Cross-Layer Rate Aware Network Coding for Wireless Communications Systems

Final Project Summary Report 2018

Fiona Vi Nguyen[1]

The University of New South Wales at the Australian Defence Force Academy

Network Coding (NC) is a relatively new research area that has been demonstrated to improve network performance by utilising the concept of packet combinations to increase data throughput. The aim of this project is to reduce the delivery time (in seconds) of messages to wireless users using instantly decodable network coding (IDNC) with physical layer rate awareness. Previous work involving IDNC systems focused on optimising parameters from an upper layer network perspective and reduced physical layer influences to simple erasure channel probabilities. This simplification is an unrealistic representation of empirical channel conditions. These conditions result in different transmission rates between a base station (BS) and the users in a network. Adapting a cross layer NC scheme, which considers the heterogeneity of transmission rates, is a more realistic approach to the problem. This is currently a very limited area of research. The project was divided into three phases, each phase comprising of the following stages: theoretical scoping, algorithm development, simulation, and analysis. Phase 1 was the development of a point-to-multipoint rate adapted IDNC system utilising a *single-layer* maximum weighted clique selection policy. Phase 2 was the implementation of a rate adapted *multi-layer* scheme based on the concept of prioritising *decisive users* who will most likely extend the overall completion time. Phase 3 was the application of the best performing algorithm to a multipoint-to-multipoint situation involving two coordinated base stations. Phase 1 results showed the proposed algorithm's mean completion time improved by approximately 71.76% (for several users and packets) when compared to an equivalent simple selection scheme which did not combine packets for transmission. Phase 2 results found that the *multi-layer* rate aware scheme performed 10.22% worse than the Phase 1's *single-layer* scheme. Phase 3 was the application of the *single-layer* scheme to a Multi-Base Station (MBS) scenario which saw an improvement 50.34% when compared to a scheme which did not combine packets. The development of an effective rate adapted *single-layer* heuristic solution (applied to a MBS scenario as well) is the major contribution to the research area.

Contents

[1]SBLT, School of Engineering & Information Technology. ZEIT4501

I. Introduction

This section is divided into two parts. Part A will provide background information about NC and the field of study. Part B will detail the project objectives and contributions to the research area.

A. Literature Review and Motivation

The Network Coding (NC) paradigm was first introduced at the turn of the century in the seminal paper by Alshwede *et al.* [1] which forever changed the approach to communication systems and handling of data streams. Conventional networks operate on the fundamental principle that data streams are independent, although sharing the same network resources [2]. Essentially, if two packets (or two data streams) are fed into a node in a network, the output of that node can only be one packet at a time. Thus, two transmissions would be required to output those two input packets. As an analogy, information transfer in a network is akin to how cars (packets or data streams) are individual entities that are travelling on the same highway (shared network resources) where the roads become a single lane approaching a traffic light (node) – only one car at a time can drive through the traffic light. This basic principle is the foundation of today's general network functions, routing, and data storage [3]. NC opposes this principle as it is premised on the fact that input packets can be outputted as a combination. For example, linear network coding (LNC) linearly combines data represented as vectors or numbers over a finite field [1]. Another way to combine packets (in the form of bits) is by exploiting exclusive-OR (XOR) logic [4]. XOR computations consume minimal computational power yet allow NC to dramatically increase the throughput of a network [3]. This advantage is another reason why NC has garnered the attention of researchers. This project will be using XOR logic as the packet combination method.

NC treats nodes as not just mere switches or relays, but as points in a network that are capable of providing an encoding opportunity so that multiple input data streams can be combined to a single output – thus increasing *throughput* and minimising *completion time*. Throughput is defined as the number of bits that pass through a point in a network every second [19]. Thus, the completion time is inversely proportional to throughput. To illustrate how NC increases throughput, Fig.1 shows a comparison between traditional network information flow and NC. The traditional method requires more transmissions (higher completion time) as packets (a & b) can only be transmitted one at a time. NC permits XOR combinations and thus allows the base station (BS) to complete the transmission in one less step compared to the traditional method.

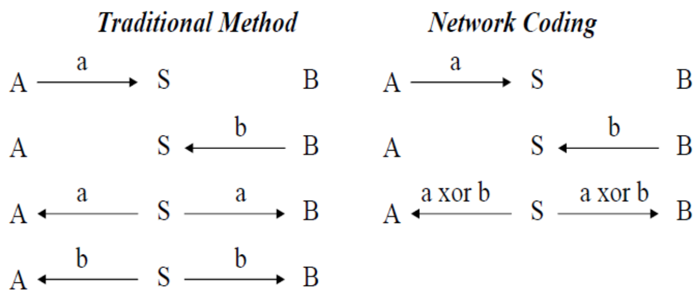

Figure 1. Simple NC example. 'A' and 'B' can be considered two wireless users communicating to a base station 'S' which then exchanges the information between the two users. [2].

Fig. 1 is only a very basic example of a network coded transmissions. NC has been adapted to significantly more complicated networks involving potentially multiple receivers (users), multiple base stations, and a significantly larger number of packets. As one can imagine, the choice of packet combinations for each transmission is vast. Since the objective of communication networks is to transmit the maximum amount of information in the shortest time possible, packet combinations (the 'encoding' of packets) is a necessary factor to consider as it greatly influences the completion time. How these decisions are made depends on the NC method which is implemented. This involves deciding: how the missing packets will be represented; and the selection policy (algorithm) used to determine which packets to combine. Missing packet representation can be achieved using structures such as directed/undirected graphs, line graphs, subtree decompositions, steiner trees [3].

In the recent years, there has been extensive research into the optimisation of online NC packet selection algorithms focused on improving throughput and minimising decoding delays of broadcast erasure channel networks [5]-[9]. These studies consider the problem for both the point-to-multipoint and multipoint-to-multipoint cases. In Ref. [5]-[7], the studies propose algorithms that prioritise packet selection for each NC transmission and consider the effect of channel conditions – aspects which were not considered in Ref. [8]-[9]. In studies [5]-[7], the minimum decoding and completion time problems were studied for a subclass of NC called *instantly decodable network coding* (IDNC). This subclass of NC ensures that all coded packets include at most one missing source packet for each receiver [6]. This requirement allows for instant packet recovery when applicable to the receiver, which is a property that linear and random NC lack [5]. IDNC packet encoding is implemented using binary XOR which allows for the instant packet recovery to be possible and thus guarantees fast, low complexity decoding. When receivers receive a relevant packet, the missing packet can be deduced by cancelling out the already received packets. This also means that IDNC is suited for order-insensitive applications where the order of transmitted packets is irrelevant. XOR encoding removes the need for matrix inversions in the decoding stage which is a computational bottleneck in linear and random NC [9]. The encoding and decoding complexity and large decoding delay of random linear network coding makes it unsuitable for applications prioritizing completion times [13]. The advantages of IDNC allow the scheme to be more suitable for applications such as real-time multimedia streaming, roadside to vehicle safety message broadcasts, and sending coordinated commands to sensors [6].

Ref. [5]-[7] consider channel conditions from a simplistic view and only model physical layer conditions as packet erasure probabilities which remain constant for all transmissions. Coded combinations from the base station (BS) are also assumed to be transmitted with the same physical layer rate and thus take fixed durations of time to complete [10]. Reducing the problem to such a simple representation allows transmissions and packet combinations to be solved essentially, solely in the network layer thus removing the problem of simulating complicated empirical scenarios. This modeling of physical layer conditions is unrealistic as it is known that users in a network can receive messages from the base station at different rates. The heterogeneity of transmission rates ultimately affects which packets should be combined; the transmission rate chosen to send the combined packet; and the time duration needed to deliver the message [11]. To improve upon the simple erasure channel modeling of physical layer conditions, Ref. [10]-[14], incorporate rate adaptation as a major factor that affects the entire packet selection process. These studies introduce a *cross layer approach* to the problem and propose a *multi-layer packet selection design* based on *decisive users*. Rate adapted NC schemes is a very limited area of research.

B. Project Aim and Contributions

The aim of this project was to develop a *cross-layer* packet selection algorithm that reduced the physical delivery time (in seconds) of messages to wireless users using instantly decodable network coding (IDNC), incorporating physical layer rate awareness. The focus was to incorporate physical layer factors (i.e. transmission rate capabilities of receivers) into the packet selection process so that packets can be transmitted at a faster rate thus reducing the overall completion time. Results of the project found that the developed *single-layer* approach of Phase 1 was more effective than the heuristic version of the *multi-layer decisive user* scheme proposed in the latest rate aware papers [10]-[11]. In addition to outperforming the *multi-layer* scheme, the *single-layer* approach was a significantly simpler algorithm. This allows the scheme to be more easily applied to complex scenarios such as the multipoint-to-multipoint case. The development of this *single-layer* packet selection algorithm and applying it to a Multi-Base Station (MBS) case was the major contribution of this project.

The rest of this report is organised as follows. Section II will provide in-depth details about the system model and the developed algorithms for each phase. Section III will cover the results of the project and provide a detailed analysis. Finally, Section IV will conclude the report and Section V will provide recommendations for future work.

II. Development of Algorithms

This section will be divided into five parts. Part A will provide a general overview of the system model. Part B will explain the subclass of NC called IDNC and discuss the formulation of an IDNC graph which is used to determine which packet combinations to send for each transmission. Part C will discuss *Algorithm 1*: the *single-layer* selection policy which uses a maximum weighted clique search that operates on the IDNC graph. Part D will discuss *Algorithm 2*: the heuristic version of the *multi-layer decisive user* scheme proposed in the latest rate aware papers [10]-[11]. Finally, Part E will discuss *Algorithm 3*: the coordinated Multi-Base Station scheme based on Alg.1 involving two BS. Fig. 2 above summarises the development of the algorithms in a basic timeline. Alg. 01 and Alg. 02 are simply the

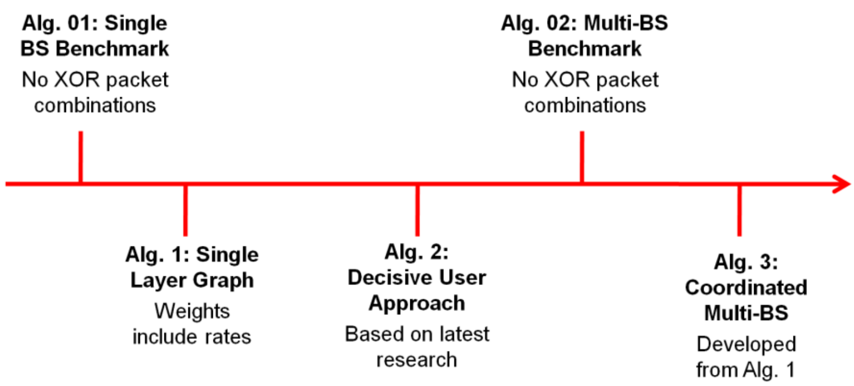

Figure 2. Timeline of Algorithm Development.

benchmark cases which represent the 'traditional' information flow where packets are not combined using NC (packets are singularly transmitted).

A. Basic System Model

Before discussing the algorithms developed in this project, it is important to clarify the basic aspects of the system model which were derived from papers [5]-[7], [10]-[11]. This will provide some context when elaborating on the NC schemes. The basic system model revolves around a wireless sender (base-station) that is required to send a message consisting of *N* source packets (the message) to a set of *M* users (receivers). It is assumed that receiver channels will still retain erasure probabilities that are unchanged for all transmissions. However, each receiver is designated a maximum *achievable capacity* rate which indicates the maximum transmission rate for a successful transmission to occur, notwithstanding erasure probabilities. A user can only receive a transmission if the selected rate is less than or equal to its achievable capacity rate ($R(t) \leq R_u(t)$). It is assumed that users will have the same maximum *achievable capacity* rate for all transmissions and that the BS can adjust its modulation scheme to transmit at any selected rate $R(t)$.

Ref. [20]-[22] provides information on how to model the physical layer attributes of the system (i.e. determining the rates of users). It is assumed that users have the same maximum achievable capacity rate for all transmissions from the same base station. Each user in the network will have an *achievable capacity* rate at a given by the formula in Ref. [20], [22]:

$$R_u = B \times log_2\left(1 + \frac{Pt|h_u|^2}{\sigma^2}\right) \tag{1}$$

where B is the channel bandwidth (assumed to be fixed); Pt is the *transmit power* of the BS (fixed); h_u is the complex channel gain from the BS for a user; and σ^2 is the *Gaussian noise variance*. σ^2 is equivalent to $2 \times B \times N0$ where $N0$ is the noise power spectral density which is set to -174dBm/Hz Ref. [20]-[22]. Eq. (1) was derived (Ref. [22]) from the *Shannon Capacity* equation (Eq. (2) below) which determines the theoretical highest data rate for a noise channel [19]. The fraction $\frac{Pt|h_u|^2}{\sigma^2}$ in Eq. (1) is simply the signal-to-noise ratio (SNR). Both Eq. (1) and Eq. (2) indicate that the higher the SNR, the higher the capacity of the channel (in terms of bits/second – the rate). Ref [20] uses Rayleigh random distribution to find the complex channel gains.

$$Capacity = bandwidth \times log_2(1 + SNR) \tag{2}$$

Communication commences when the *N* packets are broadcast to the users in an *initial transmission phase*. Due to erasure channel probabilities, not all packets will be successfully received. Each user then provides feedback to the sender – ACK if a packet is received, NAK for lost packets. This information results in each user *i* being characterised by two sets of packet information:

- *Has* set (H_i): The set of successfully received packets.
- *Wants* set (W_i): The set of lost packets.

The sender (BS) stores this information in a *state feedback matrix* (SFM) where the rows represent the users *M* and the columns represent the source packets *N*. Each cell of the matrix will use bits to indicate whether a user has received or lost a packet. Bit '0' will indicate successful transmission and bit '1' will indicate a lost packet. Mathematically, the SFM is **F** = $[f_{ij}], \forall\ i \in M,\ j\ \in N$ such that:

$$f_{ij} = \begin{cases} 0 & j \in H_i \\ 1 & j \in W_i \end{cases} \tag{3}$$

After the *initial transmission phase*, the *recovery transmission phase* commences which consists of all subsequent transmissions concluding only when all source packets *N* have been successfully received by all users *M* – i.e. when the SFM shows a matrix of all zeroes. For each transmission in this phase, the chosen NC scheme will decide on the combination of missing source packets to transmit. After each transmission, the SFM is updated to reflect the current state of received/lost packets. The *completion time* is generally represented by the number of transmissions required to complete all transmissions and achieve a SFM of all zeroes. In the case of a *cross layer approach rate aware* scheme, *completion time* is better defined as the total physical time (seconds) it takes to complete the *recovery transmission phase* (in other words, achieve a SFM of all zeroes).

MATLAB was used at the simulation platform for the project.

B. Instantly Decodable Network Coding (IDNC)

IDNC is a class of NC which has been extensively researched and applied to a variety of network situations in the recent years due to its low complexity encoding/decoding and fast decoding potential which is ideal for real-time applications [5]-[7], [10]-[16]. Fast decoding is achieved because IDNC packet combinations are encoded using XOR binary logic which draws little computational power [3]. Additionally, when users receive the packet combination, it must be instantly decoded and not stored for future processing. If the packet has no relevance to the user, it is simply discarded. This implies that receivers can be simplified as no buffers are needed to store coded packets for future processing [6], allowing design costs of receivers to be economical. In order for XOR logic to be utilised, IDNC ensures that all packet combinations cannot have more than one wanted packet for each user [5]. The instantaneous packet recovery of IDNC is one of its most desirable properties which makes it a more favourable NC scheme when compared to its counterparts such as random linear network coding which requires complex decoding.

Random network coding (RNC) [17] and opportunistic network coding (ONC) [18] are two distinct areas of interest in NC literature. IDNC is considered a subset of ONC because the scheme operates by taking advantage of the relationships between wanted packets and its users to transmit combinations which maximise throughput. RNC combines packets using coefficients and ONC exploits receiver's side information for packet selection [7], [17], [18]. Although RNC has gained attention in literature, encoding and decoding complexity and large decoding delay tolerance makes it unsuitable for applications prioritizing completion times [13]. Thus, IDNC was utilised for the project.

Referencing Section II.A, all recovery phase transmission packets that are successfully received for each receiver *i* are classified as one of the following possibilities [6]:

- *Non-innovative*: The received packet contains no source packets in the *Wants* set of the user.
- *Instantly Decodable*: The received packet contains only one source packet in the *Wants* set of the user.
- *Non-Instantly Decodable*: The received packet contains two or more source packets in the *Wants* set of the user. Thus, user cannot extract wanted packets using XOR decoding. Packet is discarded.

An IDNC encoded packet *addresses* a receiver if that packet contains a source packet from the *Wants* set of the user.

In order to implement a packet selection policy, an *IDNC graph* is developed using the bit '1's in the SFM (i.e. the wanted packets) as vertices (v_{ij}) of the undirected graph. The IDNC graph ($\mathcal{G}$) is a tool used to define all possible instantly decodable packet combinations [5]. Edges exist between two vertices if *strictly one* of the following conditions is satisfied:

- *C1*: Two vertices were induced by the loss of the same packet but just different users.
- *C2*: The requested packet for each user is in the *Has* set of the other.

Visually speaking, the locations of the bit '1s' (the vertices) relative to each other in the SFM for each condition are as follows (recalling that the rows are the users and the columns are the packets):

$$C1{:}\begin{bmatrix}1 & 0\\ 1 & 0\end{bmatrix} \text{ or } \begin{bmatrix}0 & 1\\ 0 & 1\end{bmatrix} \qquad C2{:}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \text{ or } \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$$

Thus, the edges of the IDNC graph indicate coding opportunities.

C. Algorithm 1: Single Layer Graph Approach

Algorithm 1 is based on a simple online selection algorithm called *Maximum Weighted Clique Search* (Ref. [5]) which is shown to outperform random and greedy selection algorithms which have similar computational complexity. This selection policy is implemented upon the IDNC graph ($\mathcal{G}$) to determine the best packet combination to transmit which will reduce the overall completion time. The "best" packet combination is indicated by the *maximum weighted clique*. A *clique* is defined a subset of vertices in an undirected graph such that each vertex is connected to every other vertex in the subset. The *maximum clique* is the clique with the largest number of vertices. Thus, transmitting the packets of a maximum clique will address the highest number of users. *Greedy selection* algorithms select any maximum clique to transmit. However, Ref. [5] explains that a more strategic clique selection would be a far better approach to minimising completion time, which is the primary objective of the project. Thus, the concept of weighted vertices is introduced. One can infer that the maximum weighted clique is the maximum clique with the heaviest weighted vertices. Study [5] explains that a vertex's weight reflects the reception success probability of its receiver as well as the reception success probability of its neighboring vertices in the IDNC graph.

A slightly modified scheme was developed from Ref. [5] which incorporates a user's maximum achievable capacity rate (R_u) into the vertex weights. This ensures that the *maximum weighted clique* consists of vertices of users who also have the highest rates. This idea was derived from Ref. [20] which incorporated rates into the weights of vertices but for a conflict IDNC graph as opposed to the IDNC coding opportunity graph.

To further elaborate on how the *Maximum Weighted Clique Search* was conducted, it is important to emphasise that a **SFM** matrix was used to represent the lost and received packets for all users. An **adjacency** matrix was used to represent the existence of edges between vertices (v_{ij}) stored in the **vertex** matrix. This was how the IDNC graph was represented in programming code. An adjacency function was created to generate an *adjacency indicator* ($a_{ij,kl}(s)$) where a bit '1' signified if the IDNC graph edge conditions were satisfied and a bit '0' signified if vertices were not instantly decodable.

Before discussing about vertex weights, it is necessary to clarify the concept of *priority measure* ($\tau_i(s)$). The 's' variable is simply referring to the state of the SFM. During calculations, 's' can be considered (for ease of understanding) as the current transmission. Further details about the state space representation can be found in [5]. The higher the priority measure for a receiver *i* at state *s*, the more likely that vertices of this user should be addressed first. This will prevent further delays in the overall completion time. Priority measure can be calculated using the following equation:

$$\tau_i(s) = R_u \times \frac{|W_i(s)|}{1-p_i} = user\ rate\ \times \frac{total\ number\ of\ wanted\ packets\ for\ the\ receiver}{1-probability\ of\ erasure\ for\ the\ receiver's\ channel} \quad (4)$$

Eq. (4) differs to the equation provided in Ref. [5] as the user rate is included in the equation so that the vertices of users with a higher achievable capacity rate are coded together. A receiver will have a large $\tau_i(s)$ if the receiver is missing a larger number of packets, has a more lossy channel (higher probability of erasure), and has a higher achievable capacity rate.

$\Delta_{ij}(s)$ is the weighted degree of a vertex. A large weighted vertex degree indicates that a vertex is connected to a large number of vertices belonging to receivers with large $\tau_i(s)$ values [5].

$$\Delta_{ij}(s) = \sum_{\forall v_{kl} \in \mathcal{G}(s)} a_{ij,kl}(s)\tau_k(s) \quad (5)$$

Thus, vertex weights ($w_{ij}(s)$) are calculated using the formula:

$$w_{ij}(s) = \tau_i(s)\Delta_{ij}(s) = \tau_i(s) \times sum\ of\ adjacent\ vertices'\tau_k(s) \quad (6)$$

MATLAB functions were created to perform each of above calculations. The *maximum weighted clique search* can be summarised in the following steps:

1. Scan SFM for the location of all bit '1's (all wanted packets). Store locations in a **vertex** matrix.
2. Create empty **clique** matrix which will contain vertices of the maximum weighted clique.
3. Determine $\tau_i(s)$ for all users. Store values in a matrix.
4. Create **adjacency** matrix.
5. Determine adjacent vertices and fill **adjacency** matrix.
6. Determine the weights of all vertices. Store values in a matrix.
7. Find the heaviest (or next heaviest) weighted vertex in the matrix.
8. Add vertex to **clique** matrix.
9. Update the IDNC graph by removing any vertices not connected to all vertices currently stored in the **clique** matrix.
10. Repeat steps 5-9 until all vertices have been tested and the IDNC graph is reduced to the maximum weighted clique.
11. Set transmission rate to lowest rate in the clique (rates were not considered in the selection scheme).

When the maximum weighted clique is found, the transmission rate is set to the vertex with the lowest rate. The packets of the maximum weighted clique are transmitted at the set rate and the SFM is updated. The clique search is repeated and the *recovery transmission phase* is complete when the SFM returns all zeroes. The *completion time* in seconds for each transmission is calculated by dividing the packet size with the selected rate.

D. Algorithm 2: Multi-Layer Decisive User Approach

Algorithm 2 is the heuristic version of the algorithm proposed in the latest rate aware papers [10]-[11]. These studies propose an *optimal* solution to the problem of minimising completion time using IDNC with physical layer rate awareness. The scheme is premised on the concept of *k-th decisive users*. Decisive users ($\mathcal{K}_{R(t)}$) is the set of users who can increase the total *maximum anticipated completion time*. The basic idea is that decisive users (in descending order of criticality) should be prioritised and served first. *Criticality* is determined based on the minimum number of consecutive non-instantly decodable packets required for that user to extend the overall completion time. A user becomes part of this set if they satisfy this basic equation:

$$\mathcal{K}_{R(t)} = \{u \in \mathcal{U} \,|C_u(t) \geq C_u{}^*(t-1)\} \quad (4)$$

Equation (4) uses the current *anticipated individual completion time* ($C_u(t)$) of a user and determines if it is greater than highest $C_u(t)$ value in the previous transmission ($C_u{}^*(t-1)$), therefore determining that the user is capable of increasing the overall *maximum anticipated completion time*. $C_u(t)$ is essentially used as a predictor of the future.

$C_u(t)$ is defined as the completion time of the *u*-th user *after* the *t*-th transmission if that user were to receive only successful instantly decodable receptions for all future transmissions (hence why it is called the *anticipated* completion time). $C_u(t)$ is approximated using the following equation from Ref. [10]:

$$C_u(t) = \left(\frac{N|\mathcal{F}|}{\tilde{R}_u(t)} + \mathcal{T}_u(t)\right) \quad (5)$$

where $N|\mathcal{F}|$ indicates the number of bits in the message (N = no. of bits per frame, $\mathcal{F}$ = no. of frames); $\tilde{R}_u(t)$ is the *harmonic mean* of all the rates of previously successful instantly decodable transmissions; $\mathcal{T}_u(t)$is the accumulated *time delay* (number of received transmission which were not instantly decodable). In simple terms, this approach to calculating *anticipated individual completion time* is based on looking at the history of transmissions. If a user has received many transmissions which were 'useless' to them (i.e. non-instantly decodable or not received) then the *anticipated individual completion time* would of course increase. Additionally, the higher the rates of the successful transmissions, the smaller the $\frac{N|\mathcal{F}|}{\tilde{R}_u(t)}$ factor becomes and therefore the lower the $C_u(t)$.

A user is part of the *k-th decisive user set* ($\mathcal{K}_R^k$) if they will cause an increase in the overall completion time if they were to experience $k \geq 1$ consecutive time delay increases (i.e. they experience a minimum of k sequential non-instantly decodable transmissions at a rate R after the *t*-th transmission). This is determined using the equation:

$$\mathcal{K}_R^k = \left\{u \notin \mathcal{K}_R^{k-1} | C_u(t-1) + \frac{kN}{R} \geq C_u{}^*(t-1)\right\} \quad (6)$$

Ref. [10]-[11] suggest that rate aware IDNC (RA-IDNC) sub-graphs are then created sequentially for every *k-th decisive user* set starting with $\mathcal{K}^1 = \cup_{R\in\mathcal{R}} \{\mathcal{K}_R^1\}$ (highest criticality) where R is the achievable capacity R_u of each user. Users in $\mathcal{K}^1$ are used to create the RA-IDNC graph $\mathcal{G}^1$. The maximum clique of $\mathcal{G}^1$is to be found first. Since finding the maximum clique is a NP-hard problem [6] (hence it is an *optimal* solution), it was decided that a standard IDNC graph should be implemented where the vertex weights incorporate the user rates (Alg.1). When the maximum weighted clique of the 1st decisive user set is found, the transmission rate R is set to the lowest rate of the users.

According to study [10], the maximum clique (**M**^{1}) of $\mathcal{G}^1$ is part of the global clique solution **M** (**M** = **M**^{1}) which is the maximum clique of the overall RA-IDNC graph $\mathcal{G}$. The rate R of the vertices in **M**^{1} becomes the selected rate for all subsequent $\mathcal{K}_R^k$. $\mathcal{K}_R^2$ users are then determined. $\mathcal{G}^2$(**M**) vertices are created from the users of $\mathcal{K}_R^2$and must be connected to all vertices in **M**. Paper [11] suggests that after two (future) transmissions, vertices in $\mathcal{G}^2$(**M**) have equal weights of $\log(R/N)$ because of the potential of becoming decisive users at that stage. Thus, in order to heuristically find the maximum clique, rates are not considered in the vertex weights so Eq. (4) for the calculation of priority measure is modified to be:

$$\tau_i(s) = \frac{|W_i(s)|}{1-p_i} = \frac{total\ number\ of\ wanted\ packets\ for\ the\ receiver}{1-probability\ of\ erasure\ for\ the\ receiver's\ channel} \quad (7)$$

The maximum clique of $\mathcal{G}^2$(**M**) is **M**^{2} which is merged with **M** (at this stage is **M**^{1}) to form the updated global clique **M = M**^{1}∪**M**^{2}. This process is repeated for all remaining *k-th decisive user* sets.

The following steps will provide a synopsis of how the algorithm was implemented.

- For each transmission during the <u>Initial Transmission Phase</u> (transmission rate is set to the lowest rate out of all the users)
  1. For all users, update the rate *harmonic mean* if transmission was successfully decodable to that user.
  2. For all users, update the *delay accumulation* if transmission was not received or not decodable.
  3. For the last transmission of this phase, calculate the *anticipated completion time* $C_u(t)$ for all users and find the maximum one (Eq. (5)). This will be $C_u{}^*(t-1)$ for the first transmission of the recovery phase.
- For each transmission during the <u>Recovery Transmission Phase</u>
  1. Calculate the *anticipated completion time* $C_u(t)$ for all users (Eq. (5)).
  2. Create a matrix called **already_kth** which tracks which users have already joined a k-th decisive set.
  3. Determine which users are part of the 1-st decisive user set (k = 1) (Eq. (6)) where transmission rate *R* is the achievable capacity of each user (R_u). Mark off **already_kth**.
  4. Create an IDNC graph $\mathcal{G}^1$ using these users.
  5. Find the maximum weighted clique (**M**^{1}) using Alg. 1 and Eq. (4) for the *priority measure* calculations.
  6. Set transmission rate R to the lowest achievable rate of the **M = M**^{1} clique.
  7. Find the users of the next k-th decisive user set using transmission rate R. Mark off **already_kth**.
  8. Create an IDNC graph $\mathcal{G}^k$ using these users.
  9. Find the maximum weighted clique (**M**^{k}) using Alg. 1 and Eq. (7) for the *priority measure* calculations. Vertices of this clique must be connected to all vertices in the global clique (**M**).
  10. Add **M**^{k} to **M** global clique.
  11. Repeat steps 7-10 until all users have been ticked off in **already_kth**.
  12. Transmit **M** global clique at rate R. Update *harmonic mean* and *delay accumulation* for each user.

E. Algorithm 3: Coordinated Multi-Base Station Rate Aware Network Coding

Using Ref. [12] as a basis, Alg. 1 was implemented on a multipoint-to-multipoint with a modified IDNC graph (in this project, two base stations (BS) are used) which will be called the BS-IDNC graph. Two BS's entails that each user will have a different *achievable capacity rate* R_u for each BS. This is where the advantage of a coordinated multipoint-to-multipoint system lies.

Say there are three users (U1, U2 and U3) and two base stations (BS1 and BS2) and every user is missing packet P1 (size = 1 bit). Users U1 and U2 have a maximum achievable rate of 2 bits/sec for BS1 but U3 only has an achievable rate of 1 bits/sec. In a singular BS scenario, BS1 would transmit at rate 1 bits/sec to reach U3 (completion time = packet size / rate = 1 sec). However, in the multi-BS case, U3 can receive transmissions from BS2 at a rate of 2 bits/sec. Thus, BS1 can transmit P1 to U1 and U2 and BS2 can transmit to U3 – all at a rate of 2 bits/sec (therefore completion time is only 0.5 seconds).

In order to incorporate this into the IDNC graph, two vertices ($v_{\boldsymbol{b},ij}$) will be created for every bit '1' in the SFM – representing BS 1 ($v_{\boldsymbol{1},ij}$) and BS 2 ($v_{\boldsymbol{2},ij}$). An important attribute of a multipoint-to-multipoint system is that a user cannot be served by more than one BS during a transmission. This means that for an edge to exist between two vertices in the BS-IDNC graph, before considering the IDNC conditions *C1* and *C2* of Section II.B, it must be determined whether the two vertices are for the same base station. If so, then the IDNC conditions *C1* and *C2* apply. If not, then an edge exists if the users of the vertices are different. These conditions will be called *C01* and *C02* respectively. The bullet points below will summarise all the conditions for an edge to exist between vertices in the BS-IDNC graph.

- *C01*: Base stations are the same. Apply *C1* and *C2* of Section II.B.
    - *C1*: Two vertices were induced by the loss of the same packet but just different users.
    - *C2*: The requested packet for each user is in the *Has* set of the other.
- *C02:* Base stations are different.
    - *C3:* Users must be different.

With the BS-IDNC established, the Maximum Weighted Clique Search Selection Policy of Algorithm 1 can be applied in the same manner as discussed in Section II.B with the rates of each vertex (unique to each BS) incorporated in the *priority measures* (unique to each BS) and therefore in the vertex weights.

III. Simulation Results & Analysis

This section will explain the results of simulating all the algorithms discussed in Section II and comparing the performances of all the algorithms.

A. Simulation Parameters

Table 1 lists the system parameters used to simulate all algorithms. The constant values were obtained from Ref. [5], [20] and [21]. Packet size was set to 10 Mbits based on the generated rates (using Eq. (1)) which were fluctuating around 10 Mbits/second.

MATLAB was used as the programming platform for the simulations.

Channel Bandwidth	*B*	200 KHz
Base Station Transmit Power	*Pt*	20 W
Complex Channel Gain	h_u	Rayleigh Random Distribution
Noise Spectral Density	*N0*	-174 dBm/Hz
Probability of Channel Erasure	*p*	0.05-0.3 (random)

Table 1. Physical Layer parameters used to generate rates.

B. Simulation of Single Base Station Case

Figs. 3-4 below show the results of simulating Alg. 1 and Alg. 2 against the benchmark Alg. 01 where packets were singularly transmitted. Each case was simulated extensively to achieve the average mean completion times.

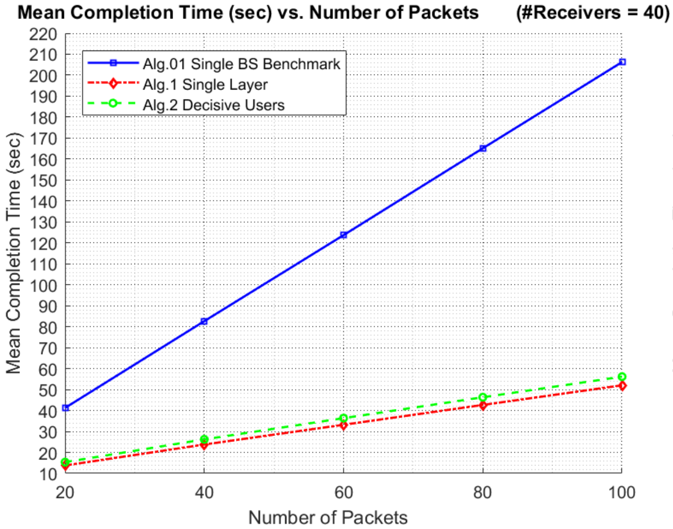

Figure 3. Graph comparing MATLAB Simulation Results Alg. 01, Alg. 1 and Alg. 2 across a number of packets with number of receivers set to 40.

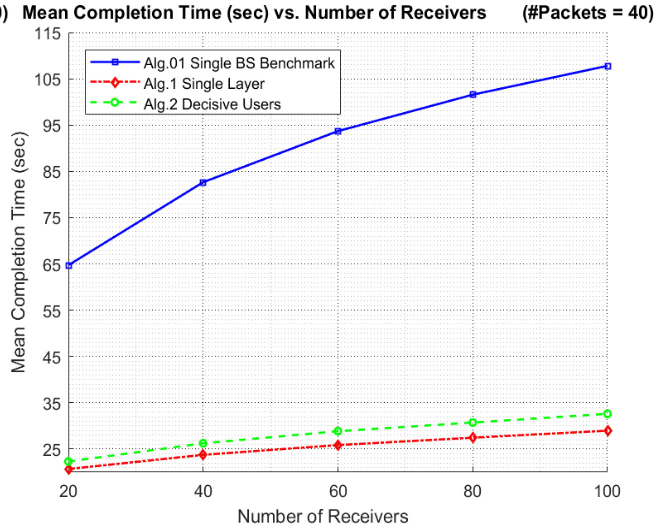

Figure 4. Graph comparing MATLAB Simulation Results Alg. 01, Alg. 1 and Alg. 2 across a number of receivers with number of packets set to 40.

Alg. 01 conducted a "weighted clique search" in the same manner as Alg. 1 but there were no edges in the graph of vertices for the instant decodability condition (*C2* of Section II.B). The "weighted clique search" simply meant that the packets, which were demanded by the highest number of users with the worst channel conditions and highest achievable capacity rates, would be transmitted first. Packets were not combined for transmission and only singular packets were transmitted (essentially representing traditional communications). This provided a

good comparison to show the effectiveness of NC in minimising completion time. Fig. 3 shows the comparison of mean completion times across a number of packets and Fig. 4 shows the comparison across a number of receivers. Both figures clearly show that Alg. 1 and Alg. 2 outperform the conventional information flow case (Alg. 01), thus showing the effectiveness of NC. However, when comparing between Alg. 1 and Alg. 2, it can be seen in both graphs that Alg. 1 outperforms Alg. 2. Alg. 1 was developed to be a significantly simpler *single-layer* algorithm compared to the *multi-layer decisive user* scheme proposed in the latest papers [10]-[11], though was proven to perform better in overall completion time (sec). With an increasing number of packets, the Alg. 1 showed an average improvement factor of 71.97% compared to Alg. 01 and 9.57% when compared to Alg. 2.
For an increasing number of receivers, Alg. 1 showed an average improvement factor of 71.55% compared to Alg. 01 and 10.87% when compared to Alg. 2. Thus, Alg. 1 was chosen to be implemented on the more complex case involving coordinated base stations.

C. Simulation of Coordinated Multi-Base Station Case

Figs. 5-6 below show the results of simulating Alg. 3 against the benchmark Alg. 02 where packets were singularly transmitted.

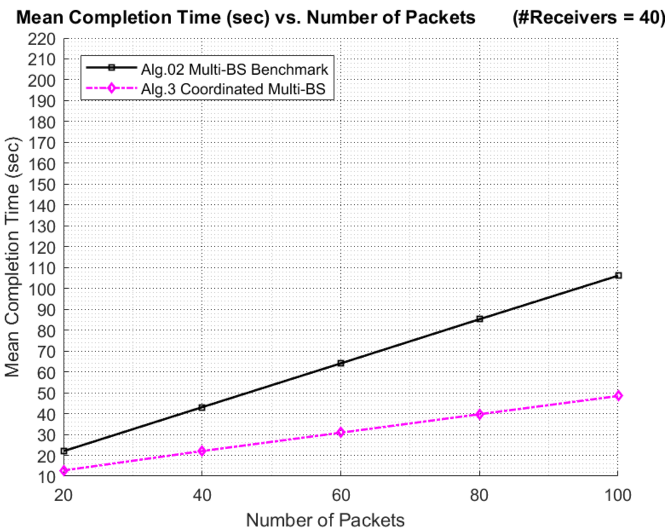

Figure 5. Graph comparing MATLAB Simulation Results Alg. 02 and Alg. 3 across a number of packets with number of receivers set to 40.

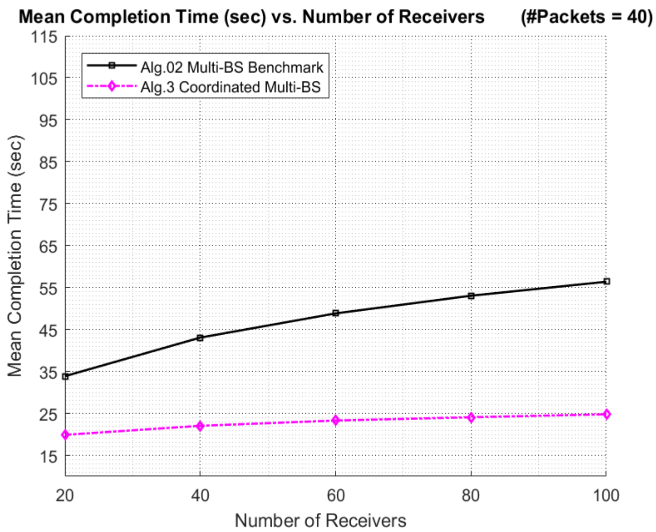

Figure 6. Graph comparing MATLAB Simulation Results Alg. 01 and Alg. 3 across a number of receivers with number of packets set to 40.

Alg. 02 conducted a "weighted clique search" in the same manner as Alg. 3 but, like Alg. 01, there were no edges in the graph of vertices for the instant decodability condition (*C2* of Section II.E). Fig. 5 shows the comparison of mean completion times across a number of packets and Fig. 6 shows the comparison across a number of receivers. As expected, Alg. 3 outperforms the benchmark case. With an increasing number of packets, the Alg. 3 showed an average improvement factor of 50.10% compared to Alg. 02. For an increasing number of receivers, Alg. 3 showed an average improvement factor of 50.60% compared to Alg. 02.

D. Combined Overall Results

Figs. 7-8 below are all the graphs from Sections III.B and III.C on the same plots. It can be seen that for the Multi-Base Station scenario, the improvement factor is not as great as the difference between Alg. 01 and Alg. 1 (71.76%). This is because for Alg. 02, there are two base stations which each can transmit a singular packet (which can be different). Thus, more users can be addressed at each transmission as the load can be considered *shared* between the two coordinated base stations. This is why even as benchmark cases, Alg. 02 significantly outperforms Alg. 01. If the number of base stations were to increase, there would be an even greater apparent difference. Figs. 7-8 also show that Alg. 3 outperforms Alg. 1 by an average improvement factor of 7.18% (for increasing number of packets) and by 9.52% (for increasing number of receivers). It was expected that there would be a greater improvement but this might be due to the fact that the generated achievable capacity rates did not vary much (fluctuating around 10Mbits/sec) thus the advantages of using multiple base stations did not shine through. If there were a greater variation in rates (as demonstrated in the example in Section II.E), it is likely that there would be a greater difference between Alg. 3 and Alg. 1. Nonetheless, the final results show that the developed *single-layer* scheme outperformed the proposed *multi-layer decisive user* scheme in Ref [10]-[11] and was easily applied to a more complex scenario involving two coordinated base stations.

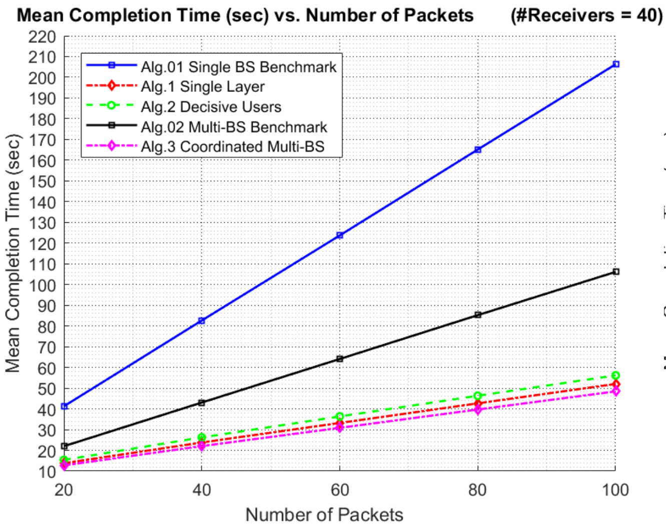

Figure 7. Graph comparing MATLAB Simulation Results for all algorithms across a number of packets with number of receivers set to 40.

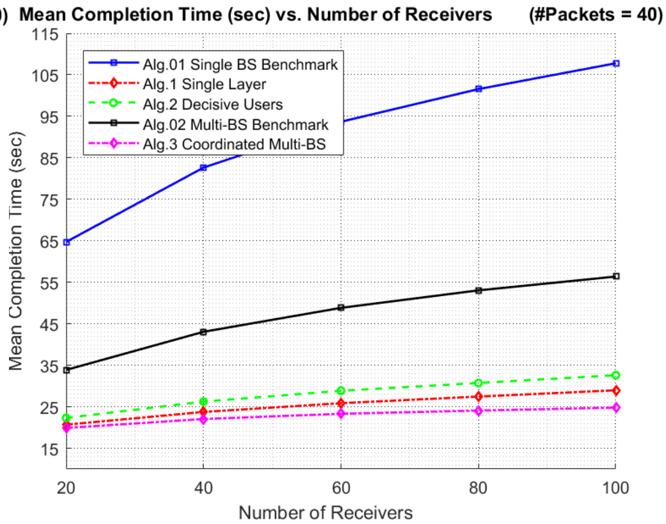

Figure 8. Graph comparing MATLAB Simulation Results for all algorithms across a number of receivers with number of packets set to 40.

IV. Conclusions

The aim of this project was to develop a *cross-layer* packet selection algorithm that reduced the physical delivery time (in seconds) of messages to wireless users using instantly decodable network coding (IDNC), incorporating physical layer rate awareness. Previous work is primarily focused on optimising parameters in the network layer to increase throughput while representing physical layer conditions as simple erasure channel models. Adapting a cross layer NC scheme, which considers the heterogeneity of transmission rates, was a more realistic approach to the problem. The proposed overarching idea of this project was that receiver capacity rates should be considered as a major factor when deciding packet combinations. Algorithm 1 provided an efficient and simple *single-layer* maximum weighted clique search where the maximum achievable capacity rates of receivers were incorporated in the vertex weights. The *multi-layer decisive user* scheme proposed in studies [10]-[11] was then developed as Alg. 2 using the heuristic maximum weighted clique searches of Alg. 1. Results found that Alg. 1 outperformed Alg. 2 by 10.22%, even though Alg. 1 was a simpler algorithm. When compared to the benchmark case representing conventional information flow (singular packet transmission), Alg. 1 outperformed Alg. 01 by 71.76%. Alg. 3 was then developed by firstly modifying the standard IDNC graph to become a multi-base station catered BS-IDNC graph. Alg. 3 was then implemented using the newly developed graph. Results found that Alg. 3 outperformed the Multi-Base Station benchmark case (Alg. 02) by 50.34%. When compared to Alg. 1, Alg. 3 performed better by an average of 8.35%. Overall, the project was a success. The development of a *simplified* cross-layer rate aware network coding algorithm which reduced the overall physical completion is the major contribution of this project. Alg. 1 could potentially be implemented in future complex NC applications such as in coordinated device-to-device communications.

V. Recommendations

Future work could include implementing the developed Alg. 1 to more complex NC scenarios such as coordinated device-to-device communication or cloud offloading networks. In regards to the multi-base station scenario, it would be interesting to see Alg. 3 implemented using more than two base stations. Additionally, one could investigate the effect of varying receiver capacity rates (e.g. increase noise power or have varying complex channel gains over time). In order to better model the empirical world, it is recommended that a better model is used to determine channel erasure probabilities instead of randomly generating a probability.

Acknowledgments

I would like to acknowledge my supervisor, Dr. Neda Aboutorab, for her consistent support and guidance throughout this project. I would also like to thank Ms. Kameliya Kaneva for helping me with channel capacity rates.

References

[1] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, “Network information flow,” *IEEE Transactions on Information Theory*, Vol. 46, No. 4, July 2000, pp. 1204-1216.

[2] C. Fragouli, J. Y. L. Boudec, and J. Widmer, “Network Coding: An Instant Primer,” *ACM SIGCOMM Computer Communication Review*, Vol. 36, No. 1, Jan. 2006.

[3] C. Fragouli, and E. Soljanin, “Network Coding Fundamentals,” *Foundations and Trends® in Networking*, Vol. 2, No. 1, 2007, pp. 1-133.

[4] K. A. Agha, Y. Benefattoum, and S. Martin, *Network Coding*, ISTE Ltd, John Wiley & Sons Inc., London, 2012, pp.1-24.

[5] S. Sorour and S. Valaee, "On Minimizing Broadcast Completion Delay for Instantly Decodable Network Coding," *2010 IEEE International Conference on Communications*, Cape Town, 2010, pp. 1-5.

[6] S. Sorour and S. Valaee, "Minimum Broadcast Decoding Delay for Generalized Instantly Decodable Network Coding," *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, Miami, FL, 2010, pp. 1-5.

[7] S. Sorour and S. Valaee, "Completion Delay Minimization for Instantly Decodable Network Codes," *IEEE/ACM Transactions on Networking*, Vol. 23, No. 5, Oct. 2015, pp. 1553-1567.

[8] J. K. Sundararajan, D. Shah, M. Medard, “Online Network Coding for Optimal Throughput and Delay – Three Receiver Case,” *International Symposium on Information Theory and its Applications ISITA2008*, Auckland, New Zealand, Dec. 2008, pp. 1-6.

[9] P. Sadeghi, R. Shams, and D. Traskov, “An optimal adaptive network coding scheme for minimizing decoding delay in broadcast erasure channels,” *EURASIP Journal of Wireless Communications and Networking*, April 2010, pp. 1–14.

[10] A. Douik S. Sorour T. Y. Al-Naffouri M.-S. Alouini "Rate aware instantly decodable network codes" *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1-6.

[11] A. Douik, S. Sorour, T. Y. Al-Naffouri and M. S. Alouini, "Rate Aware Instantly Decodable Network Codes," in *IEEE Transactions on Wireless Communications*, vol. 16, no. 2, Feb. 2017, pp. 998-1011.

[12] M. Saif, A. Douik and S. Sorour, "Rate aware network codes for coordinated multi base-station networks," *2016 IEEE International Conference on Communications (ICC)*, Kuala Lumpur, 2016, pp. 1-7.

[13] M. S. Karim, A. Douik, S. Sorour and P. Sadeghi, "Rate-aware network codes for completion time reduction in device-to-device communications," *2016 IEEE International Conference on Communications (ICC)*, Kuala Lumpur, 2016, pp. 1-7.

[14] M. S. Karim, A. Douik and S. Sorour, "Rate-Aware Network Codes for Video Distortion Reduction in Point-to-Multipoint Networks," in *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, Aug. 2017, pp. 7446-7460.

[15] A. Arefi, M. Khabbazian, M. Ardakani and G. Bansal, "Blind Instantly Decodable Network Codes for Wireless Broadcast of Real-Time Multimedia," in *IEEE Transactions on Wireless Communications*, Vol. 17, No. 4, Apr. 2018, pp. 2276-2288.

[16] A. Douik, S. Sorour, T. Y. Al-Naffouri and M. S. Alouini, "Decoding-Delay-Controlled Completion Time Reduction in Instantly Decodable Network Coding," in *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, Mar. 2017, pp. 2756-2770.

[17] T. Ho, R. Koetter, M. Medard, D. R. Karger and M. Effros, "The benefits of coding over routing in a randomized setting," *IEEE International Symposium on Information Theory, ISIT 2003*, 2003, pp. 442.

[18] D. Katabi, S. Katti, Wenjun Hu, H. Rahul and M. Medard, "On Practical Network Coding for Wireless Environments," *2006 International Zurich Seminar on Communications*, Zurich, 2006, pp. 84-85.

[19] B. A. Forouzan, *Data Communications and Networking*, 5th ed., McGraw-Hill, New York, 2013, pp.82, 85, 523.

[20] K. Kaneva, N. Aboutorab, S. Sorour, M. Reed, “Cross-Layer Offloading in Fog-RANs using Device Cooperation and Network Coding” *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, 2018, pp. 1-6.

[21] Y. N. Shnaiwer, S. Sorour, T. Y. Al-Naffouri and S. Al-Ghadhban, "Cross-Layer Cloud Offloading Using Fog Radio Access Networks and Network Coding," *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, Kansas City, MO, 2018, pp. 1-6.

[22] E. A. Lee and D. G. Messerschmitt, *Digital Communication*, 2nd ed., Kluwer Academic Publishers, The Netherlands, 1994, pp. 348-350.