UC Santa Cruz
UC Santa Cruz Previously Published Works

Title
Application of Index Coding in Information-Centric Networks

Permalink
https://escholarship.org/uc/item/1575p169

ISBN
978-1-4799-6959-3

Authors
Kiskani, Mohsen Karimzadeh
Sadjadpour, Hamid

Publication Date
2015-02-01

DOI
10.1109/iccnc.2015.7069479

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Application of Index Coding in Information-Centric Networks

Mohsen Karimzadeh Kiskani[†] and Hamid Sadjadpour[‡]
Department of Electrical Engineering, University of California, Santa Cruz[†]
Suns-Tech Corporation, Milpitas, California[‡]
{mohsen}@soe.ucsc.edu, {hamid}@suns-tech.com

***Abstract*—The index coding problem relates to transmission policies when the source node broadcasts encoded data to users with side information. This paper extends the index coding problem to cases when the source node can reach users through multihop communications. The new approach is called Modified Index Coding (MIC) which can be applied to both wireless and wired networks. We demonstrate the benefits of our approach by applying MIC to Information-Centric Networks (ICN). We demonstrate that the combination of ICN and MIC requires a hybrid caching scheme that includes both central and distributed caching to support two different goals. The approach results in a combination of conventional caching in ICN and a new distributed caching scheme across nodes in the network. Our analysis demonstrates that capacity improvement can be achieved by the new architecture. Simulation results compare the capacity improvement to traditional ICN architecture.**

I. Introduction

Index Coding (IC) [1], [2] focuses on wireless architectures in which the transmitter utilizes coding in order to take advantage of broadcast nature of the wireless channel. For example in satellite communications, combination of coding at the transmitter side with caching at the receivers achieves higher capacity gains compared to traditional schemes. The original IC problem was based on the assumption that the channel is a broadcast channel and the source can reach all the nodes in one transmission. We will modify this concept to accommodate other classes of channels such as wired or multihop wireless networks. Figure 1 demonstrates an example of IC problem with source having six messages and each node N_1 to N_6 needs one message and has a subset of all messages. For example, node N_4 needs message m_4 while it has prior side information m_5 and m_6. The objective is to find an optimal encoding scheme that allows all nodes to receive their required messages with minimum number of transmissions. If all the nodes in figure 1 are within transmission range of the source, then the source can send only two encoded messages of $m_1 + m_2 + m_3$ and $m_4 + m_5 + m_6$ to allow all nodes to retrieve their requested messages instead of six transmissions. For example, node N_4 can add m_5 and m_6 to encoded message $m_4 + m_5 + m_6$ in order to recover m_4.

Information Centric Network (ICN) architectures [3]–[8] were introduced based on the premise that in most Internet applications, users are interested in accessing the content regardless of the location of delivery as long as the information is secure. ICN focuses on the content delivery without any consideration on where the content is obtained. The key question that ICN attempts to address is how to securely deliver huge amount of contents that are distributed in different locations and requested by many users. ICN addresses this question by utilizing a naming architecture where the content is retrieved by its name and defining a naming taxonomy that makes the content independent of its source or location. Further, it allows the contents to be cached in the network, preferably close to the destinations. This unique content recovery requires content-based routing in order to find the content in the network using appropriate name resolution infrastructure to map a name to one copy of the content. This new approach has provided significant benefits for content delivery in the network at the expense of additional overhead to keep track of content locations in different caches. A natural question that we intend to address is that "how can we take advantage of caching in ICN and modify IC concept in order to increase the overall throughput capacity of the network?"

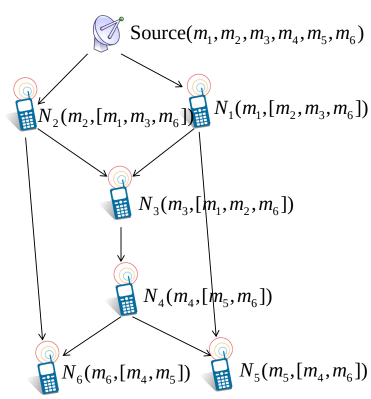

Fig. 1. *An example of index coding problem. In this figure, the source node can either broadcast to all the nodes (IC problem) or communicate with nodes using multihop communications (MIC problem).*

The rest of the paper is organized as follows. In Section II, we focus on defining the modified index coding (MIC) and the new hybrid caching schemes. Section III describes the proposed ICN architecture. In section IV, we prove that MIC can be efficiently utilized to increase the capacity of the ICN. Simulation results in section V demonstrate the throughput capacity improvement by combining MIC with ICN. Section VI concludes the paper.

II. Model and Problem formulation

A. Modified Index Coding (MIC)

The IC problem [1], [2] was originally formulated for wireless broadcast channels such as satellite communication applications. In this paper, we assume the source node can access the receiver nodes through multihop communications. The new scheme, called Modified Index Coding (MIC) [9] technique, can be applied to different types of networks such as wireless multihop communication network, wired network or a hybrid network consisting of both wired and wireless channels. For example, in a corporate environment, we can have information transported in a wired medium while the last hop is a wireless access point with nodes connected to the infrastructure through a wireless modem. Such scenarios are becoming more common in medium and large size corporate campuses.

Definition 1. *A Modified Index Code* $\mathbb{MIC}(\mathbb{M},\mathbb{N})$ *consists of a set of k messages* $\mathbb{M} = \{m_1, \ldots, m_k\}$ *and a set of receiver nodes* $\mathbb{N}$*. Each node* N_i *stores a subset of messages called* $\mathbb{L}_i \subseteq \mathbb{M}$ *and requests one message*[1] m_i *that node* N_i *does not have, i.e.,* $m_i \in M$ *and* $\mathbb{L}_i \subseteq \mathbb{M} \setminus \{m_i\}$*. Each message can be divided into* n *packets, i.e.,* $m_i = \{m_{i1}, \ldots, m_{in}\}$*. Each packet also belongs to an alphabet taken from a q-ary finite Field* Γ*. Therefore, we have* $m_i = \{m_{i1}, \ldots, m_{in}\} \in \Gamma^n$*. We further define* $\aleph = \{m_{11}, \ldots, m_{1n}, \ldots, m_{k1}, \ldots, m_{kn}\} \in \Gamma^{nk}$*. At any given time,* $k_1 \leq k$ *receiver nodes are requesting some messages. The source defines groups of receiver nodes* $k_1^1, \ldots, k_1^p$*, where* $k_1^1 + \ldots + k_1^p = k_1$*. Modified index code for* $\mathbb{MIC}(\mathbb{M},\mathbb{N})$ *is a function* $f_i : \Gamma^{nk} \to \Gamma^{\ell_i}$*, for an integer value of* ℓ_i *and groups of nodes* k_1^i *that satisfies for each receiver node* $N_i = (m_i, \mathbb{L}_i) \in \mathbb{N}$ *within this group* k_1^i *a function* $\Phi_i : \Gamma^{\ell_i + n|\mathbb{L}_i|} \to \Gamma^n$ *such that the desired message can be decoded for that particular node, i.e.,* $\Phi_i(f_i(\aleph), \mathbb{L}_i) = m_i, \forall \aleph \in \Gamma^{nk}$*.*

Note that in this new definition, we multicast different coded messages to these p groups, each one consists of k_1^i receiver nodes where $1 \leq i \leq p$. IC definition subsumes MIC since broadcast is a special case of multicast when the set of receiver nodes include the entire network. Under the new definition, we can apply MIC for both wireless and wired networks. MIC is utilized to design a new ICN architecture.

Figure 1 can be used for an example of MIC. Suppose that any two nodes can communicate only when there is an arrow between them. For instance, the source node in Figure 1 can only communicate with nodes N_1 and N_2. In this example, $p = 2$ and $k_1^1 = \{N_1, N_2, N_3\}$ and $k_1^2 = \{N_4, N_5, N_6\}$. The source node multicasts two encoded messages of $m_1 + m_2 + m_3$ and $m_4 + m_5 + m_6$ to k_1^1 and k_1^2 groups respectively. These two encoded messages are the minimum number of transmissions (optimum) that will achieve the desired outcome. However, in general the problem of finding the best encoding strategy is an NP-hard problem.

[1] Extension of this definition to multiple requests is straightforward.

B. Hybrid Caching

One of the main features of ICN is the ability of the network to cache the requested contents in the network at different locations in order to serve the users with lower latency and improve the throughput capacity in the process by bringing the contents closer to the users. This feature seems to be very attractive in separating the content from any unique source node in order to find the nearest content to the client node. However, this by itself creates certain challenges for network designers. One major challenge is how to locate the closest cached content in the network? Another challenge is to design a caching policy that will increase the throughput capacity while reducing the latency. We introduce a new hybrid caching technique that requires minimum overhead related to locating stored cache contents.

In our architecture, we use caching for two different purposes. We cache the contents in some locations in the network in order to provide it to users similar to the original approach in ICN. However, most users have significant storage capacity that is not used. For example, it is now common for a laptop to have Terabyte of storage not utilized by majority of the users. We use this enormous distributed storage capacity for improving the data distribution in ICN architecture. We propose that each user shall keep any data object that is requesting from the network. Therefore, each user allocates a predefined portion of its storage to keep the data objects that it has already obtained. By the discussion that we had in the previous section related to MIC, it should be clear that the data stored by different users throughout the network will be used to extract the desired message when the node receives a combination of multiple messages. The encoded message is multicasted by the local cache system that is serving these nodes. The contents that are cached by the nodes will not be used for transmission to other requesting nodes unlike current ICN architectures. The problem with distributed caching is the significant overhead associated to this approach. We suggest that the requested contents by different users should always come from the local cache system or from source node that has the content. In our proposed architecture, caching are used for two different purposes as described above.

Note that by taking advantage of the MIC, we need one multicast session to replace k_1^i unicast sessions. It is easy to see [10] that one multicast session always consumes less channel bandwidth in the network than k_1^i unicast sessions. This reduction in network resource usage can be very helpful specially when the size of contents are large. Another advantage of this architecture is the fact that since the local router that caches the contents knows what contents each node has stored before, there is no need to update each cache (node) information in the network. As long as the local router can keep track of the cached contents in different nodes, then the overhead is very small. Note that since each node receives the requested content via this local router, then that information is already available to the local router.

III. Proposed ICN Architecture

In this section, we will describe how to take advantage of the MIC concept in order to derive a new architecture for ICN. We assume each group of nodes in the network is served by a unique router that also caches the information. In this context, if a node requests a content, this request will be directed toward that particular router. The router either sends the information directly to the node or finds the source for the requested content. Further, we assume when a node receives a content, it will keep this content in its cache. In Figure 2, all the routers that are shaded are responsible for serving different groups of nodes. The selection of these routers is based on the number of nodes that are connected to that router either directly or through multiple hops. In general, there is a trade-off between latency, speed of router and the maximum number of nodes assigned to a router.

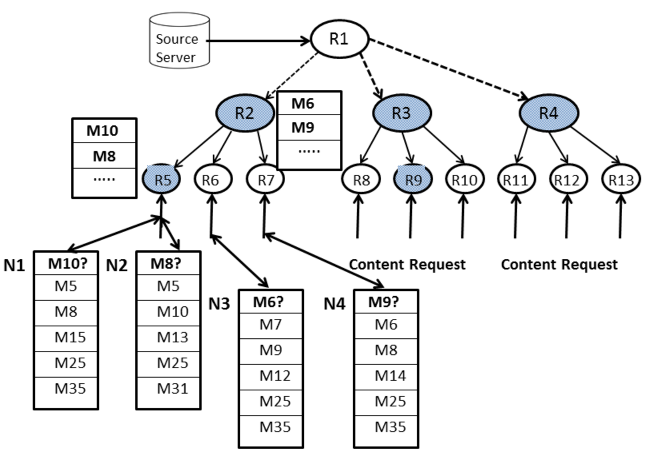

Fig. 2. *Example of serving nodes by routers. Router R_5 serves nodes N_1 and N_2 and router R_2 serves nodes N_3 and N_4. Solid lines represent one hop and dotted lines represent multiple hops.*

When a content is delivered to a node, the node will keep a copy of this content in its cache. Now let's assume each node has a subset of contents in its cache (see figure 2). When some of these nodes request different[2] contents from the local router, this router network encodes [11] the requested contents by utilizing modified index coding technique to minimize the total number of transmissions to serve all these nodes. The optimum encoding selection is an NP-hard problem. We will introduce some sub-optimal approach for the encoding scheme.

If the content is not available in the serving router, then the router will request the content from the source (or another router on the way toward the source). Once it receives the content, it will encode the received content along with other requested contents and multicast it to the requesting nodes. This router also keeps one copy of the content in its cache. As we can see, under the new architecture, we do not use an aggressive caching approach that each router or node caches the contents but rather a subset of the routers cache the contents. The assumption here is that most of the contents that are being requested by a node in each group of nodes, will likely be requested by another node in that group. This is particularly true since most of the content request popularity are heavy-tailed and have a distribution close to Zipf distribution. Prior studies [12], [13] have shown that multiple layer caching or cooperative caching does not provide significant improvement for Zipfian distribution. Recent study [14] has suggested caching scheme that takes advantage of this distribution and caches at the edges of the network. Our approach has some similarities to the technique proposed in [14] by suggesting that it is sufficient to cache in a subset of routers at the edges of the network.

Figure 2 demonstrates an example for our proposed ICN architecture. Let nodes N_1 and N_2 request messages m_{10} and m_8 respectively. Both these nodes are served by router R_5. When these nodes send request to this router, the router will multicast $m_{10}+m_8$ to these two nodes. Node N_1 can add the received encoded message with m_8 to obtain m_{10} and node N_2 can similarly obtain m_8. Nodes N_3 and N_4 are served by router R_2 via routers R_6 and R_7 respectively. R_2 multicasts m_6+m_9 and each node can retrieve its requested data. All these operations are carried in Galois Field. It is quite possible that more complicated combinations of messages are sent by routers in order for nodes to decode their requested messages.

As long as the caching policies of the users are known by the router, the router knows for each user which contents are being stored and which contents are evicted after the user reaches its maximum caching capacity. This clearly requires additional processing power for each router that is involved in caching but it also reduces the overhead. There exists another overhead associated to a node requesting a content. Since each user has an assigned local router to serve that user, the request is always directed toward that router. Clearly, the overhead associated to finding the content in the network by the local router is similar to the current network architectures. Therefore, our proposed architecture simplifies the overhead and content routing challenges in ICN systems.

IV. ICN Capcity Improvement using MIC

In this section, we study the problem of capacity improvement in the proposed combination of ICN architecture and MIC. We assume the content popularity follows a Zipfian-like distribution which is supported by many studies [15], [16]. MIC provides additional capacity gain when a subset of contents have higher popularity among nodes such as in Zipfian-like distribution.

In the remainder of this section, we assume that the network is a hybrid network with the last hop is between a wireless router (like 802.11) and mobile users. Similar to many studies on index coding (IC) [1], [17], [18] that demonstrate dependency graph is a useful tool for analysis of these networks, we take advantage of this concept in this paper.

[2]Some nodes can request the same content.

Definition 2. (Dependency Graph): *Given an instance of index coding problem, the dependency graph $G(V,E)$ is defined as follows:*

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

It is known from [17] that if we choose the right encoding vectors, for any vertex disjoint cycle in the dependency graph we can save at least one transmission. Therefore, the number of vertex-disjoint cycles in the dependency graph can serve as a lower bound for the number of saved transmissions in any IC problem. The same result also holds for an MIC problem since MIC is similar to the IC problem and a dependency graph can be defined for each subnetwork.

Assume that we have an ICN system that is utilizing MIC. Let's assume the set of contents available in the entire network as $M = \{m_1, m_2, ..., m_k\}$ with m_1 being the most popular content and m_k being the least popular content in the network[3]. Also, assume that the users $N = \{N_1, N_2, ..., N_l\}$ are being served by a specific router R and user N_i is requesting content with popularity index r_i in the current time instant. For the sake of simplicity of calculations, let's assume that each user has a cache of fixed size δ in which, the contents with indices $C_i = \{c_{i1}, ..., c_{i\delta}\}$ are stored.

As suggested by [15], [16], [19], we can assume a Zipfian distribution with parameter s for content popularity distribution in the network. This means that the probability that N_i requests any content with popularity index r_i at any time instant is

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

where $H_{k,s}$ is the k^{th} generalized harmonic number with parameter s defined as

$$H_{k,s} = \sum_{j=1}^{k} \frac{1}{j^s}. \quad (2)$$

Lemma 1. *When $s > 1$, for every $0 < \epsilon < 1$, there exists an integer $h = \Theta(1)$ with respect to k such that for every i,*

$$Pr[r_i \le h] \ge 1 - \epsilon. \quad (3)$$

Proof. Based on the Zipfian distribution assumption, this probability is qual to

$$\Pr[r_i \le h] = \frac{H_{h,s}}{H_{n,s}}. \quad (4)$$

If $s > 1$, we have $H_{n,s} < H_{\infty,s} = \zeta(s)$ where $\zeta(.)$ denotes the Reimann Zeta function. If we choose h to be the first integer such that $H_{h,s} \ge \zeta(s)(1-\epsilon)$, we are guaranteed to have $\Pr[r_i \le h] \ge 1-\epsilon$. Notice that h can be chosen independently of n, i.e., $h = \Theta(1)$. □

[3]Popularity decreases with index number.

Remark 1. *If $0 \le s \le 1$, to make sure that $Pr[r_i \le h] \ge 1-\epsilon$, the value of h should grow with n but the growth rate is so slow that we can still treat h as a constant number with respect to n and use lemma 1 for practical purposes.*

Therefore, based on lemma 1 and remark 1, if h is chosen a sufficiently large integer, with probability close to one, all users are requesting contents with maximum popularity index h. Before going further, we will bring up the following lemma from [20].

Lemma 2. Let $d_c > 1$ and $l \ge 24d_c$, then any graph $G^l_{f(l,d_c)}$ with l nodes and at least $f(l,d_c) = (2d_c-1)l - 2d_c^2 + d_c$ edges contains d_c disjoint cycles or $2d_c - 1$ vertices of degree $l-1$ (its structure is then uniquely determined). [4]

Proof. The proof is in [20]. □

The fact that there are strong correlation between cached contents and new requests due to Zipfian distribution of contents, it is clear that MIC will provide some capacity improvement. We will now demonstrate the efficiency of applying MIC to ICN in the following theorem.

Theorem 1. *For large values of h and l, using MIC in ICN can save on average $\Omega(lp_0)$ transmissions for any router serving l nodes in any time slot where*

$$p_0 = \frac{h^{-s}}{H_{k,s}}. \quad (5)$$

Proof. The dependency graph $G(V,E)$ in our problem is composed of l vertices $N_1, N_2, ..., N_l$ which corresponds to the l nodes served by a local router. Note that the existence of an edge in dependency graph depends on the probability that a node is requesting a content that another node has already cached. Therefore, this is a non-deterministic graph with some probability for the existence of each edge between any 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 the content r_i requested by N_i is cached in node N_j. If we assume these two probabilities are independent, then the probability of a directed edge (N_i, N_j) in E is at least

$$\Pr[(N_i, N_j) \in E] \ge \frac{r_i^{-s}}{H_{k,s}}. \quad (6)$$

Using lemma 1 and remark 1, when the value of h is large enough, the probability that r_i is less than h is very close to one. This means that with a probability close to one, the edge presence probability in equation (6) can be lower bounded by p_0. Therefore, the maximum number of vertex-disjoint cycles in $G(V,E)$ can be lower bounded by the maximum number of vertex-disjoint cycles in an Erdős-Réyni random graph $G'(l,p_0)$ with l nodes and edge presence probability p_0. Now we can use lemma 2 to find a lower bound on the number of vertex disjoint cycles in $G'(l,p_0)$. This in turn will

[4]Clearly, the theorem is valid when the number of edges is more than $f(l,d_c)$.

give us a lower bound on the number of vertex-disjoint cycles in $G(V,E)$.

Notice that $G'(l,p_0)$ is an Erdős-Réyni random graph and it can have a maximum of $l(l-1)$ directed edges. However, since every edge in this graph exists with a probability of p_0, the expected value of the number of edges is $l(l-1)p_0$. This means that if d_c is chosen to be an integer such that

$$l(l-1)p_0 \geq (2d_c-1)l - 2d_c^2 + d_c, \tag{7}$$

then on average, $G'(l,p_0)$ will have d_c disjoint cycles. We can easily verify that $d_c = \frac{lp_0}{2}$ satisfies (7). Therefore, with a probability close to one (for large enough values of h), the dependency graph $G(V,E)$, on average has at least $\Omega(lp_0)$ vertex disjoint cycles. This can be directly applied to prove the theorem. □

Theorem 2. *The acheived lower bound in theorem 1 is a tight order bound of* $\Theta(l)$.

Proof. Notice that the maximum number of vertex-disjoint cycles in any graph with l vertices is $\frac{l}{2}$. However, theorem 1 proves that the maximum number of vertex-disjoint cycles in our graph is lower bounded by $\Omega(lp_0)$. This suggests that this is indeed a tight order bound. □

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 consider the problem of finding a clique of size k_1 in the dependency graph. A clique of size k_1 in the dependency graph has an interesting interpretation since such a clique means that there exist a set of k_1 users $N_b = \{N_{b_1}, N_{b_2}, ..., N_{b_{k_1}}\}$ such that for every $1 \leq i \leq k_1$ and every $1 \leq j \leq k_1$ when $j \neq i$, we have $r_{b_i} \in C_{b_j}$. This means that a simple linear index code $\sum_{i=1}^{k_1} m_{b_i}$ can be used by the local router to send the content m_{b_i} to user N_{b_i} for all $1 \leq i \leq k_1$ in just one transmission. Each user will then be able to decode the requested message using its cached contents. The following theorem proves that the existence of a clique of size k_1 is independent of the total number of contents in the network k and only depends on the popularity index s.

Theorem 3. *Let's define the content index requested by node i in j^{th} cache space as $c_{i,j} \in C_i$. We assume the content request probability follows a Zipfian distribution and the users request independent contents in different time slots*[5]. *Then the probability of finding a set of k_1 users $N_b = \{N_{b_1}, N_{b_2}, ..., N_{b_{k_1}}\} \subseteq \mathbb{N}$ for which a single index code can be used to transmit the requested content m_{b_i} with index number r_{b_i} to N_{b_i} for $1 \leq i \leq k_1$ is independent of the total number of contents in the network.*

Proof. The probability that a specific set of users $\{N_{b_1}, N_{b_2}, ..., N_{b_{k_1}}\}$ form a clique of size k_1 is given by

$$P_{b_1,b_2,...,b_{k_1}} = \text{Pr}[r_{b_i} \in C_{b_j} \text{ for } 1 \leq \forall i,j \leq k_1, j \neq i\ ], \tag{8}$$

where C_{b_j} is the set representing the contents that node N_{b_j} caches. Assuming that the users are requesting contents independently of each other, this probability can be simplified as

$$P_{b_1,b_2,...,b_{k_1}} = \prod_{i=1}^{k_1} \prod_{j=1, j\neq i}^{k_1} \text{Pr}[r_{b_i} \in C_{b_j}]. \tag{9}$$

Since the contents that are requested by each user in different time slots are independent of each other, in the steady state when the caches of all nodes are filled we have

$$\text{Pr}[r_{b_i} \in C_{b_j}] = 1 - \text{Pr}[r_{b_i} \notin C_{b_j}] = 1 - \prod_{u=1}^{\delta} \text{Pr}[r_{b_i} \neq c_{b_j,u}]. \tag{10}$$

Notice that we assume a Zipfian distribution for the content request in the ICN network, therefore, $\text{Pr}[r_{b_i} = c_{b_j,u}]$ will be equal to the probability that user N_{b_j} requests content with index r_{b_i} in u^{th} cache space. Hence

$$\text{Pr}[r_{b_i} = c_{b_j,u}] = \frac{{r_{b_i}}^{-s}}{H_{n,s}} \tag{11}$$

Therefore, combining equations (11) and (10), we arrive at

$$\text{Pr}[r_{b_i} \in C_{b_j}] = 1 - \left(1 - \frac{{r_{b_i}}^{-s}}{H_{n,s}}\right)^{\delta}. \tag{12}$$

Equation (9) can be simplified as

$$P_{b_1,b_2,...,b_{k_1}} = \prod_{i=1}^{k_1} \left(1 - \left(1 - \frac{{r_{b_i}}^{-s}}{H_{n,s}}\right)^{\delta}\right)^{k_1-1}. \tag{13}$$

In order to have a clique of size k_1, we need to include the possibility over all $\binom{l}{k_1}$ groups of k_1 users. Therefore, the probability of having a clique of size k_1 (which we denote by P) should be summed up over all of these choices.

$$\begin{aligned} P &= \sum_{b_1,b_2,...,b_{k_1} \subseteq \mathbb{N}} P_{b_1,b_2,...,b_{k_1}} \\ &= \sum_{b_1,b_2,...,b_{k_1} \subseteq \mathbb{N}} \prod_{i=1}^{k_1} \left(1 - \left(1 - \frac{{r_{b_i}}^{-s}}{H_{n,s}}\right)^{\delta}\right)^{k_1-1} \end{aligned} \tag{14}$$

Note that for any $0 \leq x \leq 1$ and positive integer δ we have

$$1 - (1-x)^{\delta} \geq x. \tag{15}$$

Hence,

$$\begin{aligned} P &\geq \sum_{b_1,b_2,...,b_{k_1} \subseteq \mathbb{N}} \prod_{i=1}^{k_1} \left(\frac{{r_{b_i}}^{-s}}{H_{n,s}}\right)^{k_1-1}, \\ &= \frac{\sum_{b_1,b_2,...,b_{k_1} \subseteq \mathbb{N}} \prod_{i=1}^{k_1} r_{b_i}^{-s(k_1-1)}}{H_{n,s}^{k_1(k_1-1)}}. \end{aligned} \tag{16}$$

In order to simplify this expression, we use the *elementary symmetric polynomial* notation. If we have a vector $V_l =$

[5] A user will not request a content that it has already cached.

$(v_1, v_2, ..., v_l)$ of length l, then the k_1-th degree elementary symmetric polynomial of these variables is denoted as

$$\sigma_{k_1}(V_l) = \sigma_{k_1}(v_1, ..., v_l) = \sum_{1 \le i_1 < i_2 < .. < i_{k_1} \le l} v_{i_1} ... v_{i_{k_1}}. \quad (17)$$

Using this notation and by defining $Y_l = (r_1^{-s(k-1)}, r_2^{-s(k-1)}, ..., r_l^{-s(k-1)})$, we can write

$$P \ge \frac{\sigma_{k_1}(Y_l)}{H_{n,s}^{k_1(k_1-1)}}. \quad (18)$$

Notice that since the content request probability follows a Zipfian distribution, we have

$$\Pr[r_j \le h] = \frac{H_{h,s}}{H_{n,s}}. \quad (19)$$

Therefore, for a specific group of users $N_{b_1}, N_{b_2}, ..., N_{b_{k_1}}$, the probability that they all request contents from the top h most popular contents is given by

$$\Pr[r_{b_1} \le h, ..., r_{b_{k_1}} \le h] = \prod_{j=1}^{k_1} \Pr[r_{b_j} \le h] = \left(\frac{H_{h,s}}{H_{n,s}}\right)^{k_1}. \quad (20)$$

Using lemma 1, we can verify that for values of $h = \Theta(1)$ with respect to n, the ratio $\frac{H_{h,s}}{H_{n,s}}$ can be very close to one. This, along with the fact that l can possibly be much larger than k_1, means that with a very high probability, there exists for each set of users $\{N_{b_1}, N_{b_2}, ..., N_{b_{k_1}}\}$ that the requests come only from the h most popular contents. This implies that with a very high probability, $\sigma_{k_1}(Y_l) \ge \binom{l}{k_1} h^{-k_1 s(k_1-1)}$. Also, notice that

$$H_{n,s} < H_{\infty,s} = \zeta(s) < \infty. \quad (21)$$

Therefore, the lower bound of (18) is obtained as

$$P \ge \binom{l}{k_1} \left(\frac{h^{-s}}{\zeta(s)}\right)^{k_1(k_1-1)}. \quad (22)$$

This lower bound does not depend on the total number of contents in the network (k), and only depends on l, h, s and k_1. The results means that regardless of the amount of contents in the network, there is always a constant lower bound for the probability of finding a clique of size k_1. This implies that MIC can actually be very practical in large networks. The result also indicates that the probability of a clique tends exponentially to zero as k_1 increases. In practice, it is usually sufficient to look for cliques of sizes 2, 3, and 4.

□

V. Simulations

In this section, we demonstrate the performance of the new architecture compared to the original ICN without using MIC. Figure 3 shows the probability that a specific content that the users are requesting be available in the endpoint router. As this plot suggests, the probability of content availability in the endpoint router approaches one as the Zipfian parameter is increased. Notice that this probability goes to one regardless of the number of available contents in the network, number of users and other factors. However, figure 3 suggests that this probability is slightly higher when the number of contents is smaller and/or the router has a larger cache size. The fact that many requested contents have been already cached in the router implies that some nodes also store them in their caches. Therefore, we predict that the introduction of MIC to ICN architecture will be very useful. For these simulations, R denotes the size of endpoint router cache and U denotes the size of user's cache.

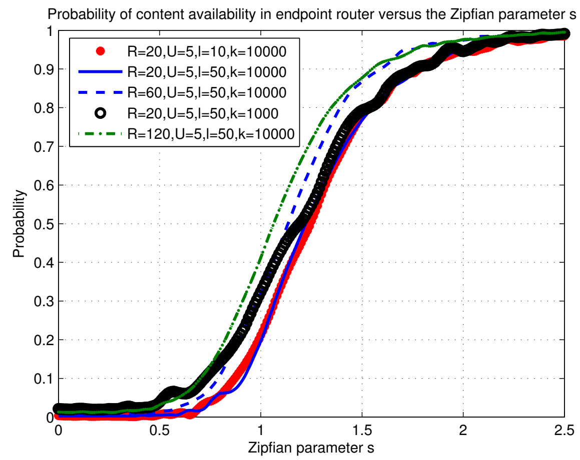

Fig. 3. *Probability of requesting a content that is already available in the edge router cache versus the Zipfian parameter s. In this figure, R denotes the size of local router cache, U is the size of user's cache, l is the number of users served by the router and k is the total number of contents in the network.)*

Figure 4, shows the simulation results for four different sets of parameters. In this figure, we have plotted the effective average number of packets sent per transmission. Note that one transmission can contain more than one package as it may serve multiple users. We have assumed that the users are requesting contents based on a Poisson distribution. In each time slot, the local router updates the dependency graph based on the received requests from users and also removes some of the edges for the contents that it has transmitted. The algorithm is a simple heuristic approach that at each time slot, the local router first searches for a clique in the dependency graph. If we can find a clique of size k_1 in the dependency graph, we can save $k_1 - 1$ transmissions by transmitting a simple index code to all the users in the clique. If there is no clique in the dependency graph, then we will search for cycles. As mentioned before, each cycle can save one transmission per cycle. Note that this is a simple heuristic suboptimal approach and finding the best solution is beyond the scope of this paper. In fact, to find the actual benefit that we can achieve by using MIC, we need to find the optimal rate for the index coding problem. Note that finding a clique of maximum size in the dependency graph or the optimal index coding rate is an NP-hard problem. For our simulation, we have searched for cliques and cycles of maximum size four. Even with this simple algorithm, we were able to show that the MIC can close to double the average number of packets per transmission in each time slot for certain values of the Zipfian parameter. Clearly, the optimal MIC rate is larger than the results obtained by our simple algorithm. Note that the users are using Least Recently Used (LRU) caching policy for eviction of overflow contents.

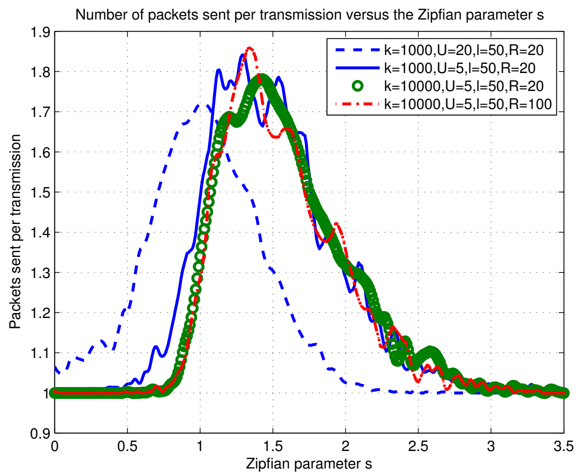

Fig. 4. *Average number of packets transmitted in each time slot when using MIC as a function of the Zipfian parameter s using a suboptimal search. The baseline is ICN with no MIC which is equivalent of one content per transmission. In this figure, R denotes the size of local router cache, U is the size of user's cache, l is the number of users served by the router and k is the total number of contents in the network.*

When s is a small value, then the distribution of content request is close to uniform distribution. Under this condition, the dependency graph is very sparse because there is a small probability that a user requests a content that is already cached by another user. Clearly, there is no benefit for using MIC in this case. Similarly, when s is a very large number, most users are asking for the same content and therefore, the router broadcasts the content to all of them which is equivalent of average of one content per transmission. The main benefit of MIC happens for medium values of s between 0.5 and 2 which is usually the case in practical networks. Note that an ICN with no MIC, will have always one content per transmission which is the baseline.

VI. Conclusion

This paper introduces *Modified Index Coding (MIC)* approach. MIC is more general than original index coding problem which can be applied to both wireless multi-hop communications and wired networks. We also applied MIC to information centric networks (ICN). The result was a hybrid caching scheme in ICN that consists of a caching at the routers and distributed caching at the nodes. The purpose of caching in routers is similar to the original concept of caching in ICN but we transmit encoded messages instead of the data itself. The purpose of distributed caching is to have replicas of the popular messages so that when we transmit new requested messages, we can combine multiple messages in order to serve multiple users with a single transmission. There are many research problems that were not addressed in this paper. For example, security, content routing and caching policies for these two types of caching are some problems that can be investigated in the future. We also focused on the linear network coding while nonlinear codes may outperform linear codes.

References

[1] Y. Birk and T. Kol, “Coding on demand by an informed source (iscod) for efficient broadcast of different supplemental data to caching clients,” *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2825–2830, 2006.

[2] ——, “Informed-source coding-on-demand (iscod) over broadcast channels,” in *INFOCOM'98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, 1998, pp. 1257–1264.

[3] “Nebula project,” *http://nebula.cis.upenn.edu*.

[4] “expressive internet architecture project,” *http://www.cs.cmu.edu/ xia/*.

[5] “Named data networking,” *http://named-data.net/*, August 2010.

[6] “Psirp: publish-subscribe internet routing paradigm,” *http://psirp.org/*, Jan. 2008.

[7] L. Peng, S. Guo, S. Yu, and A. V. Vasilakos, “Codepipe: An opportunistic feeding and routing protocol for reliable multicast with pipelined network coding,” in *Proceedings of IEEE INFOCOM*, 2012, pp. 100–108.

[8] W. Quan, X. Changqiao, A. Vasilakos, J. Guan, H. Zhang, and L. A. Grieco, “Tb 2 f: Tree-bitmap and bloom-filter for a scalable and efficient name lookup in content-centric networking,” in *IFIP Networking*, 2014.

[9] H. R. Sadjadpour, “A new design for information centric networks,” in *IEEE 48th Annual Conference on Information Sciences and Systems (CISS)*, 2014, pp. 1–6.

[10] Z. Wang, H. R. Sadjadpour, J. Garcia-Luna-Aceves, and S. S. Karande, “Fundamental limits of information dissemination in wireless ad hoc networks-part i: single-packet reception,” *Wireless Communications, IEEE Transactions on*, vol. 8, no. 12, pp. 5749–5754, 2009.

[11] R. Yeung, S. Li, N. Cai, and Z. Zhang, “Network coding theory (foundations and trends in communications and information theory),” *New York: Now*, 2006.

[12] D. Beaver, S. Kumar, H. C. Li, J. Sobel, P. Vajgel *et al.*, “Finding a needle in haystack: Facebook's photo storage.” in *OSDI*, vol. 10, 2010, pp. 1–8.

[13] A. Wolman, M. Voelker, N. Sharma, N. Cardwell, A. Karlin, and H. M. Levy, “On the scale and performance of cooperative web proxy caching,” *ACM SIGOPS Operating Systems Review*, vol. 33, no. 5, pp. 16–31, 1999.

[14] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. M. Maggs, K. Ng, V. Sekar, and S. Shenker, “Less pain, most of the gain: Incrementally deployable icn,” *ACM SIGCOMM conference*, Hong Kong, August 13-16, 2013.

[15] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon., “I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system,” in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement.*, 2007.

[16] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, “Web caching and zipf-like distributions: Evidence and implications,” in *Proceedings of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'99.), Vol. 1*, 1999.

[17] M. A. R. Chaudhry, Z. Asad, A. Sprintson, and M. Langberg, “On the complementary index coding problem,” in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, 2011.

[18] S. El Rouayheb, A. Sprintson, and C. Georghiades, “On the index coding problem and its relation to network coding and matroid theory,” *Information Theory, IEEE Transactions on*, vol. 56, no. 7, pp. 3187–3195, 2010.

[19] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, “On the implications of zipf's law for web caching,” 1998.

[20] P. Erdős and L. Posa, “On the maximal number of disjoint circuits of a graph,” *Publicationes Mathematicae Debrecen 3-12*, Sep. 1962.