

## Research Article

# Network Coding for Backhaul Offloading in D2D Cooperative Fog Data Networks

Ben Quinton  and Neda Aboutorab 

University of New South Wales, Campbell, ACT 2612, Australia

Correspondence should be addressed to Neda Aboutorab; n.aboutorab@unsw.edu.au

Received 26 January 2018; Accepted 19 March 2018; Published 22 April 2018

Academic Editor: Xuyun Zhang

Copyright © 2018 Ben Quinton and Neda Aboutorab. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Future distributed data networks are expected to be assisted by users cooperation and coding schemes. Given the explosive increase in the end-users' demand for download of the content from the servers, in this paper, the implementation of instantly decodable network coding (IDNC) is considered in full-duplex device-to-device (D2D) cooperative fog data networks. In particular, this paper is concerned with designing efficient transmission schemes to offload traffic from the expensive backhaul of network servers by employing IDNC and users cooperation. The generalized framework where users send request for multiple packets and the transmissions are subject to erasure is considered. The optimal problem formulation is presented using the stochastic shortest path (SSP) technique over the IDNC graph with induced subgraphs. 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 offloading of the backhaul servers and thus saving valuable servers' resources. It is also shown that the performance of the proposed heuristic algorithm is very close to the optimal solution with much lower computational complexity.

## 1. Introduction

With the modern advancements of wireless communications, wireless networks have seen an explosion in data traffic over the past decade [1]. 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 [1, 2]. 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 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 [1]. 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, 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 [3–7]. The idea of distributing resources to the edge of a network is known as “fog” networking [8]. Fog networking is motivated by distributing the “popular” content that is demanded by the end-users closer to the edge of the network or even the end-Users. It is expected that with the proactive (i.e., without the end-users request) diffusing of the popular content from large network backhauls and distributing it in a “fog” of low-cost geographically close caches to the end-users will help to serve user download requests, in turn dramatically improving the network performance and quality-of-service [9]. Using this approach not only can the users' requests 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. These wireless devices can form an autonomous cooperative local network in a “geographically close” region where they can communicate and share files without the need for a centralized server. For example, this scenario may occur when coworkers are using wireless devices to update files stored in the cloud (e.g., Dropbox), or when users, in the subway, are interested in watching the same trending and popular video. Under these conditions, the benefits of communicating over a local network can be utilized to not only 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) as an efficient approach to further assist and improve the offloading of the backhaul servers while providing a faster and more reliable service to the users has attracted interest in recent years. NC, initially introduced in [13] by Ahlswede et al., 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 through users cooperation as well as coded servers transmissions, thus maximizing the backhaul offloading. In general, NC can be defined as a communication technique where information is encoded at the transmitter or the intermediate nodes in the network and then decoded by the users. NC has shown the ability to improve throughput, reduce delays, and provide more robust networks [14–18].

Multiple areas of study have focused on various types of network coding, where this paper will focus on opportunistic network coding (ONC) [19], in particular instantly decodable network coding (IDNC) [14]. 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 [4, 14, 20, 21] show through heuristic algorithms utilizing IDNC results in 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, the studies have focused on centralized PMP and noncooperative scenarios. Only limited results exist on implementing IDNC in a cooperative setting where there is a focus on reducing the number of downloads required from the backhaul servers. For instance, the authors’ previous work [22] considers the problem of designing network coding schemes for cooperative fog networks with a somewhat limited set of numerical analysis and under simplified assumptions such as single file request from the users as well as the case with no channel erasures. A *generalized* network coded cooperative D2D-enabled fog architecture under more realistic assumptions (i.e., multiple file requests from the users over multiple time epochs as well as channel erasures in the users-to-users and servers-to-users transmission links) is considered in this paper, where an attractive solution that aims to minimize the number of downloads from the backhaul servers in a cooperative fog

data network is presented. More specifically, this paper aims to address the following question: *How should files be encoded (using IDNC) amongst users in generalized cooperative D2D-enabled transmissions over multiple time epochs, such that the remaining requests from the users (if any) can be delivered (employing IDNC) with a minimum number of transmissions from the backhaul servers?*

To address the question above, in this paper we have utilized the stochastic shortest path (SSP) technique to study the maximum backhaul offloading problem in the cooperative network coded fog data networks. The problem is first modelled using graphical representation proposed in [22], namely, the IDNC graph with induced subgraphs. This graph representation overcomes the limitations of the conventional graphical representation [23], as it is not compatible to implement with a system model having full-duplex communications and the additional constraints that arise from D2D-enabled communications. With this graph modelling of the system, the optimal solution is formulated using the SSP technique and shown to be NP-hard and not applicable for real-time applications [24]. As the optimal solution is NP-hard, a greedy heuristic algorithm is proposed, employing a maximum weighted vertex search over the graph model. Simulation results show that the proposed algorithm significantly outperforms the conventional uncooperative IDNC approach in reducing the downloads required from the servers of the cooperative fog data networks.

The main contributions of the paper can be summarized as follows:

- (i) We solve the backhaul servers offloading problem in generalized network coded cooperative fog data networks by utilizing the IDNC graph with induced subgraphs as well as the SSP technique under the general assumptions where users can demand multiple files over multiple time epochs and the users and servers wireless channels are subject to erasures. This, to the best of our knowledge, did not exist in the literature before.
- (ii) After theoretically formulating the maximum offloading problem in the generalized network coded cooperative fog data networks and guided by the properties of the optimal solution, we present a computationally simple heuristic graph-based algorithm to find network coded transmissions amongst the users and from the servers that efficiently reduce the number of transmissions required from the network servers.
- (iii) We also present the complexity analysis of the proposed heuristic algorithm and the optimal solution and show that the complexity of the proposed algorithm is much lower than the optimal solution.
- (iv) We then assess the performance of the proposed algorithm through extensive simulations as well as comparison with the optimal performance. The simulations and comparison results confirm the effectiveness of the proposed algorithm and the fact that its performance is very close to the optimal performance.

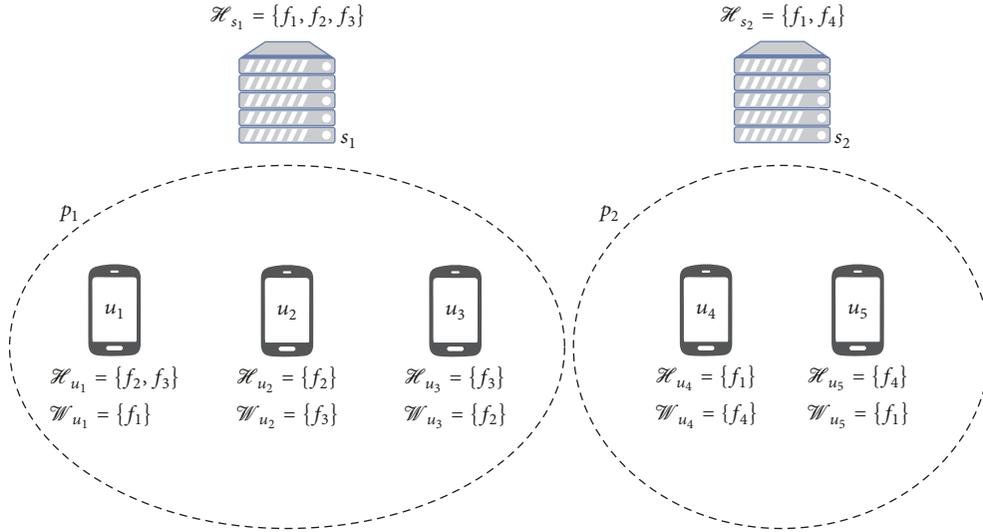


FIGURE 1: An illustration depicting the system model for a distributed storage network, showing two servers, six users, and two proximity based wireless networks, where users can conduct cooperative D2D communications.

The organization of the paper is as follows: System model and mathematical notation are presented in Section 2. In Section 3, the SSP-based problem formulation for the generalized cooperative fog data network model is proposed. Then the proposed heuristic algorithm is provided in Section 5. The simulation results and complexity analysis are presented in Sections 6 and 7, respectively. Lastly, the paper is concluded in Section 8.

## 2. System Model

A distributed wireless data network model is illustrated in Figure 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 some files from a library of files defined as  $\mathcal{F} = \{f_1, \dots, f_{N_f}\}$  with  $N_f$  files that are collectively stored at the servers. The servers are defined in the set  $\mathcal{S} = \{s_1, \dots, s_{N_s}\}$  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 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, where the users are capable of full-duplex cooperative D2D communications. The proximity regions are therefore defined as the proximity set  $\mathcal{P} = \{p_1, \dots, p_{N_p}\}$  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. Unlike previous works [22, 23] that consider ideal transmissions amongst the users

and from the servers, in this manuscript, we have made the more realistic assumption that the users-to-users and servers-to-users’ communications links are subject to random erasure.

In order to leverage IDNC, it is assumed that the users have received files in previous time epochs, where a user  $u_i$  has partially downloaded some of the files from a transmitted frame which constitutes the users’ *Has* set  $\mathcal{H}_{u_i}$ . 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; following this initial transmission some users will have received only a portion of the files requested due to channel erasure. Furthermore, the remaining files wanted by user  $u_i$  in the frame form the user’s *Wants* set, denoted as  $\mathcal{W}_{u_i}$ . The remaining files (if any) that are neither in the users’ *Wants* or *Has* sets are defined as the users’ *Lacks* set by  $\mathcal{B}_{u_i}$ . 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 whether the user receives 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, where there is now the possibility for IDNC transmissions with cooperative D2D communications supported by the servers. With the servers now aware of the state of the network, the servers will generate a *state feedback matrix* (SFM) to represent the network state. The SFM is generated as follows; generate the matrix  $\mathbf{F} = [\gamma_{ij}]$ ,  $\forall u_i \in \mathcal{U}, f_j \in \mathcal{F}$  where

$$\gamma_{ij} = \begin{cases} 0, & j \in \mathcal{H}_{u_i} \\ 1, & j \in \mathcal{W}_{u_i} \\ -1, & j \in \mathcal{B}_{u_i} \end{cases} \quad (1)$$

$$\mathbf{F} = \text{Users } (u_i) \begin{matrix} \xrightarrow{\text{Files } (f_j)} \\ \left( \begin{array}{cccc} \gamma_{1,1} & \gamma_{1,2} & \cdots & \gamma_{1,j} \\ \gamma_{2,1} & \gamma_{2,2} & \cdots & \gamma_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{i,1} & \gamma_{i,2} & \cdots & \gamma_{i,j} \end{array} \right) \end{matrix}$$

The SFM is generated for the recovery transmission phase and is shown in (1). Finally, when channel erasure is considered the probability of erasure between the receiving user  $u_i$  and the transmitter (either user  $u_j$  or server  $s_j$ ) is defined as  $e_{ij}$  (also define  $q_{ij} = 1 - e_{ij}$ ).

### 3. Problem Formulation

As discussed earlier, we consider the backhaul offloading maximization problem in cooperative fog data networks where the users in the network may demand multiple files and the recovery transmissions can occur over multiple time epochs. The network state is considered to be known after the initial transmission phase. The model will allow for D2D communications to take priority, where all remaining requests will be transmitted by the servers on orthogonal channels (assuming the servers have unlimited capacity). In this model, it is also assumed that the recovery transmissions (both users-to-users and servers-to-users transmissions) will be subject to erasures (i.e., for the probability of channel erasure to occur during any transmission).

### 4. IDNC Graph with Induced Subgraphs Model

To be able to formulate the optimal solution to the above-mentioned problem, the IDNC graph with induced subgraphs model [22] is employed in this paper. This IDNC graph model represents all the possible files that can be XORed together to create a network coded transmission that can be decoded by the targeted end-users (or transmissions not requiring any network coding). To form the model, firstly the graphs of interest in our considered system model are defined as follows: Graph  $\mathcal{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_{N_p}\}$  with the subgraph  $\mathcal{G}_i$  representing each discrete proximity network, as well as graph  $\Psi = \{\Psi_1, \dots, \Psi_{N_s}\}$  that is representing all servers, where the subgraph  $\Psi_i$  represents each individual server. To construct each of the graphs previously mentioned, proceed as follows:

- (1) Generate Vertex Set: vertices are generated from a server and user perspective under the two following conditions:
  - (i) Generate a vertex set for every server  $s_i$  in  $\mathcal{S}$  that is represented in the subgraph  $\Psi_i$ , generating the

vertices  $v_{ijk}$ ,  $\forall s_i \in \mathcal{S}$  and  $f_k \in (\mathcal{H}_{s_i} \cap \mathcal{W}_{u_j})$ . The vertices of the subgraph are defined as  $\Psi_{i(ijk)}$ .

- (ii) Generate a vertex set for every user  $u_i \in \mathcal{P}_n$  in  $\mathcal{U}$  that is represented in the subgraph  $\mathcal{G}_n$ , generating the vertices  $v_{ijk}$ ,  $\forall u_i \in \mathcal{U}$  and  $f_k \in (\mathcal{H}_{u_i} \cap \mathcal{W}_{u_j})$  on the conditions  $u_i \neq u_j$  and both  $u_i, u_j \in \mathcal{P}_n$ . The vertices of the subgraph are defined as  $\mathcal{G}_{n(ijk)}$ .
- (2) Generate Coding Opportunity Edges: in each individual subgraph in  $\mathcal{G}$  and  $\Psi$ , two vertices  $v_{ijk}$  and  $v_{lmn}$  are connected with an edge if they satisfy one of the following two conditions:
    - (i)  $f_k = f_n$ ,  $u_j \neq u_m$  and  $u_i = u_l$  if in  $\mathcal{G}$  (or  $s_i = s_l$  if in  $\Psi$ ), meaning the two requested files are the same, and these files are requested by two different users.
    - (ii)  $f_n \in \mathcal{H}_{u_j}$  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, the graphs implemented represent coding or transmission opportunities from a user/server viewpoint. To further create a global awareness, induced subgraphs (subgraphs of  $\mathcal{G}$  and  $\Psi$ ) are incorporated 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  $\mathcal{L}$ ).
- (3) Generate Induced Subgraphs: the two induced subgraphs described are generated as follows:
    - (i) The first set of induced subgraphs are defined as  $\mathcal{K} = \{\mathcal{K}_1, \dots, \mathcal{K}_D\}$ ; these are a subset of both graphs  $\mathcal{G}$  and  $\Psi$ , where the subgraphs may contain the null-set of either  $\mathcal{G}$  or  $\Psi$  but not both. To generate the subgraph  $\mathcal{K}_j$ , each vertex  $v_{ijk}$  in both  $\mathcal{G}$  and  $\Psi$  will form a member of the subgraph  $\mathcal{K}_j$  for every vertex that has the same user  $u_j$  and file  $f_k$ .
    - (ii) The next set of induced subgraphs are defined as  $\mathcal{L} = \{\mathcal{L}_1, \dots, \mathcal{L}_E\}$ ; these are a subset of both graphs  $\mathcal{G}$  and  $\Psi$ . 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  $f_k \neq f_n$ .

In Figure 2, an illustration depicts the implementation of IDNC with the prescribed theoretical graph model for the example shown in Figure 1. In this example, it is helpful to show independently the IDNC subgraphs for each individual proximity network in subgraphs  $\mathcal{G}_i$ . While in subgraphs  $\Psi_i$ , the potential coded transmissions are shown in maximal cliques (a clique is a subset of the graph, where every distinct pair of vertices in the induced subgraph are pairwise adjacent. A maximal clique is one that cannot be

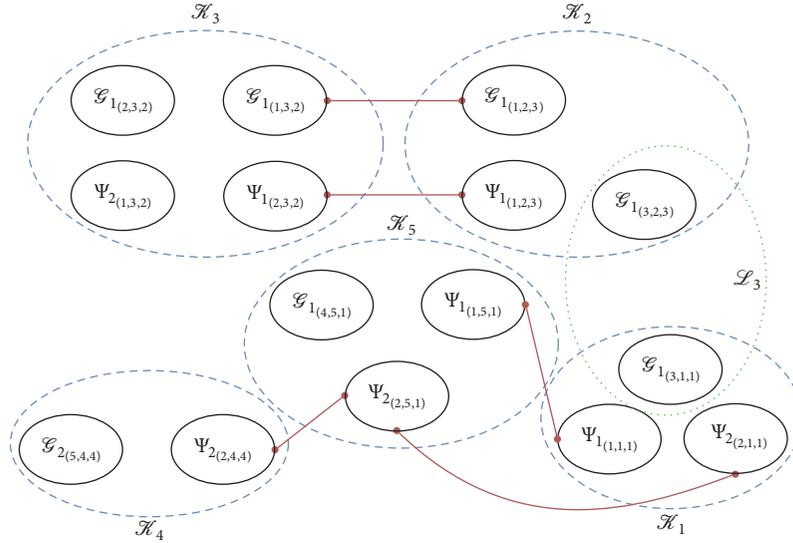


FIGURE 2: A visualization of the theoretical graph model proposed for system model shown in Figure 1. The figure shows coding opportunities represented by edges, transmission conflicts represented in subgraphs  $\mathcal{K}_j$ , and limitation of one transmission per user represented in subgraph  $\mathcal{L}_3$ .

a subset of a larger clique [25]) for each server  $s_i$ . In the graph model shown in Figure 2, it is clear that there is no interconnection between the graphs of  $\mathcal{G}$  and  $\Psi$  (no edges connecting vertices). Therefore, the induced subgraphs approach allows the representation of particular conditions that need to be accounted for in the network setting. The graph  $\mathcal{K}$  represents a set of subgraphs that ensures conflict-free transmissions. Additionally, the graph  $\mathcal{L}$  contains a set of subgraphs that ensures users in a proximity network will not transmit more than once in the current time epoch.

**4.1. The Proposed Optimal Problem Formulation Using SSP.** In order to formulate the optimal solution using the IDNC graph with induced subgraphs model that represents the global state of the network, a shortest stochastic path problem (SSP) technique is employed. SSP is in fact a special case of the well known Markov decision processes (MDP) [26] where MDPs are problems where a sequence of decisions are made in stochastic environments, such as is the network model with channel erasures. In SSP problem, different possible situations that the system could encounter are modelled as states. In each state  $s$ , the system must select an action  $a$  from an action space that will charge it an immediate cost  $c(s, a)$ . The terminating condition of the system can be thus represented as a zero cost absorbing state. Once an action  $a$  is taken at state  $s$ , the system can move to a state  $s'$  with probability  $P_a(s, s')$ , which only depends on the current state and the taken action. An SSP policy  $\pi = [\pi(s)]$  is a mapping from state space to action space that associates a given action with each of the states. The optimal policy  $\pi^*$  is the one that minimizes the cumulative mean cost until the absorbing state is reached.

To formulate the SSP problem, the following definitions are required [14]:

- (1) *State space  $\mathcal{S}$* : the state space  $\mathcal{S}$  is represented by all the possibilities that the SFM may represent. At each state  $s$ , the SFM will represent the *Lacks*, *Wants* and *Has* sets of all users. The state  $s_a$  (the state where the system has reached absorption) is defined for when all users have received their wanted packets such that  $|\mathcal{W}_{u_i}(s_a)| = 0, \forall i \in U$ .
- (2) *Action spaces  $\mathcal{A}(s)$* : at each state  $s$  the system can represent there is a set of actions  $\mathcal{A}$ , that is, a selection of maximal cliques from both graphs  $\mathcal{G}(s)$  and  $\Psi(s)$  that are constructed from the SFM at state  $s$ . All actions  $a \in \mathcal{A}(s)$  at state  $s$  are defined by first selecting a set of maximal cliques  $\Gamma_{\mathcal{G}(s)}$  from graph  $\mathcal{G}(s)$ . Then for all vertices in  $\Gamma_{\mathcal{G}(s)}$  that are also a subset of the graphs  $\mathcal{K}(s)$  and  $\mathcal{L}(s)$  that are to be removed from the graphs  $\Psi(s)$  and  $\mathcal{G}(s)$ . The set of maximal cliques selected in  $\Gamma_{\mathcal{G}(s)}$  must result in removing all vertices in  $\mathcal{G}(s)$  after this process. Finally, a set of maximal cliques  $\Gamma_{\Psi(s)}$  is selected from the remaining graph  $\Psi(s)$ , ensuring all users are targeted in the current time epoch.
- (3) *State-Action transition probabilities*: in order to define the state-action transitions probability  $P_a(s, s')$  for the action  $a$  it is useful to introduce two new sets:

$$\begin{aligned} \mathcal{X} &= \{i \in \mathcal{T}(\Gamma_{\mathcal{G}(s)}), i \in \mathcal{T}(\Gamma_{\Psi(s)}) \mid |\mathcal{W}_{u_i}(s')| \\ &< |\mathcal{W}_{u_i}(s)|\} \\ \mathcal{Y} &= \{i \in \mathcal{T}(\Gamma_{\mathcal{G}(s)}), i \in \mathcal{T}(\Gamma_{\Psi(s)}) \mid |\mathcal{W}_{u_i}(s')| \\ &= |\mathcal{W}_{u_i}(s)|\}, \end{aligned} \quad (2)$$

where  $\mathcal{T}(\Gamma_{\mathcal{G}(s)})$  are the targeted users in the maximal clique that is selected from the graph  $\mathcal{G}(s)$ .

Also  $\mathcal{T}(\Gamma_{\Psi(s)})$  is defined as the targeted users in the maximal clique that is selected from the graph  $\Psi(s)$ . Therefore, the state-action probability is given as

$$P_a(s, s') = \prod_{i \in \mathcal{Y}} q_{ij} \prod_{i \in \mathcal{X}} e_{ij}. \quad (3)$$

- (4) *State-action costs*: in the problem formulation the aim is to reduce the amount of transmissions from the servers-to-users. Accordingly, transmissions that occur from using cooperative transmissions will incur no cost. Moreover, for every state any state-action transition will incur a cost of 1 for each orthogonal channel used by the servers. For example, if an action at state  $s$  required the servers to transmit on a total of three orthogonal channels collectively, the cost incurred will be 3. Therefore, the amount of maximal cliques that are in the set  $\Gamma_{\Psi(s)}$  will directly determine the cost, which is the same as the cardinality of set  $\Gamma_{\Psi(s)}$  or equivalently  $c(s, a) = |\Gamma_{\Psi(s)}|$ , as user-to-user transmissions come at zero cost.

The optimal policy  $\pi^*$  of an SSP is the one that minimizes the cumulative mean cost until completion is reached. This policy is given in

$$\pi^*(s) = \arg \min_{a \in \mathcal{A}(s)} \left\{ c(s, a) + \sum_{s' \in \mathcal{S}(s, a)} P_a(s, s') V_{\pi^*}(s') \right\}, \quad (4)$$

where  $V_{\pi^*}(s)$  is the expected cumulative cost until the system reaches absorption. From the SSP problem formulation, it is clear that the optimal policy requires checking through all possible actions (i.e., maximal cliques), which is proven to be NP-hard [24].

## 5. The Proposed Greedy Heuristic Algorithm

In this section, a greedy heuristic approach is proposed that can be solved in real-time and aims to 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 this scheme will, on average, give a good approximation to it. An attractive feature of the graph-based formulation proposed in Section 3 is that a maximum weighted vertex search under a greedy policy can be directly applied on the graph model.

As discussed already, if an optimal solution is NP-hard there is a requirement for a suboptimal solution for real-time communications. Figures 3 and 4 demonstrate by example the impact that state-action transitions may have depending on the action selected from the current state. Figure 3 shows the case of selecting action  $a_1$ , where the associated cost is zero. Here we see that if  $e_1$  and  $e_2$  are large there is a greater chance that the packets will not be received. Therefore, it is desirable that a user that has a poor channel (high probability of erasure) with the server to be targeted with cooperative transmissions. Furthermore, Figure 4 shows a subset of consecutive actions and the resulting state and cost;

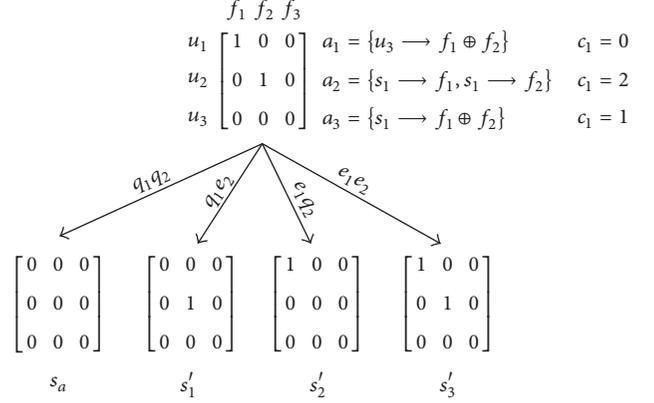


FIGURE 3: Example SFM at the beginning of the recovery transmissions, shown is a subset of possible actions that can be taken at the current time epoch. SFM transition shows dependence on channel erasure for a given action, in this example that is  $a_1$ .

here clearly the shortest (lowest cost) path to absorption is the ones that incur zero cost.

Therefore, it will be desirable for users to serve themselves with the greatest chance of serving as many users simultaneously as possible in each time epoch. To represent this a vertex  $v_{ijk}(s)$  in either graph  $\mathcal{S}$  or graph  $\Psi$  will have a large weight  $w_{ijk}(s)$  if it has a large number of adjacent vertices, which themselves have a large number of adjacent vertices. Additionally, as shown in the motivating example it is preferable to place further (higher) weighting on transmissions for users with high probability of erasure with the server.

Therefore, a weight  $\tau$  is first introduced for each user in  $\mathcal{U}$  that is proportional to a users' likely mean completion time (dependent on user-to-server channel). The weight also varies depending on the state the system is in; therefore define the weight for each state  $s$

$$\tau_{u_i}(s) = \frac{|\mathcal{W}_i(s)|}{1 - e_{ij}}, \quad (5)$$

where (5) is large for a user with a large *Wants* set and a large probability of channel erasure for user-to-server. Now that a weight is defined in (5), at each state  $s$  the set of vertices adjacent to  $v_{ij}(s)$  is defined as the set of vertices in  $\mathcal{N}(s)$ . From this the weighted vertex degree  $\Delta_{ij}(s)$  can be calculated as

$$\Delta_{ij}(s) = \sum_{\forall v_{kl} \in \mathcal{N}(s)} \tau_{u_k}(s), \quad (6)$$

where (6) defines a value for each vertex that is proportional to how many adjacent vertices it has. Moreover, it describes the potential that the particular vertex has in relative terms of coding potential with other users. Finally, the multiplication of both weights will yield the final weight for each vertex and is defined in

$$w_{ij}(s) = \tau_{u_i}(s) \Delta_{ij}(s), \quad (7)$$

where a large weight  $w_{ij}(s)$  for the vertex  $v_{ij}(s)$  is reflected by a large weight  $\tau_{u_i}(s)$  and also has a large number of connected

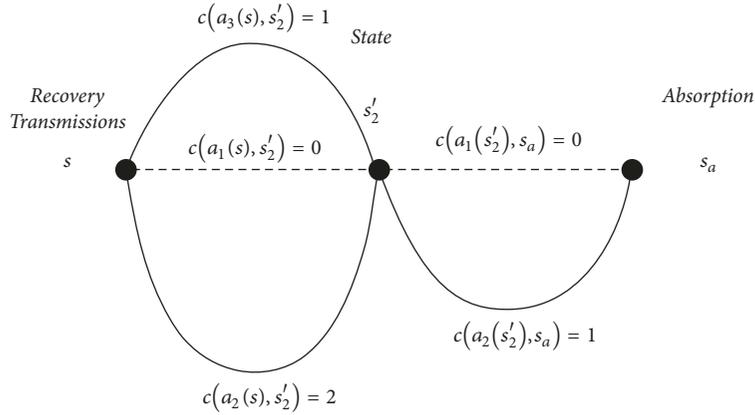


FIGURE 4: Visualization of a SSP, here only state-action probabilities of no erasure are considered. The diagram shows the transition from state  $s$  to absorption state  $s_a$  that takes multiple paths, where the shortest cost to absorption will be taking the actions of 0 cost at both states  $s$  and  $s'_2$ .

vertices that are also connected to a large number of vertices represented by  $\Delta_{ij}(s)$ .

Once the weights are calculated for all the vertices in the graph, the greedy heuristic search will select the vertex  $v_{ijk}$  with the largest weighting, or between those with the same largest weight with equal probability. The algorithm then removes all nonadjacent 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 (resulting in a maximal clique). Finally, once a maximal clique listing is found and removed, the whole procedure is iterated until no more vertices are left in the graph  $\mathcal{G}$ .

At this stage, the algorithm has removed all possible D2D cooperation's available, with the aim of minimizing the amount of downloads from the servers. Therefore, the remaining vertices in graph  $\Psi$  need to be served, which were not served locally from D2D cooperation. The exact same procedure can be conducted on the remaining vertices in graph  $\Psi$ , where each maximal clique represents one download from a server and this continues until all vertices are removed from the graph. Once all vertices have been removed from the graph  $\Psi$  the system will have reached absorption and all users will have received the file in their *Wants* sets.

It is worth mentioning that as users are required to receive multiple files (multiple time epochs required) and there is a chance of channel erasure in transmission, after each sequence of transmissions the SFM needs to be updated. The updated SFM should then be again represented in the graph model where the algorithm will repeat. The process continues until all users have received all wanted files and the system reaches absorption.

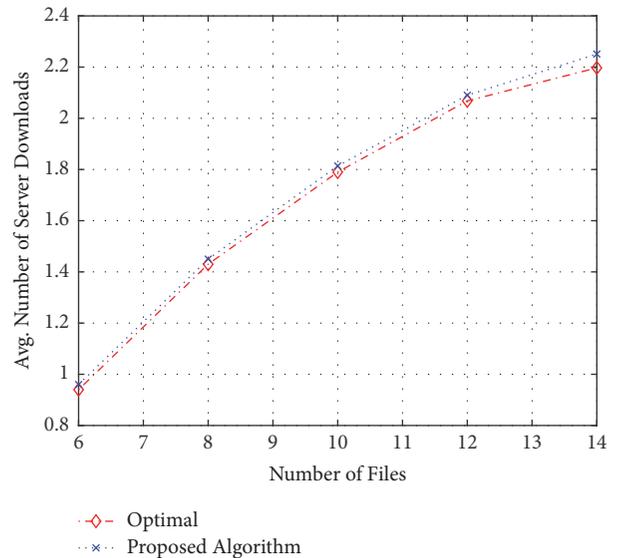


FIGURE 5: Performance of algorithm against the optimal solution.

## 6. Simulation Results

In this section, the simulation results are presented for the proposed algorithm in a cooperative D2D setting in comparison with a uncooperative decentralized conflict-free IDNC approach that was incorporated in [21]. The aim of both approaches is to reduce the number downloads from the servers. The cooperative approach is a natural next step in the evolution and adoption of F-RAN in the future wireless communications systems to maximize backhaul offloading.

Firstly, in Figure 5 the performance of the proposed heuristic algorithm is compared with the optimal solution. The optimal solution is calculated by a brute force algorithm that checks over all possible IDNC combinations. The simulation results are presented for a small network setting, where there is one server and four users requesting one file with two files in each user's *Has* set. It can be seen from

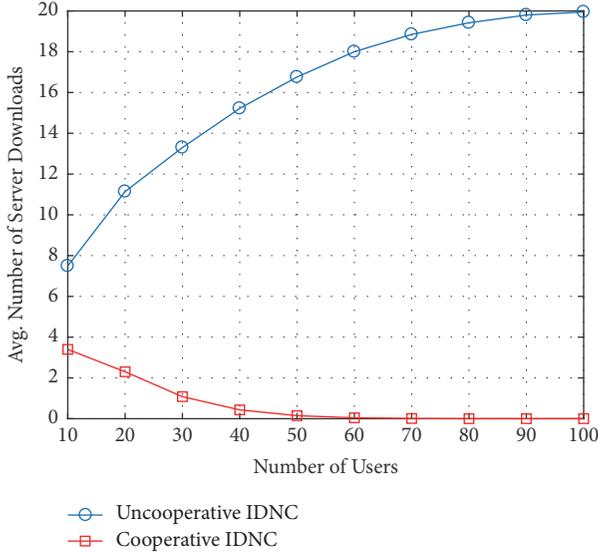


FIGURE 6: The average number of downloads required from the servers as a function of the number of users (erasure free case).

the results shown in Figure 5 that the proposed algorithm shows a good approximation to the optimal solution, with a small divergence for an increasing amount of files. This divergence from the optimal solution is expected, as the network size increases the divergence from the optimal solution is expected to increase.

It is also worth mentioning that the proposed heuristic algorithm can be solved with a much lower computational complexity and in real-time. However, as expected, there will be a tradeoff here between performance and complexity (nevertheless, the performance of the proposed heuristic is very close to the optimal one).

In the rest of this section, we assume there are two servers available with total coverage of all users in the network, while in the cooperative model a dual network is considered, where the users are split evenly between the two proximity networks  $p_1$  and  $p_2$  (similar to Figure 1). In the simulation results presented in Figures 6 and 7, 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. It is also assumed that the transmissions are erasure free. A summary of the data sets used (when fixed) in the simulations can be seen in Table 1.

In Figure 6, the average number of downloads required from the servers for a fixed number of files in the transmission frame of  $N_f = 20$  is shown, 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

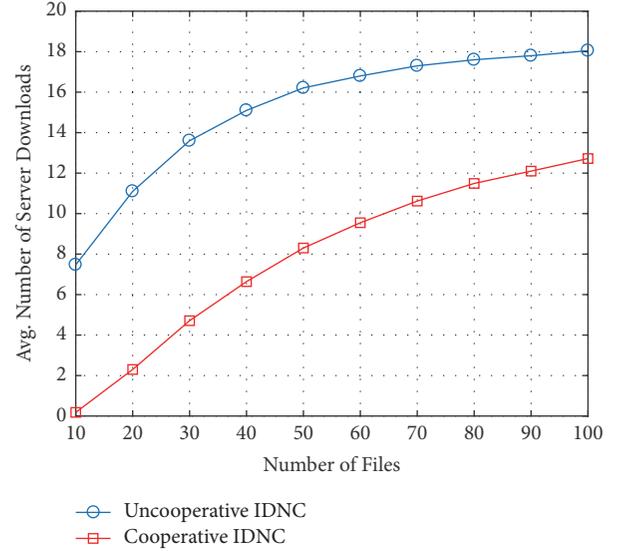


FIGURE 7: The average number of downloads required from the servers as a function of the number of files (erasure free case).

the network size increases there is significant improvement, where there is 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 approaches 60 in this network setting, that is, 30 users in each D2D-enabled network.

In Figure 7 number of users are fixed to 20, while varying the amount of files per transmission frame. Figure 7 shows the results for cooperative versus uncooperative IDNC transmission schemes. In both cases, it can be seen as the number of files increases, both schemes show a similar increase on the number of downloads required from the servers. Although the two schemes tend to converge if an asymptotic limit is considered, 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 that a user can serve another user in a cooperative D2D-enabled network is reduced.

In the next two simulations presented in Figures 8 and 9, D2D cooperation with channel erasure is considered. In both simulations erasure between user-to-server transmissions is evenly distributed from  $[0, 0.03]$ , while the transmissions from user-to-users are also evenly distributed from  $[0, 0.01]$ . In this case, it is assumed that it is more likely that users will be closer geographically and therefore will in general have a better chance for a successful transmission. Firstly in Figure 8, the number of files in the frame is fixed to 20, while all users will demand two random files from the frame. It can be seen in Figure 8 that as the number of users increases the cooperative approach significantly outperforms the uncooperative approach. Moreover, as the number of users increases the improvement also increases. In this case, the model shows a performance increase range of approximately 150–300% from 10 to 100 users.

TABLE 1: A summary of the data sets used in the simulations, the stated figures are given for the case when the parameter is fixed in the simulation. In all simulations, the number of proximity D2D enabled networks ( $N_p$ ) is fixed to 2.

Servers ( $N_s$ )	Files ( $N_f$ )	Users ( $N_u$ )	Has set $ \mathcal{H}_u $	Wants set $ \mathcal{W}_u $
2	20	20	2	1

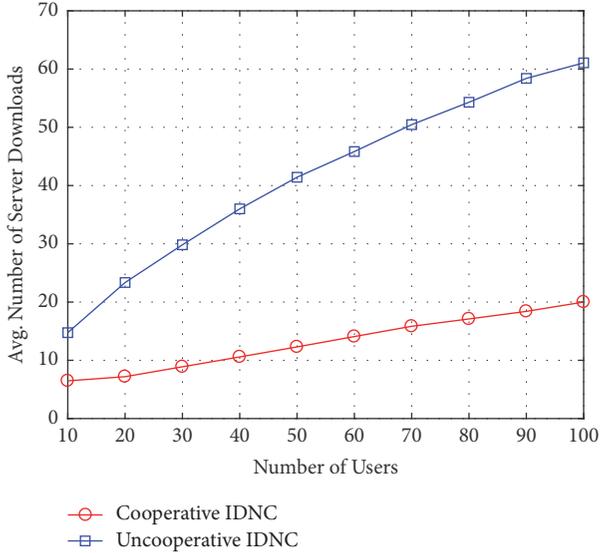


FIGURE 8: The average number of downloads required from the servers as a function of the number of users in the network (channels are subject to erasure).

In Figure 9, the number of users in the network is fixed at 20 while the number of files in the frame is varied. The results in Figure 9 show that there is significant offloading of the network backhaul, although it appears that the two approaches will tend to converge for a very large number of files. This result is intuitively expected; this is because as the number of files increases with a fixed number of users, the likelihood of the users' ability to diffuse the wanted packets is diminished. Nevertheless, the cooperative approach shows significant ability to reduce the traffic from servers.

## 7. Complexity Analysis

To gain an appreciation of the performance of the proposed heuristic algorithm in comparison to the optimal solution, we will investigate their worst-case time complexities. Firstly, we will analyze the worst-case time complexity of the optimal solution. As the optimal solution to the problem (the optimal policy) requires checking through all possible actions (i.e., finding all maximal cliques) we would require some algorithm to execute this. One popular method to achieve this, which is generally more efficient than other algorithms, is the Bron-Kerbosch (B-K) algorithm [27]. To find the worst-case time complexity, we need to find the vertex set size of the entire graph. The size of the server graph ( $\Psi$ ) will be of the order  $O(N_u N_f)$ , while the size of the graph representing potential cooperative transmissions ( $\mathcal{E}$ ) is of the order of  $O(N_s N_u N_f)$ . Therefore, by utilizing the B-K

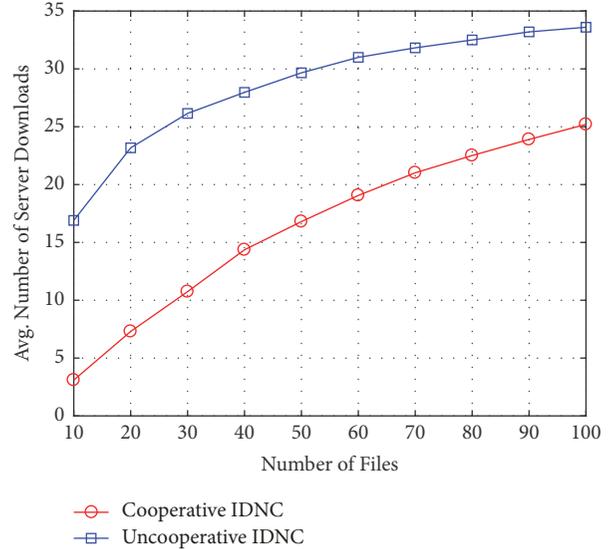


FIGURE 9: The average number of downloads required from the servers as a function of the number of files (channels are subject to erasure).

algorithm to solve find the optimal solution, the worst-case time complexity would be  $O(3^{(O(N_u N_f) + O(N_s N_u N_f))/3})$  [28].

Now we will consider the worst-case time complexity of the proposed heuristic algorithm in Section 5. The first stage of the heuristic algorithm is to scan the entire graph  $\mathcal{E}$ , while assigning a weight to each vertex, and then to conduct a maximum weighted vertex search. This process is then repeated until the maximum weighted maximal clique is found. There are  $N_u N_f$  vertices in graph  $\mathcal{E}$ , where finding the corresponding vertex with the maximum weight will require  $N_u N_f - 1$  operations in the first iteration. The next iteration would require  $N_u N_f - 2$  and so on until the final iteration would require  $N_u N_f - N_u$  operations. Hence, the total number of operations that is required will be  $\sum_{i=1}^{N_u} (N_f N_u - i) = (1/2)(N_f^2 N_u^2 + N_f N_u)$ . Therefore, the worse-case time complexity to find the cooperative communications amongst users is  $O(N_f^2 N_u^2)$ . Once the initial graph has been served (that is, the cooperative transmissions completed), there are still potentially  $N_{uc}$  unserved clients that could not be served in a D2D communication in the current time epoch and need to be served by the servers. Again, we proceed with the same algorithm as before, conducting a maximal weighted vertex search on graph  $\Psi$ ; however the order of the graph is now  $O(N_s N_{uc} N_f)$ . Since a maximal clique (or maximal independent set) cannot contain more than  $N_u$  vertices, the worst-case time complexity requires  $N_u$  of the above iterations. Therefore, the computational complexity for the graph

TABLE 2: Magnitude of worst-case complexity for each graph in the proposed algorithm.

Scenario	$S_1$	$S_2$	$S_3$
Graph $\mathcal{G}$	40000	$10^6$	$1.44 \times 10^6$
Graph $\Psi$	1280	2880	24000

TABLE 3: Parameters used in each scenario generated in Table 2, where each case has one server.

Scenario	$S_1$	$S_2$	$S_3$
$N_f$	20	20	60
$N_u$	10	50	20
$N_{uc}$	8	12	10

$\Psi$  (i.e., graph representing servers' transmissions to users that are not targeted by the D2D cooperative transmissions) will be  $O(N_u[N_f N_{uc} N_s + \log(N_f N_{uc} N_s)]) = O(N_s N_{uc}^2 N_f)$ . Thus, the entire algorithm's worst-case time complexity to serve all users will be  $O(N_f^2 N_u^2) + O(N_s N_{uc}^2 N_f)$ . Given that in reality the number of servers  $N_s$  in the network is much smaller than the number of users  $N_u$  and also the number of unserved users  $N_{uc}$  is much smaller than  $N_u$ , the worst-case complexity of the proposed algorithm can be estimated to be of order  $O(N_f^2 N_u^2)$ . Table 2 shows the numerical values for the magnitude of worst-case complexities of graphs  $\mathcal{G}$  and  $\Psi$  in the proposed algorithm under the scenarios summarized in Table 3.

From the results shown in Table 2, it can be seen that the worst-case time complexity of solving the proposed algorithm will reduce to just that of graph  $\mathcal{G}$ , that is, the cooperative transmissions. Therefore, the worst-case time complexity of the proposed algorithm will be  $O(N_f^2 N_u^2)$ , far more efficient than the B-K algorithm. Moreover, it can be seen that the worst-case time complexity of the proposed algorithm will also be independent of the number of servers in the network, assuming that  $N_u \gg N_s$ .

## 8. Conclusion and Future Work

In this paper, the problem of offloading the expensive backhaul of data network servers through a network coded cooperative D2D network model is investigated. The problem is formulated using the IDNC induced subgraph model, where the optimal solution requires finding maximal cliques of multiple graphs. The paper considers the generalized system model where users can demand multiple files and the transmission channels are subject to erasure. Having formulated the optimal solution using the SSP technique, it is found that the optimal solution is intractable and not solvable in real-time. Therefore, a greedy heuristic algorithm is employed using a maximum weighted vertex search approach on the IDNC graph with induced subgraphs. This heuristic when compared with the optimal solution for a small network showed to be a good approximation to the optimal solution. The simulation results show a significant improvement over the conventional method that incorporates IDNC

in a distributed fashion without D2D enabled cooperation.

The first recommendation for further work would be to consider a system model with imperfect feedback from the users. In this scenario, we would remove the assumption that the servers receive perfect feedback on the reception status of the transmitted files. In reality, there is the likelihood that the feedback itself may be corrupted; therefore a more efficient online algorithm could consider these uncertainties. Lastly, it would also be beneficial to consider the asymptotic bounds to the problem.

## Disclosure

This research paper is an extension of the work in the conference paper "Offloading of Fog Data Networks with Network Coded Cooperative D2D communications" [22] that was awarded the "Best Paper Award" in the 9th EAI International Conference on Mobile Networks and Management (EAI MONAMI 2017).

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update," Tech. Rep., 2016.
- [2] M. Ji, G. Caire, and A. F. Molisch, "Wireless device-to-device caching networks: basic principles and system performance," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 1, pp. 176–189, 2016.
- [3] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A Survey on network codes for distributed storage," *Proceedings of the IEEE*, vol. 99, no. 3, pp. 476–489, 2011.
- [4] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.
- [5] D. S. Papailiopoulos, J. Luo, A. G. Dimakis, C. Huang, and J. Li, "Simple regenerating codes: Network coding for cloud storage," in *Proceedings of the IEEE Conference on Computer Communications, INFOCOM 2012*, pp. 2801–2805, USA, March 2012.
- [6] A. A. Al-Habob, S. Sorour, N. Aboutorab, and P. Sadeghi, "Conflict free network coding for distributed storage networks," in *Proceedings of the IEEE International Conference on Communications, ICC 2015*, pp. 5517–5522, UK, June 2015.
- [7] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, 2014.
- [8] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the 1st ACM Mobile Cloud Computing Workshop, MCC 2012*, pp. 13–15, Finland, August 2012.
- [9] R. Tandon and O. Simeone, "Cloud-aided wireless networks with edge caching: Fundamental latency trade-offs in fog Radio Access Networks," in *Proceedings of the 2016 IEEE International*

- Symposium on Information Theory, ISIT 2016*, pp. 2029–2033, esp, July 2016.
- [10] N. Golrezaei, A. Molisch, A. G. Dimakis, and G. Caire, “Femto-caching and device-to-device collaboration: A new architecture for wireless video distribution,” *IEEE Communications Magazine*, vol. 51, no. 4, pp. 142–149, 2013.
- [11] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, “FemtoCaching: wireless content delivery through distributed caching helpers,” *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, 2013.
- [12] M. A. Maddah-Ali and U. Niesen, “Decentralized Coded Caching Attains Order-Optimal Memory-Rate Tradeoff,” *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1029–1040, 2015.
- [13] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, “Network information flow,” *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.
- [14] S. Sorour and S. Valaee, “On minimizing broadcast completion delay for instantly decodable network coding,” in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, pp. 1–5, Cape Town, South Africa, May 2010.
- [15] S. Sorour and S. Valaee, “Minimum broadcast decoding delay for generalized instantly decodable network coding,” in *Proceedings of the 53rd IEEE Global Communications Conference, GLOBECOM 2010*, USA, December 2010.
- [16] E. Drinea, C. Fragouli, and L. Keller, “Delay with network coding and feedback,” in *Proceedings of the 2009 IEEE International Symposium on Information Theory, ISIT 2009*, pp. 844–848, Republic of Korea, July 2009.
- [17] P. Sadeghi, D. Traskov, and R. Koetter, “Adaptive network coding for broadcast channels,” in *Proceedings of the 2009 Workshop on Network Coding, Theory and Applications, NetCod '09*, pp. 80–85, Switzerland, June 2009.
- [18] L. Keller, E. Drinea, and C. Fragouli, “Online broadcasting with network coding,” in *Proceedings of the 2008 4th Workshop on Network Coding, Theory, and Applications, NetCod 2008*, Hong Kong, January 2008.
- [19] 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, 2011.
- [20] P. Baran, “On distributed communications networks,” *IEEE Transactions on Communications*, vol. 12, no. 1, pp. 1–9, 1964.
- [21] Y. Shnaiwer, S. Sorour, P. Sadeghi, N. Aboutorab, and T. Y. Al-Naffouri, “Network-Coded Macrocell Offloading in Femtocaching-Assisted Cellular Networks,” *IEEE Transactions on Vehicular Technology*, 2017.
- [22] B. Quinton and N. Aboutorab, “Offloading of fog data networks with network coded cooperative d2d communications,” in *Proceedings of the In 9th EAI International Conference on Mobile Networks and Management (EAI Monami 2017)*, pp. 1–13, EAI, 2017.
- [23] Y. N. Shnaiwer, S. Sorour, N. Aboutorab, P. Sadeghi, and T. Y. Al-Naffouri, “Network-coded content delivery in femtocaching-assisted cellular networks,” in *Proceedings of the 58th IEEE Global Communications Conference, GLOBECOM 2015*, usa, December 2015.
- [24] C. S. Edwards and C. H. Elphick, “Lower bounds for the clique and the chromatic numbers of a graph,” *Discrete Applied Mathematics*, vol. 5, no. 1, pp. 51–64, 1983.
- [25] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, Macmillan Press, New York, NY, USA, 1976.
- [26] M. L. Puterman, “Markov decision processes,” in *Stochastic Models*, vol. 2 of *Handbooks Oper. Res. Management Sci.*, pp. 331–434, North-Holland, Amsterdam, 1990.
- [27] C. Bron and J. Kerbosch, “Algorithm 457: finding all cliques of an undirected graph,” *Communications of the ACM*, vol. 16, no. 9, pp. 575–577, 1973.
- [28] E. Tomita, A. Tanaka, and H. Takahashi, “The worst-case time complexity for generating all maximal cliques and computational experiments,” *Theoretical Computer Science*, vol. 363, no. 1, pp. 28–42, 2006.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

