

# Network Coding for Video Distortion Reduction in Device-to-Device Communications

Mohammad S. Karim, *Student Member, IEEE*, Sameh Sorour, *Member, IEEE*, and  
Parastoo Sadeghi, *Senior Member, IEEE*

## Abstract

In this paper, we study the problem of distributing a real-time video sequence to a group of partially connected cooperative wireless devices using instantly decodable network coding (IDNC). In such a scenario, the coding conflicts occur to service multiple devices with an immediately decodable packet and the transmission conflicts occur from simultaneous transmissions of multiple devices. To avoid these conflicts, we introduce a novel IDNC graph that represents all feasible coding and transmission conflict-free decisions in one unified framework. Moreover, a real-time video sequence has a hard deadline and unequal importance of video packets. Using these video characteristics and the new IDNC graph, we formulate the problem of minimizing the mean video distortion before the deadline as a finite horizon Markov decision process (MDP) problem. However, the backward induction algorithm that finds the optimal policy of the MDP formulation has high modelling and computational complexities. To reduce these complexities, we further design a two-stage maximal independent set selection algorithm, which can efficiently reduce the mean video distortion before the deadline. Simulation results over a real video sequence show that our proposed IDNC algorithms improve the received video quality compared to the existing IDNC algorithms.

## Index Terms

Real-Time Video Streaming, Markov Decision Process, Network Coding, Device-to-Device (D2D) Communications.

## I. INTRODUCTION

There is a sharp increase in the demand for high quality content over wireless networks. The simultaneous increase in the popularity of smart devices with improved computational, storage

M. S. Karim and P. Sadeghi are with the Australian National University, Australia ({mohammad.karim, parastoo.sadeghi}@anu.edu.au). S. Sorour is with the King Fahd University of Petroleum and Minerals, Saudi Arabia (samehsorour@kfupm.edu.sa).

and connectivity capabilities is expected to play an important role in addressing the increased throughput demand of wireless networks. This leads to a heterogeneous network architecture, where smart devices use two wireless interfaces simultaneously. One interface communicates with the central station using a *long-range wireless technology*, e.g., GSM, WiMAX or LTE, and the other interface communicates with other smart devices using a *short-range wireless technology*, e.g., Bluetooth or 802.11 adhoc mode. The usage of a short-range wireless technology has numerous practical advantages [1]–[5]. First, it offloads the central station to serve additional devices and increase the throughput of the network. Second, it increases the coverage zone of the network as devices can communicate to other devices via intermediate devices. Third, it reduces the cost associated with the deployment of new infrastructure required for the growing network size and devices' throughput demand. Finally, short-range channels provide more reliable delivery of the packets compared to the long-range channels due to small distances between the devices.

In this paper, we are interested in distributing a real-time video sequence to a group of partially connected cooperative wireless devices. Such a real-time video sequence has two distinct characteristics [6], [7]. First, it has unequally important packets such that some packets contribute more to the video quality compared to other packets. Second, it has a hard deadline such that the packets need to be decoded on-time to be usable at the applications. The video packets are broadcasted from a central station to the devices over long-range wireless channels. However, the devices receive partial content in those transmissions due to erasures in wireless channels. To recover the missing packets, the devices communicate with each other using their short-range wireless channels. Moreover, depending on the location of a device, it can be connected to all other devices directly (i.e., single-hop transmission) or via intermediate devices (i.e., multi-hop transmissions). Fig. 1 shows an example of a heterogeneous wireless network where devices use their cellular and short-range interfaces simultaneously.

Network coding has shown great potential to improve quality of services for video streaming applications in wireless networks [8]–[16]. In particular, random linear network coding (RLNC) minimizes the number of transmissions required for wireless broadcast of a set of packets [14]–[16]. However, this throughput benefit of RLNC comes at the expense of high decoding delay, high packet overhead, and high encoding and decoding complexities. On the other hand, *instantly decodable network coding* (IDNC) has drawn significant attention due to its several attractive properties [17]–[23]. IDNC generates coded packets that are immediately decodable at the devices. This instant decodability property allows a progressive improvement in the video quality as the

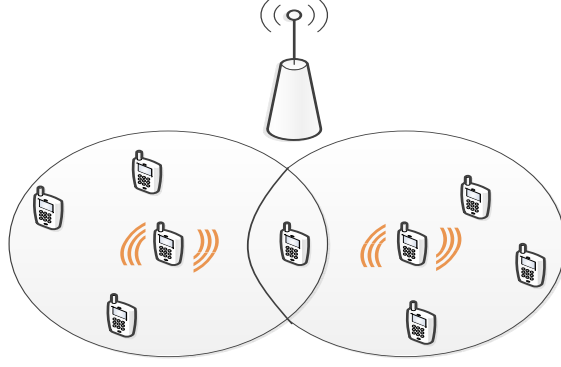


Fig. 1: Devices use their cellular and short-range interfaces simultaneously.

devices decode more packets. Furthermore, the encoding process of IDNC is performed using simple XOR operations. This reduces packet overhead required for coefficient reporting. The decoding process of IDNC is also performed using XOR operations, which is suitable for implementation in small devices.

In this paper, we are interested in designing an efficient IDNC framework that minimizes the mean video distortion before the deadline in a partially connected device-to-device (D2D) network. In such scenarios, IDNC framework needs to take into account the unequal importance of video packets, hard deadline, erasures of wireless channels, and coding and transmission conflicts in making decisions. In this context, our main contributions can be summarized as follows:

- We introduce a novel IDNC graph that represents both coding and transmission conflicts of a partially connected D2D network with one common transmission channel. The representation of transmission conflicts along with the well-known coding conflicts in one graph were suggested in [24], [25] for distributed storage and femtocaching-assisted networks for transmissions over orthogonal channels. However, the representation of transmission and coding conflicts in one graph for a partially connected D2D network with devices all transmitting over one common channel is not trivial and is novel to this paper. Indeed, this novel graph representation has to account for the coverage zones of different devices, potential collisions over the common channel, each device cannot transmit and receive concurrently and the packet reception at a device is subject to interference from simultaneous transmissions of multiple devices.
- Using the video characteristics and the new IDNC graph, we formulate the problem of minimizing the mean video distortion before the deadline as a finite horizon Markov decision process (MDP) problem. Our MDP formulation is a sequential decision making process in which the

decision is made at the current time slot and takes into account the coding opportunities at the successor time slots so that the devices experience the minimum video distortion at the end of the deadline. The Markov decision process was also used in [6], [21] for point to multi-point networks, where the central station always transmits packets to the devices. However, the MDP formulation for a partially connected D2D network is different compared to those in [6], [21] since it takes into account the fact that a set of devices transmit XOR packet combinations simultaneously and another set of devices receive a single transmitted packet (i.e., free from transmission conflicts) from the transmitting devices.

- We further design a two-stage maximal independent set (TS-MIS) selection algorithm, which has much lower modelling and computational complexities compared to the MDP formulation. This is a greedy approach since it makes decision at the current time slot without going through all possible future situations before the deadline. However, this algorithm is designed following the properties of the minimum video distortion problem in a partially connected D2D network.
- We use a real video sequence to evaluate the performance of different algorithms. Simulation results show that our proposed IDNC algorithms improve the received video quality compared to the IDNC algorithms in [20], [26], [27] that were not particularly designed for a real-time video sequence and a partially connected D2D network.

The rest of this paper is organized as follows. We discuss the related works in Section II. The system model is described in Section III. Section IV defines the novel IDNC graph. We formulate the minimum video distortion problem into an MDP framework in Section V and design a TS-MIS selection algorithm in Section VI. Section VII describes the calculations for the importance of individual video packet. Simulation results are presented in Section VIII. Finally, Section IX concludes the paper.

## II. RELATED WORKS

In this section, we first discuss the related network coding schemes designed for point to multi-point (PMP) networks (i.e., the central station is responsible to transmit all packets to all devices) and then discuss the related network coding schemes designed for fully connected D2D networks (i.e., each device is directly connected to all other devices) and partially connected D2D networks as considered in this paper.

### A. Point to Multi-Point (PMP) Networks

Numerous IDNC schemes have been developed to meet different requirements of video streaming applications [6], [7], [19]–[23]. In particular, the authors in [19], [20] considered IDNC for wireless broadcast of a set of packets and serviced the maximum number of devices with any new packet in each time slot. Moreover, the authors in [21] addressed the problem of minimizing the number of time slots required for broadcasting a set of packets in IDNC systems and formulated the problem into a stochastic shortest path (SSP) framework. However, the works in [19]–[21] neither considered explicit packet delivery deadline nor considered unequal importance of video packets.

Several other works including [6], [7], [22], [23] considered video streaming applications with unequally important packets. The work in [22] proposed an IDNC scheme that is asymptotically throughput optimal for the three-device system subject to sequential packet delivery deadline constraints. Moreover, the works in [6], [7] determined the importance of each video packet based on its contribution to the video quality and proposed IDNC schemes to maximize the overall video quality at the devices. The aforementioned works [6], [7], [19]–[23] developed IDNC schemes for conventional PMP networks, which are fundamentally different from partially connected D2D networks considered in this paper.

### B. Fully Connected D2D Networks

The network coded D2D communications have drawn a significant attention over the past several years to take advantages of both network coding and devices' cooperation. The works in [28]–[30] incorporated algebraic network coding for D2D communications at the packet level. In particular, the authors in [28] provided upper and lower bounds on the number of time slots required for recovering all the missing packets at the devices. Furthermore, the authors in [29] proposed a randomized algorithm that has a high probability of achieving the minimum number of time slots. However, the works in [28]–[30] neither considered erasure channels nor considered addressing the hard deadline for high importance video packets.

Several other works including [26], [31], [32] adopted IDNC for D2D communications. In [31], [32], the authors selected a transmitting device and its XOR packet combination to service a large number of devices with any new packet in each time slot. Moreover, the authors in [26] prioritized packets based on their contributions to the video quality as in [6], [7] and proposed a joint device and packet selection algorithm that maximizes the overall video quality after the current time slot. The aforementioned works [26], [28]–[32] developed network coding schemes

for a fully connected D2D network. This fully connected D2D network is not always practical due to the limited transmission range of devices. Consequently, in this paper, we consider a partially connected D2D network, which is more general and includes the fully connected D2D network as a special case. Unlike a single transmitting device in a fully connected D2D network, multiple devices can transmit simultaneously in a partially connected D2D network without causing transmission conflicts.

### C. Partially Connected D2D Networks

In the context of partially connected networks, the related works to our work are [27], [33]–[35]. In particular, the authors in [33] provided various necessary and sufficient conditions that characterize the number of transmissions required to recover all missing packets at all devices. The authors in [34] continued the work in [33] and showed that solving the minimum number of transmissions problem exactly or even approximately is computationally intractable. Moreover, the authors in [33], [34] adopted algebraic network coding in large finite fields. Unlike the works in [33], [34], we consider erasure channels, XOR based network coding, explicit packet delivery deadline and unequal importance of video packets.

The works in [27], [35] adopted IDNC for a partially connected D2D network and addressed the problem of servicing a large number of devices with any new packet in each time slot. However, these works are not readily compatible with the real-time video sequence that has a hard deadline and unequally important video packets. In contrast to [27], [35], we introduce a novel IDNC graph that represents all feasible coding and transmission conflict-free decisions in one unified framework and develop an efficient IDNC framework that prioritizes the distribution of high importance video packets to all devices before the deadline.

## III. SYSTEM MODEL

We consider a wireless network with a set of  $M$  devices  $\mathcal{M} = \{R_1, \dots, R_M\}$ .<sup>1</sup> Each device in  $\mathcal{M}$  is interested in receiving a set of  $N$  source packets  $\mathcal{N} = \{P_1, \dots, P_N\}$ . Packets are transmitted in two phases. The first phase consists of the initial  $N$  time slots, in which a central station (e.g., a base station) broadcasts the packets from  $\mathcal{N}$  in an uncoded manner. However, a subset of devices from  $\mathcal{M}$  receive each broadcasted packet due to erasures in long-range wireless channels. We assume that at least one device from  $\mathcal{M}$  receives each broadcasted packet.

<sup>1</sup>Throughout this paper, we use calligraphic letters to denote sets and their corresponding capital letters to denote the cardinalities of these sets.

The second phase starts after  $N$  time slots (referred to as a *D2D phase*), in which the devices cooperate with each other to recover their missing packets using short-range wireless channels. There is a limit on the number of allowable time slots  $\Theta$  used in the D2D phase as the deadline for delivering  $N$  packets expires after  $\Theta$  D2D time slots. This deadline constraint arises from the minimum delivery delay requirement in real-time video streaming applications. At any D2D time slot  $t \in [1, 2, \dots, \Theta]$ , we can compute the number of remaining time slots for delivering  $N$  packets as,  $Q = \Theta - t + 1$ . A device can either transmit or listen to a packet in each D2D time slot.

We consider a partially connected network, where a device is connected to another device directly (i.e., single hop) or via intermediate devices (i.e., multiple hops). The packet reception probabilities of all channels connecting all pairs of devices is stored in an  $M \times M$  *symmetric connectivity matrix* (SCM)  $\mathbf{Y} = [y_{i,k}]$ ,  $\forall (R_i, R_k) \in \mathcal{M}$ , such that:

$$y_{i,k} = \begin{cases} 1 - \epsilon_{i,k} & \text{if } R_i \text{ is directly connected to } R_k, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

$$y_{i,i} = 1, \quad \forall R_i \in \mathcal{M}. \quad (2)$$

Here, a packet transmission from device  $R_i$  to device  $R_k$  is subject to an independent Bernoulli erasure with probability  $\epsilon_{i,k}$ . We assume reciprocal channels such as  $\epsilon_{i,k} = \epsilon_{k,i}$ . A channel connecting a pair of devices is independent, but not necessarily identical, to another channel connecting another pair of devices. In fact, a device  $R_i \in \mathcal{M}$  is directly connected to a subset of devices in  $\mathcal{M}$  depending on the location of the device in the network.

**Example 1.** An example of SCM with  $M = 4$  devices is given as follows:

$$\mathbf{Y} = \begin{pmatrix} 1 & 0.84 & 0 & 0 \\ 0.84 & 1 & 0.75 & 0 \\ 0 & 0.75 & 1 & 0.91 \\ 0 & 0 & 0.91 & 1 \end{pmatrix}. \quad (3)$$

The SCM in (3) represents a line network shown in Fig. 2. In this example, device  $R_1$  is not directly connected to device  $R_3$  and thus,  $y_{1,3} = 0$ . Moreover, device  $R_1$  is directly connected to device  $R_2$  with packet reception probability  $y_{1,2} = 1 - \epsilon_{1,2} = 0.84$ .

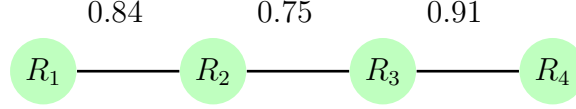


Fig. 2: A line network corresponding to SCM in (3).

**Definition 1.** (Coverage Zone) The coverage zone of transmitting device  $R_i$  (denoted by  $\mathcal{Y}_i$ ) is defined as the set of neighboring devices that are directly connected to it using short-range wireless channels. In other words,  $\mathcal{Y}_i = \{R_k \mid y_{i,k} \neq 0\}$ .

**Definition 2.** (Transmission Conflict) A transmission conflict is experienced by a device when it belongs to the coverage zones of multiple transmitting devices. In other words, when two neighboring devices  $R_i$  and  $R_r$  of device  $R_k$  transmit simultaneously, their transmissions will collide and device  $R_k$  will not be able to receive any of these transmissions successfully.

After each time slot, the reception status of all packets at all devices is stored in an  $M \times N$  global status matrix (GSM)  $\mathbf{F} = [f_{k,l}]$ ,  $\forall R_k \in \mathcal{M}, P_l \in \mathcal{N}$ , such that:

$$f_{k,l} = \begin{cases} 0 & \text{if packet } P_l \text{ is received by device } R_k, \\ 1 & \text{if packet } P_l \text{ is missing at device } R_k. \end{cases} \quad (4)$$

**Example 2.** An example of GSM with  $M = 4$  devices and  $N = 3$  packets is given as follows:

$$\mathbf{F} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}. \quad (5)$$

According to the GSM  $\mathbf{F}$ , the following two sets of packets can be attributed to each device  $R_k \in \mathcal{M}$  at any given time slot  $t$ :

- 1) The *Has set* ( $\mathcal{H}_k$ ) of device  $R_k$  is defined as the set of packets that are successfully received by device  $R_k$ . In (5), the Has set of device  $R_1$  is  $\mathcal{H}_1 = \{P_3\}$ .
- 2) The *Wants set* ( $\mathcal{W}_k$ ) of device  $R_k$  is defined as the set of packets that are missing at device  $R_k$ . In other words,  $\mathcal{W}_k = \mathcal{N} \setminus \mathcal{H}_k$ . In (5), the Wants set of device  $R_1$  is  $\mathcal{W}_1 = \{P_1, P_2\}$ .

The cardinalities of  $\mathcal{H}_k$  and  $\mathcal{W}_k$  are denoted by  $H_k$  and  $W_k$ , respectively. The set of devices having *non-empty Wants sets* is denoted by  $\mathcal{M}_w$ . This set can be defined as:  $\mathcal{M}_w = \{R_k \mid \mathcal{W}_k \neq \emptyset\}$ .



At any given time slot  $t$ , a device  $R_k$  in  $\mathcal{M}_w$  belongs to one of the following two sets:

- The *critical set* of devices ( $\mathcal{C}$ ) is defined as the set of devices with the number of missing packets being greater than or equal to the number of remaining  $Q$  time slots (i.e.,  $W_k \geq Q, \forall R_k \in \mathcal{C}$ ).
- The *non-critical set* of devices ( $\mathcal{A}$ ) is defined as the set of devices with the number of missing packets being less than the number of remaining  $Q$  time slots (i.e.,  $W_k < Q, \forall R_k \in \mathcal{A}$ ).

In fact,  $\mathcal{C}(t) \cup \mathcal{A}(t) = \mathcal{M}_w(t)$ .

**Definition 3.** (*Instantly Decodable Packet*) A transmitted packet is instantly decodable for device  $R_k$  if it contains exactly one source packet from  $\mathcal{W}_k$ .

**Definition 4.** (*Targeted Device*) Device  $R_k$  is targeted by transmitting device  $R_i$  with packet  $P_l$  at time slot  $t$  when device  $R_k$  belongs to the coverage zone of a single transmitting device  $R_i$  and will immediately decode packet  $P_l$  upon receiving the transmitted packet from device  $R_i$ .

**Definition 5.** (*Individual Completion Time*) At any time slot  $t$ , individual completion time of device  $R_k$  (denoted by  $T_{W_k}$ ) is the total number of time slots required to decode all the missing packets in  $W_k$ .

Individual completion time of device  $R_k$  for  $W_k$  missing packets can be  $T_{W_k} = W_k, W_k + 1, \dots$  depending on the number of time slots in which this device is targeted with a new packet (i.e., satisfies Definition 4) and the channel erasures experienced by this device in those transmissions.

**Definition 6.** (*Individual Completion Times of All Non-critical Devices*) At any time slot  $t$ , individual completion times of all non-critical devices (denoted by  $T_A$ ) is the total number of time slots required to deliver all the missing packets to all non-critical devices in  $\mathcal{A}$ .

**Definition 7.** (*Transmission Schedule*) A transmission schedule  $\mathcal{L} = \{\kappa(t)\}, \forall t \in \{1, \dots, \Theta\}$  is defined as the set of transmitting devices and packet combinations at every time slot  $t$  before the deadline. Furthermore,  $\mathbf{L}$  is the set of all possible transmission schedules and  $\mathcal{L} \in \mathbf{L}$ .

#### A. Centralized Protocol for Implementing the System

As a potential protocol, we now discuss the possible implementation processes of the IDNC system in a centralized fashion.<sup>2</sup> In this case, the central station forms the SCM  $\mathbf{Y}$  and the GSM

<sup>2</sup>A distributed approach can be adopted to make a decision at each device separately. Many works on distributed approaches were referred in [1], [35].

$\mathbf{F}$ , and coordinates the global decision making process in each time slot.

1) *Coverage Zone*: The devices exchange Hello messages among themselves in order to determine their coverage zones (i.e., neighbouring devices). Each device broadcasts one bit Hello message. Other  $O(M - 1)$  neighboring devices generate one bit response message. Consequently, a device discovers its coverage zone using  $M$  bits. The coverage zones of all  $M$  devices in the network can be discovered using  $M^2$  bits. Since the locations of all devices in the network are static with respect to the delivery deadline of the video sequence, the communication overhead of  $M^2$  bits is required only once.

2) *Packet Reception Probability*: In this paper, the network coding is performed at the network layer. With an efficient channel coding performed at the physical layer, an abstraction of channel model at the network layer is often considered, where a transmitted packet is either received or lost with an average erasure probability. This channel erasure probability is a slowly changing parameter in the network and can be estimated based on the test (or the past) packet reception performance over the channel. Once the packet reception probabilities connecting a device to other devices are estimated, the device sends this information to the central station. A channel erasure probability can be represented using  $\lceil \log_2 100 \rceil$  bits, where 100 is the maximum erasure probability in percentage. Since each of  $M$  devices sends  $M - 1$  channels' information connecting this device to other  $M - 1$  devices, the overall communication overhead is  $M^2 \lceil \log_2 100 \rceil$  bits. Using this information, the central station forms the SCM  $\mathbf{Y}$ .

3) *GSM Update*: Each device sends a positive/negative acknowledgement to the central station indicating a received/lost packet. Note that a device needs to use one bit to acknowledge a received packet. Since there are  $M$  devices in the network, the overall communication overhead from feedback is  $M$  bits per time slot. With the feedback reception, the central station updates the GSM  $\mathbf{F}$  in each time slot.

4) *Centralized Decision*: In each time slot, the central station selects a set of transmitting devices and their packet combinations using an IDNC algorithm. It then informs the transmitting devices separately about the packet combinations and uses the indices of individual packets. In fact, a packet combination can be formed XORing  $O(N)$  individual packets. The central station sends a bitmap of  $N$  bits to each transmitting device, where the entries with 1's are the indices of the source packets that are XORed together. In a partially connected D2D network, there can be  $O(\frac{M}{2})$  transmitting devices since a device cannot receive and transmit simultaneously. The overall communication overhead to inform  $O(\frac{M}{2})$  transmitting devices about their packet combinations is  $\frac{MN}{2}$  bits, which

is negligible compared to the typical size of a packet in wireless networks.

### B. Importance of Individual Packet

The importance of individual packet in a video sequence can be determined by the source and can be marked on a special field of the packet header. This field can be part of the real-time transport protocol (RTP) header or the network coding header [7]. To compute the importance of packet  $P_l$ , we follow a similar approach as in [6], [7] and decode the entire video sequence with this packet missing and assign the resulting distortion to the importance value of this packet. This is an approximation as the actual distortion of a packet depends on the reception status of prior and subsequent packets at the devices. Having defined the importance of individual packets, we calculate the individual video distortion of device  $R_k$  at time slot  $t$  as:

$$d_k^{(t)} = \sum_{P_l \in \mathcal{W}_k} \delta_{k,l} \quad (6)$$

where  $\delta_{k,l}$  is the importance of missing packet  $P_l$  at device  $R_k$ . Here, we consider that distortions caused by the loss of multiple packets at a device are additive, which is accurate for sparse losses. Nonetheless, these approximations allow us to separate the total distortion of a video sequence into a set of distortions corresponding to individual packets and optimize the decisions for individual packets. To compute the received video quality at the devices, we capture the correlations of the packets in a video sequence. We use these correlations to compute the actual video distortion at a device resulting from its missing packets at the end of the deadline. These practical aspects in computing the received video quality at the devices will be further explained in Section VII.

## IV. NOVEL IDNC GRAPH

In this section, we define a novel IDNC graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$  to represent both coding and transmission conflicts in one unified framework and select a set of transmitting devices and their XOR packet combinations in each D2D time slot. A transmission conflict occurs due to the simultaneous transmissions from multiple devices to a device in their coverage zones. Moreover, a coding conflict occurs due to the instant decodability constraint.

<table><tr><td>0</td></tr><tr><td>1</td></tr></table>	0	1	<table><tr><td>1</td></tr><tr><td>0</td></tr><tr><td>0</td></tr></table>	1	0	0	<table><tr><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td></tr></table>	0	1	0	0	1	0	<table><tr><td>0</td></tr><tr><td>0</td></tr></table>	0	0
0																
1																
1																
0																
0																
0	1															
0	0															
1	0															
0																
0																
$\mathbf{F}_1$	$\mathbf{F}_2$	$\mathbf{F}_3$	$\mathbf{F}_4$													

Fig. 3: Four LSMs for four devices corresponding to SCM in (3) and GSM in (5)

#### A. Vertex Set

To define vertex set  $\mathcal{V}$  of IDNC graph  $\mathcal{G}$ , given GSM  $\mathbf{F}$  at time slot  $t$ , we form an  $Y_i \times H_i$  *local status matrix (LSM)*  $\mathbf{F}_i = [f_{k,l}]$ ,  $\forall R_k \in \mathcal{Y}_i, P_l \in \mathcal{H}_i$ , for a device  $R_i \in \mathcal{M}$  such that<sup>3</sup>:

$$f_{k,l} = \begin{cases} 0 & \text{if packet } P_l \text{ is received by device } R_k, \\ 1 & \text{if packet } P_l \text{ is missing at device } R_k. \end{cases} \quad (7)$$

Note that the rows in LSM  $\mathbf{F}_i$  represent the devices which are in the coverage zone of device  $R_i$  and the columns in LSM  $\mathbf{F}_i$  represent the packets in the Has set of device  $R_i$  which are used for forming a transmitted packet from device  $R_i$ . Fig. 3 shows four LSMs for four devices corresponding to SCM in (3) and GSM in (5).

We generate a vertex for a missing packet in each LSM at IDNC graph  $\mathcal{G}$ . In fact, for each LSM  $\mathbf{F}_i, \forall R_i \in \mathcal{M}$ , a vertex  $v_{i,kl}$  is generated for a packet  $P_l \in \{\mathcal{H}_i \cap \mathcal{W}_k\}, \forall R_k \in \mathcal{Y}_i$ .<sup>4</sup> In other words, a vertex is generated for a missing packet of another device in  $\mathcal{Y}_i$ , which also belongs to the Has set  $\mathcal{H}_i$  of potential transmitting device  $R_i$ . Note that a missing packet at a device can generate more than one vertex in graph  $\mathcal{G}$  since that packet can be present in multiple LSMs. Once the vertices are generated in IDNC graph  $\mathcal{G}$ , two vertices  $v_{i,kl}$  and  $v_{r,mn}$  are adjacent (i.e., connected) by an edge due to either a coding conflict or a transmission conflict.

#### B. Coding Conflicts

Two vertices  $v_{i,kl}$  and  $v_{r,mn}$  are adjacent by an edge due to a coding conflict if one of the following two conditions holds:

<sup>3</sup>The number of devices in the coverage zone of device  $R_i$  is  $Y_i = |\mathcal{Y}_i|$ .

<sup>4</sup>Note that vertex  $v_{i,kl}$  represents a transmission from device  $R_i \in \mathcal{M}$  to a neighboring device  $R_k \in \mathcal{Y}_i$  with packet  $P_l$ .

- **C1:**  $P_l \neq P_n$  and  $R_k = R_m$ . In other words, two vertices are induced by different missing packets  $P_l$  and  $P_n$  at the same device  $R_k$ .
- **C2:**  $R_k \neq R_m$  and  $P_l \neq P_n$  but  $P_l \notin \mathcal{H}_m$  or  $P_n \notin \mathcal{H}_k$ . In other words, two different devices  $R_k$  and  $R_m$  require two different packets  $P_l$  and  $P_n$ , but at least one of these two devices does not possess the other missing packet. As a result, that device cannot decode a new packet from an XOR combination of  $P_l \oplus P_n$ .

### C. Transmission Conflicts

Two vertices  $v_{i,kl}$  and  $v_{r,mn}$  are adjacent by an edge due to a transmission conflict if one of the following three conditions holds:

- **C3:**  $R_i \neq R_r$  and  $R_k = R_m \in \{\mathcal{Y}_i \cap \mathcal{Y}_r\}$ . In other words, two vertices representing the transmissions from two different devices  $R_i$  and  $R_r$  to the same device  $R_k$  in the coverage zones of both transmitting devices  $R_i$  and  $R_r$ . This prohibits transmissions from two different devices to the same device in the common coverage zone and prevents interference at that device from multiple transmissions.
- **C4:**  $R_i \neq R_r$  and  $R_k \neq R_m$  but  $R_k \in \{\mathcal{Y}_i \cap \mathcal{Y}_r\}$  or  $R_m \in \{\mathcal{Y}_i \cap \mathcal{Y}_r\}$ . In other words, two vertices representing the transmissions from two different devices  $R_i$  and  $R_r$  to two different devices  $R_k$  and  $R_m$ , but at least one of these two devices  $R_k$  and  $R_m$  is in the coverage zones of both transmitting devices  $R_i$  and  $R_r$ . This prohibits transmission from device  $R_r$  to device  $R_m$  in the case of transmission from device  $R_i$  to device  $R_k$ , and vice versa.
- **C5:**  $R_i \neq R_r$  but  $R_i = R_m$  or  $R_r = R_k$ . In other words, two vertices representing the transmissions from two different devices  $R_i$  and  $R_r$ , but at least one of these two devices  $R_i$  and  $R_r$  is targeted by the other device. This prohibits transmission from a device in the case of that device is already targeted by another device, and vice versa. In other words, a device cannot be a transmitting device and a targeted device simultaneously.

### D. Maximal Independent Sets

With this graph representation, we can define all feasible coding and transmission conflict-free decisions by the set of all maximal independent sets in IDNC graph  $\mathcal{G}$ .

**Definition 8.** (*Independent Set*) An independent set or a stable set in a graph is a set of pairwise non-adjacent vertices.

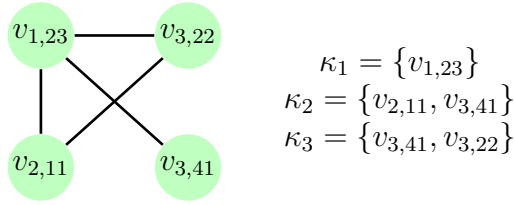


Fig. 4: IDNC graph corresponding to SCM in (3) and GSM in (5).

**Definition 9.** (*Maximal Independent Set*) A maximal independent set (denoted by  $\kappa$ ) is an independent set that cannot be extended by including one more vertex without violating pairwise non-adjacent vertex constraint. In other words, a maximal independent set is an independent set that is not subset of any larger independent set [36].

Each device can have at most one vertex in a maximal independent set  $\kappa$  representing either a transmitting device or a targeted device. Moreover, the selection of a maximal independent set  $\kappa$  is equivalent to the selection of a set of transmitting devices  $\mathcal{Z}(\kappa) = \{R_i | v_{i,kl} \in \kappa\}$  and a set of targeted devices  $\mathcal{X}(\kappa) = \{R_k | v_{i,kl} \in \kappa\}$ . Each of the selected transmitting devices forms a coded packet by XORing the source packets identified by the vertices in  $\kappa$  representing transmission from that device.

**Example 3.** The new IDNC graph  $\mathcal{G}$  corresponding to SCM in (3) and GSM in (5) is shown in Fig. 4. The maximal independent sets of this graph are also listed in this figure.

## V. MINIMUM VIDEO DISTORTION PROBLEM FORMULATION

In this section, we first define the minimum mean video distortion problem and then formulate the problem into a finite horizon Markov decision process (MDP) framework.

### A. Problem Description

We now discuss the characteristics of the minimum video distortion problem and infer that it is a sequential decision making problem. In such a problem, the decision is made at the current time slot and needs to take into account all possible GSMs and their coding opportunities at the successor time slots before the deadline. First, some packets are needed to be exchanged via multiple hops before the deadline due to the partial connectivity in the network. Therefore, the decision at the current time slot needs to consider that some devices are able to quickly relay their received packets to a large number of other devices in the successor time slots due to having large coverage zones.

Second, it is not always possible to target all the devices with a new packet due to the instant decodability constraint. Moreover, servicing the largest number of devices with a new packet in the current time slot may reduce the coding opportunities at the successor time slots, and results in delivering a small number of packets to the devices before the deadline. Therefore, the decision at the current time slot needs to take into account the coding opportunities at the successor time slots before the deadline. Finally, the hard deadline constraint may limit the number of delivered packets to the devices. Therefore, the decision maker needs to be adaptive to the deadline so that the received video packets before the deadline contribute to the maximum video quality at the devices.

Based on all aforementioned aspects, we can infer that our problem is a sequential decision making problem that not necessarily minimizes the mean video distortion after the current time slot, but rather it achieves the minimum mean video distortion at the end of the deadline. Moreover, due to the random nature of channel erasures, our system is a stochastic system, in which there are many possible outcomes resulting from a chosen maximal independent set at the current time slot. To define the minimum video distortion problem, let us consider  $d_k(\mathcal{L})$  and  $\mathcal{H}_k(\mathcal{L})$  are the individual video distortion and the Has set of device  $R_k$  at the end of the deadline for a given transmission schedule  $\mathcal{L}$ . Moreover,  $d_k^{(0)}$  is the initial individual video distortion of device  $R_k$  before starting the D2D phase and can be computed following (6). With these results, we define the problem of minimizing the mean video distortion at the end of the deadline as a transmission schedule selection problem such that:

$$\begin{aligned}
\mathcal{L}^* &= \arg \min_{\mathcal{L} \in \mathbf{L}} \left\{ \frac{\sum_{R_k \in \mathcal{M}} d_k(\mathcal{L})}{M} \right\} \\
&= \arg \min_{\mathcal{L} \in \mathbf{L}} \left\{ \sum_{R_k \in \mathcal{M}} d_k(\mathcal{L}) \right\} \\
&= \arg \min_{\mathcal{L} \in \mathbf{L}} \left\{ \sum_{R_k \in \mathcal{M}} \left( d_k^{(0)} - \sum_{P_l \in \mathcal{H}_k(\mathcal{L})} \delta_{k,l} \right) \right\} \\
&= \arg \max_{\mathcal{L} \in \mathbf{L}} \left\{ \sum_{R_k \in \mathcal{M}} \sum_{P_l \in \mathcal{H}_k(\mathcal{L})} \delta_{k,l} \right\}.
\end{aligned} \tag{8}$$

The optimization problem in (8) can be formulated using a finite horizon Markov decision process and the optimal transmission schedule can be found using the backward induction algorithm, which

will be shown in the following two subsections.

### B. MDP Formulation

We formulate the problem of minimizing the mean video distortion before the deadline as a finite horizon Markov decisions process (MDP) problem, which models our decision based stochastic dynamic systems with a finite number of steps.

- 1) *Horizon*: The number of time slots  $\Theta$  used in the D2D phase, over which the decisions are made. The MDP problem is a finite horizon problem with  $\Theta$  time slots.
- 2) *State Space*  $\mathcal{S}$ : States are defined by all possibilities of GSM  $\mathbf{F}$  that may occur during the D2D phase. GSM corresponding to state  $s \in \mathcal{S}$  is represented by  $\mathbf{F}(s)$ . We can characterize each state  $s$  according to its Has and Wants vectors,  $\mathbf{h}(s) = [H_1(s), \dots, H_M(s)]$  and  $\mathbf{w}(s) = [W_1(s), \dots, W_M(s)]$ . The state at the starting of the D2D phase is denoted by  $s_a$  and its Has and Wants vectors are denoted by  $\mathbf{h}(s_a) = [H_1(s_a), \dots, H_M(s_a)]$  and  $\mathbf{w}(s_a) = [W_1(s_a), \dots, W_M(s_a)]$ .

Given GSM  $\mathbf{F}$  is an  $M \times N$  binary matrix, the size of the state space is  $|\mathcal{S}| = O(2^{MN})$ . However, the devices receive a subset of packets from  $\mathcal{N}$  in the initial  $N$  time slots from the central station. We can conclude that the size of the state space for D2D phase is  $|\mathcal{S}| = 2^{MN} - 2^{(\sum_{R_i \in \mathcal{M}} H_i(s_a))}$ .

- 3) *Action Space*  $\mathcal{A}(s)$ : The action space for each state  $s$  consists of the set of all possible maximal independent sets in IDNC graph  $\mathcal{G}(s)$ . The size of the action space for a given state  $\mathbf{F}(s)$  is  $|\mathcal{A}(s)| = O(3^{|\mathcal{V}|/3})$  [36], where  $|\mathcal{V}|$  is the size of the vertex set  $\mathcal{V}$  in graph  $\mathcal{G}(s)$ .
- 4) *State-Action Transition Probability*  $\mathcal{P}_a(s, \acute{s})$ : The state-action transition probability  $\mathcal{P}_a(s, \acute{s})$  for an action  $a = \kappa(s)$  can be defined based on the possibilities of the variations in GSM  $\mathbf{F}(s)$  from state  $s$  to the successor state  $\acute{s}$ . With action  $\kappa(s)$ , the system transits to the successor state  $\acute{s}$  depending on the targeted devices in  $\kappa(s)$  and the packet reception probabilities of the targeted devices. In other words, successor state  $s' \in \mathcal{S}(s, a)$  such that  $\mathcal{S}(s, a) = \{\acute{s} | \mathcal{P}_a(s, \acute{s}) > 0\}$ . To define  $\mathcal{P}_a(s, \acute{s})$ , we first introduce the following two sets:

$$\mathcal{T} = \{R_k | R_k \in \mathcal{X}(\kappa), W_k(\acute{s}) = W_k(s) - 1\} \quad (9)$$

$$\tilde{\mathcal{T}} = \{R_k | R_k \in \mathcal{X}(\kappa), W_k(\acute{s}) = W_k(s)\} \quad (10)$$

Here, the first set  $\mathcal{T}$  includes the targeted devices whose Wants sets have decreased from



state  $s$  to the successor state  $\acute{s}$  due to successful packet receptions. The second set  $\tilde{\mathcal{T}}$  includes the targeted devices whose Wants sets have remained unchanged due to packet losses. Using these two sets and considering all transmissions are independent of each other, we can express  $\mathcal{P}_a(s, \acute{s})$  as follows:

$$\mathcal{P}_a(s, \acute{s}) = \prod_{R_k \in \mathcal{T}: v_{i,kl} \in \kappa(s)} (1 - \epsilon_{i,k}) \times \prod_{R_k \in \tilde{\mathcal{T}}: v_{i,kl} \in \kappa(s)} (\epsilon_{i,k}) \quad (11)$$

5) *State-Action Reward*: Having required the minimum mean video distortion at the end of the deadline, at state  $s$ , the expected reward  $\bar{r}_k(s, a)$  of action  $a = \kappa(s)$  on each device  $R_k \in \mathcal{M}_w(s)$  is defined as the expected video distortion reduction at device  $R_k$  at the successor state  $s'$ . We can calculate the expected reward of action  $a = \kappa(s)$  on each targeted device  $R_k \in \mathcal{X}(a)$  as  $\bar{r}_k(s, a | v_{i,kl} \in \kappa(s)) = \delta_{k,l}(1 - \epsilon_{i,k})$ . On the other hand, we can define the expected reward of action  $a = \kappa(s)$  on each ignored device  $R_k \in \{\mathcal{M}_w(s) \setminus \mathcal{X}(a)\}$  as  $\bar{r}_k(s, a | R_k \in \mathcal{M}_w(s) \setminus \mathcal{X}(a)) = 0$ . With these results, the total expected reward of action  $a \in \mathcal{A}(s)$  over all the devices in  $\mathcal{M}_w(s)$  can be calculated as:

$$\bar{r}(s, a) = \sum_{R_k \in \mathcal{M}_w(s)} \bar{r}_k(s, a) = \sum_{R_k \in \mathcal{X}(a): v_{i,kl} \in \kappa(s)} \delta_{k,l}(1 - \epsilon_{i,k}). \quad (12)$$

### C. MDP Solution Complexity

An MDP policy  $\pi = [\pi(s)]$  is a mapping from state space to action space that specifies an action to each of the states. Every policy is associated with a value function  $V_\pi(s)$  that gives the expected cumulative reward at the end of the deadline, when the system starts at state  $s$  and follows policy  $\pi$ . It can be recursively expressed as [37]:

$$V_\pi(s) = \bar{r}(s, a) + \sum_{s' \in \mathcal{S}(s, a)} \mathcal{P}_a(s, \acute{s}) V_\pi(s'), \quad \forall s \in \mathcal{S}. \quad (13)$$

Here,  $\mathcal{S}(s, a)$  is the set of successor states to state  $s$  when action  $a = \kappa(s)$  is taken following policy  $\pi(s)$ . The solution of a finite horizon MDP problem is an optimal policy  $\pi^*(s)$  at state  $s$  that maximizes the expected cumulative reward at the end of the finite number of time slots and can be defined as [37]:

$$\pi^*(s) = \arg \max_{a \in \mathcal{A}(s)} \{V_\pi(s)\}, \quad \forall s \in \mathcal{S}. \quad (14)$$

The optimal policy can be computed iteratively using the backward induction algorithm (BIA). From the modeling perspective, BIA requires to define all state-action transition probabilities and rewards of all transitions. From the computational perspective, it has complexity of  $O(|\mathcal{S}|^2|\mathcal{A}|)$ . Based on the sizes of  $\mathcal{S}$  and  $\mathcal{A}(s)$  described in our MDP formulation, we conclude that finding the optimal policy using BIA is computationally complex, especially for systems with large numbers of devices  $M$  and packets  $N$ . Therefore, in the following section, we design a low-complexity IDNC algorithm that can efficiently reduce the mean video distortion before the deadline.

## VI. TWO-STAGE MAXIMAL INDEPENDENT SET SELECTION ALGORITHM

In this section, we propose a two-stage maximal independent set (TS-MIS) selection algorithm that eliminates the need for using BIA (a dynamic programming approach) and reduces both modeling and computational complexities. This is a greedy approach since it selects an action in a given state without going through all the successor states. However, this approach follows the characteristics of our sequential decision making problem and reduces the mean video distortion at the end of the deadline. The main aspects of this approach are summarized as follows:

- We prioritize the critical devices over the non-critical devices in making decisions. If a non-critical device is ignored at the current time slot  $t$ , it is still possible to deliver all its missing packets in the remaining  $Q - 1$  time slots. On the other hand, a critical device already has a larger number of missing packets compared to the remaining time slots. Therefore, if a critical device is ignored at the current time slot  $t$ , it will receive a smaller subset of its missing packets at the end of the deadline.<sup>5</sup>
- To prioritize the critical devices, we partition the IDNC graph  $\mathcal{G}$  into critical graph  $\mathcal{G}_c$  and non-critical graph  $\mathcal{G}_a$ . The critical graph  $\mathcal{G}_c$  includes the vertices representing transmissions from all devices to the critical devices. Similarly, the non-critical graph  $\mathcal{G}_a$  includes the vertices representing transmissions from all devices to the non-critical devices.
- It may not be possible to deliver all the missing packets to the critical devices before the deadline due to their large numbers of missing packets. Consequently, we select a critical maximal independent set  $\kappa_c^*$  over critical graph  $\mathcal{G}_c$  that delivers the high importance packets to a subset of, or if possible, all critical devices.
- It is still possible to deliver all the missing packets to the non-critical devices before the deadline due to their small numbers of missing packets. Consequently, we select a non-critical maximal

<sup>5</sup>Note that a non-critical device at time slot  $t$  can become a critical device at the successor time slot  $t + 1$  and have a high priority compared to other devices.

independent set  $\kappa_a^*$  over non-critical graph  $\mathcal{G}_a$  that increases the probability of delivering all the missing packets to all non-critical devices before the deadline. However,  $\kappa_a^*$  is selected without violating the independent set constraint (thus, prohibiting coding and transmission conflicts) for the targeted critical devices in  $\kappa_c^*$ .

#### A. Maximal Independent Set Selection Algorithm over Critical Graph

In this sub-section, we select a critical maximal independent set  $\kappa_c^*$  over critical graph  $\mathcal{G}_c$  that minimizes the sum video distortion of all critical devices after the current time slot  $t$ . Let us define  $\mathcal{X}_c(\kappa_c)$  as the set of targeted critical devices in  $\kappa_c$  and  $d_k^{(t+1)}(\kappa_c)$  as the expected individual video distortion of critical device  $R_k \in \mathcal{C}(t)$  at time slot  $t+1$  due to selecting  $\kappa_c$ . This can be expressed as:

$$d_k^{(t+1)}(\kappa_c) = \begin{cases} d_k^{(t)} & \text{if } R_k \in \mathcal{C}(t) \setminus \mathcal{X}_c(\kappa_c), \\ d_k^{(t)} - \delta_{k,l}(1 - \epsilon_{i,k}) & \text{if } R_k \in \mathcal{X}_c(\kappa_c) : v_{i,kl} \in \kappa_c \end{cases} \quad (15)$$

Here, the first term represents the ignored critical device for which the distortion value will remain unchanged from time slot  $t$  to time slot  $t+1$ . The second term represents the expected distortion reduction in the targeted critical device from time slot  $t$  to time slot  $t+1$ . Let  $D^{(t+1)}(\kappa_c)$  be the sum of individual video distortion of all critical devices after time slot  $t$ . We now express the expected sum video distortion of all critical devices after time slot  $t$  as:

$$\begin{aligned} \mathbb{E}[D^{(t+1)}(\kappa_c)] &= \sum_{R_k \in \mathcal{C}(t)} \mathbb{E}[d_k^{(t+1)}(\kappa_c)] \\ &= \sum_{R_k \in \{\mathcal{C}(t) \setminus \mathcal{X}_c(\kappa_c)\}} d_k^{(t)} + \sum_{R_k \in \mathcal{X}_c(\kappa_c)} d_k^{(t)} - \delta_{k,l}(1 - \epsilon_{i,k}). \end{aligned} \quad (16)$$

We now formulate the problem of minimizing the sum video distortion of all critical devices as a critical maximal independent set  $\kappa_c^*$  selection problem over critical graph  $\mathcal{G}_c$  such that:

$$\begin{aligned}
\kappa_c^* &= \arg \min_{\kappa_c \in \mathcal{G}_c} \mathbb{E}[D^{(t+1)}(\kappa_c)] \\
&= \arg \min_{\kappa_c \in \mathcal{G}_c} \left\{ \sum_{R_k \in \{\mathcal{C}(t) \setminus \mathcal{X}_c(\kappa_c)\}} d_k^{(t)} + \sum_{R_k \in \mathcal{X}_c(\kappa_c)} d_k^{(t)} - \delta_{k,l}(1 - \epsilon_{i,k}) \right\} \\
&= \arg \max_{\kappa_c \in \mathcal{G}_c} \left\{ \sum_{R_k \in \mathcal{X}_c(\kappa_c)} \delta_{k,l}(1 - \epsilon_{i,k}) \right\}.
\end{aligned} \tag{17}$$

In other words, the problem of minimizing the sum video distortion of all critical devices is equivalent to finding the maximum weighted independent set in the critical graph  $\mathcal{G}_c$ . In this paper, we use the Bron-Kerbosch algorithm to find  $\kappa_c^*$  among all maximal independent sets in  $\mathcal{G}_c$  [38]. The complexity of the Bron-Kerbosch algorithm of a graph with  $|\mathcal{V}|$  vertices is  $O(3^{|\mathcal{V}|/3})$ .<sup>6</sup> In the following two sub-sections, we first derive the probability that the individual completion times of all non-critical devices meet the deadline and then select a non-critical maximal independent set  $\kappa_a^*$ .

### B. Probability that the Individual Completion Time Meets Deadline

At any given time slot  $t$ , we select a non-critical maximal independent set that increases the probability of delivering all missing packets to all non-critical devices before the deadline. To select such an independent set, we compute the probability that the individual completion times of all non-critical devices meet the deadline. The computation of this probability is simple since it is computed separately for each non-critical device and does not take into account the interdependence of devices' packet reception captured in the GSM. In fact, we trade-off some accuracy in calculation for much more computational simplicity.

To derive the probability, we first consider a special scenario with a single non-critical device  $R_k$  and assume that it is targeted with a new packet in each time slot. The probability of individual completion time  $T_{W_k}$  of device  $R_k$  being equal to  $W_k + x$ ,  $x \in [0, 1, \dots, Q - W_k]$  can be expressed using negative binomial distribution as:

$$\mathbb{P}[T_{W_k} = W_k + x] = \binom{W_k + x - 1}{x} (\bar{\epsilon}_k)^x (1 - \bar{\epsilon}_k)^{W_k}, \tag{18}$$

where,  $\bar{\epsilon}_k$  is the average of the channel erasure probabilities connecting device  $R_k$  to other devices. In other words,  $\bar{\epsilon}_k = \frac{\sum_{R_i \in \mathcal{I}} \epsilon_{i,k}}{|\mathcal{I}|}$ , where  $\mathcal{I} = \{R_i | y_{i,k} \neq 0, R_i \neq R_k\}$ . This average erasure probability

<sup>6</sup>To select a maximal independent set with much lower computational complexity, a greedy vertex search approach can be adopted following [21], which has a tolerable performance degradation.

represents that device  $R_k$  can receive its missing packets from any other neighboring device in the remaining time slots. Consequently, the probability that the individual completion time  $T_{W_k}$  of non-critical device  $R_k$  is less than or equal to the remaining  $Q$  time slots can be expressed as:

$$\mathbb{P}[T_{W_k} \leq Q] = \sum_{x=0}^{Q-W_k} \mathbb{P}[T_{W_k} = W_k + x]. \quad (19)$$

We now consider a scenario with a set of non-critical devices  $\mathcal{A}$  and assume that all non-critical devices are targeted with a new packet in each time slot. This is an ideal scenario and defines a lower bound on individual completion time of each non-critical device. Consequently, we can compute an upper bound on the probability that individual completion time of each non-critical device meets the deadline. However, this ideal scenario will not occur in practice since the transmitting devices cannot benefit from their own transmissions and the instant decodability constraint limits the number of targeted devices in each time slot. We can still use this probability upper bound as a metric in designing our computationally simple IDNC algorithms.

With the aforementioned ideal scenario, at any D2D time slot  $t$ , we can compute the upper bound on the probability that individual completion times of all non-critical devices in  $\mathcal{A}(t)$  are less than or equal to the remaining  $Q$  time slots (denoted by  $\hat{\mathbb{P}}^{(t)}[T_A \leq Q]$ ) as:

$$\hat{\mathbb{P}}^{(t)}[T_A \leq Q] = \prod_{R_k \in \mathcal{A}(t)} \sum_{x=0}^{Q-W_k} \mathbb{P}[T_{W_k} = W_k + x]. \quad (20)$$

In the following sub-section, we use expression (20) as a metric of selecting a non-critical maximal independent set in each time slot.

### C. Maximal Independent Set Selection Algorithm over Non-critical Graph

Once a critical maximal independent set  $\kappa_c^*$  is selected over critical graph  $\mathcal{G}_c$ , there may exist vertices belonging to the non-critical devices in non-critical graph  $\mathcal{G}_a$  that can form even a bigger maximal independent set. When the selected new vertices are non-adjacent to all vertices in  $\kappa_c^*$ , the corresponding non-critical devices are targeted without creating coding or transmission conflicts for the targeted critical devices in  $\kappa_c^*$ . Therefore, we first extract non-critical subgraph  $\mathcal{G}_a(\kappa_c^*)$  of vertices in  $\mathcal{G}_a$  that are non-adjacent to all the vertices in  $\kappa_c^*$  and then select non-critical maximal independent set  $\kappa_a^*$  over subgraph  $\mathcal{G}_a(\kappa_c^*)$ .

Let us define  $\mathcal{X}_a(\kappa_a)$  as the set of targeted non-critical devices in  $\kappa_a$  and  $W_k^{(t+1)}(\kappa_a)$  as the expected number of missing packets at a non-critical device  $R_k \in \mathcal{A}(t)$  at time slot  $t + 1$  due to

selecting  $\kappa_a$ . This can be expressed as:

$$W_k^{(t+1)}(\kappa_a) = \begin{cases} W_k^{(t)} & \text{if } R_k \in \mathcal{A}(t) \setminus \mathcal{X}_a(\kappa_a), \\ (W_k^{(t)} - 1)(1 - \epsilon_{i,k}) + (W_k^{(t)})(\epsilon_{i,k}) & \text{if } R_k \in \mathcal{X}_a(\kappa_a) : v_{i,k} \in \kappa_a \end{cases} \quad (21)$$

Here, the first term represents the ignored non-critical device for which the number of missing packets will remain unchanged from time slot  $t$  to time slot  $t + 1$ . The second term represents the targeted non-critical device for which the number of missing packets can be either  $W_k - 1$  with the packet reception probability  $(1 - \epsilon_{i,k})$  or  $W_k$  with the channel erasure probability  $\epsilon_{i,k}$ . With  $\kappa_a$  selection at time slot  $t$ , let  $\hat{\mathbb{P}}^{(t+1)}[T_A \leq Q - 1]$  be the resulting upper bound on the probability that individual completion times of all non-critical devices in  $\mathcal{A}(t)$ , starting from the successor time slot  $t + 1$ , are less than or equal to the remaining  $Q - 1$  time slots. We can express probability  $\hat{\mathbb{P}}^{(t+1)}[T_A \leq Q - 1]$  as:

$$\begin{aligned} \hat{\mathbb{P}}^{(t+1)}[T_A \leq Q - 1] &= \prod_{R_k \in \mathcal{X}_a(\kappa_a)} (\mathbb{P}[T_{W_k-1} \leq Q - 1] \cdot (1 - \epsilon_{i,k}) + \mathbb{P}[T_{W_k} \leq Q - 1] \cdot (\epsilon_{i,k})) \\ &\times \prod_{R_k \in \mathcal{A} \setminus \mathcal{X}_a(\kappa_a)} \mathbb{P}[T_{W_k} \leq Q - 1] \end{aligned} \quad (22)$$

In the first product, we compute the probability that a targeted non-critical device receives its  $W_k - 1$  or  $W_k$  missing packets in the remaining  $Q - 1$  time slots. Moreover, in the second product, we compute the probability that an ignored non-critical device receives its  $W_k$  missing packets in the remaining  $Q - 1$  time slots. We now formulate the problem of maximizing probability  $\hat{\mathbb{P}}^{(t+1)}[T_A \leq Q - 1]$  as a non-critical maximal independent set  $\kappa_a^*$  selection problem over non-critical subgraph  $\mathcal{G}_a(\kappa_c^*)$  such that:

$$\begin{aligned} \kappa_a^* &= \arg \max_{\kappa_a \in \mathcal{G}_a(\kappa_c^*)} \left\{ \hat{\mathbb{P}}^{(t+1)}[T_A \leq Q - 1] \right\} \\ &= \arg \max_{\kappa_a \in \mathcal{G}_a(\kappa_c^*)} \left\{ \prod_{R_k \in \mathcal{X}_a(\kappa_a)} (\mathbb{P}[T_{W_k-1} \leq Q - 1] \cdot (1 - \epsilon_{i,k}) + \mathbb{P}[T_{W_k} \leq Q - 1] \cdot (\epsilon_{i,k})) \right. \\ &\quad \times \left. \prod_{R_k \in \mathcal{A} \setminus \mathcal{X}_a(\kappa_a)} \mathbb{P}[T_{W_k} \leq Q - 1] \right\} \end{aligned} \quad (23)$$

In other words, the problem of maximizing probability  $\hat{\mathbb{P}}^{(t+1)}[T_A \leq Q - 1]$  is equivalent to finding all maximal independent sets in the non-critical subgraph  $\mathcal{G}_a(\kappa_c^*)$ , and selecting the maximal independent set among them that results in the maximum probability  $\hat{\mathbb{P}}^{(t+1)}[T_A \leq Q - 1]$ . Similar

---

**Algorithm 1:** Two-Stage Maximal Independent Set (TS-MIS) Selection Algorithm
 

---

**Construct** IDNC graph  $\mathcal{G}$  according to all LSMs  $\mathbf{F}_i, \forall R_i \in \mathcal{M}$ ;

Partition  $\mathcal{G}$  into  $\mathcal{G}_c$  and  $\mathcal{G}_a$  according to the critical and the non-critical devices;

Initialize  $\kappa_c^* = \emptyset$  and  $\kappa_a^* = \emptyset$ ;

**if**  $\mathcal{G}_c \neq \emptyset$  **then**

    | Select  $\kappa_c^* = \arg \max_{\kappa_c \in \mathcal{G}_c} \left\{ \sum_{R_k \in \mathcal{X}_c} \delta_{k,l} (1 - \epsilon_{i,k}) \right\}$  ;

**end**

Update subgraph  $\mathcal{G}_a(\kappa_c^*)$ ;

**if**  $\mathcal{G}_a(\kappa_c^*) \neq \emptyset$  **then**

    | Select  $\kappa_a^* = \arg \max_{\kappa_a \in \mathcal{G}_a(\kappa_c^*)} \left\{ \hat{\mathbb{P}}^{(t+1)}[T_A \leq Q - 1] \right\}$ ;

**end**

**Set**  $\kappa^* \leftarrow \kappa_c^* \cup \kappa_a^*$ ;

---

to Section VI-A, we use the Bron-Kerbosch algorithm to find  $\kappa_a^*$  among all maximal independent sets in  $\mathcal{G}_a(\kappa_c^*)$ .

The final maximal independent set  $\kappa^*$  is the union of two maximal independent sets  $\kappa_c^*$  and  $\kappa_a^*$  (i.e.,  $\kappa^* = \{\kappa_c^* \cup \kappa_a^*\}$ ). All the vertices in  $\kappa^*$  determines a set of transmitting devices. Each of the selected transmitting devices forms a coded packet by XORing the source packets identified by the vertices in  $\kappa^*$  representing transmission from that device. The proposed two-stage maximal independent set (TS-MIS) selection algorithm is summarized in Algorithm 1.

## VII. CALCULATIONS FOR PACKET IMPORTANCE OF A REAL VIDEO SEQUENCE

In this section, we first discuss the H.264/SVC video test sequence used in this paper and then provide details about the calculations for individual packet importance. We use a standard video sequence, *Soccer* [39]. This sequence is in common intermediate format (CIF, i.e.,  $352 \times 288$ ) and has 300 frames with 30 frames per second (fps). We encode the sequence using the JSVM 9.19.14 version of H.264/SVC codec [40], [41] while considering the temporal scalability of the video sequence.<sup>7</sup> The size of each group of pictures (GOP) is 8 frames, which results in 38 GOPs for the video sequence. As shown in Fig. 5, each GOP consists of a sequence of I, P and B frames that are encoded into four video layers. We use the identical shade to represent the frames of the same video layer and the darker shades to represent the more important video layers. Moreover, we use arrows to illustrate the dependency between frames in a GOP. The GOP shown in Fig. 5

<sup>7</sup>Note that our proposed IDNC framework is general and can be applied to a single layer H.264/AVC video sequence considered in [7], [26].

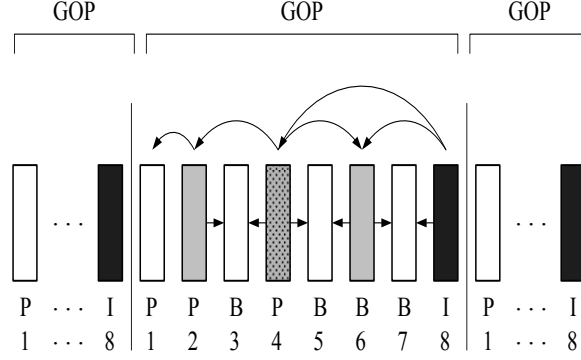


Fig. 5: A closed GOP with 4 layers and 8 frames (a sequence of I, P and B frames).

is a closed GOP, where the decoding of frames inside the GOP is independent of frames outside the GOP [42].

We use 1500 bytes as the packet length. This is the largest allowed packet over Ethernet. We allocate 1400 bytes for video information and the remaining 100 bytes for all the header information. Given the encoded I frame (i.e., the first layer) composed of  $\sigma$  bytes, the required number of packets for this frame and layer can be calculated as  $\lceil \frac{\sigma}{1400} \rceil$ . Here, the ceiling function  $\lceil \cdot \rceil$  represents the additional padding bits that are inserted into the last packet of the layer to make it 1500 bytes. The average number of packets in the first, second, third and fourth video layers over 38 GOPs are 8.35, 3.11, 3.29 and 3.43, respectively. This means on average 8.35 packets are required to decode the first layer, which consists of a single I frame. This frame is discarded at the devices if all the packets of this frame are not received before the deadline. For a GOP of interest, given that the number of frames per GOP is 8, the video frame rate is 30 frames per second, the transmission rate is  $\lambda$  bits per second and a packet length is  $1500 \times 8$  bits, the allowable number of total time slots for a GOP is fixed and can be computed as:  $\frac{8\lambda}{1500 \times 8 \times 30}$ .

In this paper, we use the average *peak-signal-to-noise ratio* (PSNR) as the performance metric for the video quality of our encoded video sequence *Soccer*. Similar to the work in [42], we obtain  $\alpha_{f_i, f_j}$  for  $1 \leq f_i, f_j \leq 300$ , which represents the PSNR if uncompressed  $f_i$  frame is replaced by compressed  $f_j$  frame. We calculate the average PSNR of each GOP, if the first  $\ell$  layers of four video layers are decodable ( $0 \leq \ell \leq 4$ ).<sup>8</sup> Moreover, the frames of the undecodable layers of the current GOP are replaced by the nearest frames in time of decodable layers of the current GOP or the previous GOP. This results in concealing the errors in the video sequence. For example, the

<sup>8</sup>Note that the  $\ell$ -th layer of a scalable video can be decoded only if all packets in the first  $\ell$  layers are received before the deadline.



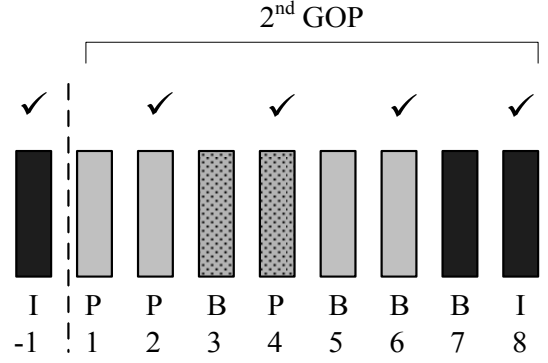


Fig. 6: The nearest decoded frames are used to conceal the loss of undecoded frames.

average PSNR of the second GOP can be calculated as:

$$\bar{\alpha}_2 = \frac{\sum_{f_i \in \mathcal{B}} \alpha_{f_i, f_i} + \sum_{f_i \notin \mathcal{B}} \alpha_{f_i, f_j}}{8} \quad (24)$$

where,  $\mathcal{B}$  is the set of frames of the decodable layers of the second GOP.

**Example 4.** *Let us consider the GOP shown in Fig. 5. We assume that the fourth layer of the second GOP is lost due to missing a packet of that layer at the end of the deadline. The resulting error concealment is shown in Fig. 6 and the resulting average PSNR can be computed as:*

$$\bar{\alpha}_2 = \frac{\alpha_{f_1, f_2} + \alpha_{f_2, f_2} + \alpha_{f_3, f_4} + \alpha_{f_4, f_4} + \alpha_{f_5, f_6} + \alpha_{f_6, f_6} + \alpha_{f_7, f_8} + \alpha_{f_8, f_8}}{8} \quad (25)$$

**Remark 1.** (*PSNR without Error*) *The average PSNR of the encoded Soccer sequence over 38 GOPs is 35.64 decibel (dB) if there is no error in the sequence.*

## VIII. SIMULATION RESULTS

In this section, we present the simulation results comparing the performance of the BIA that solves the formulated MDP problem and the TS-MIS algorithm to the following algorithms.

- ‘Fully Connected Distortion (FCD)’ algorithm [26] that considers a fully connected network and uses IDNC to minimize the mean video distortion in each time slot. This algorithm first determines the importance of individual packet according to its contribution to the overall video quality. It then selects a transmitting device and its XOR packet combination that minimizes the mean video distortion after the current time slot.
- ‘Partially Connected Blind (PCB)’ algorithm [27] that considers a partially connected network and uses IDNC to serve the maximum number of devices with any new packet in each time

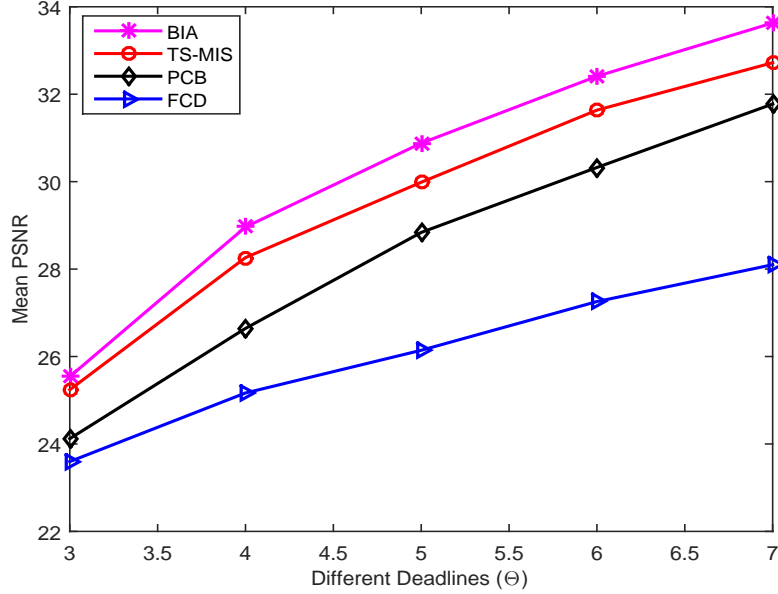


Fig. 7: Mean PSNR versus different deadlines  $\Theta$ .

slot. This algorithm selects a set of transmitting devices and their XOR packet combinations while ignoring the hard deadline and the unequal importance of video packets. This problem was addressed in [31] for a fully connected D2D network and in [20] for a PMP network.

We first consider a line network with  $M = 4$  devices described in (3) and encode four video layers of *Soccer* video sequence into four different packets, i.e.,  $N = 4$ . As discussed in Section V-C, the modelling and computational complexities of the BIA scale with the size of the state space  $|\mathcal{S}|$ , which is  $O(2^{16})$  even for  $M = N = 4$ . Moreover, as discussed in Section III, the central station uses the initial  $N$  time slots. Due to erasures in long-range wireless channels, at the beginning of the D2D phase, each device holds between 45% and 55% of  $N$  packets in all scenarios. Note that these percentages of initial received packets are arbitrary and reflect the erasures in long-range wireless channels.

**Definition 10.** (*Mean PSNR Calculation*) *The mean PSNR is calculated by taking average of the received PSNR at all  $M$  devices at the end of the deadline.*

Fig. 7 shows the mean PSNR achieved by different algorithms against the different number of allowable D2D time slots  $\Theta$  (i.e., different deadlines). From this figure, we can see that our proposed BIA and TS-MIS algorithms quickly increase the received PSNR at the devices with increasing the deadlines. Indeed, both BIA and TS-MIS algorithms use the new IDNC graph to make

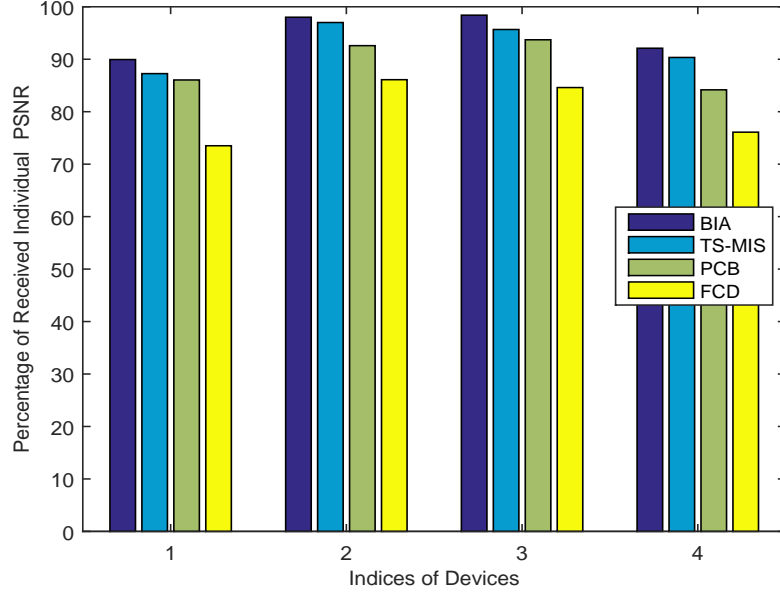


Fig. 8: Histogram showing the percentage of received PSNR at individual devices before the deadline.

coding and transmission conflict-free decisions and exploit the characteristics of a real-time video sequence. This figure also shows that the performance of the FCD and PCB algorithms considerably deviates from the BIA and TS-MIS algorithms. FCD algorithm selects a single transmitting device and its packet combination without exploiting the possibility of simultaneous transmissions from multiple devices. Moreover, FCD algorithm does not capture the aspects of the hard deadline and the channel erasures in making decisions. On the other hand, PCB algorithm exploits the possibility of simultaneous transmissions from multiple devices, but targets a large number of devices with any new packet in each time slot.

Fig. 8 shows the histogram obtained by different algorithms for the same line network (for  $M = N = 4$  and  $\Theta = 7$ ). This histogram illustrates the percentage of received PSNR before the deadline at individual devices separately. From this histogram, we can see that all devices receive an acceptable video quality at the end of the deadline (i.e.,  $\Theta = 7$  D2D time slots). Moreover, devices  $R_2$  and  $R_3$  experience a slightly better video quality compared to devices  $R_1$  and  $R_4$  since these are the intermediate devices in the line network shown in Fig. 2.

Having shown the performance of the BIA and TS-MIS algorithms for a simple line network, we now consider more general partially connected networks and show the performance of the TS-MIS algorithm. We use the *Soccer* video sequence discussed in Section VII, where the packet length is

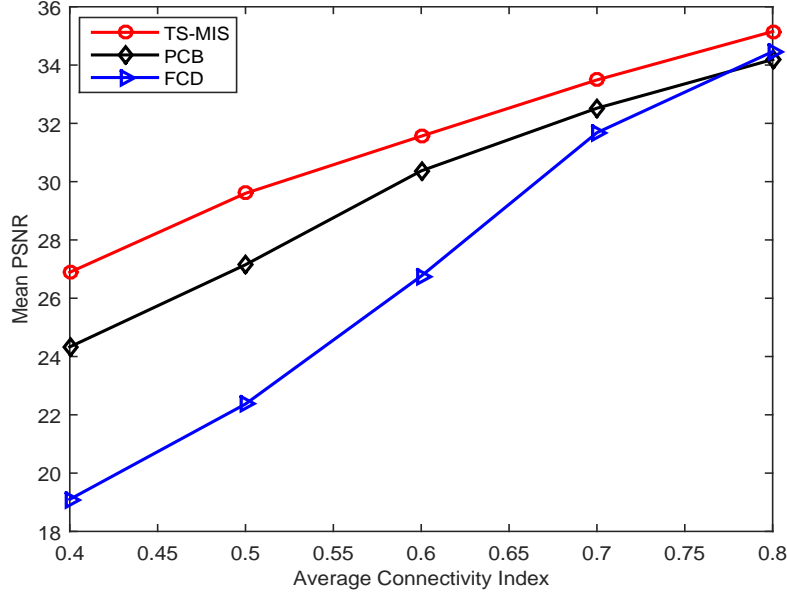


Fig. 9: Mean PSNR versus different average connectivity indices  $\bar{y}$ .

1500 bytes and each video layer is encoded into multiple packets. In SCM Y, if a pair of devices are directly connected, the packet reception probability over the channel is in the range  $[0.65, 0.9]$ . We compute the average connectivity index in the network as  $\bar{y} = \frac{\sum_{(R_i, R_k)} y_{i,k}}{M \times M}$ , which represents the average packet reception probability over all short-range channels. In the case of a fully connected network, the average connectivity index is  $\bar{y} = 0.8$ .

Fig. 9 shows the mean PSNR achieved by different algorithms against different average connectivity indices  $\bar{y}$  (for  $M = 15$  devices and  $\Theta = 17$  D2D time slots). From this figure, we can see that our proposed TS-MIS algorithm outperforms the FCD algorithm in all cases, even in the case of a fully connected network, i.e.,  $\bar{y} = 0.8$ . In fact, our proposed TS-MIS algorithm adopts a decision that not necessarily minimizes the mean video distortion after the current time slot but rather reduces the mean video distortion at the end of the deadline. Moreover, the decisions of the TS-MIS algorithm are adaptive to the number of remaining time slots. In particular, when the number of remaining time slots is large and all devices are non-critical devices, generally as in the case of the beginning of the D2D phase, the algorithm increases the probability of delivering all the packets to all devices. On the other hand, when the number of remaining time slots is small and all devices are critical devices, generally as in the case of the end of the D2D phase, the algorithm minimizes the mean video distortion after the current time slot. Finally, the algorithm mixes both decisions when some devices are critical devices and some are non-critical devices, in which case

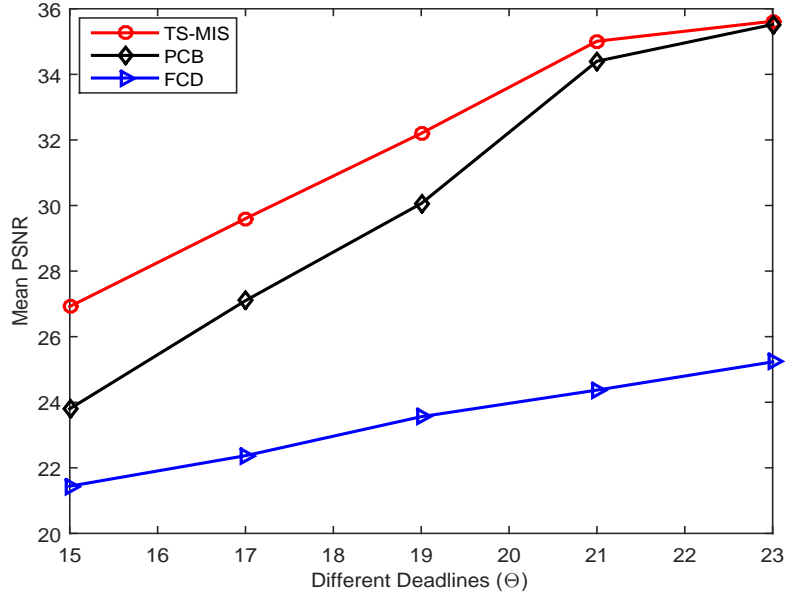


Fig. 10: Mean PSNR versus different number of allowable time slots  $\Theta$

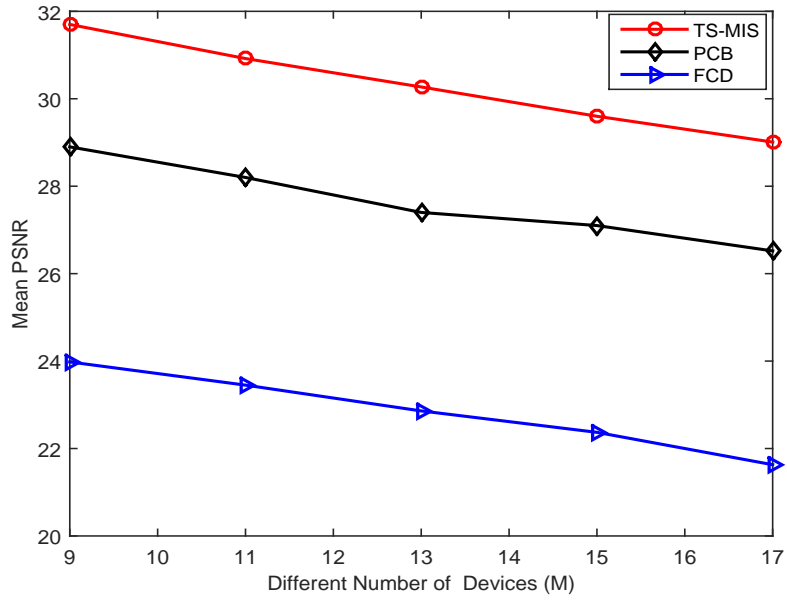


Fig. 11: Mean PSNR versus different number of devices  $M$

it prioritizes the critical devices since they will receive one less packet with each ignored time slot at the end of the deadline. From this figure, we can also see that the performance of the PCB algorithm considerably deviates from the TS-MIS algorithm since PCB algorithm does not address the hard deadline for the high importance video packets.

Fig. 10 and Fig. 11 show the mean PSNR achieved by different algorithms against different deadlines  $\Theta$  (for  $\bar{y} = 0.5$  average connectivity index and  $M = 15$  devices) and different number of devices  $M$  (for  $\bar{y} = 0.5$  average connectivity index and  $\Theta = 17$  D2D time slots), respectively. As expected, our proposed TS-MIS algorithm outperforms the FCD and PCB algorithms in all scenarios. In fact, our proposed TS-MIS algorithm makes decisions by taking into account the unequal importance of video packets, hard deadline, erasures of wireless channels, coding and transmission conflicts. Note that we have used another video sequence *Foreman* in the simulations and observed the similar results as in the case of *Soccer*.

## IX. CONCLUSION

In this paper, we developed an efficient IDNC framework for distributing a real-time video sequence to a group of cooperative wireless devices in a partially connected network. In particular, we introduced a novel IDNC graph that represents all feasible coding and transmission conflict-free decisions in one unified framework. Using the new IDNC graph and the characteristics of a real-time video sequence, we formulated the problem of minimizing the mean video distortion before the deadline as a finite horizon MDP problem. Since solving the formulated MDP problem was computationally complex, we further designed a TS-MIS selection algorithm that efficiently solves the problem with much lower complexity. Simulation results over a real video sequence showed that our proposed IDNC algorithms improve the received video quality compared to existing IDNC algorithms. Future research direction is to extend our proposed IDNC framework to a non-cooperative system, where the devices are selfish and pursue to minimize their individual video distortions before the deadline.

## REFERENCES

- [1] L. Al-Kanj, Z. Dawy, and E. Yaacoub, "Energy-aware cooperative content distribution over wireless networks: Design alternatives and implementation aspects," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1736–1760, 2013.
- [2] H. Khamfroush, P. Pahlavani, D. E. Lucani, M. Hundeboll, and F. H. Fitzek, "On the coded packet relay network in the presence of neighbors: Benefits of speaking in a crowded room," in *IEEE International Conference on Communications (ICC)*, 2014, pp. 1928–1933.
- [3] J. El-Najjar, H. M. AlAzemi, and C. Assi, "On the interplay between spatial reuse and network coding in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 2, pp. 560–569, 2011.
- [4] N. J. Hernandez Marciano, J. Heide, D. E. Lucani, and F. H. Fitzek, "On the throughput and energy benefits of network coded cooperation," in *IEEE 3rd International Conference on Cloud Networking (CloudNet)*, 2014, pp. 138–142.
- [5] N. Abedini, S. Sampath, R. Bhattacharyya, S. Paul, and S. Shakkottai, "Realtime streaming with guaranteed qos over wireless d2d networks," in *Fourteenth ACM international symposium on Mobile ad hoc networking and computing*, 2013, pp. 197–206.

- [6] D. Nguyen, T. Nguyen, and X. Yang, "Multimedia wireless transmission with network coding," in *IEEE Packet Video*, 2007, pp. 326–335.
- [7] H. Seferoglu and A. Markopoulou, "Video-aware opportunistic network coding over wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 5, pp. 713–728, 2009.
- [8] M. Yan and A. Sprintson, "Weakly secure network coding for wireless cooperative data exchange," in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2011, pp. 1–5.
- [9] M. H. Firooz and S. Roy, "Data dissemination in wireless networks with network coding," *IEEE Communications Letters*, vol. 17, no. 5, pp. 944–947, 2013.
- [10] Q. Zhang, J. Heide, M. V. Pedersen, and F. H. Fitzek, "MBMS with user cooperation and network coding," in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2011, pp. 1–6.
- [11] X. Wang, C. Yuen, T. Li, W. Song, and Y. Xu, "Minimizing transmission cost for third-party information exchange with network coding," 2014.
- [12] Z. Dong, X. Wang, S. H. Dau, and C. Yuen, "Delay minimization for relay-based cooperative data exchange with network coding," in *IEEE 78th Vehicular Technology Conference (VTC Fall)*, 2013, pp. 1–5.
- [13] W. Zeng, C. T. Ng, and M. Médard, "Joint coding and scheduling optimization in wireless systems with varying delay sensitivities," in *9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2012, pp. 416–424.
- [14] A. Tassi, C. Khirallah, D. Vukobratovic, F. Chiti, J. S. Thompson, and R. Fantacci, "Resource allocation strategies for network-coded video broadcasting services over lte-advanced," *IEEE Trans. Veh. Technol.*, vol. 64, no. 5, pp. 2186–2192, 2015.
- [15] N. Thomos, E. Kurdoglu, P. Frossard, and M. Van der Schaar, "Adaptive prioritized random linear coding and scheduling for layered data delivery from multiple servers," *IEEE Trans. Multimedia*, vol. 17, no. 6, pp. 893–906, 2015.
- [16] D. Vukobratovic and V. Stankovic, "Unequal error protection random linear coding strategies for erasure channels," *IEEE Trans. Commun.*, vol. 60, no. 5, pp. 1243–1252, 2012.
- [17] X. Li, C.-C. Wang, and X. Lin, "Optimal immediately-decodable inter-session network coding (idnc) schemes for two unicast sessions with hard deadline constraints," in *49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2011, pp. 784–791.
- [18] C. Zhan, V. C. Lee, J. Wang, and Y. Xu, "Coding-based data broadcast scheduling in on-demand broadcast," *IEEE Trans. Wireless Commun.*, vol. 10, no. 11, pp. 3774–3783, 2011.
- [19] L. Keller, E. Drinea, and C. Fragouli, "Online broadcasting with network coding," in *Fourth Workshop on Network Coding, Theory and Applications (NetCod)*, 2008, pp. 1–6.
- [20] A. Le, A. S. Tehrani, A. G. Dimakis, and A. Markopoulou, "Instantly decodable network codes for real-time applications," in *International Symposium on Network Coding (NetCod)*, 2013, pp. 1–6.
- [21] S. Sorour and S. Valaee, "Completion delay minimization for instantly decodable network codes," *IEEE/ACM Trans. Netw.*, 2014.
- [22] X. Li, C.-C. Wang, and X. Lin, "On the capacity of immediately-decodable coding schemes for wireless stored-video broadcast with hard deadline constraints," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 5, pp. 1094–1105, 2011.
- [23] M. Muhammad, M. Berioli, G. Liva, and G. Giambene, "Instantly decodable network coding protocols with unequal error protection," in *IEEE International Conference on Communications (ICC)*, 2013, pp. 5120–5125.
- [24] A. Al-Habob, S. Sorour, N. Aboutorab, and P. Sadeghi, "Conflict free network coding for distributed storage networks," in *IEEE International Conference on Communications (ICC)*, 2015, pp. 1–5.
- [25] Y. Shnaiwer, S. Sorour, T. Al-Naffouri, P. Sadeghi, and N. Aboutorab, "Network coded content delivery in femtochaching-assisted cellular networks," in *IEEE Global Communications Conference (GLOBECOM)*, 2015, pp. 1–5.

- [26] Y. Keshtkarjahromi, H. Seferoglu, R. Ansari, and A. Khokhar, "Content-aware instantly decodable network coding over wireless networks," in *International Conference on Computing, Networking and Communications*, 2015, pp. 1–5.
- [27] A. Douik, S. Sorour, T. Y. Al-Naffouri, H.-C. Yang, and M.-S. Alouini, "Delay reduction in multi-hop device-to-device communication using network coding," in *IEEE International Symposium on Network Coding (NetCod)*, 2015, pp. 1–5.
- [28] S. El Rouayheb, A. Sprintson, and P. Sadeghi, "On coding for cooperative data exchange," in *IEEE Information Theory Workshop (ITW)*, 2010, pp. 1–5.
- [29] A. Sprintson, P. Sadeghi, G. Booker, and S. El Rouayheb, "A randomized algorithm and performance bounds for coded cooperative data exchange," in *IEEE International Symposium on Information Theory Proceedings (ISIT)*, 2010, pp. 1888–1892.
- [30] N. Milosavljevic, S. Pawar, S. El Rouayheb, M. Gastpar, and K. Ramchandran, "Deterministic algorithm for the cooperative data exchange problem," in *IEEE International Symposium on Information Theory Proceedings (ISIT)*, 2011, pp. 410–414.
- [31] N. Aboutorab, P. Sadeghi, and S. E. Tajbakhsh, "Instantly decodable network coding for delay reduction in cooperative data exchange systems," in *IEEE International Symposium on Information Theory (ISIT)*, 2013, pp. 3095–3099.
- [32] M. S. Karim, N. Aboutorab, A. A. Nasir, and P. Sadeghi, "Decoding delay reduction in network coded cooperative systems with intermittent status update," in *IEEE Information Theory Workshop (ITW)*, 2014, pp. 391–395.
- [33] T. A. Courtade and R. D. Wesel, "Coded cooperative data exchange in multihop networks," *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 1136–1158, 2014.
- [34] M. Gonen and M. Langberg, "Coded cooperative data exchange problem for general topologies," in *IEEE International Symposium on Information Theory Proceedings (ISIT)*, 2012, pp. 2606–2610.
- [35] S. E. Tajbakhsh, P. Sadeghi, and N. Aboutorab, "Instantly decodable network codes for cooperative index coding problem over general topologies," in *IEEE 2014 Australian Communications Theory Workshop (AusCTW)*, 2014, pp. 84–89.
- [36] D. B. West *et al.*, *Introduction to graph theory*. Prentice hall Upper Saddle River, 2001, vol. 2.
- [37] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2009, vol. 414.
- [38] 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.
- [39] "Test video sequences (retrieved june 2014)." [Online]. Available: <ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/>
- [40] "Joint scalable video model (jsvm) reference software, version 9.19.14," 2011. [Online]. Available: <http://www.hhi.fraunhofer.de/en/fields-of-competence/image-processing/research-groups/image-video-coding/svc-extension-of-h264avc/jsvm-ref>
- [41] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h. 264/avc standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, 2007.
- [42] M. Esmaeilzadeh, P. Sadeghi, and N. Aboutorab, "Random linear network coding for wireless layered video broadcast: General design methods for adaptive feedback-free transmission," 2014. [Online]. Available: <http://arxiv.org/abs/1411.1841>