Multihop Caching-Aided Coded Multicasting for the Next Generation of Cellular Networks

Mohsen Karimzadeh Kiskani†, and Hamid R. Sadjadpour†,

***Abstract*—Next generation of cellular networks deploying wireless distributed femtocaching infrastructure proposed by Golrezaei et. al. are studied. By taking advantage of multihop communications in each cell, the number of required *femtocaching helpers* is significantly reduced. This reduction of femtocaches is achieved by using the underutilized storage and communication capabilities in the User Terminals (UTs), which results in reducing the deployment costs of distributed femtocaches.**

A multihop index coding technique is proposed to code the cached contents in helpers to achieve order optimal capacity gains. This can serve as an efficient content delivery algorithm for the solution provided by Golrezaei et. al. As an example, we consider a wireless cellular system in which contents have a popularity distribution. It has been shown that if the contents follow a high content reuse popularity distribution, our approach can replace many unicast communication with multicast communication. We will prove that simple linear index codes found by heuristics based on graph coloring algorithms can achieve order optimal capacity under Zipfian content popularity distribution.

***Index Terms*—Cellular Networks, Femtocaching, Index Coding, Distributed Caching.**

I. Introduction

With the recent pervasive surge in using wireless devices for video and high speed data transfer, it seems eminent that the current wireless cellular networks cannot be robust solutions to the ever-increasing wireless bandwidth utilization problem. Researchers have been recently focused on laying down the fundamental grounds for future cellular networks to overcome such problems.

Deploying home size base stations is proposed as a solution in [7]. The solution in [7] is based on the idea of femtocells in which many small cells are deployed throughout the network to cover the whole network. Deploying many cells in the network with reliable backhaul links requires significant capital investment. Therefore, Golrezaei et. al. [14] proposed femtocaching as an alterante solution to overcome this problem. In this approach, in every cell along with the main base station, smaller base stations with low-bandwidth backhaul links and high storage capabilities are deployed to create a wireless distributed caching infrastructure. These small base stations which are called caching helpers (or simply helpers), will store popular contents in their caches and use their caches to serve *User Terminal (UT)* requests. Therefore, in networks with high content reuse, the backhaul utilization will be significantly reduced using this approach. If the requested content is not available in the helper's cache, UTs can still download the content using their low-bandwidth backhaul link to the base station. The proposed technique in [14] requires deployment of large number of femtocaches in order to cover all the nodes in the network.

On the other hand, it is well-known that web content request popularity follows Zipfian-like distributions [6]. This content popularity distribution implies that few popular contents are widely requested by the network UTs. We assume UTs store their requested contents and therefore, helpers can multicast multiple requests by taking advantage of coding to reduce the total number of transmissions.

In this paper, we propose to use index coding to code the contents in helpers before transmission. Index coding is a source coding technique proposed in [3] which takes advantage of UTs' side information in broadcast channels to minimize the required number of transmissions. In index coding, the source (e.g. base station) design codes based on the side information stored in requesting nodes. The coded information will be broadcasted to the UTs that will use coded information together with their cached contents to decode the desired content. Index coding improves bandwidth utilization by minimizing the number of required transmissions. We propose to extend index coding approach from broadcast one-hop communication to multihop scenarios which will be explained in details.

Our main motivation to use index coding is the high storage availability in UTs that is mainly unused to improve the achievable throughput of the future wireless cellular networks. Current improvements in high density storage systems has made it possible to have personal devices with Terabytes of storage capability. This ever increasing trend promises future personal wireless devices with huge under-utilized storage capabilities. Future wireless devices can use their storage capability to store the contents that they have already requested. In an index coding setting, when a UT is requesting a content, it can receive a coded content which is multicasted to many UTs and then each UT uses the information in its cache to decode its requested content from the received coded content. There is an important equivalence between index coding and network coding as stated in [10], [11] and therefore, the results in this paper can also be stated based on a network coding terminology.

We will prove that index coding can be efficiently used to encode the contents by helpers under a Zipfian distribution model. The encoded contents can be relayed through multiple hops to all the UTs being served by that helper.

The optimal index coding solution is an NP-Hard problem

M. K. Kiskani† and H. R. Sadjadpour† are with the Department of Electrical Engineering, University of California, Santa Cruz. Email: {mohsen, hamid}@soe.ucsc.edu

[18]. However, we will show that even using linear index codes can result in order optimal capacity gains in these networks. We believe that this coding technique can serve as a complement to the solution proposed in [14]. As clearly articulated in [21], in any caching problem we are faced with two phases of cache placement and cache delivery. While [14] proposes efficient cache placement algorithms, we will be focusing on efficient delivery methods for their solution. We will show that the problem of delivery in their approach can be efficiently addressed by using index coding in the helpers and relaying the coded contents in the network.

Recent discussions on standards for future wireless cellular networks are focused on providing high bandwidth for Device-to-Device communications (D2D). Examples of such bandwidths can be found in the recent proposal for IEEE 802.11ad standard (up to 60GHz [2]) and the millimeter-wave proposal in future 5G networks which can potentially enable up to 300GHz of D2D communications [1], [4]. In our proposal, this potential abundant D2D bandwidth can be effectively utilized to relay the coded contents inside an ad hoc network which is being served by a helper. It is in fact such excessive storage and bandwidth capabilities of future wireless systems that make our solution feasible. Therefore, our solution can significantly improve the capacity and performance of future mobile (or vehicular) networks by reducing the handover probability between the helper nodes.

Deploying many femtocaching helpers is not economically efficent. Furthermore, in networks in which smaller numbers of UTs are requesting contents, the deployed helpers may not be fully utilized. On the other hand, since UTs have significant D2D capabilities, they can efficiently participate in content delivery through multihop D2D communications. In our proposal, we suggest to deploy few helpers which can deliver the contents to neighboring UTs through multihop D2D communications. This can reduce the network deployment and maintenance costs.

The rest of this paper is organized as follows. Section II reviews the related works and section III describes the proposed network model. This network model is based on the network model proposed in [14] with the addition of using multihop communications and index coding. In section IV, we will explain the scaling laws of capacity improvement using index coding and relaying to send the coded contents to the UTs. Section V demonstrates that index coding algorithms can achieve order optimal gains. Section VI shows the simulation results and the paper is concluded in section VII.

II. Related work

The problem of caching when a server is transmitting contents to clients was studied in [21] from an information theoretic point of view. The authors introduced two phases of *cache placement* and *content delivery*. For a femtocaching solution, efficient cache placement algorithms are proposed in [14] for helpers. In this paper, we focus on efficient content delivery algorithms through index coding and multihop D2D communications for a femtocaching solution.

There has been significant research on index coding since it was proposed by [3]. The practical implementation of index coding for wireless applications was proposed in [8] by proposing cycle counting methods. A dynamic index coding solution for wireless broadcast channels is proposed in [23]. In [23], a wireless broadcast station is considered and a simple set of codes based on cycles in the dependency graph is provided. They show the optimality of these codes for a class of broadcast relay problems. In this paper, we prove that codes based on cycles can acheive order optimal capacity gains in networks with Zipfian content request distribution.

Approximating index coding solution is proved [3] to be even an NP-Hard problem. However, efficient heuristics has been proposed in [9] some of which are based on well-known graph coloring algorithms. Other references like [11] and [10] have studied the connections and equivalence between index coding and network coding. Tran et al. [24] have studied a single hop wireless link from a network coding approach and have shown similar results to index coding. Ji et al. [16] studied theoretical limits of caching in D2D communication networks.

Study of coding techniques in networks with high content reuse has recently attracted the attention of researchers. Montpetit et al. [22] studied the applications of network coding in Information-Centric Networks (ICN). Wu et al. [25] studied network coding in Content Centric Networks (CCN) which is an implementation of ICN. Leong et al. [19] proposed a linear programming formulation to deliver contents optimally in today's IP based Content Delivery Networks (CDN) using network coding. Llorca et at. [20] proposed a network-coded caching-aided multicasting technique for efficient content delivery in CDNs. Our work has focused on the extension of Index coding to multihop communication and to show that linear codes can achieve order optimal capacity gains in such networks. Our work can be also formulated based on network coding approach.

III. Network Model

In this section, we will explain the network model that we will study. We assume a network model for future wireless cellular networks in which femtocaching helpers with significant storage capabilities are deployed throughout the network to assist the efficient delivery of contents to UTs through reliable D2D communications. Such helpers are characterized with low rate backhaul links which can be wired or wireless. They will also have localized, high-bandwidth communication capabilities. With their significant storage capacity, in a network with high content reuse, many of the content requests for the UTs inside a cell can be satisfied directly by the helpers. This approach allows a wireless distributed caching infrastructure to satisfy most of the content requests and reduce the traffic load of the main base station in a cell.

Each helper is serving a wireless ad-hoc network in which the UTs are utilizing high bandwidth D2D communication techniques such as millimeter wave and IEEE 802.11ad technologies. This high bandwidth D2D communication enables the UTs to relay data from a nearby helper to all the UTs that are within transmission range. We assume that the path lifetime between the helper and UTs is longer than the time required to

transmit the content. For large files and when UTs are moving fast, one solution is to divide the content into smaller files and treat each file separately. The paper assumes that the network is connected which can be justified by the large number of UTs that will be available in the future wireless cellular networks.

Unlike baseline single hop D2D communication approach discussed earlier, we assume communications between UTs and helpers through multihop D2D communication. Multihop communication allows abundant D2D bandwidth to be effectively utilized for delivering contents from helpers to UTs.

Further, we take advantage of index coding and the side information cached in UTs to significantly reduce the required number of helpers and consequently, reduce the infrastructure maintenance and deployment costs. To demonstrate the effectiveness of using multihop communications, we consider similar assumptions as in [14] with a macro base station placed in the center of a cell with radius 400 meters serving 1000 UTs and a transmission range of 100 meters [1] for D2D communication. As shown in Figure 1, with only 4 helpers uniformly located in the cell, 100% and 80% of nodes are covered with 3 and 2 hop communications respectively. Covering all the UTs in the same cell with only one hop communication requires up to 45 helpers [14]. Clearly, for a

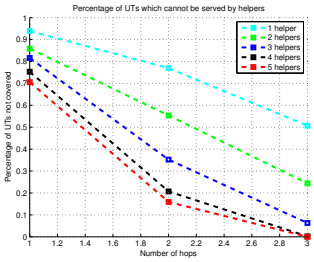

Fig. 1. Pertentage of UTs not covered versus the maximum number of hops traveled. This simulation is carried over a cell with radius 400 meters and with a communication range of 100 meters.

vehicle or a mobile UT operating in the cell, the handover probability will be significantly reduced if the number of helpers shrinks from 45 to 4.

We assume that n UTs denoted by $\mathbb{N} = \{N_1, N_2, ..., N_n\}$ are being served by a helper. There are m contents $\mathbb{M} = \{M_1, M_2, ..., M_m\}$ available with M_1 as the most popular content and M_m as the least popular content in the network . Let's assume UT N_i requests a content with popularity index r_i in the current time interval. Each UT has a cache of fixed size δ in which contents with indices $C_i = \{c_{i1}, ..., c_{i\delta}\}$ are stored. Therefore, we assume that UT N_i caches contents $M_{c_{i1}}, M_{c_{i2}}, \ldots, M_{c_{i\delta}}$ and the set of cached content indices in N_i is represented by C_i. Therefore, if we denote the set of cached contents in UT N_i by $\mathcal{C}_i$, then we have $\mathcal{C}_i = \{M_j \mid j \in C_i\}$. The requested content index is shown by r_i.

Let's assume that we have n UTs each with a set of side information $\mathcal{C}_i$. The formal definition of index code described in [3] is given below.

Definition 1. An *index code* on a set of n UTs each with a side information set $\mathcal{C}_i \subseteq \mathbb{M}$, and a requesting content $M_{r_i} \in \mathbb{M}$ for $i = 1, 2, \ldots, n$ is defined as a set of codewords in $\{0,1\}^l$ toghether with

1) An encoding function $\mathcal{E}$ mapping inputs in $\{0,1\}^m$ to codewords and
2) A set of decoding functions $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$ such that $\mathcal{D}_i(\mathcal{E}(\mathbb{M}), \mathcal{C}_i) = M_{r_i}$ for $i = 1, 2, \ldots, n$.

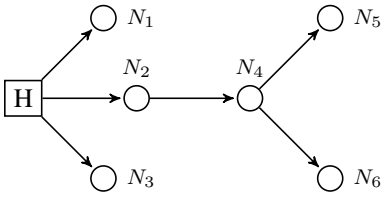

Fig. 2. Example of a wireless multihop network being served by the helper H. Each arrow represents a link with high bandwidth D2D communication capability.

In the above definition, the length of the index code l denotes the number of required transmissions to satisfy all the content requests of the n UTs. The encoding function $\mathcal{E}$ is applied to the contents by the helper and the decoding functions $\mathcal{D}_i$s are applied individually by UTs to decode their desired contents from the encoded content using their cached information.

Figure 2 shows a helper H serving 6 UTs N_1, N_2, N_3, N_4, N_5, and N_6. Let's assume UTs N_1, N_2 and N_5 request contents M_3, M_1 and M_4 while storing $\{M_1, M_4\}$, $\{M_3, M_4\}$ and $\{M_1, M_3\}$ respectively. Using index coding requires 3 channel usages while without index coding, we need 5 channel usages. This is true since using index coding, the helper creates the XOR combination of contents M_1, M_3 and M_4 as $M_1 \oplus M_3 \oplus M_4$ and broadcasts this coded content to its neighboring UTs. Either of N_1 and N_2 can immediately reconstruct their requested content from this coded content that they have received. For instance, N_1 which is requesting M_3 can decode its requested content by using it's cached information and XOR operation on the encoded message to retrieve the requested content M_3, i.e., $(M_1 \oplus M_3 \oplus M_4) \oplus M_1 \oplus M_4 = M_3$.

After N_2 receives $M_1 \oplus M_3 \oplus M_4$, it relays it forward to the node N_4 which again simply broadcasts it to N_5. UT N_5 will again use XOR operation to decode it's desired content M_4 by adding it's cached contents to the coded content that it has received, i.e., $(M_1 \oplus M_3 \oplus M_4) \oplus M_1 \oplus M_3 = M_4$. Therefore, to satisfy all the requests, the helper only needs to broadcast $M_1 \oplus M_3 \oplus M_4$ and then N_2 and N_4 need to relay this coded content to N_5. This way using index coding, only 3 transmissions are needed.

To satisfy the content requests through multihop D2D but without index coding, 5 transmissions are needed as the helper should transmit M_3 to N_1, M_1 to N_2, M_4 to N_2, and N_2 relays M_4 to N_4 and N_4 relays it to N_5.

We assume a Zipfian distribution with parameter $s > 1$ for content popularity distribution in the network. This means that the probability that UT N_i requests any content with index r_i

at any time instant is given by

$$\Pr[N_i \text{ requests content with index } r_i] = \frac{r_i^{-s}}{H_{m,s}}, \quad (1)$$

where $H_{m,s} = \sum_{j=1}^{m} \frac{1}{j^s}$ denotes the m^{th} generalized harmonic number with parameter s.

Remark 1. This paper only focuses on Zipfian content request probability with parameter $s > 1$ which is a *non-heavy-tailed* probability distribution. Our results are correct for any type of non-heavy-tailed probability distribution and the extension to heavy-tailed probability distribution is the subject of future work.

Dependency graph is a useful analytical tool [3], [8], [11] that is widely used in index coding literature.

Definition 2. *(Dependency Graph):* Given an instance of an index coding problem, the dependency graph[1] $\overrightarrow{G}(V, E)$ is defined as

- Each UT N_i corresponds to a vertex in V, $N_i \in V$, and
- There is a directed edge in E from N_i to N_j if and only if N_i is requesting a content that is already cached in N_j.

This dependency graph does not represent the actual physical links between UTs in the network. This is a virtual graph in which each edge represents the connection between a UT that is requesting a content and a UT that caches this content. As discussed in [8], [23], every cycle in the dependency graph is representative of a connection between UTs and it can save one transmission. For every clique in the dependency graph, all the requesting UTs in the clique can be satisfied by a simple linear XOR index code. The complement of dependency graph is called *conflict graph*. This graph is of significant interest since any clique in the dependency graph gives rise to an independent set[2] in the conflict graph. Therefore, well-known graph coloring algorithms over conflict graph can be used to find simple linear XOR index codes. The dependency and conflict graphs in our network are random directed graphs. In the next section, we will use the properties of these graphs to find the capacity gains and propose simple index coding solutions.

To prove our results in this paper, we have used Least Recently Used (LRU) or Least Frequently Used (LFU) cache policies. Similar results can be produced for other caching policies. LRU caching policy assumes that most recently requested contents are kept in the cache. In LRU caching, in each time slot the content that is requested in the previous time slot is stored in the first cache location. If this content was already available in the cache, it is moved from that location to the first cache location. If the content was not available, then the content that is least recently used is discarded and the most recently requested content is cached in the first cache location. Any other content is relocated to a new cache location such that the contents appear in the order that they have been requested.

[1]Dependency graph is a directed graph.

[2]An independent set is a set of vertices in a graph for which none of the vertices are connected by an edge.

In LFU caching policy, the contents are cached based on their request frequency. Highly popular contents are stored in the first locations of the cache and contents with lower request frequency are cached in the bottom locations of the cache.

Computing the probability of having content with index r_i in cache C_j, $\Pr[r_i \in C_j]$, turns out to be complicated for LRU or LFU caching policies. A simple lower bound on this probability for LRU can be found by noticing that $\Pr[r_i \in C_j]$ is greater than the probability that UT N_j have requested the content with index r_i in the most recent time slot and therefore it is located at the top of the cache. This lower bound can be derived using equation (1).

$$\begin{aligned} \Pr[r_i \in C_j] &= \Pr[c_{j1} = r_i] + \sum_{l=2}^{\delta} \Pr[c_{jl} = r_i] \\ &\geq \Pr[c_{j1} = r_i] = \frac{r_i^{-s}}{H_{m,s}}. \end{aligned} \quad (2)$$

To prove that the same lower bound holds for LFU, notice that $\Pr[r_i \in C_j]$ is larger than the same probability when the cache size is $M = 1$. Therefore,

$$\Pr[r_i \in C_j] \geq \Pr[r_i \in C_j \mid M = 1] = \Pr[c_{j1} = r_i] = \frac{r_i^{-s}}{H_{m,s}}. \quad (3)$$

In the subsequent sections, we will use these lower bounds to prove our results.

In this paper, we will state our results in terms of order bounds. To avoid any confusion, we use the following order notations [17]. We denote $f(n) = \mathrm{O}(g(n))$ if there exist $c > 0$ and $n_0 > 0$ such that $f(n) \leq cg(n)$ for all $n \geq n_0$, $f(n) = \Omega(g(n))$ if $g(n) = \mathrm{O}(f(n))$, and $f(n) = \Theta(g(n))$ if $f(n) = \mathrm{O}(g(n))$ and $g(n) = \mathrm{O}(f(n))$.

IV. Order optimal capacity gain

In this section, we will prove that index coding can significantly decrease the number of transmissions in the network. We will specifically use the Zipfian content distribution in the underlying content distribution network. To do so, we will first state and prove the following lemma.

Lemma 1. Let's consider a Zipfian content distribution with parameter $s > 1$ and parameter $h_\epsilon = \epsilon^{\frac{1}{1-s}}$ where $0 < \epsilon < 1$. For every i with popularity index r_i that is less than h_ϵ, the request probability is at least $1 - \epsilon$.

Proof. Based on the Zipfian distribution assumption, using equation (1), the probability that the requested content has a popularity of at most h_ϵ is equal to

$$\Pr[r_j \leq h_\epsilon] = \frac{H_{h_\epsilon,s}}{H_{m,s}}. \quad (4)$$

In order to satisfy $\Pr[r_j \leq h_\epsilon] \geq 1 - \epsilon$, we should have

$$\begin{aligned} \epsilon \geq 1 - \Pr[r_j \leq h_\epsilon] &= 1 - \frac{H_{h_\epsilon,s}}{H_{m,s}} = \frac{1}{H_{m,s}} \sum_{j=h_\epsilon+1}^{m} j^{-s} \\ &= \frac{1}{H_{m,s}} \sum_{i=1}^{\lfloor \frac{m}{h_\epsilon} \rfloor - 1} \sum_{j=ih_\epsilon+1}^{(i+1)h_\epsilon} j^{-s} + \frac{1}{H_{m,s}} \sum_{j=\lfloor \frac{m}{h_\epsilon} \rfloor h_\epsilon + 1}^{m} j^{-s}. \end{aligned} \quad (5)$$

If we have

$$\epsilon \geq \frac{1}{H_{m,s}} \sum_{i=1}^{\lfloor \frac{m}{h_\epsilon} \rfloor} \sum_{j=ih_\epsilon+1}^{(i+1)h_\epsilon} j^{-s}, \tag{6}$$

then the inequality in (5) will certainly hold since the right hand side in (6) is larger than the right hand side in (5). Note that, for $ih_\epsilon + 1 \leq j \leq (i+1)h_\epsilon$ we have $j^{-s} < (ih_\epsilon)^{-s}$. Hence,

$$\sum_{j=ih_\epsilon+1}^{(i+1)h_\epsilon} j^{-s} < \sum_{j=ih_\epsilon+1}^{(i+1)h_\epsilon} (ih_\epsilon)^{-s} = h_\epsilon (ih_\epsilon)^{-s} = i^{-s} h_\epsilon^{1-s} \tag{7}$$

This means that if

$$\epsilon \geq h_\epsilon^{1-s} \frac{1}{H_{m,s}} \sum_{i=1}^{\lfloor \frac{m}{h_\epsilon} \rfloor} i^{-s}, \tag{8}$$

then (6) holds since the right hand side of inequality (6) is smaller than the right hand side of inequality (8) as shown by (7). Notice that since,

$$\frac{1}{H_{m,s}} \sum_{i=1}^{\lfloor \frac{m}{h_\epsilon} \rfloor} i^{-s} = \frac{\sum_{i=1}^{\lfloor \frac{m}{h_\epsilon} \rfloor} i^{-s}}{\sum_{i=1}^{\lfloor \frac{m}{h_\epsilon} \rfloor} i^{-s} + \sum_{\lfloor \frac{m}{h_\epsilon} \rfloor + 1}^{m} i^{-s}} < 1,$$

if h_ϵ is chosen such that

$$\epsilon \geq h_\epsilon^{1-s}, \tag{9}$$

then all of the inequalities in equations (5), (6), (7) and (8) will be valid and hence $\Pr[r_j \leq h_\epsilon] \geq 1 - \epsilon$. Therefore, in order to have $\Pr[r_j \leq h_\epsilon] \geq 1 - \epsilon$, it is enough to choose h_ϵ such that (9) is valid. Hence, if h_ϵ is chosen to be at least equal to

$$h_\epsilon = \epsilon^{\frac{1}{1-s}}, \tag{10}$$

then we have $\Pr[r_j \leq h_\epsilon] \geq 1 - \epsilon$. Notice that the choice of h_ϵ in (10) is such that it only depends on ϵ and is independent of m. □

For instance for $s = 2$ and $\epsilon = 0.01$, h_ϵ can be chosen as 100 (regardless of the size of m). This implies that for a Zipfian distribution with $s = 2$, 100 highly popular contents among any large number of contents would account for 99% of the total content requests. Therefore, if h_ϵ is chosen as in equation (10), with a probability of at least $1 - \epsilon$ all content requests have popularity index of at most h_ϵ. Now define p_ϵ as

$$p_\epsilon \triangleq \frac{h_\epsilon^{-s}}{H_{m,s}} = \frac{\epsilon^{-\frac{s}{1-s}}}{H_{m,s}}. \tag{11}$$

Based on above discussion, in our instance of index coding dependency graph, with a probability of at least $1 - \epsilon$ edges are present with a probability of at least p_ϵ. We will discuss this in more detail later in the proof for Theorem 1. Notice that for large values of m, p_ϵ is also independent of m since in that case $H_{m,s} \approx \zeta(s)$ and p_ϵ only depends on ϵ and s.

As stated in [8], if we choose the right encoding vectors for any index coding problem, for any vertex disjoint cycle in the dependency graph we can save one transmission. Therefore, the number of vertex-disjoint cycles[3] in the dependency graph can serve as a lower bound for the number of saved transmissions in any index coding problem. Number of vertex disjoint cycles is also used in [23] as a way of finding the lower bound for index coding gain. To count the number of vertex-disjoint cycles in our random dependency graph, we will use the following lemma originally proved as Theorem 1 in [12].

Lemma 2. Let $d > 1$ and $v \geq 24d$ be integers. Then any graph $G^v_{f(v,d)}$ with v vertices and at least $f(v,d) = (2d-1)v - 2d^2 + d$ edges contains d disjoint cycles or $2d-1$ vertices of degree $v - 1$.[4]

Note that the dependency graph is a directed graph and in order to use Lemma 2, we need to construct an undirected graph. Let's denote the directed and undirected random graphs on n vertices and edge presence probability p_ϵ by $\overrightarrow{G}(n, p_\epsilon)$ and $G(n, p_\epsilon)$, respectively. In a directed graph $\overrightarrow{G}(n, p_\epsilon)$, the probability that two vertices are connected by two opposite directed edges is p_ϵ^2. Therefore, we can build an undirected graph $G(n, p_\epsilon^2)$ with the same number of vertices and an edge between two vertices if there are two opposite directed edges in the directed graph $\overrightarrow{G}(n, p_\epsilon)$ between these two UTs. Hence, $\overrightarrow{G}(n, p_\epsilon)$ essentially contains a copy of $G(n, p_\epsilon^2)$. Note that there are some edges between UTs in $\overrightarrow{G}(n, p_\epsilon)$ that do not appear in $G(n, p_\epsilon^2)$. This fact was also observed in [15]. Therefore, a lower bound on the number of disjoint cycles for $G(n, p_\epsilon^2)$ implies a lower bound on the number of disjoint cycles for $\overrightarrow{G}(n, p_\epsilon)$.

In the following theorems, we will use Lemma 2 to prove that using index coding to code the contents can be very efficient.

Theorem 1. Assume all UTs are utilizing LRU or LFU caching policies for a Zipfian content request distribution with parameter $s > 1$. Index coding can save $\Omega(np_\epsilon^2)$ transmissions for any helper serving n UTs with a probability of at least $1 - \epsilon$ for any $0 < \epsilon < 1$.

Proof. Consider a Zipfian distribution with parameter $s > 1$ and let $0 < \epsilon < 1$ be fixed. The dependency graph $\overrightarrow{G}(V, E)$ in our problem is composed of n vertices $N_1, N_2, \ldots, N_n$ which correspond to the n UTs that are served by a helper. Note that the existence of an edge in dependency graph depends on the probability that a UT is requesting a content and another UT has already cached that content[5]. Therefore, this is a non-deterministic graph with some probability for the existence of each edge between the two vertices. In this non-deterministic dependency graph, the probability of existence of edge (N_i, N_j) in E is equal to the probability that content r_i requested by N_i, is already cached in N_j. Therefore, with

[3]These are the cylces that do not have any common vertex.

[4]Clearly, this lemma is valid when the number of edges is more than $f(v,d)$.

[5]This edge has no relationship with the actual physical link between two UTs.

LRU or LFU caching policy assumption and using equations (2) and (3), we arrive at

$$\Pr[(N_i, N_j) \in E] = \Pr[r_i \in C_j] \geq \frac{r_i^{-s}}{H_{m,s}}. \quad (12)$$

Using Lemma 1 for any $0 < \epsilon < 1$, if h_ϵ is chosen as $h_\epsilon = \epsilon^{\frac{1}{1-s}}$, then with a probability of at least $1-\epsilon$, any requested content has a popularity index r_i less than h_ϵ. This means that with a probability of at least $1-\epsilon$, the edge presence probability in equation (12) can be lower bounded by p_ϵ. Therefore, with a probability of at least $1-\epsilon$, maximum number of vertex-disjoint cycles in our directed dependency graph $\overrightarrow{G}(V,E)$ can be lower bounded by the maximum number of vertex-disjoint cycles in an Erdős-Réyni random graph $\overrightarrow{G}(n,p_\epsilon)$ with n vertices and edge presence probability p_ϵ. Now we can use Lemma 2 and undirected graph $G(n,p_\epsilon^2)$ as explained earlier to find a lower bound on the number of vertex disjoint cycles in $\overrightarrow{G}(n,p_\epsilon)$. This in turn, will give us a lower bound on the number of vertex-disjoint cycles in $\overrightarrow{G}(V,E)$.

Note that $G(n,p_\epsilon^2)$ is an undirected Erdős-Réyni random graph on n vertices and edge presence probability p_ϵ^2. This graph has a maximum of $n(n-1)$ undirected edges. However, since every undirected edge in this graph exists with a probability of p_ϵ^2, the expected value of the number of edges in graph $G(n,p_\epsilon^2)$ is $n(n-1)p_\epsilon^2$. This means that if d in Lemma 2 with $v=n$ is chosen to be an integer such that

$$n(n-1)p_\epsilon^2 \geq (2d-1)n - 2d^2 + d, \quad (13)$$

then on average, $G(n,p_\epsilon^2)$ will have either d disjoint cycles or $2d-1$ vertices of degree $n-1$. For the purpose of our paper we can easily verify that for large enough values of n, $d^\star = \lfloor \frac{np_\epsilon^2}{24} \rfloor$ satisfies equation (13) (Notice that the condition $24d^\star \leq n$ in Lemma 2 is also met). Therefore based on Lemma 2, the graph $G(n,p_\epsilon^2)$ either has at least $d^\star = \lfloor \frac{np_\epsilon^2}{24} \rfloor$ disjoint cycles or $2d^\star - 1 = 2\lfloor \frac{np_\epsilon^2}{24} \rfloor - 1$ vertices with degree $n-1$. As mentioned before, $\overrightarrow{G}(n,p_\epsilon)$ essentially contains a copy of $G(n,p_\epsilon^2)$. Consequently, $\overrightarrow{G}(n,p_\epsilon)$ either has at least $d^\star = \lfloor \frac{np_\epsilon^2}{24} \rfloor$ disjoint cycles or $2d^\star - 1 = 2\lfloor \frac{np_\epsilon^2}{24} \rfloor - 1$ vertices with degree $n-1$. The number of vertices in graph $\overrightarrow{G}(n,p_\epsilon)$ is n. Therefore, the latter case gives rise to a situation where there are $2d^\star - 1 = 2\lfloor \frac{np_\epsilon^2}{24} \rfloor - 1$ vertices which are connected to any other vertex in $\overrightarrow{G}(n,p_\epsilon)$ through undirected edges. This condition results in having a clique of size $2\lfloor \frac{np_\epsilon^2}{24} \rfloor - 1$ in $\overrightarrow{G}(n,p_\epsilon)$.

In summary, $\overrightarrow{G}(n,p_\epsilon)$ has either $d^\star$ disjoint cycles or it contains a clique of size $2d^\star - 1$. Hence, with a probability of at least $1-\epsilon$, the dependency graph $\overrightarrow{G}(V,E)$ on average has either $d^\star$ disjoint cycles or it contains a clique of size $2d^\star - 1$. In either of these cases $d^\star = \lfloor \frac{np_\epsilon^2}{24} \rfloor$ transmissions can be saved using index coding. This proves the theorem. □

Theorem 2. Index coding through cycle counting and clique partitioning can save $\Theta(n)$ transmissions in a network with n UTs and Zipfian content request distribution with parameter $s > 1$.

Proof. In Theorem 1, we proved that in a network with Zipfian content request distribution and for a fixed $0 < \epsilon < 1$, with a probability of at least $1-\epsilon$, the index coding dependency graph either has $d^\star = \lfloor \frac{np_\epsilon^2}{24} \rfloor$ disjoint cycles or it contains a clique of size $2d^\star - 1 = 2\lfloor \frac{np_\epsilon^2}{24} \rfloor - 1$. Consider the following situations,

1) The dependency graph has $d^\star = \lfloor \frac{np_\epsilon^2}{24} \rfloor$ disjoint cycles. In this case, for a fixed ϵ, p_ϵ is a constant which does not depend on n. Hence, cycle counting can result in at least $d^\star = \lfloor \frac{np_\epsilon^2}{24} \rfloor = \Omega(n)$ transmission savings. This is a lower bound on the number of transmission savings.
   On the other hand, notice that the maximum number of vertex-disjoint cycles in any graph with n vertices cannot be greater than $\frac{n}{2}$ as shown in Figure 3. Therefore, the maximum number of transmission savings using cycle counting is $\frac{n}{2} = O(n)$. This is an upper bound on the number of saved transmissions.
   Hence, in this case the number of saved transmissions through cycle counting is upper bounded as $O(n)$ and lower bounded as $\Omega(n)$ which proves that the number of saved transmissions scales as $\Theta(n)$ in this case.
2) The dependency graph contains a clique of size $\mathrm{k}^\star = 2d^\star - 1 = 2\lfloor \frac{np_\epsilon^2}{24} \rfloor - 1$. Through clique partitioning we will therefore be able to save at least $\mathrm{k}^\star - 1$ transmissions by only sending one linear XOR index code to the requesting UTs which form that specific clique. Hence, through clique partitioning, we will be able to save at least $\mathrm{k}^\star - 1 = 2d^\star - 2 = 2\lfloor \frac{np_\epsilon^2}{24} \rfloor - 2 = \Omega(\frac{np_\epsilon^2}{12}) = \Omega(n)$ transmissions by sending only one transmission to the UTs forming that specific clique. This is a lower bound on the number of saved transmissions.
   On the other hand, if the dependency graph is a perfectly complete graph on n nodes which means that every requested content is available in all other UTs' caches, then all the requested transmissions can be satisfied by one transmission which is a linear index code equal to XOR combination of all requested contents. In this case all the content requests can be satisfied by only one transmission. Hence, the number of transmission savings is equal to $n - 1 = O(n)$. Notice that this is the maximum number of transmission savings since we at least need 1 transmission to satisfy all content requests. This means that the number of transmission savings is upper bounded by $\mathrm{O}(n)$. Since we have already proved that the number of transmission savings is lower bounded by $\Omega(n)$, it follows that the number of transmission savings in this case will also scale as $\Theta(n)$.

□

We can further prove that many properties of the dependency graph are independent of the number of contents in the network. This implies that the properties of the dependency graph are mainly dominated by the most popular contents in the network. As an example of these properties, we can

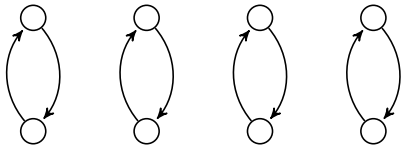

Fig. 3. Example of a dependency graph on $n = 8$ nodes with maximum possible number of vertex disjoint cycles $\frac{n}{2} = 4$.

consider the problem of finding a clique of size k in the dependency graph. A clique of size k in the dependency graph has an interesting interpretation since such a clique means that there exist a set of k UTs $N_b = \{N_{b_1}, N_{b_2}, ..., N_{b_k}\}$ such that for every $1 \leq i \leq k$ and for every $1 \leq j \leq k$, when $j \neq i$ we have $r_{b_i} \in C_{b_j}$. This means that the simple XOR-based linear index code $\sum_{i=1}^{k} M_{b_i}$ can be used by the helper to send the content M_{b_i} to UT N_{b_i} for every $1 \leq i \leq k$ in just one multicast transmission. Each UT will then be able to decode the requested content using its cached contents. The following theorem proves that the probability of existence of a clique of size k is lower bounded by a value which is independent of the total number of contents in the network, m, and only depends on the popularity index s.

Theorem 3. If LRU or LFU caching policy is used and the content request probability is Zipfian distribution, then the probability of finding a set of k UTs $N_b = \{N_{b_1}, N_{b_2}, ..., N_{b_k}\} \subseteq \mathbb{N}$ for which a single linear index code (XOR operation) can be used to transmit the requested content r_{b_i} to N_{b_i} for $1 \leq i \leq k$ can be lower bounded by a value that with a probability close to one is independent of the total number of contents in the network.

Proof. The probability that a specific set of UTs $\{N_{b_1}, N_{b_2}, ..., N_{b_k}\}$ form a clique of size k is

$$P_{b_1,b_2,...,b_k} = \Pr[r_{b_i} \in C_{b_j} \text{ for } 1 \leq \forall i, j \leq k, j \neq i\ ]. \quad (14)$$

Assuming that the UTs are requesting contents independently of each other, this probability can be simplified as

$$P_{b_1,b_2,...,b_k} = \prod_{i=1}^{k} \prod_{j=1, j\neq i}^{k} \Pr[r_{b_i} \in C_{b_j}]. \quad (15)$$

Using equations (2) and (3), we arrive at

$$\Pr[r_{b_i} \in C_{b_j}] \geq \frac{{r_{b_i}}^{-s}}{H_{m,s}}. \quad (16)$$

Equation (15) can be lower bounded as

$$P_{b_1,b_2,...,b_k} \geq \prod_{i=1}^{k} \left(\frac{{r_{b_i}}^{-s}}{H_{m,s}}\right)^{k-1}. \quad (17)$$

The probability to have a clique of size k is computed by considering all $\binom{n}{k}$ groups of k UTs. Hence, the probability of having a clique of size k denoted by P_k is given by

$$P_k = \sum_{b_1,b_2,...,b_k \subseteq N} P_{b_1,b_2,...,b_k} \geq \sum_{b_1,b_2,...,b_k \subseteq N} \prod_{i=1}^{k} \left(\frac{{r_{b_i}}^{-s}}{H_{m,s}}\right)^{k-1}$$
$$= \frac{\sum_{b_1,b_2,...,b_k \subseteq N} \prod_{i=1}^{k} {r_{b_i}}^{-s(k-1)}}{H_{m,s}^{k-1}}. \quad (18)$$

In order to simplify this expression, we use the *elementary symmetric polynomial* notation. If we have a vector $V_n = (v_1, v_2, ..., v_n)$ of length n, then the k-th degree elementary symmetric polynomial of these variables is denoted as

$$\sigma_k(V_n) = \sigma_k(v_1, ..., v_n) = \sum_{1 \leq i_1 < i_2 < .. < i_k \leq n} v_{i_1} ... v_{i_k}. \quad (19)$$

Using this notation and by defining

$$Y_n = (r_1^{-s(k-1)}, r_2^{-s(k-1)}, ..., r_n^{-s(k-1)}), \quad (20)$$

we have

$$P_k \geq \frac{\sigma_k(Y_n)}{H_{m,s}^{k-1}}. \quad (21)$$

Since the content request probability follows a Zipfian distribution, we have

$$\Pr[r_j \leq h_\epsilon] = \frac{H_{h_\epsilon,s}}{H_{m,s}}. \quad (22)$$

Therefore, for a specific group of UTs $N_{b_1}, N_{b_2}, ..., N_{b_k}$, the probability that they all request contents from the top h_ϵ most popular contents is given by

$$\Pr[r_{b_1} \leq h, ..., r_{b_k} \leq h_\epsilon] = \prod_{j=1}^{k} \Pr[r_{b_j} \leq h_\epsilon] = \left(\frac{H_{h_\epsilon,s}}{H_{m,s}}\right)^k. \quad (23)$$

We have already proved in Lemma 1 that for large values of m and $h_\epsilon = \epsilon^{\frac{1}{1-s}}$, the ratio $\frac{H_{h_\epsilon,s}}{H_{m,s}}$ is greater than $1 - \epsilon$. Besides this, the fact that n is most likely much larger than k, means that with a very high probability, for each set of UTs $\{N_{b_1}, N_{b_2}, ..., N_{b_k}\}$, the requests come only from the h_ϵ most popular contents. This implies that with a high probability, $\sigma_k(Y_n) \geq \binom{n}{k} h_\epsilon^{-ks(k-1)}$. Also, notice that

$$H_{m,s} < H_{\infty,s} = \zeta(s) < \infty. \quad (24)$$

Therefore, with a probability close to one, P_k can be lower bounded as

$$P_k \geq \binom{n}{k} \left(\frac{h_\epsilon^{-ks}}{\zeta(s)}\right)^{k-1}. \quad (25)$$

This lower bound does not depend on m and only depends on n, ϵ, s and k. □

Theorem 3 states that regardless of the number of contents in the network, there is always a constant lower bound for the probability of finding a clique of size k. The result hints the potential use of linear index coding in these networks. In the next section, we will prove that linear index coding can indeed be very useful and can be used to construct codes acheiving order optimal capacity gains.

Remark 2. The above capacity improvement is found for a traditional single hop index coding scenario. For our proposed multihop setup, similar gains still hold. In our proposed setting, we consider communications for a small number of hops and therefore multihop communication can only affect the capacity gain by a constant factor and the order bound results will not be affected.

V. Heuristics acheiving order optimal capacity

Both optimal and approximate solutions [3], [18] for the general index coding problem are NP-hard problems. Some efficient heuristic algorithms for the index coding problem were proposed [9] which can provide near optimal solutions. In some of these heuristic algorithms, the authors reduce the index coding problem to the graph coloring problem.

Notice that every clique in the dependency graph of a specific index coding problem, can be satisfied with only one transmission which is a linear combination of all contents requested by the UTs corresponding to the clique. Therefore, solving the clique partitioning problem, which is the problem of finding a clique cover of minimum size for a graph [13], yields a simple linear index coding solution. The minimum number of cliques required to cover a graph can be regarded as an upper bound on the minimum number of index codes required to satisfy the UTs. Index coding rate is defined as the minimum number of required index codes to satisfy all the UTs. Since lower index coding rates translate into higher values of transmission savings (or index coding gains)[6] as discussed in [8], the number of transmission savings found in the clique partitioning problem is in fact a lower bound on the total number of transmission savings found from the optimal index coding scheme (or the optimal index coding gain).

On the other hand, solving the clique partitioning problem for any graph $G(V, E)$ is equivalent to solving the graph coloring problem for the complement graph $\bar{G}(V, \bar{E})$ which is a graph on the same set of vertices V but containing only the edges that are not present in E. This is true because every clique in the dependency graph, gives rise to an independent set in the complement graph. Therefore, if we have a clique partitioning of size χ in the dependency graph, we have χ distinct independent sets in the complement graph. In other words, the chromatic number of the complement graph is χ.

The above argument allows us to use the rich literature on the chromatic number of graphs to study the index coding problem. In fact, any graph coloring algorithm running over the conflict graph can be directly used to obtain an achievable index coding rate. If running such an algorithm over the conflict graph results in a coloring of size χ, this coloring gives rise to a clique cover of size χ in the dependency graph and an index coding of rate χ with index coding gain of $n-\chi$ which is a lower bound for the total number of transmission savings using the optimal index code[7]. Therefore, considering the chromatic number of the conflict graph, we can find a lower bound on the asymptotic index coding gain. To do so, we use the following theorem from [5],

Theorem 4. For a fixed probability p, $0 < p < 1$, almost every random graph $G(n, p)$ (a graph with n UTs and the edge presence probability of p) has chromatic number,

$$\chi_{G(n,p)} = -\left(\frac{1}{2} + o(1)\right) \log(1-p) \frac{n}{\log n} \tag{26}$$

We will now use Theorem 4 and the designed undirected graph $G(n, p_\epsilon^2)$ to find the number of transmission savings using a graph coloring based heuristic in our network.

Theorem 5. Using a graph coloring algorithm, in a network with n UTs almost surely gives us a linear index code with gain

$$l = \Theta\left(n + \left(\frac{1}{2} + o(1)\right) \frac{n}{\log n} \log p_\epsilon^2\right). \tag{27}$$

Proof. Assume that a helper is serving n UTs where n is a large number. As discussed in Theorem 4, the index coding gain is lower bounded by $n - \chi$ where χ is the chromatic number of the conflict graph. However, notice that on average the chromatic number of our non-deterministic conflict graph is upper bounded by the chromatic number of an undirected random graph with edge existence probability of $1 - p_\epsilon^2$. To prove this, notice that in the dependency graph, the probability of edge existence between two vertices is at least p_ϵ which implies that the number of edges in the dependency graph is on average greater than or equal to the number of edges in a directed Erdos-Reyni random graph $\overrightarrow{G}(n, p_\epsilon)$. However, we know that the number of edges in $\overrightarrow{G}(n, p_\epsilon)$ is at least equal to the number of edges in an undirected Erdos-Reyni random graph $G(n, p_\epsilon^2)$. Therefore, the conflict graph which is the complement of dependency graph, on average has less edges compared to a random graph with edge existence probability of $1 - p_\epsilon^2$ and consequently, its chromatic number cannot be greater than the chromatic number of $G(n, 1 - p_\epsilon^2)$. Given these facts, the index coding gain is lower bounded by $n - \chi_{G(n,1-p_\epsilon^2)}$. Since $1 - p_\epsilon^2$ is fixed, Theorem 4 implies that the chromatic number of the conflict graph is equal to

$$\chi_{G(n,1-p_\epsilon^2)} = -\left(\frac{1}{2} + o(1)\right) \log p_\epsilon^2 \ \frac{n}{\log n} \tag{28}$$

This proves that the index coding gain is lower bounded by $\Omega(n \times \{1 + \left(\frac{1}{2} + o(1)\right) \frac{1}{\log n} \log p_\epsilon^2\})$ which asymptotically tends to n. However, the maximum index coding gain of n UTs is also n. Therefore, this coding gain is also a tight bound. □

Remark 3. Theorem 5 presents the index coding gain using a graph coloring algorithm which only counts the number of cliques in the dependency graph. The gain in Theorem 1 counts the number of disjoint cycles in the dependency graph. Theorem 2 proves that index coding gain in Theorem 1 is $\Theta(n)$ which means that it is order optimal. Theorem 5 is also proving the same result. Therefore, a graph coloring algorithm can acheive order optimal capacity gains.

Remark 4. We have shown our proposed algorithm in pseudo-code in Algorithm 1. As we mentioned earlier, in our solution we only focus on content delivery. For cache placement, we assume that the greedy approximate algorithm proposed in

[6] In a dependency graph of n UTs with the index coding rate of χ, the number of saved transmissions, $n - \chi$, is called the index coding gain.

[7] Notice that since the optimal index coding rate is upper bounded by the size of the minimum clique cover (which is equal to the chromatic number of the conflict graph), the value of transmission savings that we can achieve using the optimal index code is lower bounded by $n - \chi$.

Algorithm 1 Multihop Caching-Aided Coded Multicasting

procedure CACHE PLACEMENT
  Use greedy cache placement algorithm in [14] to populate helper caches.
end procedure
procedure HELPER ASSIGNMENT
  Use a greedy algorithm to assign the closest helper which has the requested content, to the UT.
end procedure
procedure CONTENT DELIVERY
  Form the dependency graph and conflict graph.
  Color the conflict graph by a graph coloring heuristic to find independent sets in dependency graph.
  for every independet set S found **do**
    Helper multicasts coded XOR of requested contents $\oplus_{k\in S}\ M_{r_k}$.
  end for
end procedure
procedure CONTENT DECODING
  UT j XORs some of it's cached contents with the received coded content $\oplus_{k\in S}\ M_{r_k}$ to decode M_{r_j}
end procedure

[14] is used to populate helper caches. For helper assignment, the greedy algorithm is used. In other words, we suggest that the closest helper which has the content in its cache be assigned to the UT. During the content delivery phase, we use our proposed heuristic for index coding. We use a graph coloring heuristic for the conflict graph and find independent sets in the dependency graph. Then for each independent set only one multicast transmission is needed which will be sent by the helper. Our main contribution is to propose better content delivery algorithm compared to the baseline. In the decoding phase, UTs use their cached contents and the received content to decode their desired contents.

VI. SIMULATIONS

In this section, we will show our simulation results. To show the performance of our coding technique, we have plotted the simulation results for five different sets of parameters in Figure 4. In this simulation, we assume that index coding is done in the packet level. We plotted the average packets sent in each transmission. We have assumed that the UTs are requesting contents based on a Poisson distribution with an average rate of $\frac{1}{n}$. This way we can assure that the average total request rate is one and we are efficiently using the time resource without generating unstable queues. The optimum solution is an NP-hard problem. However, we used a very simple heuristic algorithm to count the number of cliques and cycles of maximum size 4. Even with this simple algorithm, we were able to show that the index coding can double the average number of packets per transmission in each time slot for certain values of the Zipfian parameter. Clearly, optimal index coding or more sophisticated algorithms can achieve better results compared to what we obtained by our simple algorithm.

For small values of s, the content request distribution is close to uniform, the dependency graph is very sparse and there is little benefit of using index coding. For large values of s, most UTs are requesting similar contents which results in broadcasting the same content to all nodes which is equal to one content per transmission. The main benefit of index coding happens for values of s between 0.5 and 2 which is usually the case in practical networks. Note that a wireless distributed caching system with no index coding, will always have one content per transmission.

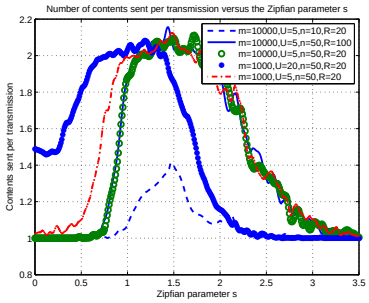

Fig. 4. Average number of contents transmitted in each time slot when using index coding as a function of the Zipfian parameter s using a suboptimal search.

Figure 5 compares the average number of requests satisfied per time slot by a helper between our proposed scheme and the baseline approach. For this simulation, a Zipfian content request distribution with parameter $s = 2$ is assumed. The cell radius is 400 meters, D2D transmission range is assumed to be 100 meters and 1000 UTs are considered in this figure similar to the simulations in [14] and Figure 1. Notice that with the baseline approach at least 27 helpers are required to cover the whole network nodes and 24, 20, 14 and 10 helpers can cover 97%, 93%, 78% and 62% of the network nodes, respectively. Using multihop D2D communications with a maximum of three hops, 4 helpers can cover the whole network while 3 helpers can cover 95% of the network nodes. We have shown that even with our very simple heuristic algorithm, multihop D2D can significantly improve the helper utilization ratio. Note that in baseline approach, each helper can at most transmit one content per transmission, however, our approach can satisfy more than one request per transmission by taking advantage of the side information that is stored in nodes' caches. As the number of requests per user increases, there are more possibilities of creation of cliques of large size which results in increasing the efficiency of helper nodes. The simulation up to probability of 0.3 is ready and attached.

VII. CONCLUSION

An efficient order optimal content delivery approach is proposed for future wireless cellular systems. We take advantage of femtocaches [14] and multihop communication using index coding. We proved that using index coding is very efficient technique by taking advantage of side information stored in UTs. Further, it was shown that under Zipfian content distribution, linear index coding could be order optimal. A heuristic graph coloring algorithm is proposed that achieves

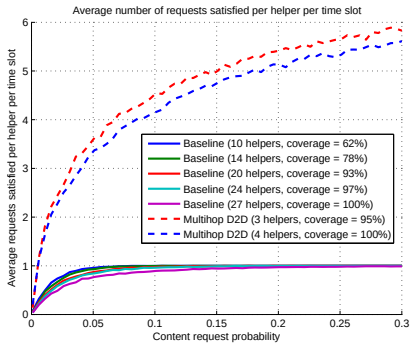

Fig. 5. Comparing our proposed and baseline solution on average number of requests satisfied per time slot per helper versus the content request probability.

order optimal capacity bounds. Our simulation result demonstrates the gains that can be achieved with this approach.

Our proposed scheme improves the efficiency of femtocaches by taking advantage of multihop communications and index coding. One challenge of multihop communication is connectivity when nodes are moving fast. One solution is to divide the contents into equal smaller chunks and treat each chuck as a content. Another challenge is the additional delay imposed by using multihop communications. However, it is not readily clear that baseline approach provides lower delays. The reason is the fact that by using multihop communication, larger portion of the cell can be covered when the same number of femtocaches are used in both approaches as shown in Figure 1. A more detailed comparison of mobility and delay in single hop and multihop communication is the subject of future study. Security, overhead, routing are other issues that should be investigated in future work.

REFERENCES

[1] http://spectrum.ieee.org/telecom/wireless/millimeter-waves-may-be-the-future-of-5g-phones.

[2] Amendments in IEEE 802.11ad™ enable multi-gigabit data throughput and groundbreaking improvements in capacity. https://standards.ieee.org/news/2013/802.11ad.html, 2013. [Online; accessed 9-October-2015].

[3] Ziv Bar-Yossef, Yitzhak Birk, TS Jayram, and Tomer Kol. Index coding with side information. *Information Theory, IEEE Transactions on*, 57(3):1479–1494, 2011.

[4] Federico Boccardi, Robert W Heath, Aurelie Lozano, Thomas L Marzetta, and Petar Popovski. Five disruptive technology directions for 5G. *Communications Magazine, IEEE*, 52(2):74–80, 2014.

[5] Béla Bollobás. The chromatic number of random graphs. *Combinatorica*, 8(1):49–55, 1988.

[6] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. Web caching and Zipf-like distributions: Evidence and implications. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 126–134, New York, NY, 1999. IEEE.

[7] Vikram Chandrasekhar, Jeffrey G Andrews, and Alan Gatherer. Femtocell networks: a survey. *Communications Magazine, IEEE*, 46(9):59–67, 2008.

[8] Mohammad Asad R Chaudhry, Zakia Asad, Alex Sprintson, and Michael Langberg. On the complementary index coding problem. In *Proceedings of Information Theory (ISIT), IEEE International Symposium on*, pages 244–248, Saint Petersburg, Russia, 2011. IEEE.

[9] Mohammad Asad R Chaudhry and Alex Sprintson. Efficient algorithms for index coding. In *INFOCOM Workshops, IEEE*, pages 1–4, Phoenix, AZ, 2008. IEEE.

[10] Michelle Effros, Salim El Rouayheb, and Michael Langberg. An equivalence between network coding and index coding. *Information Theory, IEEE Transactions on*, 61(5):2478–2487, 2015.

[11] Salim El Rouayheb, Alex Sprintson, and Costas Georghiades. On the index coding problem and its relation to network coding and matroid theory. *Information Theory, IEEE Transactions on*, 56(7):3187–3195, 2010.

[12] P Erdős and L Pósa. On the maximal number of disjoint circuits of a graph. *Publ. Math. Debrecen*, 9:3–12, 1962.

[13] Michael R Garey and David S Johnson. *Computers and intractability*. W. H. Freeman, San Francisco, 2002.

[14] Negin Golrezaei, Karthikeyan Shanmugam, Alexandros G Dimakis, Andreas F Molisch, and Giuseppe Caire. Femtocaching: Wireless video content delivery through distributed caching helpers. In *INFOCOM, Proceedings IEEE*, pages 1107–1115, Orlando, FL, 2012. IEEE.

[15] Ishay Haviv and Michael Langberg. On linear index coding for random graphs. In *Information Theory Proceedings (ISIT), IEEE International Symposium on*, pages 2231–2235, Cambridge, MA, 2012. IEEE.

[16] Mingyue Ji, Giuseppe Caire, and Andreas F Molisch. Fundamental limits of caching in wireless D2D networks. *Information Theory, IEEE Transactions on*, 62(2):849–869, 2016.

[17] Donald E Knuth. Big omicron and big omega and big theta. *ACM Sigact News*, 8(2):18–24, 1976.

[18] Michael Langberg and Alex Sprintson. On the hardness of approximating the network coding capacity. *Information Theory, IEEE Transactions on*, 57(2):1008–1014, 2011.

[19] Derek Leong, Tracey Ho, and Rebecca Cathey. Optimal content delivery with network coding. In *Information Sciences and Systems, CISS, 43rd Annual Conference on*, pages 414–419, Baltimore, MD, 2009. IEEE.

[20] Jaime Llorca, Antonia M Tulino, Ke Guan, and Daniel Kilper. Network-coded caching-aided multicast for efficient content delivery. In *Communications (ICC), IEEE International Conference on*, pages 3557–3562, Budapest, Hungary, 2013. IEEE.

[21] Mohammad Ali Maddah-Ali and Urs Niesen. Fundamental limits of caching. *Information Theory, IEEE Transactions on*, 60(5):2856–2867, 2014.

[22] Marie-Jose Montpetit, Cedric Westphal, and Dirk Trossen. Network coding meets information-centric networking: an architectural case for information dispersion through native network coding. In *Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications*, pages 31–36, Hilton Head, SC, 2012. ACM.

[23] Michael J Neely, Arash Saber Tehrani, and Zhen Zhang. Dynamic index coding for wireless broadcast networks. *Information Theory, IEEE Transactions on*, 59(11):7525–7540, 2013.

[24] Tuan Tran, Thinh Nguyen, Bella Bose, and Vinodh Gopal. A hybrid network coding technique for single-hop wireless networks. *Selected Areas in Communications, IEEE Journal on*, 27(5):685–698, 2009.

[25] Qinghua Wu, Zhenyu Li, and Gaogang Xie. Codingcache: multipath-aware CCN cache with network coding. In *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-Centric Networking*, pages 41–42, Hong Kong, 2013. ACM.