Offloading of Fog Data Networks with Network Coded Cooperative D2D Communications

Ben Quinton(✉) and Neda Aboutorab

University of New South Wales, Campbell, ACT 2612, Australia
quintonbj@gmail.com, n.aboutorab@unsw.edu.au

Abstract. Future fog data networks are expected to be assisted by users cooperation and coding schemes. Given the finite I/O access bandwidth of the drives in the data servers and the explosive increase in the end users' demand for download of the content from the servers, in this paper, we consider the implementation of instantly decodable network coding (IDNC) in full-duplex device-to-device (D2D) enabled cooperative distributed data networks. In particular, this paper is concerned with optimizing D2D communications with efficiently coded transmissions such that we offload traffic from the expensive backhaul of network servers. Previous works implementing IDNC have not focused on a cooperative architecture, therefore a new theoretical-graph model is proposed and the optimal problem formulation is presented. However, as the optimal solution suffers from the intractability of being NP-hard, it is not suitable for real-time communications. The complexity of the problem is addressed by presenting a greedy heuristic algorithm used over the proposed graph model. The paper shows that by implementing IDNC in a full-duplex cooperative D2D network model significant reduction in the number of downloads required from the servers can be achieved, which will result in saving valuable servers' resources.

Keywords: Instantly decodable network coding · IoT · Full-duplex
Backhaul offloading · Cooperative D2D communications
Fog storage networks

1 Introduction

With the modern advancements of wireless communications, wireless networks have seen an explosion in data traffic over the past decade [6]. This rapid demand for more data is largely attributed to video and multimedia streaming, where it is expected that three-fourths of data traffic will be consumed by video [6]. To compound this further, it is expected that the next generation of wireless networks will encapsulate the new paradigm of the internet of things (IoT). This concept

J. Hu et al. (Eds.): MONAMI 2017, LNICST 235, pp. 1–13, 2018.
https://doi.org/10.1007/978-3-319-90775-8_1

moves to further integrate more and more devices into communication networks, where it is foreseen that the IoT will add a further 50 billion heterogeneous wireless devices by 2020 [6]. Consequently, this growing demand puts further pressure on data networks, where the offloading of the servers becomes an increasingly important problem.

This ever-growing demand for real-time data, where users expect to maintain their quality-of-experience (QoE) has led to much research to address the data networks backhaul problem. Multiple areas of research have shown promising methods to deal with this problem, one such option is to distribute the data closer to the users with improved redundancy [2,7,8,11]. The idea of distributing resources to the edge of a network is known as "fog" networking [5]. Motivated by very high temporal correlation among the "popular" content demanded by end-users, it is expected that the proactive (i.e. without users requests) diffusing of such popular content from its storage and transmission clouds behind the backhaul, and caching it in a "fog" of low-cost storage units close to the end-users to serve the requests to download this content could largely improve the network performance and service quality. Using this approach not only the users' requests can be immediately and efficiently addressed, but also the access to the backhaul could be significantly offloaded [10,12].

In addition to distributing the data, with the rapid increase in the number of wireless devices, there are more and more devices in each others proximity. Such "geographically close" wireless devices form an autonomous local network over which the users can communicate and exchange files without contacting the backhaul servers. Such scenario may occur for instance when co-workers are using their tablets to share and update files stored in the cloud (e.g. Dropbox), or when users, in the subway or a mall, are interested in watching the same popular video. Under such scenario, the benefits of communicating over a local network can be utilized not only to reduce the users' download time but also offload the backhaul of the data network (i.e., minimizing the download from it).

Furthermore, network coding (NC), initially introduced in [1], can help in offloading of the backhaul servers in the considered distributed cooperative data network scenario by maximizing the number of served users in one transmission, thus maximizing the backhaul offloading. Although NC was originally implemented at the network layer, more attractive application was found at the data link layer where there is coded combinations of files to improve throughput. Multiple areas of study have focused on various types of network coding, where this paper will focus on opportunistic network coding (ONC) [14], in particular instantly decodable network coding (IDNC) [13]. This technique has recently gained much attention due to its instant decodability (as the name suggests) by using a simple XOR operation that results in reducing the computational complexity of the decoding at the end users. It also provides a significant benefit to real-time communications, where studies in [3,7,13] show through a heuristic algorithm that utilizing IDNC results in shown significant performance improvements over uncoded transmissions in both centralized point-to-multipoint (PMP) and decentralized network settings.

Although much work has focused on implementing IDNC in various network models to address the above problem, the studies have focused on centralized PMP and distributed architectures. Furthermore, there has currently been no work to consider implementing IDNC in a cooperative setting where there is a focus on reducing the number of downloads required from the network servers. A network coded cooperative D2D-enabled architecture is considered in this paper, as it provides an attractive solution to offload the servers in a distributed data networks. Therefore, in this paper we aim to address the following question: *How should we encode files amongst users in a cooperative D2D-enabled transmission, such that the remaining requests from the users (if any) can be delivered (using IDNC) with a minimum number of transmissions from the network servers?* Although much research has focused on implementing IDNC in a PMP setting and even a distributed architecture with multiple servers, these approaches cannot be directly applied to a full-duplex cooperative network with the current graph modelling technique, therefore there is a need to develop a new model.

To address the question above, we first need to model the problem with a new graphical representation, namely the IDNC graph with induced subgraphs. The new graph representation is developed due to limitations of the conventional graphical representation when we wish to implement full-duplex communications into the system model. With the new graph modelling of the system, the optimal solution is formulated and shown to be NP-hard and not applicable for real-time applications [9]. The paper then proposes an online greedy heuristic algorithm that employs a maximum weighted vertex search over the new graph model. Simulation results show that the proposed algorithm when employed over the new graph model in a full-duplex D2D-enabled environment significantly outperforms the conventional uncooperative IDNC approach in reducing the downloads required from the servers of distributed data networks.

In this paper, we first present the system model and mathematical notation in Sect. 2. In Sect. 3, we formulate the problem, where a motivating example is first presented then followed by a mathematical optimal solution to the problem utilizing the new graph model. As the solution is found to be NP-hard, we then present the proposed greedy heuristic scheme in Sect. 4, followed by simulation results and discussion in Sect. 5. Lastly, this paper is concluded in Sect. 6.

2 System Model

A distributed wireless data network model is considered in this paper and is illustrated in Fig. 1. In this model, there is a set of N_u users defined as $\mathcal{U} = \{u_1, \dots, u_{N_u}\}$. In the system model, the assumption is made that all users are capable of full-duplex communications. The users will request to receive one file in the current time epoch, from a library of files defined as $\mathcal{F} = \left\{f_1, \dots, f_{N_f}\right\}$ with N_f files that are collectively stored at the servers. The servers are defined in the set $S = \{s_1, \dots, s_{N_s}\}$ with N_s servers. All servers are assumed to have full coverage, where the users in the coverage area are denoted by $\mathcal{U}(s_i)$ and

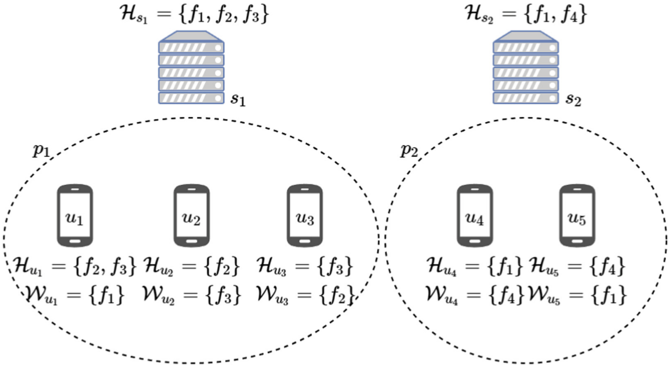

Fig. 1. An illustration depicting our system model for a distributed storage network, showing two servers, six users, and two proximity based wireless networks, where users can conduct cooperative D2D communications.

must satisfy $\mathcal{U}(s_i) \cap \mathcal{U} = \mathcal{U}$. The model shows a distributed setting where the users are in coverage of multiple servers. Also in the model, multiple proximity networks (possibly Wi-Fi or LAN) are shown. The proximity regions are defined as the proximity set $\mathcal{P} = \left\{p_1, \ldots, p_{N_p}\right\}$ with N_p proximity-enabled D2D communication networks. The proximity networks contain a subset of the users in $\mathcal{U}$, defined as $\mathcal{U}(p_i)$, that is the users in the coverage area of the proximity-enabled network p_i. It is assumed that there is no overlap of the users in each proximity set, that is, the users in each proximity network that are "geographically close" can communicate locally but not outside this network.

In our model, we assume the users have received some files in the initial transmission phase[1]. That is, a user u_i has partially downloaded some of the files from a transmitted frame which constitutes the users *Has* set $\mathcal{H}_{u_i}$. Furthermore, the remaining files wanted by user u_i in the frame form the user's *Wants* set, denoted as $\mathcal{W}_{u_i}$. Similarly, the servers will store a subset of the files in $\mathcal{F}$, however the union of all files at the servers should contain the complete set of $\mathcal{F}$ (with possible repetition). Here, a server's *Has* set is defined as $\mathcal{H}_{s_i}$. It is assumed that the servers will maintain a global knowledge of the system state during the initial transmissions, that is the users will respond with positive/negative feedback depending if they receive their files successfully or not. At completion of this phase the system will move into the recovery transmission phase.

IDNC can now be utilized to exploit users' side data to optimize the transmissions in the current time epoch. In the recovery transmission, we assume an erasure free channel, where the different users and servers will operate on orthogonal channels. It is also assumed that the servers have an unlimited capacity,

[1] This first phase of the transmission is known as the initial transmission phase. During the initial transmission the servers will attempt to serve all files to the users in the network. However, some users will have received only a portion of the files requested due to channel erasure.

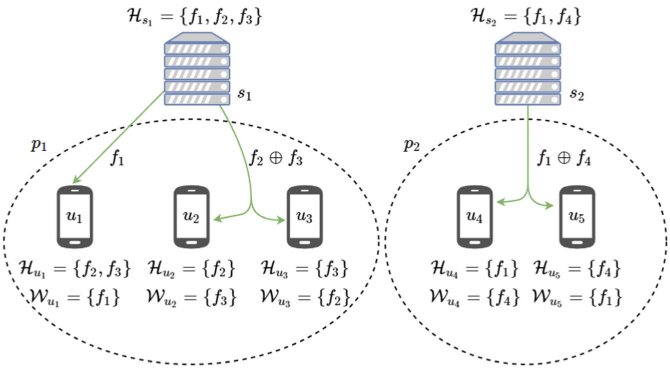

Fig. 2. Conventional IDNC approach from [2], where there is a total of three downloads required for the optimal solution.

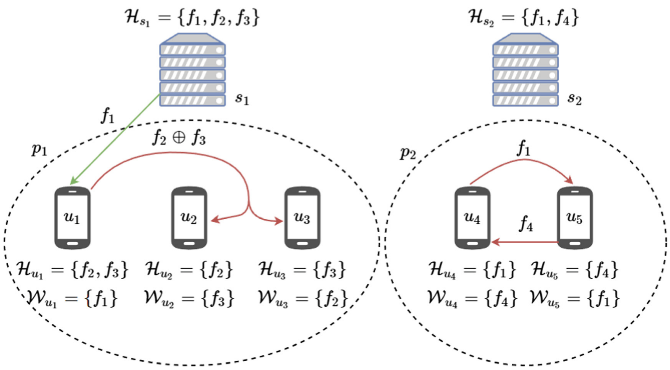

Fig. 3. A D2D-enabled approach showing the potential for offloading servers, where only one download is required.

such that after all cooperative D2D communications all requests remaining will be served by the server in the current time epoch. Here, the main goal is to optimize the selection of the files for network coding for the users and the servers, where priority is given to the cooperative D2D communications, such that we reduce the amount of downloads required from the servers.

3 Problem Formulation

3.1 Motivating Example

If we consider the system model depicted in Fig. 1, it can be shown by example that by finding the optimal solution, that is, to solve our question defined earlier, there are numerous allocations of coded transmissions that can lead to different results. For way of a motivating example, we present two different allocations:

in the first solution, we assume a conventional IDNC approach without the cooperative D2D enabled transmissions. In the second solution, we show the scenario for a network with cooperative D2D enabled communications.

Solution 1. In this solution (depicted in Fig. 2), we employ the method used in [2], this solution utilizes a conventional IDNC approach without D2D-enabled cooperation. One possible optimal solution using this method is:

- s_1 transmits f_1 to u_1 and $f_2 \oplus f_3$ to u_2 and u_3.
- s_2 transmits $f_1 \oplus f_4$ to u_4 and u_5.

This scenario results in consuming three downloads from the servers.

Solution 2. In this solution (depicted in Fig. 3), we show a scenario where a cooperative D2D setting is incorporated. One possible optimal solution is:

- s_1 transmits f_1 to u_1.
- u_1 transmits $f_2 \oplus f_3$ to u_2 and u_3.
- u_4 transmits f_1 to u_5 and u_5 transmits f_4 to u_1.

In the second solution, it can be seen that we only need one download from one of the servers, while no download is required from the other server. This approach shows that even in a small network setting, there is a download reduction of two thirds of the previous solution, freeing up valuable servers' resources.

Although much work has focused on implementing IDNC in various network models, the graph-theoretical modellings used in these cases are limited in a cooperative full-duplex environment. In previous approaches, the graph models incorporated assume that there is a clear differentiation between the sender and the receiver. In our setting, we remove this restriction and allow users the ability of full-duplex communications, therefore the existing IDNC graph models are not appropriate and there is a need for a new model.

3.2 Graph-Based Solution

To be able to formulate the optimal solution to the above problem, stated in Sect. 3, we will propose a new IDNC graph that represents coding opportunities. The IDNC graph when formulated, will represent all the possible files that can be XORed together to create a network coded transmission that can be decoded by the targeted end users. To form the model, we first define the graphs of interest in our system model as follows: Graph $\mathcal{G} = \left\{\mathcal{G}_1, \ldots, \mathcal{G}_{N_p}\right\}$ with the subgraph $\mathcal{G}_i$ representing each discrete D2D network, as well as the graph $\Psi = \{\Psi_1, \ldots, \Psi_{N_s}\}$ that is representing all servers, where the subgraph Ψ_i represents each individual server.

To construct each of the graphs previously mentioned, we proceed as follows:

Generate Vertex Set. Vertices are generated from a server and user perspective under the two conditions:

- Generate a vertex set for every server s_i in $\mathcal{S}$ that is represented in the subgraph Ψ_i, generating the vertices $v_{ijk}, \forall s_i \in \mathcal{S}$ and $f_k \in \left(\mathcal{H}_{s_i} \cap \mathcal{W}_{u_j}\right)$. The vertices of the subgraph are defined as $\Psi_{i_{(ijk)}}$.
- Generate a vertex set for every user $u_i \in p_n$ in $\mathcal{U}$ that is represented in the subgraph $\mathcal{G}_n$, generating the vertices $v_{ijk}, \forall u_i \in \mathcal{U}$ and $f_k \in \left(\mathcal{H}_{u_i} \cap \mathcal{W}_{u_j}\right)$ on the conditions $u_i \neq u_j$ and both $u_i, u_j \in p_n$. The vertices of the subgraph are defined as $\mathcal{G}_{n_{(ijk)}}$.

Generate Coding Opportunity Edges. In each individual subgraph in $\mathcal{G}$ and Ψ, we connect two vertices v_{ijk} and v_{lmn} with an edge if they satisfy one of the following two conditions:

- $f_k = f_n$, $u_j \neq u_m$ and $u_i = u_l$ if in $\mathcal{G}$ (or $s_i = s_l$ if in Ψ), meaning the two requested files are the same, and these files are requested by two different users.
- $f_n \in \mathcal{H}_{u_j}$ and $f_k \in \mathcal{H}_{u_m}$, representing a potential coding opportunity, so that when f_n and f_k are XORed both users can successfully decode and retrieve their requested file.

In the formulation so far we have incorporated graphs that represent coding opportunities from a user/server viewpoint. To further create a global awareness, we have to incorporate induced subgraphs (subgraphs of $\mathcal{G}$ and $\mathcal{H}$) that will represent the transmission conflicts (subgraph $\mathcal{K}$) and a subgraph to ensure only one transmission per user is permitted in the current time epoch (subgraph $\mathcal{L}$).

In Fig. 4, we depict the implementation of IDNC with the prescribed theoretical graph model for the example shown in Fig. 1. In our case, we show independently, firstly in subgraphs $\mathcal{G}_i$, the IDNC subgraphs for each individual proximity network. While in subgraphs Ψ_i, we show the potential coded transmissions in maximal cliques[2] for each server s_i. In the graph model shown in Fig. 4, it is clear that there is no interconnection between the graphs of $\mathcal{G}$ and Ψ (no edges connecting vertices). Therefore, we introduce the induced subgraphs approach to allow us to represent particular conditions that need to be accounted for in our network setting. We will introduce the graph $\mathcal{K}$, a set of subgraphs that ensures conflict free transmissions. Additionally, we introduce the graph $\mathcal{L}$ that contains a set of subgraphs that ensure users in a proximity network will not transmit more than once in the current time epoch.

Generate Induced Subgraphs. The two induced subgraphs as described are generated as follows:

[2] A clique is a sub-set of the graph, where every distinct pair of vertices in the induced subgraph are pairwise adjacent. A maximal clique is one that cannot be a subset of a larger clique [4].

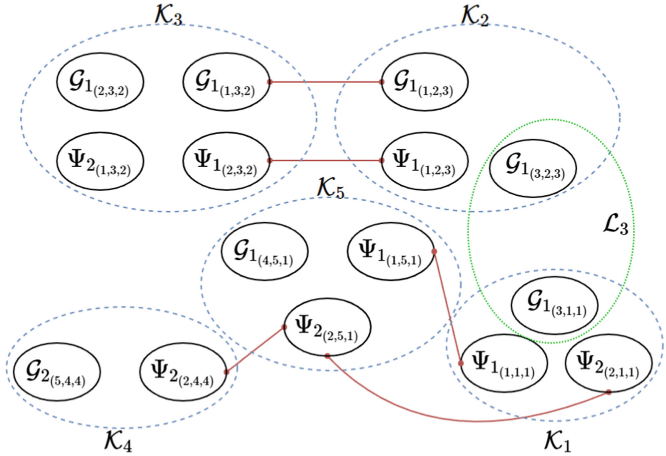

Fig. 4. A visualization of the IDNC induced subgraph methodology proposed for system model shown in Fig. 1. The figure shows coding opportunities represented by edges, transmission conflicts represented in subgraphs $\mathcal{K}_j$ and limitation of one transmission per user represented in subgraph $\mathcal{L}_3$ (that is, u_3 may only transmit to either u_2 or u_1 in the current time epoch).

- First, we define the set of induced subgraphs $\mathcal{K} = \{\mathcal{K}_1, \ldots, \mathcal{K}_D\}$ as a subset of both graphs $\mathcal{G}$ and Ψ, where the subgraphs may contain the null-set of either $\mathcal{G}$ or Ψ but not both. To generate the subgraph $\mathcal{K}_j$, each vertex v_{ijk} in both $\mathcal{G}$ and Ψ will form a member of the subgraph $\mathcal{K}_j$ for every vertex that has the same user u_j and file f_k.
- Similarly, we define the set of induced subgraphs $\mathcal{L} = \{\mathcal{L}_1, \ldots, \mathcal{L}_E\}$ as a subset of both graphs $\mathcal{G}$ and Ψ. A subgraph $\mathcal{L}_i$ is formed for any two vertices v_{ijk} and v_{lmn}, where $u_i = u_l$ but $u_j \neq u_m$ or $f_k \neq f_n$.

3.3 The Proposed Optimal Problem Formulation

In order to formulate the optimal solution we need to select the combination of disjoint maximal cliques from $\mathcal{G}$ such that when these vertices are removed, thus removing the union of the associated subgraph from Ψ, we reduce the remaining maximal cliques of Ψ. That is, we wish to minimize the number of maximal cliques in Ψ, which is equivalent to minimizing the number of downloads from the servers. Therefore, we can either find an expression to minimize the maximal cliques of Ψ, or equivalently we can minimize the number of maximal independent sets[3] of the complementary graph of Ψ, which we refer to as Ψ'. The minimum number of maximal cliques in a graph can be found by finding the chromatic number of a complementary graph [4]. Therefore, the optimal solution can be expressed in mathematically in (1)

[3] An independent set is a set of vertices in a graph, no two of which are adjacent. A maximal independent set is an independent set that is not a subset of any other independent set [4].

$$\min_{\mathcal{I}_1,\dots,\mathcal{I}_C} \mathcal{X} \left[\Psi' \backslash \coprod_{i=1}^{C} \left(\left(\mathcal{I}_i \bigcup_{j=1}^{D} \mathcal{K}_j \right) \bigcup \left(\mathcal{I}_i \bigcup_{k=1}^{E} \mathcal{L}_k \right) \right) \right] \quad (1)$$

$$\text{subject to} \quad \mathcal{I}_i \subseteq \mathcal{G}, \mathcal{I}_i \cap \Psi = \emptyset$$

$$\exists u_l \in \mathcal{U} \text{ where } \mathcal{F}(\mathcal{I}_i) \subseteq \mathcal{H}_{u_l}$$

where $\coprod$ is the disjoint set union operator and the first constraint ensures that the independent set $\mathcal{I}_i$ is selected only from the vertices that belong to the graph $\mathcal{G}$. This is to ensure the selected coded file combinations that the users serve reduces the chromatic number of the servers graph (optimal solution), as the chromatic number of the remaining graph Ψ is equal to the number of downloads required from the servers. The second constraint shown in (1) ensures that for all files selected in the independent set $\mathcal{I}_i$, denoted by $\mathcal{F}(\mathcal{I}_i)$, there exists a user that posses the files and can XOR them. If this is not satisfied then the coded transmission cannot be sent and the conditions in (1) are not met.

Solving for the optimal solution that has been presented requires that we determine the chromatic number of a graph (equivalent to finding all maximal cliques). It is well known that determining the chromatic number of a graph is proven to be NP-hard [9]. This is further compounded by the fact that we not only need to calculate the chromatic number of one graph, but we need to find the selection of independent sets such that we minimize the chromatic number of the remaining subgraph. Hence, the optimal solution is not applicable for online and real-time communications. Therefore, we will propose a heuristic scheme in the following section to solve sub-optimally.

4 The Proposed Greedy Heuristic Algorithm

In this section, we propose a greedy heuristic approach that can be solved in real-time and efficiently reduce the number of downloads from the servers. The fall back of a greedy heuristic scheme is that it does not in fact guarantee a global optimum, although we hope that this scheme will on average, give a good approximation to it.

An attractive feature of the graph-based formulation proposed in Sect. 3.2 is that we can directly apply a maximal weighted vertex search under a greedy policy on the graph model. With the graph already established from the problem formulation, we can carry out the maximum clique listing, using a maximum weighted vertex search as follows:

Firstly, we associate a weight to each vertex in the graph $\mathcal{G}$, each vertex's weight is proportional to δ_{ijk} which is the degree[4] of v_{ijk}. The weight is calculated in (2),

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

[4] The degree of a vertex (δ) in a graph is equal to the number of incident edges to that vertex [4].

Algorithm 1. Algorithm for Maximum Weight Vertex Search

Require: :
Initialisation :

- Construct Graphs $\mathcal{G}$, Ψ, $\mathcal{K}$ and $\mathcal{L}$
- $\mathcal{G}_s \leftarrow \{\mathcal{G}\}$
- $\Gamma \leftarrow \emptyset$

repeat
  $\forall v_{ijk} \in \mathcal{G}_s$: Compute w_{ijk} using (2)
  $v^*_{ijk} \Longleftarrow \text{argmax}_{v_{ijk}} w_{ijk}, \forall v_{ijk} \in \mathcal{G}_s$
  add v^*_{ijk} to Γ
  $\mathcal{G}_s \leftarrow \mathcal{G}_s \cup \ v^*_{ijk}$
until $\mathcal{G}_s = \emptyset$
return the clique listing Γ

where $\mathcal{N}(v_{ijk})$ is the set of adjacent vertices to v_{ijk}. Therefore, each vertex in graph $\mathcal{G}$ will have a large weighting if it has a large number of adjacent vertices, which themselves have a large number of adjacent vertices. The search will then select the vertex with the largest weighting, or between those with the same largest weight with equal probability. The algorithm then removes all non-adjacent vertices to v_{ijk}, and then checks if the vertex v_{ijk} belongs to a subgraph $\mathcal{K}_j$ or $\mathcal{L}_i$ and will remove *all* other vertices that are a member of either subgraph.

Secondly, the algorithm will then update all weights in $\mathcal{G}$ before selecting the next (if any) adjacent vertex in graph $\mathcal{G}$ that forms a clique with all previously selected vertices. The algorithm then continues to iterate these steps until no more vertices can be added to the clique. Finally, once a maximal clique listing is found and removed, we iterate the whole procedure until no more vertices are left in the graph $\mathcal{G}$. The steps of algorithm described is summarized in Algorithm 1.

At this stage, the algorithm has removed all possible D2D cooperations available in hopes to minimize the amount of downloads from the servers. Therefore, we now need to serve the remaining vertices in graph Ψ that were not served locally from D2D cooperation. We now conduct the exact same procedure on the remaining vertices in graph Ψ, where each maximal clique represents one download from a server and continue until all vertices are removed from the graph. Once all vertices have been removed from the graph Ψ the system will have reached absorption, that is, all users will have received the file in their *Wants* sets.

5 Simulation Results

In this section, we present our simulation results for the proposed algorithm in a cooperative D2D setting in comparison with a uncooperative decentralized conflict free IDNC approach that was incorporated in [2]. In both cases, the aim of the approaches is to reduce the number downloads from the servers.

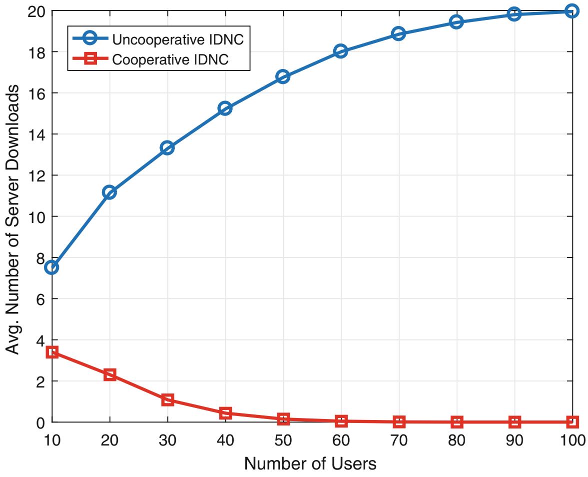

Fig. 5. The average number of downloads required from the servers as a function of the number of users.

In the simulations, each user is interested in receiving one file and has two files already received and stored in the *Has* set (when fixed), where the recovery downloads are to be completed in one time epoch. We assume each users' *Has* and *Wants* sets to be determined probabilistically, with uniform distribution over all files in the library. In all simulations, there are two servers available with total coverage of all users in the network, while in the cooperative model we consider a dual network where the users are split evenly between the two proximity networks p_1 and p_2 (similar to Fig. 1).

Firstly in Fig. 5, we show the average number of downloads required from the servers for a fixed number of files in the transmission frame of $N_f = 20$, as a function of the number of users N_u. The result shows that for the algorithm implemented for a cooperative D2D-enabled setting, as the number of users increase the average number of server downloads tends to monotonically decrease. Intuitively, this result is expected as more users in the network will result in a greater likelihood that the users can serve themselves independently from the servers, as the collective *Has* set of the users in the network will cover the files in the frame $\mathcal{F}$. Additionally, it can be seen that in comparison to a conventional uncooperative conflict-free IDNC approach, as the network size increases there is significant improvement, where we see an improvement of approximately 550% with only 20 devices in the network setting. Furthermore, approximately no downloads from the servers are required as the number of users approach 60 in this network setting, that is, 30 users in each D2D-enabled network.

Now if we consider fixing the number of users to 20, while varying the amount of files per transmission frame, we can see the results in Fig. 6 for cooperative versus uncooperative IDNC transmission schemes. In both cases, it can be seen as the number of files increase, both schemes show a similar increase on the

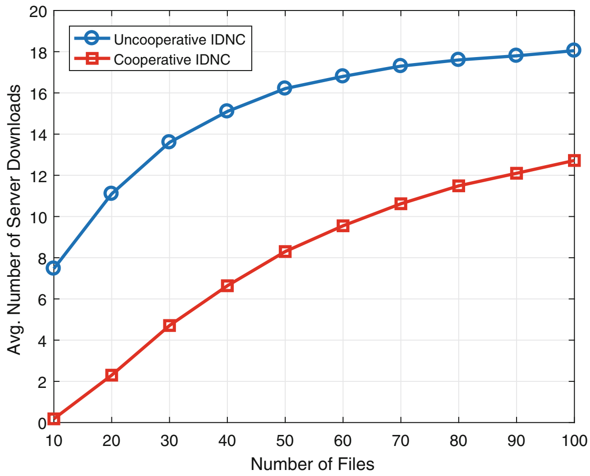

Fig. 6. The average number of downloads required from the servers as a function of the number of files.

number of downloads required from the servers. Although the two schemes tend to converge if we consider an asymptotic limit, the cooperative scheme still shows reasonable improvement of approximately 50% for up to 100 files. Again, this result is expected as increasing the number of files in a frame reduces the potential to leverage a coded transmission. Additionally, as the number of files increase the likelihood of a users ability to diffuse the wanted packets is diminished. Nevertheless, the cooperative approach still shows significant ability to reduce the number of downloads required from the network servers.

6 Conclusion

In this paper, we investigated the problem of offloading the expensive backhaul of data network servers through a network coded cooperative D2D network model. The problem was formulated using the IDNC induced subgraph model, where the optimal solution requires finding maximal cliques of multiple graphs. In the problem formulation it is found that an optimal solution is intractable and not solvable in real-time, therefore a greedy heuristic algorithm is employed using a maximum weighted vertex search approach. The paper utilizes the proposed subgraph model again in the heuristic approach, where the simulation results showed a significant improvement over the conventional method that incorporates IDNC in a distributed fashion without D2D enabled cooperation.

References

1. Ahlswede, R., Cai, N., Li, S.-Y.R., Yeung, R.W.: Network information flow. IEEE Trans. Inf. Theory **46**(4), 1204–1216 (2000)
2. Al-Habob, A.A., Sorour, S., Aboutorab, N., Sadeghi, P.: Conflict free network coding for distributed storage networks. In: 2015 IEEE International Conference on Communications (ICC), pp. 5517–5522. IEEE (2015)
3. Baran, P.: On distributed communications networks. IEEE Trans. Commun. Syst. **12**(1), 1–9 (1964)
4. Bondy, J.A., Murty, U.S.R.: Graph Theory with Applications, vol. 290. Macmillan, London (1976)
5. Bonomi, F., Milito, R., Zhu, J., Addepalli, S.: Fog computing and its role in the internet of things. In: Proceedings of 1st Edition of the MCC Workshop on Mobile Cloud Computing, pp. 13–16. ACM (2012)
6. Cisco: Cisco visual networking index: global mobile data traffic forecast update. Technical report, February 2016
7. Dimakis, A.G., Godfrey, P.B., Wu, Y., Wainwright, M.J., Ramchandran, K.: Network coding for distributed storage systems. IEEE Trans. Inf. Theory **56**(9), 4539–4551 (2010)
8. Dimakis, A.G., Ramachandran, K., Wu, Y., Suh, C.: A survey on network codes for distributed storage. Proc. IEEE **99**(3), 476–489 (2011)
9. Edwards, C.S., Elphick, C.H.: Lower bounds for the clique and the chromatic numbers of a graph. Discret. Appl. Math. **5**(1), 51–64 (1983)
10. Golrezaei, N., Molisch, A., Dimakis, A.G., Caire, G.: Femtocaching and device-to-device collaboration: a new architecture for wireless video distribution. IEEE Commun. Mag. **51**(4), 142–149 (2013)
11. Papailiopoulos, D.S., Luo, J., Dimakis, A.G., Huang, C., Li, J.: Simple regenerating codes: network coding for cloud storage. In: 2012 Proceedings of IEEE INFOCOM, pp. 2801–2805. IEEE (2012)
12. Shanmugam, K., Golrezaei, N., Dimakis, A.G., Molisch, A., Caire, G.: FemtoCaching: wireless content delivery through distributed caching helpers. IEEE Trans. Inf. Theory **59**(12), 8402–8413 (2013)
13. Sorour, S., Valaee, S.: On minimizing broadcast completion delay for instantly decodable network coding. In: 2010 IEEE International Conference on Communications (ICC), pp. 1–5. IEEE (2010)
14. Sorour, S., Valaee, S.: An adaptive network coded retransmission scheme for single-hop wireless multicast broadcast services. IEEE/ACM Trans. Netw. (TON) **19**(3), 869–878 (2011)