Network-Coded Content Delivery in Femtocaching-Assisted Cellular Networks

Yousef N. Shnaiwer*, Sameh Sorour*, Neda Aboutorab[†], Parastoo Sadeghi[‡], and Tareq Y. Al-Naffouri*[‡]

*Electrical Engineering Department, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia

[†]Research School of Engineering, The Australian National University, Canberra, Australia

[‡]CEMSE Division, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia

*{g201204420,samehsorour}@kfupm.edu.sa,[†]{neda.aboutorab,parastoo.sadeghi}@anu.edu.au,[‡]{tareq.alnaffouri}@kaust.edu.sa

Abstract—Next-generation cellular networks are expected to be assisted by femtocaches (FCs), which collectively store the most popular files for the clients. Given any arbitrary non-fragmented placement of such files, a strict no-latency constraint, and clients' prior knowledge, new file download requests could be efficiently handled by both the FCs and the macrocell base station (MBS) using opportunistic network coding (ONC). In this paper, we aim to find the best allocation of coded file downloads to the FCs so as to minimize the MBS involvement in this download process. We first formulate this optimization problem over an ONC graph, and show that it is NP-hard. We then propose a greedy approach that maximizes the number of files downloaded by the FCs, with the goal to reduce the download share of the MBS. This allocation is performed using a dual conflict ONC graph to avoid conflicts among the FC downloads. Simulations show that our proposed scheme almost achieves the optimal performance and significantly saves on the MBS bandwidth.

Index Terms—Femtocells, Femtocaching, Opportunistic Network Coding.

I. INTRODUCTION

Cellular networks are currently witnessing an explosive growth of data traffic, mainly because of the on-demand download and streaming of video files [1]. This growth is expected to further increase by two orders of magnitude in the next few years [2]. This large demand raises the need to enhance the spectrum efficiency of next-generation cellular networks [3]. One of the recent suggestions to do so in the 5G macrocell architecture is to bring the content closer to the clients, by deploying a large number of low-cost wireless nodes with large storage capacity, in which the popular files (i.e., the files which are most likely to be requested by the clients) are cached [3]. These nodes are usually referred to as femtocaches (FCs) and the approach is called *femtocaching* [2]. Since file popularity is slowly varying, this caching occurs through the expensive backhaul to the macrocell base station (MBS) with very low rate or in off-peak times. Clients can thus download such files from the FCs at peak times. Only clients that cannot be served by the FCs at a given instant (i.e. when FCs are busy serving other clients) will be allowed to download these files from the MBS to avoid undesirable video playback latencies.

The success of this new architecture depends on optimizing one (or both) of two processes, namely the content delivery process and the content placement process. The content

delivery process optimizes the files downloaded from each of the FCs, so as to minimize the expensive involvement (rate/bandwidth) of the MBS in this delivery process [4]. Complementarily, the content placement process consists of finding the optimum distribution of files or fragments of files over the FCs, so as to further enhance the delivery process. A possible cost for this enhancement is having the clients download and re-assemble fragments of their requested files from multiple FCs and the MBS [2].

Many solutions suggested for both problems in the literature are proposed under the assumption that each client deals with only one fixed FC dedicated to it [4–6]. This can result in a massive number of under-utilized FCs and could be prohibitive cost and running wise. Even when multiple clients are allowed to connect to each cache (such as in [2, 3]), it was always assumed that these clients did not previously download any of the cached files in the FCs, which contradicts with the fact of their popularity. Indeed, popularity is decided upon prior numerous requests of such files from the clients. Clearly, such previously downloaded files can enhance the delivery of new files from the FCs using opportunistic network coding (ONC) [7], thus further reducing the bandwidth consumed at the MBS. Indeed, ONC can exploit the diversity in prior downloaded files at the different clients to create coded combinations of currently requested files. Such combined files can thus be decoded at the designated clients, thus simultaneously delivering a larger number of requested files compared to simple broadcast of source files.

In this paper, we consider the content delivery problem under three practical settings. First, we assume any arbitrary (whether optimized or probabilistic) non-fragmented placement of files, thus avoiding the complexity of dealing with file fragmentation over FCs and multiple downloads per file request. Moreover, the possibility of prior file downloads at the clients is assumed. Furthermore, a no-latency constraint is imposed to guarantee the clients' quality-of-experience (QoE) in 5G networks, thus forcing the FCs and the MBS to immediately grant all the new requests for file downloads.

Given the above settings, this paper aims to find the best allocation of network-coded file downloads to the FCs, so as to minimize the bandwidth needed from the MBS to download the remaining ones. We first formulate this problem

over an ONC graph, which represents the conflicts among the transmissions of different FCs, and show that it is NP-hard. We then propose a greedy approach that maximizes the number of network-coded file downloads from the FCs, with the expectation to reduce the needed MBS bandwidth for the remaining downloads. To perform such allocation, we employ a dual conflict ONC graph to avoid conflicts among the FC downloads and serve the maximum number of requests from them.

The remainder of this paper is organized as follows. The system model is presented in Section II. The content delivery problem is formulated in Section III. Section IV introduces our proposed greedy solution. Section V presents the simulation results, and Section VI concludes the paper.

II. SYSTEM MODEL

The network model of interest is illustrated in Fig. 1. As shown in the figure, a set $\mathcal{U} = \{u_1, \dots, u_U\}$ of U clients, each requesting to download/stream one file, in the current time epoch, from a library $\mathcal{F} = \{f_1, \dots, f_F\}$ of F files. These files are all present at the MBS and also stored (with possible repetition) in the union of a set $\mathcal{C} = \{c_1, \dots, c_C\}$ of C FCs. The set $\mathcal{H}_{c_i}$ of files stored in each FC c_i is called the Has set of c_i , whereas the file requested by each client u_j constitutes its Wants set $\mathcal{W}_{u_j}$. The union of the Has sets of all the FCs is assumed to constitute the whole library. To guarantee clients QoE, all the requests should be served at the same time, and thus all the requests that are not rendered by the FCs must be served by the MBS. We also assume that each client u_j may have downloaded one or more files from $\mathcal{F}$ (other than the one currently in its Wants set) in previous time epochs, which constitutes its Has set $\mathcal{H}_{u_j}$ at the current epoch. The FCs and the MBS can thus exploit this clients' side information to employ ONC in delivering their requests in the current epoch. It is assumed that any client has a single physical wireless receiver, and thus can only download from any one of the FCs in $\mathcal{C}$ or the MBS at a time. Moreover, it is assumed that all the files are of equal lengths. Being low-cost devices with limited wireless transmission capabilities, each FC can statically transmit over only one physical channel (e.g., frequency band in FMDA, a group of subcarriers in OFDMA or a timeslot in TDMA) that is orthogonal to all the channels used by the other FCs and those of the MBS. A client scheduled to download its requested file from a specific FC must thus tune to the statically allocated channel to this FC. The MBS being a more sophisticated wireless node has many orthogonal channels to utilize dynamically. Finally, we assume that with proper modulation and detection methods and with channel error detection and correction techniques, a lossless channel model can be applied in our study.

III. PROBLEM FORMULATION

As mentioned in Section I, our ultimate target is to minimize the number of orthogonal coded downloads from the MBS. In general, when the files are requested by the clients, each of

Fig. 1: An example of our considered network model with one MBS, two FCs, and six clients.

the FCs should be allocated a subset of these files that, when combined into one coded file using ONC, could be decoded by their requesting clients. The question now is: *Which files should we allocate to each of the FCs, such that the remaining requested files (if any) can be combined (again using ONC) and delivered to the remaining clients using the minimum number of orthogonal MBS channels?*

A. Motivating Example

To clarify the above problem, let's consider the network scenario shown in Fig. 1. The example encompasses the MBS with its Has set, which constitutes the whole library, two FCs with their Has sets collectively storing the whole library, and six clients with their Has and Wants sets. The MBS is denoted as if it was a third FC to simplify the notation.

In this example, several allocations of coded file downloads can be determined. We will focus on two solutions. In what follows, $f_{k_1} \oplus f_{k_2}$ is an XOR combination of the bits of f_{k_1} and f_{k_2} .

Allocation 1:

- FC c_1 transmits $f_1 \oplus f_6$: This allows u_1 and u_6 to decode their requested files as they already have f_1 and f_6 , respectively.
- FC c_2 transmits f_2 using another statically assigned orthogonal channel, which satisfies the requests of u_4 and u_5 .

In this case, the MBS must simultaneously serve u_2 and u_3 requesting files f_3 and f_1 , respectively. Since u_2 does not have f_1 , an XOR of $f_1 \oplus f_3$ cannot be decoded at u_2 . To satisfy the no-latency constraint, the MBS must thus transmit f_3 and f_1 uncoded on separate orthogonal channels to u_2 and u_3 , respectively. Consequently, this allocation will end up consuming two channels from the MBS.

Allocation 2:

- FC c_1 transmits $f_1 \oplus f_6$, which addresses the requests of u_1 and u_6 as in the previous allocation.

- FC c_2 transmits f_3 , which satisfies the requests of u_2 .

In this case, the MBS must simultaneously serve the requests of u_3 , u_4 and u_5 requesting files f_1 , f_2 and f_2 , respectively. By looking at the Has sets of these clients, we can clearly see that they can all decode their requested files from a single coded transmission of $f_1 \oplus f_2$ by the MBS. Thus, this allocation will end up consuming only one channel from the MBS, which is half the bandwidth required by Allocation 1.

By searching over all the possible options, we can see that Allocation 2 (though not being the unique optimal solution as will be discussed later) results in the minimum number of consumed channels at the MBS. Now the interesting question is how we could systematically find the optimal solution(s) for any scenario with any size of network. In the next section, we formulate this problem as an optimization problem over an ONC graph.

B. Graph-Based Formulation

To formulate the above problem, we first need to define the ONC conflict graph $\mathcal{G}$ that represents all the possible coding conflicts (i.e. files that when XORed together cannot be decoded at their requesting clients). This ONC conflict graph is constructed as follows. Every client u_j , requesting file f_k , has only one vertex v_{jk} in the graph. Two vertices $v_{j_1 k_1}$ and $v_{j_2 k_2}$ will be set adjacent by an edge in this graph if $f_{k_1} \neq f_{k_2}$ AND either $f_{k_1} \notin \mathcal{H}_{c_{j_2}}$ OR $f_{k_2} \notin \mathcal{H}_{c_{j_1}}$. This condition occurs when two vertices represent the request of two different files by either the same client or two different clients while at least one of them does not possess the file requested by the other. Therefore, that client will not be able to extract its own requested file from an XOR of $f_{k_1} \oplus f_{k_2}$.

Fig. 2.a depicts the ONC conflict graph of the example in Fig. 1. Clearly, only adjacent vertices based on the above conditions cannot be encoded with each other. Thus, defining any independent set¹ $\mathcal{I}$ in this graph, the XOR of all the files represented in the vertices of $\mathcal{I}$ (denoted by $\mathcal{F}(\mathcal{I})$) can surely be decoded immediately by all the clients represented in the vertices of $\mathcal{I}$ (denoted by $\mathcal{U}(\mathcal{I})$). Thus, every independent set in graph $\mathcal{G}$ can consume the channel of an FC or one of the MBS channels.

Now, let $\mathcal{I}_i$, $i = 1, \dots, C$, be the independent set, whose files $\mathcal{F}(\mathcal{I}_i)$ will be allocated to FC c_i for delivery to the clients in $\mathcal{U}(\mathcal{I}_i)$. Our problem will then boil down to finding the set of independent sets $\mathcal{I}_1, \dots, \mathcal{I}_C$, so as to minimize the remaining number of independent sets in $\mathcal{G}' = \mathcal{G} \setminus \bigcup_{i=1}^C \mathcal{I}_i$. Indeed, these remaining independent sets must be served by orthogonal channels at the MBS, and thus their number must be minimized. Since the minimum number of independent sets in graph $\mathcal{G}'$ is equal to its chromatic number² $\chi(\mathcal{G}')$ [8], we

¹An independent set in a graph is a set of pairwise non-adjacent vertices.

²The chromatic number of a graph is the minimum number of colours, with which the vertices of the graph could be coloured, such that no two adjacent vertices carry the same colour.

Fig. 2: (a) An illustration of the ONC conflict graph for the scenario in Fig. 1. (b) Two possible optimal solutions for the given scenario.

can thus formulate our problem as:

$$\begin{aligned} \min_{\mathcal{I}_1, \dots, \mathcal{I}_C} \quad & \chi \left(\mathcal{G} \setminus \bigcup_{i=1}^C \mathcal{I}_i \right) \\ \text{subject to} \quad & \mathcal{F}(\mathcal{I}_i) \subseteq \mathcal{H}_{c_i}, \forall c_i \in \mathcal{C} \end{aligned} \quad (1)$$

The C constraints in (1) guarantee that all the files allocated to each of the FCs are indeed cached in that FC. If this constraint is not satisfied in any one $\mathcal{I}_i$, then an XOR of the files in $\mathcal{F}(\mathcal{I}_i)$ cannot be formed at FC c_i , which makes a solution including $\mathcal{I}_i$ invalid.

The left subfigure of Fig. 2.b depicts one optimal selection of $\mathcal{I}_1$ and $\mathcal{I}_2$ for the example in Fig. 1, resulting in Allocation 2 (explained in Section III-A) and a chromatic number of 1 for the remaining vertices. The right subfigure shows another selection of $\mathcal{I}_1$ and $\mathcal{I}_2$ achieving the same chromatic number, which means that more than one optimal allocation may exist for the same scenario.

Now, it is well known that the problem of determining the chromatic number of a graph is NP-hard [9]. This makes the optimization problem in (1) an NP-hard problem, as it not only computes a chromatic number for one graph, but needs to find an allocation that minimizes the chromatic number of the remaining subgraph. We will thus develop a heuristic to solve this problem sub-optimally in the next section.

IV. PROPOSED GREEDY APPROACH

In this section, a greedy heuristic approach is proposed to solve the problem in (1). The rationale behind this approach is first explained and then its achievement and implementation issues will be discussed.

A. Greedy Heuristic Philosophy

The philosophy of our proposed greedy heuristic is to maximize the number of file requests served by the FCs. This is equivalent to minimizing the number of files that should be served by the MBS, with the expectation that this minimum number of remaining requests could be served in the minimum number of orthogonal channels. One may think that this is intuitively true in general. However, we can easily show that this greedy approach may not necessarily result in the minimum number of orthogonal MBS channels. This comes from the fact that a smaller number of ONC-conflicting files may consume more MBS channels compared to a larger number of files that are not ONC-conflicting.

To exemplify this fact, remember our discussion of Allocation 1 and Allocation 2 in Section III-A. Allocation 1 does satisfy the policy of the greedy approach, by allowing the FCs to serve four files out of the six. This fact can be simply confirmed by a search over the ONC conflict graph in Fig. 2.a and inferring that the FCs cannot serve more than four files. Thus, Allocation 1, which is the solution of the greedy approach, will result in only two files to be transmitted by the MBS. Nonetheless, these files are conflicting and cannot be encoded together in one MBS channel. Thus, two separate MBS channels will be consumed by this solution.

On the other hand, Allocation 2 does not satisfy the policy of the greedy algorithm, as it allocates only three files (less than four) to the FCs and leaves three files to be served by the MBS. However, since these files are non-conflicting, they can be served in one channel. Even for the allocation of the right subgraph of Fig. 2.b, which allocates only two files to the FCs and four files to the MBS (i.e. representing a stronger deviation from the greedy policy), all four files left to the MBS are non-conflicting and can still be served by one MBS channel.

Thus, the proposed greedy approach may not always achieve the optimal solution of (1) in each and every scenario. However, since the problem in (1) is NP-hard, we could still use this greedy policy as a heuristic approach with the hope to achieve a very slight deviation from the optimal solution on average.

Having introduced the philosophy of the proposed greedy heuristic, we now study how we can design an algorithm to implement it in the following sections.

B. Maximizing File Downloads from FCs

One first and intuitive thought to implement the heuristic philosophy (i.e. maximize the number of files downloaded from the FCs) is to allow each FC to serve the maximum number of requests given the files in its Has set. To achieve this scheme, each FC c_i should build its own conflict graph, as described in Section 2, while only considering the files in $\mathcal{H}_{c_i}$. Clearly, this ONC conflict graph is a subset of the main problem's ONC conflict graph (as the one in Fig. 2.a), consisting only of the vertices representing requested files from $\mathcal{H}_{c_i}$. Once each FC builds its own ONC conflict graph, each one should then serve the requests corresponding to the

Fig. 3: (a) Separate ONC conflict graphs of the FCs in the example shown in (b). (b) Allocation of coded file transmissions to the FCs and the MBS using the separated ONC scheme.

maximum independent set in this graph, which represents the maximum number of requests it can possibly handle simultaneously using ONC. Although finding the maximum independent set in a graph is by itself an NP-hard problem, it has a wide variety of very efficient solvers and heuristics that can be used to find it (which will be discussed in Section IV-C).

The problem with this separate-graph scheme is that it does not take into account the possibility of redundant allocation of files to more than one FC. That is, when each FC is finding the maximum independent set in its own graph independently from the others, it may occur that two or more FCs would decide to serve the same clients. This could hinder the achievement of the heuristic policy (i.e. maximizing the number of served requests by the FCs). One example of such situation is illustrated in Fig. 3. The figure shows that one of the allocations that could result from finding the maximum independent sets in each of the FC ONC graphs is that c_1 transmits $f_1 \oplus f_2$ to address the requests of u_1 and u_3 , whereas c_2 transmits $f_1 \oplus f_4$ to address the requests of u_2 , u_3 and u_5 . This means that both FCs will simultaneously attempt to serve the request of u_3 , whereas u_3 can tune and download from only one of them. Therefore, the addressing of the request of u_3 by the other FC can be seen as an allocation conflict, which may reduce its ability to address other requests, thus overloading the MBS. This is exactly what happens in the

shown scenario, as u_4 's request is not served by the FCs. It will be thus served by the MBS costing it one channel.

To avoid such allocation conflicts and their negative effect on MBS overloading, we will use a dual conflict ONC graph described as follows. For every client u_j requesting file f_k , we will generate a set of vertices $v_{ijk} \forall c_i$ having $f_k \in \mathcal{H}_{c_i}$. In other words, the requested file f_k by client u_j is represented by several vertices in the graph, each representing an FC having f_k in its cache and is thus a candidate to address the request of u_j . Now any two vertices v_{i_1,j_1,k_1} and v_{i_2,j_2,k_2} will be set adjacent by an edge if one of the following conditions occur:

- 1) $f_{k_1} \neq f_{k_2}$ and either $f_{k_1} \notin \mathcal{H}_{c_{j_2}}$ OR $f_{k_2} \notin \mathcal{H}_{c_{j_1}}$.
- 2) $j_1 = j_2$ AND $i_1 \neq i_2$.

The first condition is clearly the same as the coding conflict condition of the original ONC graph described in Section 2. The second condition adds an allocation conflict edge between any two vertices representing the service of the same client by two different FCs. With this structure, it can be easily inferred that any independent set in this dual conflict graph will represent a full allocation of coded file downloads to the FCs without any two FCs trying to download simultaneously to the same client.

Fig. 4.a depicts the dual conflict graph of the same example in Fig. 3.b. We can clearly see that the constructed graph indeed shows both the coding and allocation conflicts in one unified framework. By finding the maximum independent set in this dual conflict graph, we achieve the coded file download allocation shown in Fig. 4.b, which is able to download all requested files from the FCs, thus costing the MBS no channels at all.

C. Implementation Issues

As clarified above, our proposed greedy heuristic approach operates in two steps:

- 1) Allocating the maximum number of file downloads to the FC by finding the maximum independent set in the dual conflict graph illustrated in Section IV-B.
- 2) Determining the minimum number of coded file downloads for the remaining requests from the MBS. This could be done by finding the minimum graph colouring of the remaining original ONC conflict graph, described in Section III, after removing the vertices representing the requests served by the FCs in the first step.

Clearly, both steps require the solution of NP-hard problems, namely the maximum independent set and minimum graph colouring problems. For small network settings, we can employ well-known solvers of such problems. One example is the use of the Bron-Kerbosch (B-K) algorithm [10], which lists all the maximal independent sets³ in graphs. Thus, we can use this algorithm in the first step to list all maximal independent sets in the dual conflict graph and choose the largest one. We can also employ it in the second step to list all independent sets of

³A maximal independent set is the one that cannot be a subset of a larger independent set.

Fig. 4: The second solution of the example showing (a) the augmented conflict graph and, (b) the packet coding scheduling for each FC.

the remaining ONC conflict graph and then select the smallest number of them that cover all the vertices.

However, the complexity of these solutions may still be prohibitive for real-time practical implementations in large network settings (i.e. with large number of FCs, clients and popular files). We thus propose the use of simple greedy vertex search (GVS) algorithm in such practical settings. For the first step, we can assign to each vertex v_{ijk} a weight w_{ijk} defined as follows:

$$w_{ijk} = (V - \delta_{ijk}) \cdot \sum_{v_{i'j'k'} \in \mathcal{N}(v_{ijk})} (V - \delta_{i'j'k'}) \quad (2)$$

where V is the total number of vertices in the graph, δ_{ijk} is the degree⁴ of vertex v_{ijk} , and $\mathcal{N}(v_{ijk})$ is the set of its adjacent vertices. Thus, a vertex will have a high weight when it has a large number of non-adjacent vertices, which themselves have large numbers of non-adjacent vertices. We can then perform a maximum weight vertex search, by picking at each iteration the vertex with the maximum weight and then removing all its adjacent (i.e. conflicting) vertices from the graph before the next iteration.

For the second step, we employ the same approach to iteratively select independent sets until no vertices remain in the graph.

⁴The degree of a vertex in a graph is the number of vertices adjacent to this vertex.

Fig. 5: Performance comparison against the optimal solution.

V. SIMULATION RESULTS

A. Comparison with Optimal Solution

In this section, simulation results are presented for a small network setting to evaluate the performance of our proposed greedy solution using the dual conflict graph with the optimal solution. The optimal solution is obtained by brute force search to solve the problem in (1). For the dual conflict graph solution, we test both the B-K and the GVS algorithms. The simulation comprises 2 FCs, 8 files, 10 clients (when fixed) and 60% of the files cached in each FC (when fixed). Each client has prior knowledge of 10% of the files.

Fig. 5 depicts the average number of orthogonal MBS channels against the number of clients and the percentage of cached files in the FCs. The main observation here is that the performance of our proposed greedy approach with the dual conflict graph is very close to the optimal solution when the B-K algorithm is employed. We can also see that the GVS algorithm results in small degradation.

B. Performance in Large Network Settings

We here test the performance of our proposed greedy solutions with both the separate FC graphs and dual conflict graph approaches for large network settings. Here only the GVS algorithm is employed. The simulation comprises 20 FCs, 50 files, 100 clients (when fixed) and 60% of the files cached in each FC (when fixed). Each client has prior knowledge of 20% of the files.

Fig. 6 depicts the designated comparison results. A significant enhancement can be observed when the dual conflict graph is employed instead of the separate FC graphs. We can also observe that 72% of the bandwidth required to download all requested files is offloaded from the MBS.

VI. CONCLUSIONS

In this paper, the problem of MBS bandwidth offloading in femtocaching-assisted wireless networks was investigated. We

Fig. 6: Performance comparison in large network settings.

first formulated the problem on a network coding graph and showed that it is an NP-hard problem. A greedy approach was then proposed to solve the problem by maximizing the number of clients served by the FCs first, and then serving the remaining clients by the MBS. A dual conflict graph and simple graph heuristics were developed to implement this greedy approach. The performance of our proposed algorithms were shown to achieve very-near optimal performance and to significantly offload the MBS bandwidth.

REFERENCES

- [1] M. Ji, G. Caire, and A. Molisch, "Optimal throughput-outage trade-off in wireless one-hop caching networks," in *IEEE International Symposium on Information Theory Proceedings (ISIT)*, 2013, July 2013, pp. 1461–1465.
- [2] K. Shanmugam, N. Golrezaei, A. Dimakis, A. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.
- [3] N. Golrezaei, A. Molisch, A. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 142–149, April 2013.
- [4] M. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.
- [5] —, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Transactions on Networking*, vol. PP, no. 99, pp. 1–1, Apr. 2014.
- [6] U. Niesen and M. Maddah-Ali, "Coded caching with nonuniform demands," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2014, pp. 221–226.
- [7] S. Sorour and S. Valaee, "An adaptive network coded retransmission scheme for single-hop wireless multicast broadcast services," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 869–878, June 2011.
- [8] M. v. Steen, *Graph theory and complex networks : an introduction*. Lexington: Maarten van Steen, 2010. [Online]. Available: <http://opac.inria.fr/record=b1130915>
- [9] C. Edwards and C. Elphick, "Lower bounds for the clique and the chromatic numbers of a graph," *Discrete Applied Mathematics*, vol. 5, no. 1, pp. 51 – 64, 1983. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0166218X8390015X>
- [10] "Bron-kerbosch maximal clique finding algorithm," <http://www.mathworks.com/matlabcentral/fileexchange/30413-bron-kerbosch-maximal-clique-finding-algorithm>, accessed: 2010-09-30.