

A New Design for Information Centric Networks

Hamid R. Sadjadpour
UC Santa Cruz
Suns-tech Corporations,
Email: hamid@suns-tech.com

Abstract—Information Centric Network (ICN) is a content-based information dissemination approach that improves the content delivery and latency. We introduce a modified architecture for ICN which can reduce the traffic by combining multiple messages and facilitate the data distribution in the network. By observing the similarities between index coding and ICN architectures, we propose to combine these two techniques to arrive at a new architecture that enhances data delivery in networks beyond original ICN scheme. By taking advantage of some concepts such as linear network coding, caching and index coding, we demonstrate that we can reduce the traffic and increase the capacity of ICN architecture by combining (network encoding) multiple messages requested by different nodes and sending them in one transmission. We demonstrate that each node will be able to extract its desired message from the combined encoded messages. To achieve this goal, we first define a modified version of index coding in order to apply index coding for both wired and wireless networks. Further, we introduce a hybrid caching scheme that includes both central and distributed caching to support two different goals. Our hybrid caching approach is a combination of conventional caching in ICN that caches the content in various network locations in order to make the content readily available to nodes and a new distributed caching scheme across nodes in the network to improve the performance of the entire system. The purpose of the second caching scheme is to allow central cache system to combine multiple contents in order to serve several client nodes simultaneously with a single transmission of encoded messages. The focus of this paper is to describe the new ICN architecture and demonstrate the advantages of the new architecture.

Keywords—Information Centric Network, Index Coding, Caching, Network Coding.

I. INTRODUCTION

The Internet has evolved from its original intent that was based on the concept of connecting two nodes for exchange of information. Generally, there are a large number of contents in the network that are being requested by different users. These contents are stored in digital format and are being distributed over the Internet. In the original architecture of Internet, end-to-end (source-destination pair) approach was pursued to deliver the contents. In the new paradigm, the focus is to access the content that is requested by users from any location. Therefore, most users are interested in accessing the content regardless of the location of delivery as long as the information is secure. Based on this observation, researchers have introduced Information Centric Network (ICN) [1]–[6] that focuses on the content delivery from any location that has cached the content. The key question that ICN attempts to address is "how to securely and efficiently deliver contents that are stored in different locations and requested by various users?"

ICN has attempted to address 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 destination so that they can be requested directly from caches. 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 contents locations in different caches.

Recently, there has been increasing interest in taking advantage of the broadcast nature of wireless channel to improve the throughput and reliability of data transmission. Many new techniques have been introduced to combine routing and coding [7], [8] to improve the throughput capacity of wireless networks. Birk and Kol [9], [10] described applications of coding with the nodes capable of caching the data. They have shown that some capacity gains can be achieved by taking advantages of caching and coding in wireless channels. This approach was named Index Coding. The original index coding 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 in this paper to accommodate other classes of the channels such as wired networks. Figure 1 demonstrates an example of index coding problem with source having six messages and each node N_1 to N_6 needs one information message and has a subset of all messages. For example, node N_1 needs message m_1 while it has prior side information m_2, m_3 , 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. In this paper, we only consider linear coding approach. If all the nodes in figure 1 are within transmission range of the source, then the source can send only two messages of $m_1 + m_2 + m_3$ and $m_4 + m_5 + m_6$ to allow all nodes to retrieve their requested messages. For example, node N_1 can add m_2 and m_3 to encoded message $m_1 + m_2 + m_3$ in order to recover message m_1 . We assume all operations are over $GF(2^n)$. This simple example shows that index coding can reduce the total number of transmissions significantly, i.e., for this example it has been reduced from 6 to only 2 transmissions. To the best of our knowledge, index coding has been introduced only for broadcast wireless channels.

In 2000, a new concept for *information flow* in networks was introduced. Traditional approaches in communication networks consider transmission of information as exchange of commodity and a data is transferred in the network by routers that only replicate the received information toward

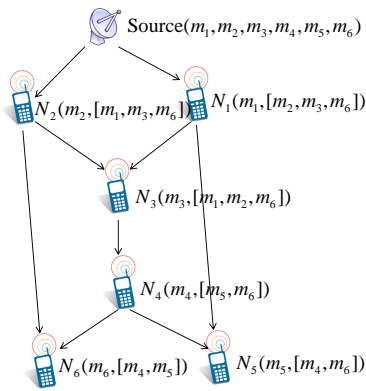


Fig. 1. An example of index coding coding problem.

the destination. However, Ahlswede et al. [11] proved that combination of information in the relays could increase the capacity of a network over the limit achieved by pure routing protocols. The new concept was termed Network Coding in literature and there has been significant work for understanding different aspects of this approach. We demonstrate that this approach can be utilized efficiently in ICN architecture to improve the data delivery. Let's again consider figure 1 and assume source node transmission range cannot reach all the nodes in the network. Further, we assume source node S desires to multicast two messages m_1 and m_2 to nodes N_5 and N_6 . The arrow between two nodes in the figure indicates that there is a channel between the two nodes. This construction is known as *butterfly network*. We assume each message is transmitted in one unit of time. Using a simple routing and nodes only doing replication of messages, it can be shown [12] that the two destination nodes can receive two messages in 8 time units while using network coding allows us to use only 6 time units to transmit these two messages. In this example and throughout the paper, we assume each channel can send one message in one time unit and the capacity of all channels and the size of all messages in the network are the same.

The rest of the paper is organized as follows. In Section II, we focus on defining the modified index coding and the caching schemes. Section III described the new ICN architecture. The discussion about this scheme and conclusion of the paper are presented in Sections IV and V respectively.

II. MODEL AND PROBLEM FORMULATION

A. Index Coding

The index coding problem [9], [10] 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 using multicast communications. The new scheme, called Modified Index Coding (MIC) technique, can be applied to different types of networks. This approach can be applied to wired networks or to a hybrid network consisting of 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 where the nodes are connected to the infrastructure through a wireless modem. Such scenarios are becoming more common in medium and large size corporate campuses.

Definition: Modified Index Code: A modified index code $\text{MIC}(\mathbb{M}, \mathbb{N})$ consists of a set of k messages $\mathbb{M} = \{m_1, \dots, m_k\}$ and a set of receiver nodes \mathbb{N} . Each node N_i has a subset of messages called $\mathbb{L}_i \subseteq \mathbb{M}$ and requests one message¹ m_i that node N_i does not have, i.e., $m_i \in \mathbb{M}$ and $\mathbb{L}_i \subseteq \mathbb{M} \setminus \{m_i\}$. Each message can be divided into n packets, i.e., $m_i = \{m_{i1}, \dots, m_{in}\}$. Each packet also belongs to an alphabet taken from a q -ary finite Field Γ . Therefore, we have $m_i = \{m_{i1}, \dots, m_{in}\} \in \Gamma^n$. We further define $\mathbb{N} = \{m_{11}, \dots, m_{1n}, \dots, m_{k1}, \dots, m_{kn}\} \in \Gamma^{nk}$. At any given time, $k_1 \leq k$ receiver nodes are requesting some messages. The source defines groups of receiver nodes k_1^1, \dots, k_1^p , where $k_1^1 + \dots + k_1^p = k_1$. Modified index code for $\text{MIC}(\mathbb{M}, \mathbb{N})$ is a function $f_i : \Gamma^{nk} \rightarrow \Gamma^{\ell_i}$, 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 $\Phi_i : \Gamma^{\ell_i + n|\mathbb{L}_i|} \rightarrow \Gamma^n$ such that the desired message can be decoded for that particular node, i.e., $\Phi_i(f(\mathbb{N}), \mathbb{L}_i) = m_i, \forall \mathbb{N} \in \Gamma^{nk}$.

Note that in this new definition, we multicast p coded messages to these p groups each one consists of k_1^i receiver nodes where $1 \leq i \leq p$. Note that index coding definition subsumes modified index coding 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 index coding concept for both wireless and wired networks. Modified index coding (MIC) concept is applied to design a new ICN architecture.

Figure 1 can be an example of MIC. Suppose that any two nodes can communicate only when there is an arrow between them. For example, 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. For this example, these two encoded messages are the minimum number of transmissions (optimum) that will achieve the desired outcome. However, in general the solution is NP-hard problem and may not be practical to obtain. We will discuss this issue briefly later.

If $n = 1$, then the index code is a scalar code and for $n > 1$, it is called a vector index code. The length of index code ℓ determines the transmission rate of the code. In theory, one would like to minimize this length which results in optimal solution for a given alphabet size q . It has been shown [13] that there is direct relationship between index coding, network coding and matroid theory. This relationship allows us to search for solutions based on existing techniques for matroids [14], [15] in literature.

B. Caching

One of the 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

¹Extension of this definition to multiple nodes is straightforward.

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 caching technique that requires minimum overhead related to locating stored cache contents.

Since any cache has limited storage capacity, we need to develop caching policies for eviction of contents when the cache runs out of storage space. There are many caching policies [16]–[19] that have been introduced in literature. The two main cache policies are Least Frequently Used (LFU) and Lease Recently Used (LRU) that their name clearly describes how eviction happens in a cache. Least Frequently Recently Used (LFRU) [16] is a technique that combines LFU and LRU by giving weights to these two policies and depending on the application, favors one policy more than the other policy. In [17], the authors present a model to approximate the hit-ratio of any multi-cache policy that has been introduced by LFRU. They also discuss how to approximate the performance of a network of caches with a dynamic traffic in the network. LRU-K basic idea [18] is to keep track of the number of K last times that a content was referenced. In [19] different caching policies for in-network caching were discussed.

As we mentioned above, in all these approaches the caching is based on the concept of storing the content in order to provide it at a later time. In our architecture, we use caching for two different purposes. We cache the contents centrally in storage systems in order to provide it to users similar to the original approach mentioned above. However most users have significant storage capacity that are not fully utilized. For example, it is now common for a laptop to have Terabyte of memory storage mostly not being utilized by majority of 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 memory to keep the data objects that it has already obtained. By the discussion that we had in previous section related to modified index coding, it should be clear that these 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 that it stores except the current requested content. The network encoded message is multicast by the central cache system that is serving these nodes. Our objective for this distributed content storage is not to send them to other clients if other client nodes request them. The problem with content distribution using distributed caching is the significant overhead associated to this approach. We suggest that the requested contents by different users should always come from the central cache system or from source node that has the content. In our new architecture, caching are used for two different purposes as described above.

Note that by taking advantage of the modified index coding and network coding concepts, we need one multicast session replacing k_1^i unicast sessions. It is well known [20] that one

multicast session always consume 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 central cache system knows what contents each node has cached before, there is no need for the extra overhead to update each cache (node) information in the network. If all nodes allocate same storage size for this purposes², then the central cache system can keep track of which data are being stored in which node. We will elaborate on this approach later.

In current network architecture, the communication is between source and destination (end-to-end concept). ICN architectures take advantage of caching to bring the content closer to the users. However, all existing ICN architectures still use point-to-point concept in delivering data from cache to destination. The only difference is that they replace the source with a cache that is closer to the destination. However, in our proposed ICN architecture, we no longer use point-to-point communication as a means of data delivery. In the new architecture, we use one to many communications to send multiple messages to multiple users from central cache system utilizing multicast communications. This is one of the main advantages of the new architecture that reduces the bandwidth usage per message in the network. To the best of our knowledge, this concept has never been used in previous ICN architectures.

III. PROPOSED ICN ARCHITECTURE

In this section, we will describe how to take advantage of index coding concept in order to derive a new architecture for Information Centric Networks. We assume each group of nodes in the network, are served by a unique router that also caches the information. In this context, serving the nodes means that 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 (as will be described in more details later) or finds the source for this content. Further, we assume when a node receives a requested content, it will keep this content in its cache. In Figure 2, all the routers that are shaded, will be 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. If we assign too many users to a single router, then this router requires more time to access which contents each node has and also the size of search space will be very large. This can cause high latency in the system. If we assign few nodes to a single router, then there is not enough diversity to combine messages in each multicast session and therefore, we are unable to fully take advantage of the new ICN architecture. For the example in figure 2, one group of users are served by router $R5$ and another group of users are served by router $R2$ via $R6$ and $R7$.

²Each node can allocate different storage size for this purpose as long as the central cache system is aware of that information. However, if all nodes allocate similar storage capacity for this purpose, it is easier for the central cache system to keep track of each node extra storage.

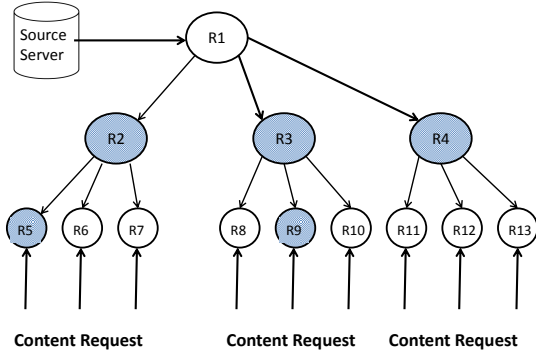


Fig. 2. Example of cache placement in a network. The shaded circles represent relays that also cache contents inside the network.

For the rest of the discussion, we assume that each node has enough storage capacity to store all requested messages. 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 3). When some of these nodes request different³ contents from the assigned router, this router try to network encode the requested contents by considering the contents that nodes store locally in order to minimize the total number of multicast transmissions to serve all these nodes. The optimum encoding selection is an NP-hard problem. However, since it has been shown that there is a direct relationship [13] between network coding, index coding and matroids, then we can develop greedy algorithms that can be close to optimum solution. It is not the intention of this paper to describe specific protocols for ICN but rather present a new architecture for ICN.

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 again apply the network coding concept for this content along with all the available contents that are requested at that time. 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 caches the contents but rather a subset of the routers cache the contents. The assumption here is that most of the contents that are requested by a node in each group of nodes, will be likely requested by other nodes in that group. This is particularly true since most of content request popularity are heavy-tailed and have a distribution close to Zipfian distribution. Therefore, our approach supports content popularity concept and recommends a technique that can improve the data delivery. Prior studies [21], [22] have shown that multiple layer caching or cooperative caching does not provide significant improvement for Zipfian distribution. For this reason, recent study [23] has suggested caching scheme that takes advantage of this distribution. However, unlike [23] that recommends caching at the edges of the network, our caching policy depends on the number of nodes that a router can serve. More specifically,

³Some nodes can request the same content.

[23] will cache the contents in routers $R5$ to $R13$ while in our approach, the routers with caching capability depends highly on the density of the network which dictates the number of users directly connected to the router. Therefore, if the number of nodes that are directly connected to a router is too small, then this scheme will not be very effective because the probability of having overlap so that we can take advantage of network coding and index coding diminishes. On the other hand, if we serve a very large number of nodes, the search dimension becomes so large that is no longer practical without incurring significant delay in the content delivery. Therefore, we would like the users that are served by each router to be large enough so that it is both economical⁴ [24] and practical.

Figure 3 demonstrates an example for our proposed ICN architecture. Let nodes $N1$ and $N2$ request messages $m10$ and $m8$ respectively. Both these nodes are served by router $R5$. When these nodes send request to this router, the router will multicast $m10 + m8$ to these two nodes. Node $N1$ can add the received encoded message with $m8$ to obtain $m10$ and node $N2$ can similarly obtain its own requested message. Note that each node has some other messages in their cache but they don't use this information for this session. Router $R5$ has all the cached messages of nodes $N1$ and $N2$ and perhaps more messages that are requested by other users. Similarly, for nodes $N3$ and $N4$ that are served by router $R2$ via routers $R6$ and $R7$ respectively, $R2$ will send $m6 + m9$ 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.

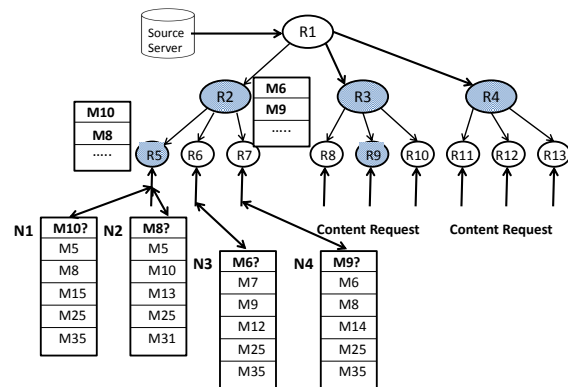


Fig. 3. Example of serving nodes by routers. Router $R5$ serves nodes $N1$ and $N2$ and router $R2$ serves nodes $N3$ and $N4$.

IV. DISCUSSION

One of the main challenges of using distributed caching in current ICN architecture is the fact that we need to have some handshaking so that the network is aware of which contents are stored by which nodes. This requires significant exchange of information that in some occasions can be comparable with

⁴Operators prefer a reasonably large number of users being served by a router with cache for economical reasons.

the data transfer bandwidth usage. There has been a recent work [25] that analytically computes this overhead using rate distortion concept. The result clearly shows that maintaining the location of contents requires overhead that for some cases can be comparable to the rate of data delivery. In our approach, we refrained from aggressive caching. Moreover, each content is registered with the router associated to a user so that when the content is delivered to that user, the router has that information and does not require any overhead for update. Further, the router will have the information about the size of each content and the size of storage that is allocated by each user for caching. Therefore, as long as the caching policy of the user is known by the router, the router knows for each user which contents are being stored and which content are evicted after the user reaches its maximum caching capacity. This clearly requires more processing power for each router that is involved in caching but it also reduces the overhead significantly. Users can also update their cache information off peak hours with their respective router⁵. The other overhead is associated to when a node requests a content. Since each user has an assigned router to serve that user, the request is always directed toward that router. Clearly, the overhead associated to finding the content in the network if it is not available locally by the router is similar to the current network architectures. Therefore, our proposed architecture simplifies the overhead and content routing challenges in ICN systems.

In many practical applications, nodes request many contents during any given period of time. Some of these contents do not have a large size. In our proposed scheme, it appears that it is beneficial for both users and central cache system to only cache contents that are larger than a minimum size. This policy will simplify the size of search space when the central cache intends to search for potential overlaps to combine multiple messages. Clearly, such trade-off can be only verified in future work by conducting comprehensive simulations. Further, in many cases, the overlapped messages that the central cache combines, may not be of the same size. The obvious solution is by zero padding the smaller size messages to make all of them the same size.

Content routing is an important research challenge in any proposed ICN architecture. Note that in our scheme, each node requests a content from a designated router. If the router has this information, then the message will be delivered to the node. However, if the router does not have the content, it will find the source that contains this information. Note that by adding this logic to the router and when the routing request is invoked, the router can use any existing routing scheme that it is currently implemented in the network to find the source. Therefore, this approach simplifies the routing in terms of backward compatibility since the network can be oblivious to the addition of ICN architecture in any sub-network. Given this advantage, the current approach can be used in a network where some sub-networks are designed based on the proposed ICN architecture while other portions of the network simply operate without any ICN capability. Since the routing will not be different for both systems, incremental addition of ICN architecture to the network is feasible. The only challenge is to have a many-to-one mapping of naming with IP address so

that when a data is requested by its name, the system knows the associated IP address⁶.

This paper introduces a new ICN architecture that is simple to implement. However, there are many questions that we did not answer. For example, since we are using two layers of caching (central and distributed), we did not address the question of what are the best policies for each caching approach? It is clear that based on the description of our two caching functions, we need to use different caching policies for these two systems. Further, there must be future studies on the trade-off between latency and capacity improvement as a function of the total number of nodes served by each router. By increasing the number of nodes for each router, we certainly can improve performance of this system while incurring some additional latency and hardware complexity at each router. The issue of security was not addressed in this paper and needs more thorough investigation. We also need to investigate what should be the window of time for a serving router to wait to receive requests from multiple nodes? Clearly, since different data contents have various QoS requirements, that will effect on the window of time to select groups of requests. This paper perhaps introduced more questions than answers by introducing the new architecture. The benefits of modeling the ICN architecture similar to an index coding problem raises the issue of how we can take advantage of the rich literature in network coding and Matroids to develop better schemes for selecting groups of active users for multicasting of encoded messages.

V. CONCLUSION

This paper introduced a new architecture for ICN based on a topic in information theory called index coding. In order to apply index coding to ICN, we introduced *Modified Index Coding (MIC)* approach. MIC is more general than original index coding problem which can be applied to networks with nodes communicating using multi-hop communications. We also introduced a hybrid caching scheme that consist of a central caching at the router and distributed caching at each node. The purpose of central caching is similar to 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. The new approach will be investigated from many aspects such as security, content routing, and performance comparison with prior schemes in the future studies.

REFERENCES

- [1] "Nebula project," <http://nebula.cis.upenn.edu>.
- [2] "expressive internet architecture project," <http://www.cs.cmu.edu/xia/>.
- [3] "Named data networking," <http://named-data.net/>, August 2010.
- [4] "Mobility first future internet architecture project," <http://mobilityfirst.winlab.rutgers.edu/>.
- [5] "Psirp: publish-subscribe internet routing paradigm," <http://psirp.org/>, Jan. 2008.

⁵For example, in a corporate environment, all these information can be updated during night when the users mostly are not active.

⁶Actually, current Cisco routers have this capability. In Cisco systems, multiple names can be associated to a single port with unique IP address.

- [6] S. Wood, J. Mathewson, J. Joy, M.-O. Stehr, M. Kim, A. Gehani, M. Gerla, H. R. Sadjadpour, and G. Luna JJ, "Iceman: A system for efficient, robust and secure situational awareness at the network edge," in *IEEE Milcom Conference*, November 2013.
- [7] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft. "Xors in the air: practical wireless network coding," in *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4. ACM, 2006, pp. 243–254.
- [8] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level network coding for wireless mesh networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 401–412, 2008.
- [9] 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.
- [10] —, "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.
- [11] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network information flow," *Information Theory, IEEE Transactions on*, vol. 46, no. 4, pp. 1204–1216, 2000.
- [12] R. Yeung, S. Li, N. Cai, and Z. Zhang, "Network coding theory (foundations and trends in communications and information theory)," *New York: Now*, 2006.
- [13] S. El Rouayheb, A. Sprintson, and C. Georghiadis, "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.
- [14] D. J. Welsh, *Matroid theory*. DoverPublications.com, 2010.
- [15] J. G. Oxley, *Matroid theory*. Oxford university press, 1992, vol. 3.
- [16] C. S. Kim, "Lrfu: A spectrum of policies that subsumes the least recently used and least frequently used policies," *IEEE transactions on Computers*, vol. 50, no. 12, pp. 1352–1361, 2001.
- [17] K. Katsaros, G. Xylomenos, and G. C. Polyzos, "Multicache: an incrementally deployable overlay architecture for information-centric networking," in *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*. IEEE, 2010, pp. 1–5.
- [18] E. J. O'neil, P. E. O'neil, and G. Weikum, "The lru-k page replacement algorithm for database disk buffering," in *ACM SIGMOD Record*, vol. 22, no. 2. ACM, 1993, pp. 297–306.
- [19] Z. Li, G. Simon, and A. Gravey, "Caching policies for in-network caching," in *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*. IEEE, 2012, pp. 1–7.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] J. Wang, Z. Hao, L. Li, and X. Xu, "Information-centric network in an isp," <http://tools.ietf.org/html/draft-li-icnrg-icn-isp-01>, 2012.
- [25] B. Azimdoost, C. Westphal, and H. Sadjadpour, "The price of updating the control plane," *Submitted for publications*.