than the no-coding method. The figure shows that



(a) (b)

Fig. 9. Fairness is equal to one divided by unfairness. Unfairness is defined as the average difference between the number of video layers that are available to each user and the average number of available video layers to the users). (a) n=5, h=4. (b) n=7, h=4.

the efficiency of the network coding increases as we increase the number of users.

Figures 8 (a) and (b) show the comparison between the average utility of the three approaches. We define the utility of a user as the fraction of the number of video layers that are available to that user and the degree of the user. The idea behind dividing by the user's degree is that, in the best case, the number of layers available to a user can be equal to its degree. It can be inferred from Figure 8 (a) that the utility of both the optimal triangular network coding and the NCCP methods are about 13% and 10% more than the best no-coding method, respectively. We change the number of caches to 7 in Figure 8 (b). By comparing Figures 8 (a) and (b), we can find that the utility in all of the methods increases as we increase the number of caches. However, increasing the number of caches has more of an effect on the NCCP method compared to that of the other approaches.

In the last experiment, we evaluate and compare the fairness of the methods. We define the unfairness (one divided by fairness) as the average difference between the number of layers that are available to each user and the average number layers that are available to the users. Figure 9 (a) shows that the fairness of the optimal triangular network coding and the NCCP approaches are up to 90% and 75% more than that of the best no-coding content placement strategy, respectively. We increase the number of cache nodes in Figure 8 (b) from 5 to 7 and evaluate the fairness of the approaches. In this setting, the fairness of the best no-coding approach is more than 56% and 50% less than that of the the best triangular network coding and the NCCP approaches, respectively.

V. Conclusion

In this paper, we study the effects of network coding on increasing the amount of available data to the users through the cache nodes in cellular networks. Providing a larger portion of the data through the caches decreases the pressure on the base station (such as 4G) and enables the base station to serve more users. We propose a network coding-based content placement method, and we compare it with the best uncoded content placement strategy and the best triangular network coding strategy. Our method not only increases the amount of video layers that are available to the users, but it also results in a fair distribution of the data.

Acknowledgment

This research was supported in part by NSF grants ECCS 1231461, ECCS 1128209, CNS 1138963, CNS 1065444, and CCF 1028167.

References

[1] N. Golrezaei, K. Shanmugam, A. Dimakis, A. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," *Arxiv preprint arXiv:1109.4179*, 2011.

[2] A. Finamore, M. Mellia, M. Munafò, R. Torres, and S. Rao, "Youtube everywhere: impact of device and infrastructure synergies on user experience," in *ACM SIGCOMM Conference on Internet Measurement*, 2011, pp. 345–360.

[3] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," in *ACM SIGCOMM*, 2010, pp. 75–86.

[4] H. Hao, M. Chen, A. Parekh, and K. Ramchandran, "A distributed multichannel demand-adaptive P2P VoD system with optimized caching and neighbor-selection," in *SPIE*, 2011.

[5] J. Wang, C. Yeo, V. Prabhakaran, and K. Ramchandran, "On the role of helpers in peer-to-peer file download systems: Design, analysis and simulation," in *IPTPS*, 2007.

[6] J. Wang and K. Ramchandran, "Enhancing peer-to-peer live multicast quality using helpers," in *IEEE ICIP*, 2008, pp. 2300–2303.

[7] H. Zhang, J. Wang, M. Chen, and K. Ramchandran, "Scaling peer-to-peer video-on-demand systems using helpers," in *IEEE ICIP*, 2009, pp. 3053–3056.

[8] Y. He and L. Guan, "Improving the streaming capacity in p2p vod systems with helpers," in *IEEE ICME*, 2009, pp. 790–793.

[9] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: practical wireless network coding," in *ACM SIGCOMM*, 2006.

[10] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[11] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782– 795, Oct 2003.

[12] P. Ostovari, J. Wu, and A. Khreishah, "Deadline-aware broadcasting in wireless networks with local network coding," in *IEEE ICNC*, Jan 2012.

[13] P. Ostovari, A. Khreishahand, and J. Wu, "Deadline-aware broadcasting in wireless networks with network coding," in *IEEE GLOBECOM*, Dec 2012.

[14] A. Khreishah, I. Khalil, P. Ostovari, and J. Wu, "Flow-based xor network coding for lossy wireless networks," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 2321–2329, 2012.

[15] A. Khreishah, C. Wang, and N. Shroff, "Rate control with pairwise intersession network coding," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 816–829, 2010.

[16] S. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[17] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[18] S. Pawar, S. Rouayheb, H. Zhang, K. Lee, and K. Ramchandran, "Codes for a distributed caching based video-on-demand system," in *Asilomar Conference on Signals, Systems, and Computers*, 2011.

[19] D. Wang, J. Liu, Q. Zhang, and F. Chen, "Cooperative coding and caching for streaming data in multihop wireless networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, no. 10, pp. 2:1–2:13, 2010.

[20] D. Koutsonikolas, Y. Hu, C. Wang, M. Comer, and A. Mohamed, "Efficient online WiFi delivery of layered-coding media using interlayer network coding," in *31st International Conference on Distributed Computing Systems (ICDCS)*, 2011, pp. 237–247.