Instantly Decodable Network Coding: From Point to Multi-Point to Device-to-Device Communications

by

Mohammad Shahedul Karim

March 2017

A thesis submitted for the degree of Doctor of Philosophy
of The Australian National University

Declaration

The contents of this thesis are the results of original research and have not been submitted for a degree to any other university or institution. Much of the work in this thesis has been published or has been submitted for publication in journals or conference proceedings as listed below:

Journal Articles and Conference Proceedings:

- M.S. Karim, P. Sadeghi, S. Sorour and N. Aboutorab, *'Instantly Decodable Network Coding for Real-Time Scalable Video Broadcast'*, EURASIP Journal on Advances in Signal Processing, pp. 1-24, 2016.

- M.S. Karim, S. Sorour and P. Sadeghi, *'Network Coding for Video Distortion Reduction in Device-to-Device Communications'*, IEEE Transactions on Vehicular Technology, pp. 1-16, 2017.

- M.S. Karim, A. Douik, P. Sadeghi and S. Sorour, *'On Using Dual Interfaces with Network Coding for Delivery Delay Reduction'*, IEEE Transactions on Wireless Communications, pp. 1-15, 2017.

- M.S. Karim, A. Douik, S. Sorour and P. Sadeghi, *'Rate-Aware Network Codes for Completion Time Reduction in Device-to-Device Communications'*, IEEE International Conference on Communications (ICC), pp. $1-7$, May 2016.

The following articles are also the results of my doctoral study but not included in this thesis:

- M.S. Karim, M. Esmaeilzadeh and P. Sadeghi, *'On Reducing Intercept Probability for Unsubscribed Video Layers using Network Coding'*, IEEE Communications Letter, pp. 1-4, 2017.

- M.S. Karim, A. Douik, and S. Sorour, *‘ Rate-Aware Network Codes for Video Distortion Reduction in Point to Multi-point Networks’*, IEEE Transactions on Vehicular Technology, pp. 1-15, 2017.

- M.S. Karim, P. Sadeghi, N. Aboutorab and S. Sorour, *‘In Order Packet Delivery in Instantly Decodable Network Coded Systems over Wireless Broadcast’*, IEEE International Symposium on Network Coding (Netcod), pp. 11-15, June 2015.

- M.S. Karim, N. Aboutorab, A.A. Nasir, and P. Sadeghi, *‘Decoding delay reduction in network coded cooperative systems with intermittent status update’*, IEEE Information Theory Workshop (ITW), pp. 391-395, November 2014.

The research work presented in this thesis has been performed under the supervision of Associate Professor Parastoo Sadeghi (The Australian National University), Dr. Neda Aboutorab (The Australian National University), Assistant Professor Sameh Sorour (University of Idaho), and some work in collaboration with Graduate Student Ahmed Douik (California Institute of Technology) as evident from the author list above. The substantial majority of this work is my own.

Mohammad Shahedul Karim
Australian National University (u4868535), March 2017

Acknowledgements

The work presented in this thesis would not have been completed without the support of a number of individuals and organizations and they are sincerely acknowledged below:

- I would like to express my gratitude to my supervisors Associate Professor Parastoo Sadeghi, Dr. Neda Aboutorab, Professor Rodney A. Kennedy and Dr. Sameh Sorour for their cordial supervision, immeasurable guidance, constructive criticism, supportive suggestions, persistent encouragement and endless patience throughout the course of my doctoral study. I am also grateful to my supervisors for being great mentors and for training me to become an independent researcher. Their constant words of wisdom and motivation have inspired me to pursue complex and challenging problems and adopt a systematic approach to solves these problems.

- I would also like to express my special thanks to my colleagues Ahmed Douik and Mohammad Esmaeilzadeh for having many discussions on the research problems and providing me constructive feedback on the research findings.

- I would like to thank the Australian National University for providing me a PhD scholarship. I feel fortunate to get the opportunity for being in the signal processing group at the Research School of Engineering. I would like to thank everyone in the group including the administrative staff because of whom I felt the friendly and relaxing environment.

- I want to thank my parents and brothers who provided me support and motivation for completing my doctoral thesis. They have always been my inspirations whenever I encounter hardships in my studies.

Abstract

The network coding paradigm enhances transmission efficiency by combining information flows and has drawn significant attention in information theory, networking, communications and data storage. Instantly decodable network coding (IDNC), a subclass of network coding, has demonstrated its ability to improve the quality of service of time critical applications thanks to its attractive properties, namely the throughput enhancement, delay reduction, simple XOR-based encoding and decoding, and small coefficient overhead. Nonetheless, for point to multi-point (PMP) networks, IDNC cannot guarantee the decoding of a specific new packet at individual devices in each transmission. Furthermore, for device-to-device (D2D) networks, the transmitting devices may possess only a subset of packets, which can be used to form coded packets. These challenges require the optimization of IDNC algorithms to be suitable for different application requirements and network configurations.

In this thesis, we first study a scalable live video broadcast over a wireless PMP network, where the devices receive video packets from a base station. Such layered live video has a hard deadline and imposes a decoding order on the video layers. We design two prioritized IDNC algorithms that provide a high level of priority to the most important video layer before considering additional video layers in coding decisions. These prioritized algorithms are shown to increase the number of decoded video layers at the devices compared to the existing network coding schemes.

We then study video distribution over a partially connected D2D network, where a group of devices cooperate with each other to recover their missing video content. We introduce a cooperation aware IDNC graph that defines all feasible coding and transmission conflict-free decisions. Using this graph, we propose an IDNC solution that avoids coding and transmission conflicts, and meets the hard deadline for high importance video packets. It is demonstrated that the proposed solution delivers an improved video quality to the devices compared to the video and cooperation oblivious coding schemes.

We also consider a heterogeneous network wherein devices use two wireless interfaces to receive packets from the base station and another device concurrently. For such network, we are interested in applications with reliable in-order packet delivery requirements. We represent all feasible coding opportunities and conflict-free transmissions using a dual interface IDNC graph. We select a maximal independent set over the graph by considering dual interfaces of individual devices, in-order delivery requirements of packets and lossy channel conditions. This graph based solution is shown to reduce the in-order delivery delay compared to the existing network coding schemes.

Finally, we consider a D2D network with a group of devices experiencing heterogeneous channel capacities. For such cooperative scenarios, we address the problem of minimizing the completion time required for recovering all missing packets at the devices using IDNC and physical layer rate adaptation. Our proposed IDNC algorithm balances between the adopted transmission rate and the number of targeted devices that can successfully receive the transmitted packet. We show that the proposed rate aware IDNC algorithm reduces the completion time compared to the rate oblivious coding schemes.

Acronyms

IDNC	Instantly decodable network coding
D2D	Device-to-device
PMP	Point to multi-point
MDP	Markov decision process
RLNC	Random linear network coding
EW-IDNC	Expanding window IDNC
NOW-IDNC	Non-overlapping window IDNC
SVC	Scalable video codec
GOP	Group of pictures
SCM	Symmetric connectivity matrix
FSM	Feedback status matrix
PSNR	Peak signal to noise ratio
DI-IDNC	Dual interface IDNC graph
C-IDNC	Cooperation aware IDNC graph
BIA	Backward induction algorithm

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Preliminaries

1.1.1 Application Requirements

Today's wireless networks carry time critical and high quality applications ranging from video, news feeds, advertisements, social network services to file download [1]. Such diverse traffic imposes various application requirements on the resource constrained wireless networks in terms of packet delivery deadline, interdependence of packets, and importance values of individual packets. These application requirements in addition to lossy wireless channels create a number of challenges for delivering data traffic to users without degrading the quality of service. In the past decade, researchers have made substantial efforts to integrate the application requirements into the wireless communications technology and protocol development [2,3]. Similarly, in this thesis, we advocate for designing transmission schemes that are aware of application requirements and can meet the quality of service requirements of users using the scarce radio resources.

First, we study time-critical and order-insensitive applications that require fast and reliable decoding of the packets [4,5]. For such applications, a decoded packet can be immediately used by the application layer irrespective of its order and partial receiving of the content can improve the experience of the end users. For example, consider a wireless sensor network with numerous agents scattered in a region. Each agent requires to process multiple commands and each command is encapsulated in a packet. In this application, fast command execution are crucial for effective coordination of the agents and therefore, new packets are required to be arrived quickly and get decoded immediately at the agents regardless of their order. Further, some applications are designed focusing on order-insensitivity to make them adaptive to the unreliable transport medium. An example of such a case is using multiple-description coding [6], in which each decoded packet brings new information and contributes to the overall quality of service. Therefore, for time-critical and order-insensitive applications, it is crucial to design transmissions schemes that allow quick and reliable decoding of individual packets.

Second, we study time-critical and order-constrained applications that require quick and reliable in-order decoding of the packets [7–9]. For such applications, a packet can be used by the application layer only if that packet and all its preceding packets are decoded on time. For example, in cloud-based applications such as remote desktop control, Google Docs and Dropbox, packets usually represent instructions that need to be executed in order and on time. Besides, transmission control protocol (TCP) delivers packets in-order to the application layer. Therefore, out-of-order received packets can rapidly flood the devices' storage buffer.* Such applications require the current and future technologies to prioritize transmitted packets following their order. However, several works demonstrate that in-order packet transmission causes a loss in the rate of packet delivery to the users in wireless networks, i.e., the throughput [7, 8]. These works also demonstrate that the throughput loss incurred by in-order packet transmissions can be reduced by striking the right balance between in-order delivery of packets and throughput maximization. Therefore, for time-critical and order-constrained applications, it is crucial to analyze the interplay between throughput and in-order decoding of packets to facilitate the design of efficient transmissions schemes.

Third, video streaming applications in wireless devices is a big share of today's wireless data traffic and is expected to have a rapid growth in the foreseeable future [10]. Such growth leads to a heterogeneous video traffic which is encoded using a wide variety of standards, such as H.264, MPEG, divx, RealVideo, and WindowsMedia [11, 12]. For example, MPEG and H.264/AVC standards are employed in RealVideo and WindowsMedia codec to encode video sequences into a single layer with different spatial resolutions or signal-to-noise ratios.

*This thesis uses device, user, node and destination interchangeably to refer to an end terminal.

From the network designers' perspective, such a single layer video sequence has two distinct properties [11, 13–15]. 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. Clearly, the different video qualities can be provided by encoding a video sequence into different versions such that each version is encoded into a single layer with a fixed format. However, the limitations of creating multiple versions of one video sequence include an increased storage at the source server, proxy caches distributed throughout the network, and streaming multiple versions into the network for adapting to the variations in the available bandwidth.

Fourth, to overcome the drawbacks of a single layer video coding, scalable encoded video is a promising solution, which provides different video formats and qualities with one encoding [13, 16–18]. In fact, a large proportion of video traffic is expected to be scalable video, e.g., H.264/SVC extension of H.264/AVC, since it can provide heterogeneous quality of services to the users over wireless networks. In scalable video coding techniques, the video frames are compressed in the form of one base layer and several enhancement layers. The base layer provides the basic video quality, and each additional enhancement layer provides quality improvement. Such a scalable video has two distinct characteristics [18]. First, it has a hard deadline before which the video layers need to be decoded to be usable at the application. Second, the video layers exhibit a hierarchical order such that a video layer can be decoded only if this layer and all its lower layers are successfully received. Even though scalable video can tolerate the loss of one or more enhancement layers, this adversely affects the video quality experienced by viewers. Therefore, for layered video applications, it is crucial to analyse the progressive decoding of layers subject to a stringent deadline to facilitate the design of efficient transmission schemes.

Having briefly discussed the significance of integrating the application requirements into the design of all future technologies and protocols, this thesis advocates for application specific solutions to fully exploit the capabilities of the future technologies.

1.1.2 Network Configurations

Point to multi-point (PMP) networks have become a cornerstone in the design of traditional wireless standards and networks, such as satellite networks, cellular access networks, WiFi-based networks and cable networks [1, 19, 20]. In these networks, a number of users receive their required contents exclusively from a central access point possessing all the contents. An example of such scenarios is a group of users who want to stream a live soccer match from a base station. In this scenario, the base station needs to deliver all packets on-time in order to prevent interruption of the stream at the users. However, wireless channels are error prone due to fading, shadowing, multipath and interference, which causes a low packet delivery ratio [1]. In fact, the average packet loss rate in harsh network conditions can be as high as $20\% - 40\%$ [21, 22]. In addition to the lossy wireless channels, the proliferation of high quality applications creates a challenge for the service providers to meet the quality of service requirements of their subscribers using the limited network infrastructure and radio resources. For these constraints, it is important to analyze the transmission efficiency in wireless PMP networks and prevent severe degradation of the throughput, delay and quality of service.

To address an increasing throughput demand, communications and networking communities have recently gone beyond the traditional PMP networks thanks to the proliferation of smart devices with improved computational, storage and connectivity capabilities. In particular, device-to-device (D2D) networks have been introduced to allow local exchange of received packets at the devices, e.g., using Bluetooth and IEEE 802.11 adhoc mode [23–25]. The advantages of using such short range technology is multifold. First, it offloads the central station to serve additional devices and increases 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 are more reliable compared to long-range cellular channels because of smaller distances between the devices.

Depending on the geographic location of the devices, they can form a fully connected D2D network or a partially connected D2D network. In the former case, all close-by devices who are within transmission range of each other form a D2D network. Therefore, to avoid interference caused by simultaneous transmissions, a single device is allowed to transmit in each time slot. In the latter case, a group of devices, who are not necessarily close-by and are connected to each other via multiple hops, form a D2D network. Due to the spatial separation, a set of devices are allowed to transmit simultaneously if there is no destructive interference by others. For these cases, the transmitting devices and their transmitted packets need to be efficiently selected in D2D networks [26, 27]. Furthermore, a device may possess a subset of packets, which are required by some devices and unwanted by others. Indeed, these challenges need to be analysed and resolved in the packet and device selection processes in order to preserve the full benefits of devices' cooperations.

While many standards and networks are developed considering a device is connected to either a PMP network or a D2D network using its single interface, it is widely expected that the smart devices with multiple interfaces will be connected to multiple networks concurrently [28–30]. This leads to a heterogeneous network architecture with coexistence of cellular and D2D networks and an increase in the download rate at individual users as compared to a single interface network. In such scenarios, the smart devices use two wireless interfaces for simultaneous reception of packets from the base station using a long-range technology, e.g., LTE, and communication with nearby devices using a short-range technology, e.g., IEEE 802.11 adhoc mode [31–35]. Given that cellular and local area channels operate concurrently using different parts of the spectrum, such heterogeneous network offers significantly higher throughput, lower delay and better quality of service compared to the single interface based networks [25, 34, 35]. However, to fully preserve these benefits, a heterogeneous network needs to systematically select a transmitted packet for the base station, a set of transmitting devices and their transmitted packets while avoiding the service to a device with the same packet by multiple senders.

In anticipation of coexisting heterogeneous networks in future, this thesis advocates for network configuration aware standards and technologies to fully unlock their benefits such as robust scheduling, increased throughput, delay reduction, increased coverage zone and high reliability.

1.1.3 Network Codes

The fundamental principle of today's networks is that the sender forwards data and the information is processed only at the end nodes. Such principle considerably limits the information contained in a transmission and the number of devices benefitting from a transmission. This also increases the possibility of missing the throughput, delay and quality of service requirements of wireless users. Network coding is a relatively new paradigm that breaks this fundamental idea and advocates for combining several incoming packets at the sender or intermediate nodes of a network to generate outgoing packets [36–39]. In linear network coding [37], to retrieve the original packets at the devices, linear operations over a finite field are performed on the combined packets. Numerous works have demonstrated the ability of network coding to achieve a high transmission efficiency compared to the conventional uncoded transmissions. For example, random linear network coding (RLNC) [17, 40–44] have been studied extensively for efficient and low complexity data delivery in wireless networks and offering significant performance gains over conventional channel codes [45–49].

Another subclass of network coding, namely the instantly decodable network coding (IDNC), exploits the natural properties of overhearing and packet reception diversity in wireless networks to form an efficient packet combination [4, 5, 9, 50–54]. For the following advantages, IDNC has been widely considered for deployment in practical systems. First, IDNC aims to provide instant packet decodability upon successful packet reception at the devices, which allows a progressive recovery of the content as the devices decode more packets. Second, the encoding process of IDNC is performed using simple XOR operations, which reduces coefficient reporting overhead. Third, the decoding process of IDNC is also performed using XOR operations, which is suitable for implementation in simple and cost-efficient devices [53, 55–58].

Even though IDNC provides a good trade-off among throughput, complexity and delay, the full potential of IDNC has not yet been unlocked. Due to the instant decodability constraint, it may not always serve all devices with a specific new packet in each transmission. Therefore, for applications with specific requirements discussed in Section 1.1.1, IDNC needs to integrate the applications features in coding decisions [4, 15]. Furthermore, in some network configurations such as D2D networks, a set of transmitting devices and their packet combinations need to be jointly determined to take full advantages of devices' cooperations and network coding [26,26]. In fact, there is a gap between the existing theory of IDNC and their direct utilization to various applications and network configurations. As a result, this thesis aims to analyze, design and optimize IDNC algorithms by addressing the challenges raised by a wide range of application requirements and network configurations. In particular, we are interested in developing IDNC frameworks by taking into account numerous application and network properties, such as layers of video, order of packets, unequal importance of packets, hard deadline, lossy wireless channels, and coding and transmission conflicts, which will be elaborated in the rest of this thesis.

Furthermore, similar to the most of the works on IDNC in literature, this thesis initially considers an abstraction of physical channel conditions such as a transmitted packet is either received or lost with an average erasure probability. Therefore, all transmissions are assumed to have a fixed transmission rate and require a fixed duration of time. Such channel model simplifies the analysis, design and optimization of IDNC solutions and facilitates the demonstration of its benefits in a variety of application requirements and network configurations. However, at the end of this thesis, we model physical channels with heterogeneous capacities, which requires intelligent selection of the transmission rate by the senders. For such scenarios, a packet transmission with a high rate will take a shorter time but will be only successfully received by the devices with high channel capacities. Therefore, to fully exploit the channel capabilities, the final chapter of this thesis demonstrates the significance of designing transmission rate aware IDNC solutions.

1.2 Thesis Contributions

The contributions of this thesis are the analysis, design and optimization of IDNC frameworks by taking into account a wide range of relevant application requirements and network configurations. In particular, we make the following four contributions:

- We study layered video aware IDNC for reducing the number of uncoded video layers before the deadline in PMP networks.

- We study content aware IDNC for reducing the video distortion in D2D networks.

- We study packet order aware IDNC for reducing the delivery delay in heterogeneous networks with coexistence of cellular and D2D networks.

- We study completion time aware IDNC for reducing the packet recovery time in D2D networks.

In the following subsections, we elaborate the aforementioned four contributions.

1.2.1 Layered Video Aware Network Codes in PMP Networks

A recent report from Cisco [10] shows that cellular data traffic is doubling in each year and the share of video traffic is expected to grow from 55% in 2014 to 80% in 2019. This continuous growth of video traffic in addition to the limited radio resources create a number of challenges for the service providers to meet the quality of service requirements for their subscribers [59]. Within this large problem space, in this study, we focus on the problem of broadcasting a scalable video sequence from a base station to a set of devices subject to a bandwidth constraint. Such scalable coding technique compresses video frames in the form of one base layer and several enhancement layers, and imposes order and deadline constraints on decoding the layers [18]. Even though scalable video can tolerate the loss of one or more enhancement layers caused by the lossy wireless channels and limited radio resources, this adversely affects the video quality experienced by viewers.

In this study, we are interested in designing an IDNC framework that maximizes the minimum number of decoded video layers over all devices before the deadline (i.e., improves fairness in terms of the minimum video quality across all devices) in wireless PMP networks. Such framework needs to carefully balance transmitted packet selection only from the base layer versus transmitted packet selection from all video layers. While the former guarantees the highest level of priority to the base layer, the latter increases the possibility of decoding a larger number of video layers before the deadline. In this context, we first derive an upper bound on the probability that the individual completion times of all devices meet the deadline. Using this probability, we design two prioritized IDNC algorithms that provide a high level of priority to the most important base layer before considering additional enhancement layers in coding decisions. Moreover, in these algorithms, we select an appropriate packet combination over a given number of video layers so that these video layers are decoded by the maximum number of devices before the deadline. We formulate this packet selection problem as a two-stage maximal clique selection problem over an IDNC graph. Simulation results over a real scalable video sequence show that our proposed algorithms improve the received video quality compared to the existing IDNC algorithms. These contributions are detailed in Chapter 3, and the results are published in [60].

1.2.2 Content Aware Network Codes in D2D Networks

The rapid growth of video streaming applications in today's networks leads to a heterogeneous video traffic that is encoded using a wide variety of techniques, such as H.264, MPEG, divx, RealVideo, and WindowsMedia [12]. A big share of these traffic is encoded using a single layer coding technique, e.g., H.264/AVC [11, 15]. Such encoded video has unequally important packets towards the video quality and a hard deadline before which the video packets need to be decoded. To deliver a high quality video to the users and address the resource limitations of cellular networks, a promising solution is D2D communications wherein devices exchange their partially possessed content with others using a short-range wireless technology, e.g., IEEE 802.11 adhoc mode [31–35]. Furthermore, due to the spatial separation, a device can communicate to other devices directly (i.e., single-hop transmission) or via intermediate devices (i.e., multi-hop transmissions). In fact, multiple devices are allowed to transmit concurrently in the D2D network given there is no destructive collision due to transmissions by others [61].

In this study, we are interested in designing a content aware IDNC framework that minimizes the mean video distortion before the deadline in D2D networks. Such framework needs to select a set of transmitting devices and their packet combinations by taking into account the unequal importance of video packets, hard deadline, erasures of wireless channels, and coding and transmission conflicts. In this context, we first introduce a cooperation aware IDNC graph that represents both coding and transmission conflicts of a D2D network with one common transmission channel. In fact, this graph representation has to account for the coverage zones of different devices, potential collisions over the common channel from simultaneous transmissions of multiple devices and the constraint that each device cannot transmit and receive concurrently. 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. We also design a greedy two-stage maximal independent set selection algorithm, which has much lower modelling and computational complexities compared to the MDP formulation. Simulation results show that our proposed IDNC algorithms improve the received video quality compared to the IDNC algorithms in [26, 52, 56]. These contributions are detailed in Chapter 4, and the results are published in [62].

1.2.3 Packet Order Aware Network Codes in Heterogeneous Networks

To address a growing traffic load, it is widely expected that next generation networks will be heterogeneous in architecture, with coexistence of cellular and D2D networks, thanks to the proliferation of the smart devices with the improved connectivity capabilities [25,29,34]. In such networks, devices use two wireless interfaces to receive packets from the base station using a long-range technology, e.g., LTE, and to transmit or receive packets from other devices using a short-range technology, e.g., IEEE 802.11 ad-hoc mode [24,31]. A big share of today's wireless traffic comes from time-critical and order-constrained applications requiring quick and reliable in-order decoding of the packets. For such applications in heterogeneous networks, it is crucial to prioritize in-order packet transmissions at the senders without incurring a significant loss in the throughput [8].

In this study, we aim to develop an efficient IDNC framework for order-constrained applications in heterogeneous networks. Such framework needs to select a packet combination for the base station, a transmitting device and its packet combination by taking into account the order of packets, erasure of wireless channels and dual interfaces of devices. In this context, we first introduce the delivery delay metric as a measure of degradation compared to the optimal in-order packet delivery to the devices. We then define the dual interface IDNC graph, where a vertex represents a possibility for a transmitting device to send a new packet to another receiving device, and an edge represents a coding or simultaneous transmission conflict. Subsequently, we show that the minimum delivery delay problem is equivalent to a maximum weight independent set selection problem over the dual interface IDNC graph, in which the weight of a vertex represents the expected delivery delay. Given the computational hardness of finding the optimal solution, we further propose a delivery delay reduction heuristic based on a greedy vertex search. Simulation results demonstrate that the proposed IDNC algorithm effectively reduces the delivery delay compared to the existing network coding algorithms in [25, 63]. These contributions are detailed in Chapter 5, and the results have been submitted for publication in [64].

1.2.4 Completion Time Aware Network Codes in D2D Networks

In the aforementioned three studies, we model the condition of physical layer channels by erasure probabilities. Therefore, all transmissions are assumed to have a fixed physical-layer rate and require a fixed duration of time. On the contrary, the wireless channels are independent, heterogeneous, and dynamic in nature, which requires intelligent selection of the transmission rate by the senders. Furthermore, a packet transmission with a high rate will take a shorter time but will be only successfully received by the devices with sufficiently high channel capacities. In contrast, a packet transmission with a lower rate is expected to be received by more devices but will take a longer time. Therefore, a balance among these conflicting effects is necessary to fully exploit the channel capabilities and improve the quality of service of applications.

In this study, we are interested in time-critical applications, in which each packet is immediately used at the application layer and must be delivered on-time. For such time-critical applications, similar to the previous three studies, we adopt IDNC for enhancing the packet transmission efficiency and expediting the packet delivery process. Furthermore, to reduce the consumption of cellular resources and meet the devices' demands, we consider a fully connected D2D network wherein devices cooperate with each other to recover their missing packets. Due to the spatial closeness and interference, a single device is allowed to transmit in each transmission in a fully connected D2D network.

In this context, we aim to design a completion time aware IDNC framework that minimizes the overall time required for recovering all missing packets at the devices in D2D networks. In particular, we first introduce a rate aware IDNC graph that defines all feasible rate and coding decisions for all potential transmitting devices. Each maximal clique of this graph represents a transmitting device, a transmission rate and an XOR packet combination. Using the rate aware graph and the properties of the optimal schedule, we design a completion time reduction heuristic that jointly selects a transmitting device, a transmission rate and an XOR packet combination. Moreover, this heuristic strikes a balance

Table 1.1: Summary of the contributions in different chapters.

	Application characteristics	Network configurations	Channel characterization	Results
Chapter 3	Layered video, deadline	PMP	Erasure probability	Enhanced the number of delivered video layers.
Chapter 4	Unequally important video packets, deadline	D2D	Erasure probability	Improved video quality.
Chapter 5	Reliable in-order packet delivery	Heterogeneous with coexistence of PMP and D2D	Erasure probability	Reduced in-order delivery delay.
Chapter 6	Quick any-order packet delivery	D2D	Channel capacities	Reduced completion time.

between the transmission rate and the number of targeted devices with a new packet in each transmission. Simulation results show that our proposed IDNC algorithm reduces the completion time compared to the network coding algorithms of [65, 66] for D2D networks. These contributions are detailed in Chapter 6, and the results are published in [67].

1.2.5 Summary of the Contributions

Table 1.1 summarizes the contributions of this thesis in terms of application requirements, network configurations, channel characterizations and results.

Although Table 1.1 shows that the proposed IDNC solutions of this thesis consider particular applications and networks, we believe that the developed analysis and optimization techniques in this thesis can be used as references for studying other applications and networks.

Chapter 2

Background and Related Work

In this chapter, we present the related works to the problems of interest in this thesis.

2.1 Network Coding

The network coding paradigm has emerged from the pioneering work in [36] and advocates for the source and intermediate nodes to perform linear operations on incoming data packets to generate outgoing data packets. In linear network coding [37], which is an important subclass of network coding, to recover the original packets from the coded packets, a set of linear equations over a finite field are solved at the devices. In multicast networks, network coding can achieve the min-cut capacity of the network to each destination [37–39].

Although network coding was originally proposed for wired networks, this breakthrough idea has inspired significant effort in practical application of network coding to wireless networks [17, 40–44]. Indeed, the broadcast nature of the wireless medium offers an opportunity for exploiting the throughput benefits of network coding. Inspired by the throughput gain in wired networks and the broadcast nature of wireless medium, a large body of works have studied network coding for wireless networks and demonstrated its benefits in increasing throughput, simplifying scheduling, reducing delay, and enhancing security [15, 32, 34, 53, 55–58, 68, 69]. To illustrate the basic concept of network coding, let us consider the following wireless example.

Example: Fig. 2.1 illustrates a bidirectional wireless topology, connecting A-R-B nodes with channel capacities are one packet per time slot. In this scenario, node B possesses packet

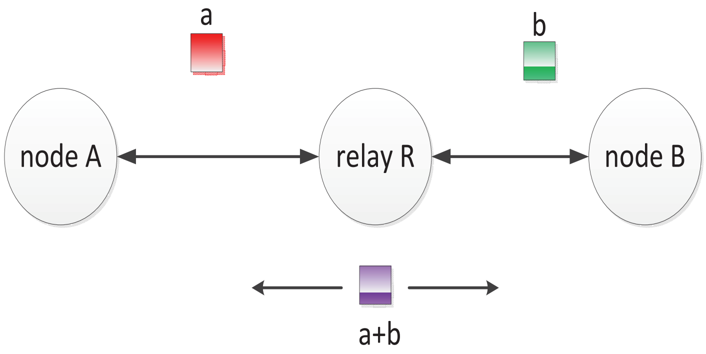


Figure 2.1: A wireless A-R-B topology with three time slots

'b' and node A possesses packet 'a'. Furthermore, node A wants to transmit its packet 'a' to node B, whereas node B wants to transmit its packet 'b' to node A via the relay node R. Nodes A and B transmit their packets in two time slots, and relay node R receives both packets. In the third time slot, node R broadcasts the coded packet $a \oplus b$ *to nodes A and B. Since nodes B and A already possesses 'b' and 'a', they decode their missing packets 'a' and 'b', respectively by re-XORing the coded packet. Here, relay node R transmits coded packet* $a \oplus b$ *in one time slot instead of packets a, b in two time slots separately. As a result, network coding reduces the number of required transmissions from four to three, and enhances throughput by* 33.3%.

Random linear network coding (RLNC) and instantly decodable network coding (IDNC), which are two attractive subclasses of network coding, have been studied extensively [17, 41–44, 52, 53, 70] for their abilities of simple extension to general networks and providing better trade-offs among bandwidth efficiency, complexity and delay. RLNC generates coded packets by selecting coefficients from a sufficiently large finite field and algebraically mixing the source packets. For a large number of source packets, the authors in [71] showed that RLNC can achieve the broadcast capacities of a wireless network. In [72], the authors studied the delay and throughput gains of RLNC in wireless PMP networks given transmitted packets are subject to channel erasures. In [73], RLNC was employed in the context of time division duplex with the aim of reducing the number of transmissions required for broadcasting a set of packets over wireless networks. As for practical evaluation of RLNC performance, [74] proposed MAC-independent opportunistic routing and encoding (MORE) and examined both unicast and multicast transmissions over wireless mesh networks. MORE allowed intermediate nodes to randomly mix packets together before forwarding them to other nodes. In fact, MORE was implemented between the IP and MAC layers and was shown to achieve throughput gain of 22-45% for unicast scenarios and 35-200% for multicast scenarios over conventional uncoded transmission schemes. However, in RLNC systems, the devices need to collect a sufficient number of independent coded packets before decoding all the original packets simultaneously. Therefore, RLNC is more attractive to the delay tolerant applications that uses a block of packets simultaneously. To provide different levels of protection to the coded bit streams of different significance, the authors in [75, 76] proposed power and bandwidth-efficient coded modulation scheme. In particular, a superposition coded modulation scheme was designed by using shaping techniques that reduce the interference between the fine-level code and the coarse-level code. The numerical results demonstrated that the superposition based scheme outperforms those without superposition coding by about 3 dB.

On the other hand, IDNC is a promising solution for time-critical applications requiring progressive decoding of packets at the devices. As discussed in Section 1.1.3, it can offer a low decoding delay by using simple XOR-based encoding and decoding operations [55, 77]. In addition to the delay reduction, in [53, 70], the authors showed that IDNC can achieve the optimal throughput for the two-device or three-device networks. In [4, 77], the authors employed IDNC to minimize the number of transmissions required for broadcasting a set of packets to a large number of devices. The throughput performance of the IDNC algorithms in [4, 77] was shown to be comparable to that of throughput optimal RLNC, while providing all the benefits of IDNC. Inspired by applications using each decoded packet immediately, in [5, 52, 70, 78], the authors adopted IDNC for wireless broadcast of a set of packets and served the maximum number of devices with a new packet in each transmission. All the aforementioned works showed that finding the optimal solution of the IDNC problem is NP-hard and therefore, proposed low-complexity heuristics by exploiting the properties of the optimal solution. Similarly, in this thesis, we aim to design simple and efficient IDNC frameworks by taking into account the application requirements and the network configurations discussed in Chapter 1.

Comprehensive reviews on RLNC and IDNC theory and applications can be found in [79, 80].

2.2 Layered Video Aware Network Codes in PMP Networks

In this section, we discuss the network coding schemes designed for layered video transmissions over PMP networks. A body of works [81–84] showed that combining the layered approach with network coding provides unequal error protection to different importance layers and improves the quality of video delivered to the users. These works used expanding window based RLNC strategies to form coded packets across different numbers of video layers and include the packets in the lower video layers into all coded packets. In particular, the authors in [43] used a probabilistic optimization approach for selecting coding windows and ensuring high decoding probabilities for the important layers. In [44], the authors considered a scalable video transmission with a hard deadline and used a deterministic optimization approach for selecting coding windows over all transmissions before the deadline.

For a layered video broadcast over PMP networks, the related IDNC works are [9, 85]. In [85], the authors proposed IDNC algorithms prioritizing a set of packets forming the base layer compared to another set of packets forming the enhancement layers. Similarly, in [9], the authors discussed the hierarchical order of video layers with motivating examples and proposed a heuristic IDNC algorithm. The heuristic in [9] struck a balance between the number of transmissions required for delivering the base layer and the number of transmissions required for delivering all video layers. However, both works [9, 85] ignored the hard deadline and did not strictly prioritize to deliver the base layer packets before the deadline in harsh network conditions. This is the scope of the study in Chapter 3.

2.3 Content Aware Network Codes in D2D Networks

In this section, we discuss the content aware network coding schemes in D2D networks. The works in [32, 35, 65, 86, 87] have recently illustrated the throughput benefits of using both network coding and device cooperation by jointly selecting a set of transmitting devices and their packet combinations. In particular, the authors in [86] analyzed algebraic network coding for D2D communications and provided upper and lower bounds on the number of time slots required for recovering all the missing packets at the devices. In [65], the authors proposed a randomized packet selection algorithm that has a high probability of achieving the minimum number of time slots. However, the works in [65, 86, 87] neither considered erasure channels nor considered addressing the hard deadline for high importance video packets.

Several other works including [56,88,89] adopted IDNC for time-critical applications and D2D networks. In [88, 89], the authors selected a transmitting device and its XOR packet combination to serve a large number of other devices with a new packet in each time slot. Inspired by the video streaming applications with hard deadline, [56] prioritized packets according to their contributions to the video quality and proposed a joint device and packet selection algorithm that maximizes the overall video quality in the current time slot. The aforementioned works [56,65,86–89] developed network coding schemes for a fully connected D2D network, wherein all devices are directly connected to other devices and a single device is allowed to transmit in each time slot.

To study network coding for more general scenarios, a body of works considered partially connected D2D networks wherein a device is connected to another device via a single hop or multiple hops [26, 27, 90, 91]. In particular, the authors in [27] studied algebraic network coding with large finite fields and provided various necessary and sufficient conditions to characterize the number of transmissions required for recovering all missing packets at all devices. The authors in [90] continued the work in [27] and showed that solving the minimum number of transmissions problem exactly or even approximately is computationally intractable. However, the works in [27, 90] did not consider erasure channels, XOR based network coding, explicit packet delivery deadline and unequal importance of video packets. For time-critical applications and simple implementation of network coding, the works in [26, 91] adopted IDNC and served a large number of devices with a new packet in each time slot. However, the works in [26, 91] are not readily usable for the transmission of a real-time video sequence that has a hard deadline and unequally important video packets. This is the scope of the study in Chapter 4.

2.4 Packet Order Aware Network Codes in Heterogeneous Networks

In this section, we discuss the network coding schemes designed for in-order packet delivery to the devices in heterogeneous networks. For time-critical and order-constrained applications, the authors in [9] proposed IDNC algorithms that allow in-order packet transmissions to the devices in PMP networks. The authors demonstrated that the order-aware IDNC algorithms provide quicker in-order packet delivery to the devices compared to the order-oblivious IDNC algorithms. In [63, 92], the authors extended the study of [9] by providing more efficient graph-based IDNC algorithms for ordered packet delivery. Nevertheless, these works [9,63,92] considered a single interface scenario wherein each device is connected to a single network and receives at most one packet in a time slot.

Several other works including [25, 59] demonstrated the performance benefits of using network coding and multiple interfaces of devices connected to multiple networks. In particular, to improve the download rate at individual devices, the authors in [59] exploited multiple interfaces with RLNC. The benefits of their proposed solution are attested by implementing the system on a testbed consisting of seven Android phones. In [25], the authors considered a heterogeneous network with coexistence of cellular and D2D networks, and employed both RLNC and IDNC to reduce the number of time slots required for broadcasting a set of packets. However, the work [25] neither considered applications with in-order packet delivery constraint nor integrated lossy channel characteristics into coding decisions. This is the scope of the study in Chapter 5.

2.5 Completion Time Aware Network Codes in D2D Networks

In this section, we discuss the completion time aware network coding schemes for both PMP and D2D networks. For time-critical and order-insensitive applications, the authors in [93,94] employed IDNC to minimize the completion time required for broadcasting a set of packets over wireless PMP networks. The authors used Markov decision process frameworks to formulate the optimization problem and show the computational intractability of the optimal solution. Furthermore, the authors designed an online heuristic by analysing the properties of the optimal formulation. On the other hand, the authors in [88,89] extended the works in [93,94] to fully connected D2D networks. In particular, the authors suggested jointly selecting the transmitting device and XOR packet combination to serve a large number of devices with a new packet in each D2D transmission. However, most of the works on IDNC in literature, including the aforementioned works in [88, 89, 93, 94] considered that coding is performed at the network layer and modeled the condition of physical layer channels by erasure probabilities. Therefore, all transmissions are assumed to have a fixed physical-layer rate and require a fixed duration of time. On the contrary, the wireless channels are dynamic with heterogeneous capacities.

The completion time and physical-layer rate aware IDNC framework proposed in this study is related to the works in [95, 96]. The authors in [95] showed that transmission rate aware IDNC decisions are more effective in reducing the completion time compared to transmission rate oblivious IDNC decisions. Concurrently, the authors in [96] considered each packet is associated with a sequential packet delivery deadline and proposed an IDNC algorithm that jointly determines the packet combination and the transmission rate. However, these works considered PMP networks with a single central transmission unit and therefore, are not readily applicable to D2D networks. This is the scope of the study in Chapter 6.

Chapter 3

Layered Video Aware Network Codes in PMP Networks

3.1 Overview

Providing high quality live video over wireless networks is a challenging problem due to the lossy wireless channels and the large bandwidth requirements of video traffic. Within this large problem space, we focus on the efficient broadcast of a scalable video over wireless PMP networks using network coding. Such a scalable video imposes an order constraint on decoding the layers and has a hard deadline before which the video layers need to be decoded [13]. Therefore, using the scarce radio resources, as many video layers as possible need to be in-order delivered to the viewers before the deadline. In fact, the video coding community advocates for taking into account the interdependence of video layers and the hard deadline in designing a video-aware transmission scheme. On the other hand, network coding community has well established that network coding offers an increased throughput, simple scheduling, delay reduction and enhanced security in wireless PMP networks [40–43, 52]. This chapter bridges the gap between these two techniques and develops a unified video-aware network coded scheme that improves both video quality and throughput.

Our work of this chapter is inspired by the recent works on scalable video transmission using RLNC in [17, 43, 44, 97]. In this chapter, we adopt XOR based IDNC to investigate its performance for scalable video transmission. In addition to the inherent encoding and decoding simplicity, the throughput performance of IDNC closely follows that of RLNC in a network with a small number of devices [93]. In particular, IDNC schemes can achieve the optimal throughput for the two-device or three-device system as shown in [53,70]. Moreover, as video streaming applications continue to proliferate, the wireless sender (e.g., the base station) often needs to support multiple simultaneously running applications with heterogenous video characteristics. Therefore, the sender can adopt IDNC to encode packets from different video sequences together. This allows immediate decoding of the received packets of different video sequences and immediate use of decoded packets at the applications, especially when one or more video sequences are encoded using multiple description coding in addition to scalable video coding.

In this chapter, we are interested in designing an efficient IDNC framework that maximizes the minimum number of decoded video layers over all devices before the deadline (i.e., improves fairness in terms of the minimum video quality across all devices). We consider that a service provider adopts a maxmin policy to improve fairness across all devices regardless of their channel conditions. With such a policy, some devices experiencing harsh channel conditions are prioritized over other devices experiencing good channel conditions with the aim of delivering an acceptable video quality to all devices. This may prevent a severe degradation of the quality of services at a device experiencing a poor channel condition.

For such scenarios, by taking into account the deadline, the coding decisions need to carefully balance coding only from the base layer versus coding from all video layers. While the former guarantees the highest level of priority to the base layer, the latter increases the possibility of decoding a large number of video layers before the deadline. In this context, our main contributions are as follows:

- We derive an upper bound on the probability that a given number of video layers are decoded by all devices before the deadline. Using this probability, we are able to approximately determine whether the broadcast of a given number of video layers can be completed before the deadline with a predefined probability.
- We design two prioritized IDNC algorithms for scalable video, namely the expanding window IDNC (EW-IDNC) algorithm and the non-overlapping window IDNC (NOW-IDNC) algorithm. EW-IDNC algorithm selects a packet combination over the first video layer and computes the resulting upper bound on the probability that the broadcast of that video layer can be completed before the deadline. Only when this proba-

bility meets a predefined threshold, the algorithm considers additional video layers in coding decisions in order to increase the number of decoded video layers at the devices. In contrast, NOW-IDNC algorithm always selects a packet combination over the first video layer without exploiting the coding opportunities by including additional video layers.

- We use a real scalable video sequence to evaluate the performance of our proposed algorithms. Simulation results show that our proposed EW-IDNC and NOW-IDNC algorithms increase the minimum number of decoded video layers over all devices compared to the IDNC algorithms in [9, 52] that are oblivious to the in-order layer decoding and deadline constraints. Furthermore, the proposed algorithms also achieve a comparable performance compared to the expanding window RLNC algorithm in [43, 44] while preserving the benefits of IDNC strategies. It is also observed that EW-IDNC achieves a similar performance in terms of maximizing the minimum number of decoded video layers and significantly better performance in terms of maximizing the mean decoded video layers as compared to NOW-IDNC algorithm.

The rest of this chapter is organized as follows. The system model and IDNC graph are described in Section 3.2. We illustrate the importance of appropriately choosing a coding window in Section 3.3 and draw several guidelines for prioritized IDNC algorithms in Section 3.4. Using these guidelines, we design two prioritized IDNC algorithms in Section 3.5. We formulate the problem of finding an appropriate packet combination in Section 3.6 and design a heuristic packet selection algorithm in Section 3.7. Simulation results are presented in Section 3.8. Finally, Section 3.9 concludes the chapter.

3.2 Scalable Video Broadcast System

3.2.1 Scalable Video Coding

We consider a system that employs the scalable video codec (SVC) extension to H.264/AVC video compression standard [13, 16]. A group of pictures (GOP) in scalable video has several video layers and the information bits of each video layer is divided into one or more packets. The video layers exhibit a hierarchical order such that each video layer can only be decoded after successfully receiving all the packets of this layer and its lower layers. The first video layer (known as the *base layer*) encodes the lowest temporal, spatial, and quality levels of the original video and the successor video layers (known as the *enhancement layers*) encode the difference between the video layers of higher temporal, spatial, and quality levels and the base layer. With the increase in the number of decoded video layers, the video quality improves at the devices.

3.2.2 Notations

Throughout this chapter, we use calligraphic letters to denote sets and their corresponding capital letters to denote the cardinalities of these sets. Let $\mathcal{N}$ be a set. Then N denotes the cardinality of the set $\mathcal{N}$, e.g., $N = |\mathcal{N}|$.

3.2.3 System Model

We consider a wireless sender (e.g., a base station or a wireless access point) that wants to broadcast a set of N source packets forming a GOP, $\mathcal{N} = \{P_1, ..., P_N\}$, to a set of M devices, $\mathcal{M} = \{U_1, ..., U_M\}$. A network coding scheme is applied on the packets of a single GOP as soon as all the packets are ready, which implies that neither merging of GOPs nor buffering of packets in more than one GOP at the sender is allowed. This significant aspect arises from the minimum delivery delay requirement in real-time video streaming. Time is slotted and the sender can transmit one packet in a time slot t. There is a limit on the total number of allowable time slots Θ used to broadcast the N packets to the M devices, as the deadline for the current GOP expires after Θ time slots. Therefore, at any time slot $t \in [1, 2, ..., \Theta]$, the sender can compute the number of remaining transmissions for the current GOP as, $Q = \Theta - t + 1$.

In the scalable video broadcast system, the sender has L scalable video layers and each video layer consists of one or more packets. Let set $\mathcal{N} = \{P_1^1, P_2^1, ..., P_{n_1}^1; ...; P_1^L, P_2^L, ..., P_{n_L}^L\}$ denote all the packets in the L video layers, with n_ℓ being the number of packets in the ℓ-th video layer. In fact, $N = \sum_{\ell=1}^{L} n_\ell$. Although the number of video layers in a GOP of a video stream is fixed, depending on the video content, n_ℓ and N can have different values for different GOPs. We denote the set that contains all packets in the first ℓ video layers as $\mathcal{N}^{1:\ell}$ and the cardinality of $\mathcal{N}^{1:\ell}$ as $N^{1:\ell}$.

In this thesis, we consider that each transmitted packet is subject to an independent Bernoulli erasure at device U_i with probability ϵ_i. Such independent erasure probability in the consecutive packet transmissions models the fast fading channel in wireless networks, wherein fading channel gain changes rapidly in an almost uncorrelated manner. Although this channel model does not describe all real-world scenarios, it indeed captures the essence of wireless broadcast and when the packet length is sufficiently long, or when time division duplex (TDM) is used to multiplex packets from different devices, so there is sufficient gap between transmitted packets of a device. On the other hand, the authors in [98] consider packet erasure correlations in designing and analysing network coding solutions. Indeed, such erasure correlations characterize and model physical channel differently, and subsequently trigger changes in coding solutions. Although interesting, such study on correlated channels is beyond the scope of my thesis.

Each device listens to all transmitted packets and feeds back to the sender a positive or negative acknowledgement (ACK or NAK) for each received or lost packet. We assume that the devices send feedback to the sender using dedicated control channels and the feedback is error-free. The sender uses such information to make efficient coding decisions. Here, we consider idealistic zero-delay feedback channels to study IDNC in video streaming applications and, subsequently, examine the benchmark performance of IDNC solutions. Such consideration of zero-delay feedback channels is more sensible for networks with short range communications, dedicated control channels and full duplex communications. However, in many practical networks such as satellite networks, there is unavoidable delay associated with feedback channels, which may impact the performance of IDNC solutions. In fact, a body of works on IDNC [51,89] show that when the feedback channels are subject to packet losses and several transmissions-delay, the degradation on the performance of IDNC solutions can be as low as $3\% - 4\%$.

After each transmission, the sender stores the reception status of all packets at all devices in an $M \times N$ *feedback status matrix (FSM)* $\mathbf{F} = [f_{i,j}],\ \forall U_i \in \mathcal{M}, P_j \in \mathcal{N}$ such that:

$$f_{i,j} = \begin{cases} 0 & \text{if packet } P_j \text{ is received by device } U_i, \\ 1 & \text{if packet } P_j \text{ is missing at device } U_i. \end{cases} \tag{3.1}$$

Example 1. *An example of FSM with $M = 2$ devices and $N = 5$ packets is given as follows:*

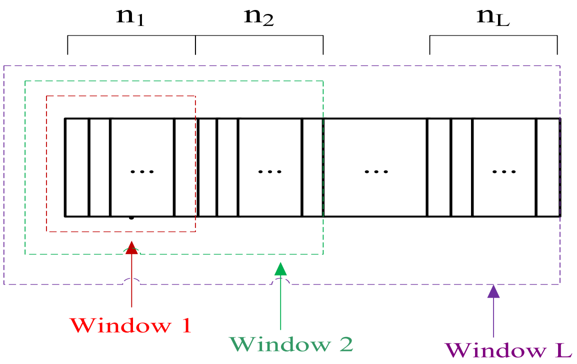


Figure 3.1: L windows for an L-layer GOP with n_ℓ packets in the ℓ-th layer.

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}. \tag{3.2}$$

In this example, we assume that packets P_1 and P_2 belong to the first (i.e., base) layer, packets P_3 and P_4 belong to the second layer and packet P_5 belongs to the third layer. Therefore, the set containing all packets in the first two video layers is $\mathcal{N}^{1:2} = \{P_1, P_2, P_3, P_4\}$.

Definition 1 (Window)**.** *A window over the first ℓ video layers (denoted by ω_ℓ) includes all the packets in $\mathcal{N}^{1:\ell} = \{P_1^1, P_2^1, ..., P_{n_1}^1, ..., P_1^\ell, P_2^\ell, ..., P_{n_\ell}^\ell\}$.*

There are L windows for a GOP with L video layers as shown in Figure 3.1. The FSM corresponding to the window ω_ℓ over the first ℓ video layers is an $M \times N^{1:\ell}$ matrix $\mathbf{F}^{1:\ell}$, which contains the first $N^{1:\ell}$ columns of the FSM $\mathbf{F}$.

Based on the FSM, the following two sets of packets can be attributed to each device U_i at any given time slot t:

- The *Has set* of device U_i in the first ℓ video layers ($\mathcal{H}_i^{1:\ell}$) is defined as the set of packets that are decoded by device U_i from the first ℓ video layers. In Example 1, the Has set of device U_2 in the first two video layers is $\mathcal{H}_2^{1:2} = \{P_1, P_4\}$.

- The *Wants set* of device U_i in the first ℓ video layers ($\mathcal{W}_i^{1:\ell}$) is defined as the set of packets that are missing at device U_i from the first ℓ video layers. In other words,

$\mathcal{W}_i^{1:\ell} = \mathcal{N}^{1:\ell} \setminus \mathcal{H}_i^{1:\ell}$. In Example 1, the Wants set of device U_2 in the first two video layers is $\mathcal{W}_2^{1:2} = \{P_2, P_3\}$.

The set of devices having *non-empty Wants sets* in the first ℓ video layers is denoted by $\mathcal{M}_w^{1:\ell}$ (i.e., $\mathcal{M}_w^{1:\ell} = \big\{U_i \big| \mathcal{W}_i^{1:\ell} \neq \varnothing\big\}$). At any given FSM $\mathbf{F}^{1:\ell}$ at time slot t, device U_i having non-empty Wants set in the first ℓ video layers (i.e., $U_i \in \mathcal{M}_w^{1:\ell}$) belongs to one of the following three sets:

- The *critical set* of devices for the first ℓ video layers ($\mathcal{C}^{1:\ell}$) is defined as the set of devices with the number of missing packets in the first ℓ video layers being equal to the number of remaining Q transmissions (i.e., $W_i^{1:\ell} = Q, \forall U_i \in \mathcal{C}^{1:\ell}$).
- The *affected set* of devices for the first ℓ video layers ($\mathcal{A}^{1:\ell}$) is defined as the set of devices with the number of missing packets in the first ℓ video layers being greater than the number of remaining Q transmissions (i.e., $W_i^{1:\ell} > Q, \forall U_i \in \mathcal{A}^{1:\ell}$).
- The *non-critical set* of devices for the first ℓ video layers ($\mathcal{B}^{1:\ell}$) is defined as the set of devices with the number of missing packets in the first ℓ video layers being less than the number of remaining Q transmissions (i.e., $W_i^{1:\ell} < Q, \forall U_i \in \mathcal{B}^{1:\ell}$).

In fact, $\mathcal{C}^{1:\ell} \cup \mathcal{A}^{1:\ell} \cup \mathcal{B}^{1:\ell} = \mathcal{M}_w^{1:\ell}$.

Definition 2 (Instantly Decodable Packet)**.** *A transmitted packet is instantly decodable for device U_i if it contains exactly one source packet from $\mathcal{W}_i^{1:L}$.*

Example 2. *Given the FSM* $\mathbf{F}$ *in* (3.2)*, assume the base station broadcasts coded packet $P_2 \oplus P_3$. In this case, coded packet $P_2 \oplus P_3$ is instantly decodable for device U_1 as it can reXOR this coded packet with previously received packet P_2 to decode the new packet P_3. On the other hand, the same coded packet $P_2 \oplus P_3$ is not instantly decodable for device U_2 as it does not possess either of these packets.*

Definition 3 (Targeted Device)**.** *A device U_i is targeted by packet P_j in a transmission when this device will immediately decode missing packet P_j upon successfully receiving the transmitted packet. In Example 2, with coded packet $P_2 \oplus P_3$ transmission, we say device U_1 is targeted by packet P_3.*

Definition 4. *(Individual Completion Time of a Single Device) At time slot t, individual completion time of device U_i for the first ℓ video layers (denoted by $T_{W_i^{1:\ell}}$) is the total number of transmissions required to deliver all the missing packets in $\mathcal{W}_i^{1:\ell}$ to device U_i.*

Individual completion time of device U_i for the first ℓ video layers can be $T_{W_i^{1:\ell}} = W_i^{1:\ell}, W_i^{1:\ell}+1, ...$ depending on the number of transmissions that device U_i is targeted with a new packet and the channel erasures experienced by device U_i in those transmissions.

Example 3. *Consider the FSM* $\mathbf{F}$ *in* (3.2) *and assume erasure-free transmissions. Let us consider the following transmission schedule in the successive four time slots:*

1. *Packet $P_1 \oplus P_2$ is transmitted in the first time slot.*
2. *Packet P_3 is transmitted in the second time slot.*
3. *Packet P_4 is transmitted in the third time slot.*
4. *Packet P_5 is transmitted in the fourth time slot.*

The evolution of the FSM after each time slot is given by:

$$\begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} \underset{t=1}{\Rightarrow} \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \underset{t=2}{\Rightarrow} \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \underset{t=3}{\Rightarrow} \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \underset{t=4}{\Rightarrow} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (3.3)$$

Let us also consider the packets of individual layers in Example 1. With the above FSM evolution, device U_2 decodes packets P_1 and P_2 of the first layer after the first transmission. Therefore, the individual completion time of device U_2 for the first video layer is $T_{W_2^{1:1}} = 1$. On the other hand, device U_2 decodes all five packets after the first two transmissions. Therefore, the individual completion time of device U_2 for all three video layers is $T_{W_2^{1:3}} = 2$.

Definition 5. *(Completion Time of all Devices) At time slot t, completion time of all devices for the first ℓ video layers (denoted by $T^{1:\ell}$) is the total number of transmissions required to deliver all the missing packets from the first ℓ video layers to all devices in $\mathcal{M}_w^{1:\ell}$.*

Example 4. *With the FSM evolution in* (3.3)*, both devices decode packets* P_1 *and* P_2 *of the first layer after the first transmission. Therefore, the completion time of all devices for the first video layer is* $T^{1:1} = 1$*. On the other hand, both devices decode all five packets after four transmissions. Therefore, the completion time of all devices for all three video layers is* $T^{1:3} = 4$ *as the transmission schedule requires four time slots to complete the reception of all packets by both devices.*

Definition 6. *(Completion Time of all Non-critical Devices) At time slot t, completion time of all non-critical devices for the first* ℓ *video layers (denoted by* $T_B^{1:\ell}$*) is the total number of transmissions required to deliver all the missing packets from the first* ℓ *video layers to all non-critical devices in* $\mathcal{B}^{1:\ell}$*.*

3.2.4 IDNC Graph and Packet Generation

We define the representation of all feasible packet combinations that are instantly decodable by a subset of, or all devices, in the form of a graph. As described in [5, 77], the IDNC graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is constructed by first inducing a vertex $v_{ij} \in \mathcal{V}$ for each missing packet $P_j \in \mathcal{W}_i^{1:L}$, $\forall U_i \in \mathcal{M}$. Two vertices v_{ij} and v_{mn} in $\mathcal{G}$ are connected (adjacent) by an edge $e_{ij,mn} \in \mathcal{E}$, when one of the following two conditions holds:

- **C1**: $P_j = P_n$, the two vertices are induced by the same missing packet P_j of two different devices U_i and U_m.
- **C2**: $P_j \in \mathcal{H}_m^{1:L}$ and $P_n \in \mathcal{H}_i^{1:L}$, the requested packet of each vertex is in the Has set of the device corresponding to the other vertex.

Definition 7. *In an undirected graph, all vertices in a clique are connected to each other with edges. A clique is maximal if it is not a subset of any larger clique [99].*

Given this graph representation, the set of all feasible IDNC packets can be defined by the set of all maximal cliques in graph $\mathcal{G}$. The sender can generate an IDNC packet for a given transmission by XORing all the source packets identified by the vertices of a selected maximal clique (denoted by κ) in graph $\mathcal{G}$. Note that each device can have at most one vertex (i.e., one missing packet) in a maximal clique κ and the selection of a maximal clique κ is equivalent to the selection of *a set of targeted devices* (denoted by $\mathcal{X}(\kappa)$).

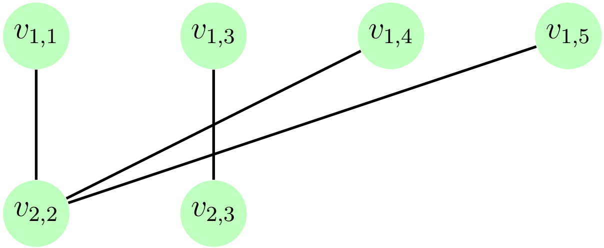


Figure 3.2: IDNC graph corresponding to FSM in (3.2).

Example 5. *The IDNC graph $\mathcal{G}$ corresponding to FSM in* (3.2) *is shown in Fig. 3.2. An example of a maximal clique of this graph is $\kappa = \{v_{1,1}, v_{2,2}\}$, in which the sender transmits $P_1 \oplus P_2$. Note that the maximal clique is just an edge, because no other larger clique exists in the graph.*

3.3 Importance of Appropriately Choosing a Coding Window

In scalable video with multiple layers, the sender needs to choose a window of video layers and the corresponding FSM to select a packet combination in each transmission. In general, different windows lead to different packet combinations and result in different probabilities of completing the broadcast of different numbers of video layers before the deadline. To further illustrate this, let us consider the following FSM with $M = 2$ devices and $N = 2$ packets at time slot t:

$$\mathbf{F} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}. \tag{3.4}$$

In this scenario, we assume that packet P_1 belongs to the first video layer and packet P_2 belongs to the second video layer. We further assume that there are two remaining transmissions before the deadline, i.e., $Q = 2$. Given two video layers, there are two windows such as $\omega_1 = \{P_1\}$ and $\omega_2 = \{P_1, P_2\}$. With these windows, the possible packet transmissions at time slot t are:

- **Case 1:** Window ω_1 leads to packet P_1 transmission since it targets device U_2 and $\mathcal{M}_w^{1:1} = \{U_2\}$.

Table 3.1: Probability expressions used in Case 1, where packet P_1 is transmitted at time slot t. With this transmission, the first layer completion probability before the deadline $\mathbb{P}[T^{1:1} \leq 2] = (1 - \epsilon_2) + \epsilon_2(1 - \epsilon_2)$ and both layers' completion probability before the deadline $\mathbb{P}[T^{1:2} \leq 2] = (1 - \epsilon_2)(1 - \epsilon_1)(1 - \epsilon_2)$ are shown.

		$P_1(t)$	-	$P_1(t)$	$P_1(t+1)$
$\mathbb{P}[T^{1:1} \leq 2]$	U_1	-	-	-	-
	U_2	$(1 - \epsilon_2)$	-	ϵ_2	$(1 - \epsilon_2)$

	$P_1(t)$	$P_2(t+1)$
$\mathbb{P}[T^{1:2} \leq 2]$	-	$(1 - \epsilon_1)$
	$(1 - \epsilon_2)$	$(1 - \epsilon_2)$

- **Case 2:** Window ω_2 leads to packet P_2 transmission since it targets devices U_1 and U_2 and $\mathcal{M}_w^{1:2} = \{U_1, U_2\}$.

(Case 1:) With packet P_1 transmitted at time slot t, we can compute the probabilities of completing the broadcast of different numbers of video layers before the deadline as follows.

- The probability of completing the first video layer broadcast before the deadline can be computed as, $\mathbb{P}[T^{1:1} \leq 2] = (1 - \epsilon_2) + \epsilon_2(1 - \epsilon_2)$. Here, $(1 - \epsilon_2)$ defines the packet reception probability at device U_2 at time slot t and $\epsilon_2(1 - \epsilon_2)$ defines the probability that packet P_1 is lost at device U_2 at time slot t and is received at device U_2 at time slot $t + 1$.

  Remark 1. *It can be stated that the missing packets of all devices need to be attempted at least once in order to have a possibility of delivering all the missing packets to all devices.*

- Using Remark 1, the sender transmits packet P_2 at time slot $t + 1$. Consequently, the probability of completing the broadcast of both video layers before the deadline can be computed as, $\mathbb{P}[T^{1:2} \leq 2] = (1 - \epsilon_2)(1 - \epsilon_1)(1 - \epsilon_2)$. This is the probability that each missing packet is received in the first attempt.

A summary of probability expressions used for Case 1 can be found in Table 3.1.

(Case 2:) With packet P_2 transmitted at time slot t, we can compute the probabilities of completing the broadcast of different numbers of video layers before the deadline as follows.

- The sender transmits packet P_1 at time slot $t + 1$. Consequently, the probability of completing the first video layer broadcast before the deadline can be computed as, $\mathbb{P}[T^{1:1} \leq 2] = (1 - \epsilon_2)$. This is the probability that packet P_1 is received at device U_2 at time slot $t + 1$.

Table 3.2: Probability expressions used in Case 2, where packet P_2 is transmitted at time slot t. With this transmission, the first layer completion probability before the deadline $\mathbb{P}[T^{1:1} \leq 2] = (1-\epsilon_2)$ and both layers' completion probability before the deadline $\mathbb{P}[T^{1:2} \leq 2] = \epsilon_1(1-\epsilon_2)(1-\epsilon_1)(1-\epsilon_2) + (1-\epsilon_1)(1-\epsilon_2)(1-\epsilon_2)$ are shown.

		$P_2(t)$	$P_1(t+1)$
$\mathbb{P}[T^{1:1} \leq 2]$	U_1	-	-
	U_2	-	$(1-\epsilon_2)$

	$P_2(t)$	$P_1 \oplus P_2(t+1)$	$P_2(t)$	$P_1(t+1)$
$\mathbb{P}[T^{1:2} \leq 2]$	ϵ_1	$(1-\epsilon_1)$	$(1-\epsilon_1)$	-
	$(1-\epsilon_2)$	$(1-\epsilon_2)$	$(1-\epsilon_2)$	$(1-\epsilon_2)$

- Using Remark 1, the sender transmits either coded packet $P_1 \oplus P_2$ or packet P_1 at time slot $t+1$. Consequently, the probability of completing both video layers' broadcast before the deadline can be computed as, $\mathbb{P}[T^{1:2} \leq 2] = \epsilon_1(1-\epsilon_2)(1-\epsilon_1)(1-\epsilon_2) + (1-\epsilon_1)(1-\epsilon_2)(1-\epsilon_2)$.

  - ⋆ $\epsilon_1(1-\epsilon_2)(1-\epsilon_1)(1-\epsilon_2)$ results from coded packet $P_1 \oplus P_2$ transmission at time slot $t+1$. The transmitted packet P_2 at time slot t can be lost at device U_1 with probability ϵ_1 and can be received at device U_2 with probability $(1-\epsilon_2)$. With this loss and reception status, the sender transmits coded packet $P_1 \oplus P_2$ to target both devices and the probability that both devices receive the transmitted packet is $(1-\epsilon_1)(1-\epsilon_2)$.

  - ⋆ $(1-\epsilon_1)(1-\epsilon_2)(1-\epsilon_2)$ results from packet P_1 transmission at time slot $t+1$. This is the probability that each missing packet is received in the first attempt.

A summary of probability expressions used for Case 2 can be found in Table 3.2. Using the results in Case 1 and Case 2, we conclude for the given time slot t:

- Packet P_1 transmission resulting from window ω_1 is a better decision in terms of completing the first video layer broadcast since $\mathbb{P}[T^{1:1} \leq 2]$ is larger in Case 1.

- Packet P_2 transmission resulting from window ω_2 is a better decision in terms of completing both video layers broadcast since $\mathbb{P}[T^{1:2} \leq 2]$ is larger in Case 2.

Remark 2. *The above example illustrates that it is not always possible to select a packet combination that achieves high probabilities of completing the broadcast of different numbers of video layers before the deadline. In general, some packet transmissions (resulting from different windows) can increase the probability of completing the broadcast of the first video layer, but reduce the probability of completing the broadcast of all video layers and vice versa.*

3.4 Guidelines for Prioritized IDNC Algorithms

In this chapter, we aim to address the problem of maximizing the minimum number of decoded video layers over all devices before the deadline. Such optimization problem can be formulated into a finite horizon Markov decision process framework and the optimal transmission schedule can be found using a dynamic programming approach. However, the works in [4,77] showed that finding the optimal IDNC schedule for wireless broadcast of a set of packets is computationally intractable due to the curse of dimensionality of the dynamic programming approach. Therefore, to address the decoded video layer maximization problem with much lower computational complexity, we draw several guidelines for the prioritized IDNC algorithms in the following three subsections.

3.4.1 Feasible Windows of Video Layers

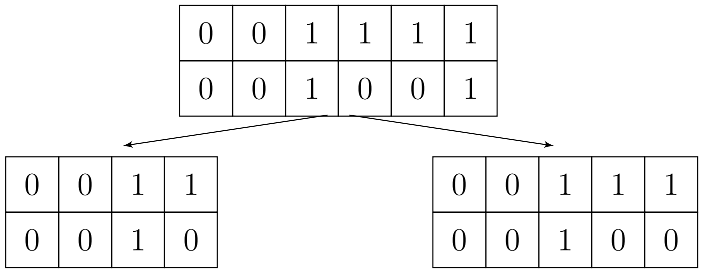
For a given FSM $\mathbf{F}$ at time slot t, we now determine the video layers which can be included in a feasible window and can be considered in coding decisions.

Definition 8 (Smallest Feasible Window)**.** *The smallest feasible window (denoted by ω_ℓ) includes the minimum number of successive video layers such that the Wants set of at least one device in those video layers is non-empty. This can be defined as, $\omega_\ell = \min\{|\omega_1|, ..., |\omega_L|\}$ such that $\exists U_i$ with $\mathcal{W}_i^{1:\ell} \neq \varnothing$.*

In this chapter, we address the problem of maximizing the minimum number of decoded video layers over all devices. Therefore, we define the largest feasible window as follows:

Definition 9 (Largest Feasible Window)**.** *The largest feasible window (denoted by $\omega_{\ell+\mu}$, where μ can be $0, 1, ..., L-\ell$) includes the maximum number of successive video layers such that the Wants sets of all devices in those video layers are less than or equal to the remaining Q transmissions. This can be defined as, $\omega_{\ell+\mu} = \max\{|\omega_\ell|, ..., |\omega_L|\}$ such that $\mathcal{W}_i^{1:(\ell+\mu)} \leq Q, \forall U_i \in \mathcal{M}$.*

Note that there is no affected device over the largest feasible window $\omega_{\ell+\mu}$ (i.e., all devices belong to critical and non-critical sets for the first $\ell+\mu$ video layers). In fact, an affected device will definitely not be able to decode all its missing packets in the remaining Q transmissions. *An exception to considering no affected device in the largest feasible window*



0	0	1	1	1	1
0	0	1	0	0	1

0	0	1	1
0	0	1	0

FSM corresponding to window ω_2

0	0	1	1	1
0	0	1	0	0

FSM corresponding to window ω_3

Figure 3.3: FSMs corresponding to the feasible windows in Example 6

is when it is the smallest feasible window, i.e., $\omega_{\ell+\mu} = \omega_\ell$*, in which case it is possible to have* $\mathcal{A}^{1:\ell}(t) \neq \varnothing$*.*

Definition 10 (Feasible Window)**.** *A feasible window includes any number of successive video layers ranging from the smallest feasible window* ω_ℓ *to the largest feasible window* $\omega_{\ell+\mu}$*. In other words, a feasible window can be any window from* $\{\omega_\ell, \omega_{\ell+1}, ..., \omega_{\ell+\mu}\}$*.*

Example 6. *To further illustrate the feasible windows, consider the following FSM at time slot t:*

$$\mathbf{F} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}. \tag{3.5}$$

In this example, we assume that packets P_1 *and* P_2 *belong to the first video layer, packets* P_3 *and* P_4 *belong to the second video layer, packet* P_5 *belongs to the third video layer and packet* P_6 *belongs to the fourth video layer. We also assume that the number of remaining transmissions Q is equal to* 3*. The smallest feasible window includes the first two video layers (i.e.,* $\omega_2 = \{P_1, P_2, P_3, P_4\}$*) and the largest feasible window includes the first three video layers (i.e.,* $\omega_3 = \{P_1, P_2, P_3, P_4, P_5\}$*). Note that the fourth video layer is not included in the largest feasible window since device* U_1 *has three missing packets in the first three layers (*$\mathcal{W}_1^{1:3} = \{P_3, P_4, P_5\}$*), which is already equal to the number of remaining three transmissions (*$W_1^{1:3} = Q = 3$*). Figure 3.3 shows the extracted FSMs from FSM in* (3.5) *corresponding to the feasible windows.*

3.4.2 Probability that the Completion Time Meets the Deadline

With the aim of designing low complexity prioritized IDNC algorithms, after selecting a packet combination over a given feasible window ω_ℓ at time slot t, we compute the resulting upper bound on the probability that the completion time of all devices for the first ℓ video layers is less than or equal to the remaining $Q-1$ transmissions (denoted by $\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell} \leq Q-1]$ and will be defined in (3.11)). Since this probability is computed separately for each device and deliberately ignores the interdependence of devices' packet reception captured in the FSM, its computation is simple and does not suffer from the curse of dimensionality as in [4, 77]. In return for computational and analytical simplicity, there is some loss in accuracy.

To derive probability $\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell} \leq Q-1]$, we first consider a scenario with one sender and one device U_i. Here, individual completion time of this device for the first ℓ layers can be $T_{W_i^{1:\ell}} = W_i^{1:\ell}, W_i^{1:\ell}+1, ...$ The probability of $T_{W_i^{1:\ell}}$ being equal to $W_i^{1:\ell}+z$, $z \in [0, 1, ..., Q-W_i]$ can be expressed using negative binomial distribution as:

$$\mathbb{P}[T_{W_i^{1:\ell}} = W_i^{1:\ell} + z] = \binom{W_i^{1:\ell} + z - 1}{z} (\epsilon_i)^z (1-\epsilon_i)^{W_i^{1:\ell}}. \tag{3.6}$$

Consequently, the probability that individual completion time $T_{W_i^{1:\ell}}$ of device U_i is less than or equal to the remaining Q transmissions can be expressed as:

$$\mathbb{P}[T_{W_i^{1:\ell}} \leq Q] = \sum_{z=0}^{Q-W_i^{1:\ell}} \mathbb{P}[T_{W_i^{1:\ell}} = W_i^{1:\ell} + z]. \tag{3.7}$$

We now consider a scenario with one sender and multiple devices in $\mathcal{M}_w^{1:\ell}$. We assume that all devices in $\mathcal{M}_w^{1:\ell}$ are targeted with a new packet in each transmission. This is an ideal scenario and defines a lower bound on individual completion time of each device. Consequently, we can compute an upper bound on the probability that individual completion time of each device meets the deadline. Although this ideal scenario is not likely to occur due to the instant decodability constraint, we can still use this probability upper bound as a metric in designing our computationally simple IDNC algorithms. Having described the ideal scenario with multiple devices, for a given feasible window ω_ℓ at time slot t, we compute the upper bound on the probability that completion time of all devices for the first ℓ video layers is less than or equal to the remaining Q transmissions as:

$$\hat{\mathbb{P}}^{(t)}[T^{1:\ell} \leq Q] = \prod_{U_i \in \mathcal{M}_w^{1:\ell}} \sum_{z=0}^{Q-W_i^{1:\ell}} \mathbb{P}[T_{W_i^{1:\ell}} = W_i^{1:\ell} + z]. \tag{3.8}$$

Due to the instant decodability constraint, it may not be possible to target all devices in $\mathcal{M}_w^{1:\ell}$ with a new packet at time slot t. After selecting a packet combination over a given feasible window ω_ℓ at time slot t, let $\mathcal{X}$ be the set of targeted devices and $\mathcal{M}_w^{1:\ell} \setminus \mathcal{X}$ be the set of ignored devices. We can express the resulting upper bound on the probability that the completion time of all devices for the first ℓ video layers, starting from the successor time slot $t+1$, is less than or equal to the remaining $Q-1$ transmissions as:

$$\begin{aligned}\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell} \leq Q-1] = \prod_{U_i \in \mathcal{X}} \{\mathbb{P}^{(t)}[T_{W_i^{1:\ell}-1} \leq Q-1].(1-\epsilon_i) + \mathbb{P}^{(t)}[T_{W_i^{1:\ell}} \leq Q-1].(\epsilon_i)\} \\ \times \prod_{U_i \in \mathcal{M}_w^{1:\ell} \setminus \mathcal{X}} \mathbb{P}^{(t)}[T_{W_i^{1:\ell}} \leq Q-1] \end{aligned} \tag{3.9}$$

- In the first product in expression (3.9), we compute the probability that a targeted device receives its $W_i^{1:\ell}-1$ or $W_i^{1:\ell}$ missing packets in the remaining $Q-1$ transmissions. Note that the number of missing packets at a targeted device can be $W_i^{1:\ell}-1$ with its packet reception probability $(1-\epsilon_i)$ or can be $W_i^{1:\ell}$ with its channel erasure probability ϵ_i.
- In the second product in expression (3.9), we compute the probability that an ignored device in the current time slot will receives its $W_i^{1:\ell}$ missing packets in the future $Q-1$ transmissions under the aforementioned ideal scenario.

By considering the packet reception and loss cases in the first product in (3.9), we can simplify expression (3.9) as:

$$\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell} \leq Q-1] = \prod_{U_i \in \mathcal{X}} \mathbb{P}^{(t)}[T_{W_i^{1:\ell}} \leq Q] \prod_{U_i \in \mathcal{M}_w^{1:\ell} \setminus \mathcal{X}} \mathbb{P}^{(t)}[T_{W_i^{1:\ell}} \leq Q-1] \tag{3.10}$$

Note that a critical and ignored device $U_i \in \{\mathcal{C}^{1:\ell} \cap (\mathcal{M}_w^{1:\ell} \setminus \mathcal{X})\}$ cannot decode all missing packets in $W_i^{1:\ell}$ in the remaining $Q-1$ transmissions since $W_i^{1:\ell}$ is already equal to Q

transmissions for a critical device. With this remark and an exceptional case of having affected devices described in Section 3.4.1, we can set:

$$\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell} \leq Q-1] = \begin{cases} 0 & \text{If } \mathcal{C}^{1:\ell} \cap (\mathcal{M}_w^{1:\ell} \setminus \mathcal{X}) \neq \varnothing \text{ or } \mathcal{A}^{1:\ell} \neq \varnothing \\ \prod_{U_i \in \mathcal{X}} \mathbb{P}^{(t)}[T_{W_i^{1:\ell}} \leq Q] \times \prod_{U_i \in \mathcal{M}_w^{1:\ell} \setminus \mathcal{X}} \mathbb{P}^{(t)}[T_{W_i^{1:\ell}} \leq Q-1] \\ & \text{Otherwise} \end{cases} \tag{3.11}$$

In this chapter, we use expression (3.11) as a metric in designing computationally simple IDNC algorithms for real-time scalable video transmissions.

3.4.3 Design Criterion for Prioritized IDNC Algorithms

In Section 3.3, we showed that some windows and subsequent packet transmissions increase the probability of completing the broadcast of the first video layer, but reduce the probability of completing the broadcast of all video layers and vice versa. This complicated interplay of selecting an appropriate window motivates us to define a new design criterion. The objective of the design criterion is to expand the coding window over the successor video layers (resulting in an increased possibility of completing the broadcast of those video layers) after providing a certain level of prioritization to the lower video layers.

Design Criterion 1. *The design criterion for the first ℓ video layers is defined as the probability $\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell} \leq Q-1]$ meets a certain threshold λ after selecting a packet combination at time slot t.*

In other words, the design criterion for the first ℓ video layers is satisfied when logical condition $\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell} \leq Q-1] \geq \lambda$ is true after selecting a packet combination at time slot t. Here, probability $\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell} \leq Q-1]$ is computed using expression (3.11) and threshold λ is chosen according to the level of prioritization desired for each video layer. In scalable video applications, each decoded layer contributes to the video quality and the layers are decoded following the hierarchical order. Therefore, the selected packet combination at time slot t requires to satisfy the design criterion following the decoding order of the video layers. In other words, the first priority is satisfying the design criterion for the first video layer (i.e., $\hat{\mathbb{P}}^{(t+1)}[T^{1:1} \leq Q-1] \geq \lambda$), the second priority is satisfying the design criterion for the first two video layers (i.e., $\hat{\mathbb{P}}^{(t+1)}[T^{1:2} \leq Q-1] \geq \lambda$) and so on. Having satisfied such a prioritized design criterion, the coding window can continue to expand over the successor video layers to increase the possibility of completing the broadcast of a large number of video layers.

Remark 3. *Threshold λ enables a tradeoff between the mean decoded video layers and the minimum decoded video layers in making decisions in each time slot. In fact, a large threshold value λ (close to 1) results in making a decision over the smallest feasible window and increasing the minimum number of decoded video layers in each time slot. On the other hand, a small threshold value λ (close to 0) results in making a decision over the largest feasible window and increasing the mean number of decoded video layers in each time slot. An intermediate threshold value λ (i.e., $0 < \lambda < 1$) enables a tradeoff between these two objectives. As a result, the service provider can adopt a threshold value λ based on its prioritized strategies.*

3.5 Prioritized IDNC Algorithms for Scalable Video

In this section, using the guidelines drawn in Section 3.4, we design two prioritized IDNC algorithms that increase the probability of completing the broadcast of a large number of video layers before the deadline. These algorithms provide unequal levels of prioritization to the video layers and adopt prioritized IDNC strategies to meet the hard deadline for the most important video layer in each transmission.

3.5.1 Expanding Window Instantly Decodable Network Coding (EW-IDNC) Algorithm

Our proposed *expanding window instantly decodable network coding* (EW-IDNC) algorithm starts by selecting a packet combination over the smallest feasible window and iterates by selecting a new packet combination over each expanded feasible window while satisfying the design criterion for the existing video layers in each window. Moreover, in EW-IDNC algorithm, a packet combination (i.e., a maximal clique κ) over a given feasible window is selected by following methods that will be described in Section 3.6 or Section 3.7.

At *Step 1 of Iteration 1*, the EW-IDNC algorithm selects a maximal clique κ over the smallest feasible window ω_ℓ. At *Step 2 of Iteration 1*, the algorithm computes the probability

Algorithm 1: Expanding Window IDNC (EW-IDNC) Algorithm

```
(Iteration 1) Consider the smallest feasible window $\omega_\ell$;
Select maximal clique $\kappa$ over window $\omega_\ell$;
Compute probability $\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell} \leq Q-1]$ using expression (3.11);
if $\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell} \leq Q-1] \geq \lambda$ and $|\omega_\ell| < |\omega_{\ell+\mu}|$ then
    Proceed to Iteration 2 and consider $\omega_{\ell+1}$;
else
    Broadcast the selected $\kappa$ at this Iteration 1;
end
(Iteration 2) Select a new maximal clique $\kappa$ over expanded window $\omega_{\ell+1}$;
Compute probability $\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell+1} \leq Q-1]$ using expression (3.11);
if $\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell+1} \leq Q-1] \geq \lambda$ and $|\omega_{\ell+1}| < |\omega_{\ell+\mu}|$ then
    Proceed to Iteration 3 and consider $\omega_{\ell+2}$;
else if $\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell+1} \leq Q-1] \geq \lambda$ and $|\omega_{\ell+1}| = |\omega_{\ell+\mu}|$ then
    Broadcast the selected $\kappa$ at this Iteration 2;
else
    Broadcast the selected $\kappa$ at the previous Iteration 1;
end
(Iteration 3) Repeat the steps of Iteration 2;
```

$\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell} \leq Q-1]$ using expression (3.11). At *Step 3 of Iteration 1*, the algorithm performs one of the following two steps.

- It proceeds to Iteration 2 and considers window $\omega_{\ell+1}$, if $\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell} \leq Q-1] \geq \lambda$ and $|\omega_\ell| < |\omega_{\ell+\mu}|$. This is the case when the design criterion for the first ℓ video layers is satisfied and the window can be further expanded.

- It broadcasts the selected κ at this Iteration 1, if $\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell} \leq Q-1] < \lambda$ or $|\omega_\ell| = |\omega_{\ell+\mu}|$. This is the case when the design criterion for the first ℓ video layers is not satisfied or the window is already the largest feasible window.

At *Step 1 of Iteration 2*, the EW-IDNC algorithm selects a new maximal clique κ over the expanded feasible window $\omega_{\ell+1}$. At *Step 2 of Iteration 2*, the algorithm computes the probability $\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell+1} \leq Q-1]$ using expression (3.11). At *Step 3 of Iteration 2*, the algorithm performs one of the following three steps.

- It proceeds to Iteration 3 and considers window $\omega_{\ell+2}$, if $\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell+1} \leq Q-1] \geq \lambda$ and $|\omega_{\ell+1}| < |\omega_{\ell+\mu}|$. This is the case when the design criterion for the first $\ell+1$ video layers is satisfied and the window can be further expanded.

- It broadcasts the selected κ at this Iteration 2, if $\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell+1} \leq Q-1] \geq \lambda$ and $|\omega_{\ell+1}| = |\omega_{\ell+\mu}|$. This is the case when the design criterion for the first $\ell+1$ video layers is satisfied but the window is already the largest feasible window. Note that when the design criterion for the first $\ell+1$ video layers is satisfied, the design criterion for the first ℓ video layers is certainly satisfied since the number of missing packets of any device in the first ℓ video layers is smaller than or equal to that in the first $\ell+1$ video layers.

- It broadcasts the selected κ at the previous Iteration 1, if $\hat{\mathbb{P}}^{(t+1)}[T^{1:\ell+1} \leq Q-1] < \lambda$. This is the case when the design criterion for the first $\ell+1$ video layers is not satisfied.

At Iteration 3, the algorithm performs the steps of Iteration 2. This iterative process is repeated until the algorithm reaches to the largest feasible window $\omega_{\ell+\mu}$ or the design criterion for the video layers over a given feasible window is not satisfied. The proposed EW-IDNC algorithm is summarized in Algorithm 1.

3.5.2 Non-overlapping Window Instantly Decodable Network Coding (NOW-IDNC) Algorithm

Our proposed *non-overlapping window instantly decodable network coding* (NOW-IDNC) algorithm always selects a maximal clique κ over the smallest feasible window ω_ℓ by following methods that will be described in Section 3.6 or Section 3.7. In fact, this algorithm broadcasts the video layers one after another following their decoding order in a non-overlapping manner. This guarantees the highest level of prioritization to the most important video layer, which has not yet been decoded by all devices.

3.6 Packet Selection Problem over a Given Window

Once a coding window ω_ℓ is given, an efficient packet combination over the coding window still needs to be determined such that all packets within the coding window are quickly decoded by all devices. Therefore, in this section, we address the problem of selecting a maximal clique κ over a given window ω_ℓ that increases the possibility of decoding those ℓ video layers by the maximum number of devices before the deadline. We first extract FSM

$\mathbf{F}^{1:\ell}$ corresponding to window ω_ℓ and construct IDNC graph $\mathcal{G}^{1:\ell}$ according to the extracted FSM $\mathbf{F}^{1:\ell}$. We then select a maximal clique κ^* over graph $\mathcal{G}^{1:\ell}$ in two stages. This approach can be summarized as follows.

- We partition IDNC graph $\mathcal{G}^{1:\ell}$ into critical graph $\mathcal{G}_c^{1:\ell}$ and non-critical graph $\mathcal{G}_b^{1:\ell}$. The critical graph $\mathcal{G}_c^{1:\ell}$ includes the vertices generated from the missing packets in the first ℓ video layers at the critical devices in $\mathcal{C}^{1:\ell}$. Similarly, the non-critical graph $\mathcal{G}_b^{1:\ell}$ includes the vertices generated from the missing packets in the first ℓ video layers at the non-critical devices in $\mathcal{B}^{1:\ell}$.

- We prioritize the critical devices for the first ℓ video layers over the non-critical devices for the first ℓ video layers since all the missing packets at the critical devices cannot be delivered without targeting them in the current transmission ($W_i^{1:\ell} = Q, \forall U_i \in \mathcal{C}^{1:\ell}$).

- If there is one or more critical devices (i.e., $\mathcal{C}^{1:\ell} \neq \varnothing$), in the first stage, we select κ_c^* to target a subset of, or if possible, all critical devices. We define $\mathcal{X}_c$ as the set of targeted critical devices who have vertices in κ_c^*.

- If there is one or more non-critical devices (i.e., $\mathcal{B}^{1:\ell} \neq \varnothing$), in the second stage, we select κ_b^* to target a subset of, or if possible, all non-critical devices that do not violate the instant decodability constraint for the targeted critical devices in κ_c^*. We define $\mathcal{X}_b$ as the set of targeted non-critical devices who have vertices in κ_b^*.

3.6.1 Maximal Clique Selection Problem over Critical Graph

With maximal clique κ_c^* selection, each critical device in $\mathcal{C}^{1:\ell}(t)$ experiences one of the following two events at time slot t:

- $U_i \in \mathcal{X}_c$, the targeted critical device can still receive $W_i^{1:\ell}$ missing packets in the exact $Q = W_i^{1:\ell}$ transmissions.

- $U_i \in \mathcal{C}^{1:\ell} \setminus \mathcal{X}_c$, the ignored critical device cannot receive $W_i^{1:\ell}$ missing packets in the remaining $Q - 1$ transmissions and becomes an affected device at time slot $t + 1$.

Let $\mathcal{A}^{1:\ell}(t + 1)$ be the set of affected devices for the first ℓ video layers at time slot $t + 1$ after κ_c^* transmission at time slot t. The critical devices that are not targeted at time slot t

will become the new affected devices, and the critical devices that are targeted at time slot t can also become the new affected devices if they experience an erasure in this transmission. Consequently, we can express the expected increase in the number of affected devices from time slot t to time slot $t+1$ after selecting κ_c^* as:

$$\begin{aligned}\mathbb{E}[A^{1:\ell}(t+1) - A^{1:\ell}(t)] &= (C^{1:\ell}(t) - |\mathcal{X}_c|) + \sum_{U_i \in \mathcal{X}_c} \epsilon_i \\ &= C^{1:\ell}(t) - \sum_{U_i \in \mathcal{X}_c} 1 + \sum_{U_i \in \mathcal{X}_c} \epsilon_i \\ &= C^{1:\ell}(t) - \sum_{U_i \in \mathcal{X}_c} (1 - \epsilon_i). \end{aligned} \tag{3.12}$$

We now formulate the problem of minimizing the expected increase in the number of affected devices for the first ℓ video layers from time slot t to time slot $t+1$ as a critical maximal clique selection problem over critical graph $\mathcal{G}_c^{1:\ell}$ such as:

$$\begin{aligned}\kappa_c^*(t) &= \arg \min_{\kappa_c \in \mathcal{G}_c^{1:\ell}} \left\{\mathbb{E}[A^{1:\ell}(t+1) - A^{1:\ell}(t)]\right\} \\ &= \arg \min_{\kappa_c \in \mathcal{G}_c^{1:\ell}} \left\{ C^{1:\ell}(t) - \sum_{U_i \in \mathcal{X}_c(\kappa_c)} (1 - \epsilon_i) \right\}. \end{aligned} \tag{3.13}$$

In other words, the solution needs to select the maximal clique in the critical IDNC graph that minimizes the expected increase in the number of affected devices.

3.6.2 Maximal Clique Selection Problem over Non-critical Graph

Once maximal clique κ_c^* is selected among the critical devices in $\mathcal{C}^{1:\ell}(t)$, there may exist vertices belonging to the non-critical devices in non-critical graph $\mathcal{G}_b^{1:\ell}$ that can form even a bigger maximal clique. In fact, if the selected new vertices are connected to all vertices in κ_c^*, the corresponding non-critical devices are targeted without affecting IDNC constraint for the targeted critical devices in κ_c^*. Therefore, we first extract non-critical subgraph $\mathcal{G}_b^{1:\ell}(\kappa_c^*)$ of vertices in $\mathcal{G}_b^{1:\ell}$ that are adjacent to all the vertices in κ_c^* and then select κ_b^* over subgraph $\mathcal{G}_b^{1:\ell}(\kappa_c^*)$.

With these considerations, we aim to maximize the upper bound on the probability that completion time of all non-critical devices for the first ℓ video layers, starting from the successor time slot $t+1$, is less than or equal to the remaining $Q-1$ transmissions (represented by $\hat{\mathbb{P}}^{(t+1)}[T_B^{1:\ell} \leq Q-1]$). We formulate this problem as a non-critical maximal clique selection problem over graph $\mathcal{G}_b^{1:\ell}(\kappa_c^*)$ such as:

$$\kappa_b^*(t) = \arg \max_{\kappa_b \in \mathcal{G}_b^{1:\ell}(\kappa_c^*)} \left\{ \hat{\mathbb{P}}^{(t+1)}[T_B^{1:\ell} \leq Q-1] \right\}. \tag{3.14}$$

By maximizing probability $\hat{\mathbb{P}}^{(t+1)}[T_B^{1:\ell} \leq Q-1]$ upon selecting a maximal clique κ_b, the sender increases the probability of transmitting all packets in the first ℓ video layers to all non-critical devices in $\mathcal{B}^{1:\ell}(t)$ before the deadline. Using expression (3.10) for non-critical devices, we can define expression (3.14) as:

$$\kappa_b^*(t) = \arg \max_{\kappa_b \in \mathcal{G}_b^{1:\ell}(\kappa_c^*)} \{ \prod_{U_i \in \mathcal{X}_b(\kappa_b)} \mathbb{P}^{(t)}[T_{W_i^{1:\ell}} \leq Q] \times \prod_{U_i \in \mathcal{B}^{1:\ell}(t) \setminus \mathcal{X}_b(\kappa_b)} \mathbb{P}^{(t)}[T_{W_i^{1:\ell}} \leq Q-1] \} \tag{3.15}$$

In other words, the solution needs to select the maximal clique in the non-critical subgraph $\mathcal{G}_b^{1:\ell}(\kappa_c^*)$ that maximizes the probability $\hat{\mathbb{P}}^{(t+1)}[T_B^{1:\ell} \leq Q-1]$ for all non-critical devices.

Remark 4. *The final maximal clique κ^* over a given window ω_ℓ is the union of two maximal cliques κ_c^* and κ_b^* (i.e., $\kappa^* = \{\kappa_c^* \cup \kappa_b^*\}$).*

It is well known that finding all the maximal cliques in a graph is NP-hard [99]. Therefore, solving the formulated packet selection problem quickly leads to high computational complexity even for systems with moderate numbers of devices and packets. To reduce the computational complexity, it is conventional to design a heuristic algorithm.

3.7 Heuristic Packet Selection Algorithm over a Given Window

Due to the high computational complexity of the formulated packet selection problem in Section 3.6, we now design a low-complexity heuristic algorithm following the formulations in (3.13) and (3.15). This heuristic algorithm selects maximal cliques κ_c and κ_b based on a greedy vertex search over IDNC graphs $\mathcal{G}_c^{1:\ell}$ and $\mathcal{G}_b^{1:\ell}(\kappa_c)$, respectively. A similar greedy vertex search approach was studied in [4, 77] due to its computational simplicity. However, the works in [4, 77] solved different problems and ignored the dependency between source packets and the hard deadline. These additional constraints considered in this chapter lead us to a different heuristic algorithm with its own features.

- If there is one or more critical devices (i.e., $\mathcal{C}^{1:\ell}(t) \neq \varnothing$), in the first stage, the algorithm selects maximal clique κ_c to reduce the number of newly affected devices for the first ℓ video layers after this transmission.

- If there is one or more non-critical devices (i.e., $\mathcal{B}^{1:\ell}(t) \neq \varnothing$), in the second stage, the algorithm selects maximal clique κ_b to increase the probability $\hat{\mathbb{P}}^{(t+1)}[T_B^{1:\ell} \leq Q - 1]$ after this transmission.

3.7.1 Greedy Maximal Clique Selection over Critical Graph

To select critical maximal clique κ_c, the proposed algorithm starts by finding a lower bound on the potential new affected devices, for the first ℓ video layers from time slot t to time slot $t + 1$, that may result from selecting each vertex from critical IDNC graph $\mathcal{G}_c^{1:\ell}$. At Step 1, the algorithm selects vertex v_{ij} from graph $\mathcal{G}_c^{1:\ell}$ and adds it to κ_c. Consequently, the lower bound on the expected number of new affected devices for the first ℓ video layers after this transmission that may result from selecting this vertex can be expressed as:

$$A_{(1)}^{1:\ell}(t+1) - A^{1:\ell}(t) = C^{1:\ell}(t) - \sum_{U_m \in \{U_i \cup \mathcal{M}_{ij}^{\mathcal{G}_c^{1:\ell}}\}} (1 - \epsilon_m). \tag{3.16}$$

Here, $A_{(1)}^{1:\ell}(t+1)$ represents the number of affected devices for the first ℓ video layers at time slot $t+1$ after transmitting κ_c selected at Step 1 and $\mathcal{M}_{ij}^{\mathcal{G}_c^{1:\ell}}$ is the set of critical devices that have at least one vertex adjacent to vertex v_{ij} in $\mathcal{G}_c^{1:\ell}$. Once $A_{(1)}^{1:\ell}(t+1) - A^{1:\ell}(t)$ is calculated for all vertices in $\mathcal{G}_c^{1:\ell}$, the algorithm chooses vertex v_{ij}^* with the minimum lower bound on the expected number of new affected devices as:

$$v_{ij}^* = \arg \min_{v_{ij} \in \mathcal{G}_c^{1:\ell}} \left\{ A_{(1)}^{1:\ell}(t+1) - A^{1:\ell}(t) \right\}. \tag{3.17}$$

After adding vertex v_{ij}^* to κ_c (i.e., $\kappa_c = \{v_{ij}^*\}$), the algorithm extracts the subgraph $\mathcal{G}_c^{1:\ell}(\kappa_c)$ of vertices in $\mathcal{G}_c^{1:\ell}$ that are adjacent to all the vertices in κ_c. At Step 2, the algorithm selects another vertex v_{mn} from subgraph $\mathcal{G}_c^{1:\ell}(\kappa_c)$ and adds it to κ_c. Consequently, the new lower bound on the expected number of new affected devices can be expressed as:

$$\begin{aligned}
A_{(2)}^{1:\ell}(t+1) - A^{1:\ell}(t) &= C^{1:\ell}(t) - \left(\sum_{U_i \in \mathcal{X}_c(\kappa_c)} (1-\epsilon_i) + \sum_{U_o \in \{U_m \cup \mathcal{M}_{mn}^{\mathcal{G}_c^{1:\ell}(\kappa_c)}\}} (1-\epsilon_o) \right) \\
&= \left(C^{1:\ell}(t) - \sum_{U_m \in \{U_i \cup \mathcal{M}_{ij}^{\mathcal{G}_c^{1:\ell}}\}} (1-\epsilon_m) \right) + \sum_{U_o \in \mathcal{M}_{ij}^{\mathcal{G}_c^{1:\ell}} \setminus (U_m \cup \mathcal{M}_{mn}^{\mathcal{G}_c^{1:\ell}(\kappa_c)})} (1-\epsilon_o) \\
&= (A_{(1)}^{1:\ell}(t+1) - A^{1:\ell}(t)) + \sum_{U_o \in \{\mathcal{M}_{ij}^{\mathcal{G}_c^{1:\ell}} \setminus (U_m \cup \mathcal{M}_{mn}^{\mathcal{G}_c^{1:\ell}(\kappa_c)})\}} (1-\epsilon_o). \qquad (3.18)
\end{aligned}$$

Since $(U_m \cup \mathcal{M}_{mn}^{\mathcal{G}_c^{1:\ell}(\kappa_c)})$ is a subset of $\mathcal{M}_{ij}^{\mathcal{G}_c^{1:\ell}}$, the last term in (3.18) is resulting from the stepwise increment on the lower bound on the expected number of newly affected devices due to selecting vertex v_{mn}. Similar to Step 1, once $A_{(2)}^{1:\ell}(t+1) - A^{1:\ell}(t)$ is calculated for all vertices in the subgraph $\mathcal{G}_c^{1:\ell}(\kappa_c)$, the algorithm chooses vertex v_{mn}^* with the minimum lower bound on the expected number of new affected devices as:

$$v_{mn}^* = \arg \min_{v_{mn} \in \mathcal{G}_c^{1:\ell}(\kappa_c)} \{A_{(2)}^{1:\ell}(t+1) - A^{1:\ell}(t)\}. \qquad (3.19)$$

After adding new vertex v_{mn}^* to κ_c (i.e., $\kappa_c = \{\kappa_c, v_{mn}^*\}$), the algorithm repeats the vertex search process until no further vertex in $\mathcal{G}_c^{1:\ell}$ is adjacent to all the vertices in κ_c.

3.7.2 Greedy Maximal Clique Selection over Non-critical Graph

To select non-critical maximal clique κ_b, the proposed algorithm extracts the non-critical IDNC subgraph $\mathcal{G}_b^{1:\ell}(\kappa_c)$ of vertices in $\mathcal{G}_b^{1:\ell}$ that are adjacent to all the vertices in κ_c. This algorithm starts by finding the maximum probability $\hat{\mathbb{P}}^{(t+1)}[T_B^{1:\ell} \leq Q-1]$ that may result from selecting each vertex from subgraph $\mathcal{G}_b^{1:\ell}(\kappa_c)$. At Step 1, the algorithm selects vertex v_{ij} from $\mathcal{G}_b^{1:\ell}(\kappa_c)$ and adds it to κ_b. Consequently, the probability $\hat{\mathbb{P}}_{(1)}^{(t+1)}[T_B^{1:\ell} \leq Q-1]$ that may result from selecting this vertex at Step 1 can be computed as:

$$\hat{\mathbb{P}}^{(t+1)}_{(1)}[T_B^{1:\ell} \leq Q-1] = \prod_{U_m \in \{U_i \cup \mathcal{M}_{ij}^{\mathcal{G}_b^{1:\ell}(\kappa_c)}\}} \mathbb{P}[T_{W_m^{1:\ell}} \leq Q] \times \prod_{U_m \in \{\mathcal{B}^{1:\ell}(t) \setminus (U_i \cup \mathcal{M}_{ij}^{\mathcal{G}_b^{1:\ell}(\kappa_c)})\}} \mathbb{P}[T_{W_m^{1:\ell}} \leq Q-1]. \quad (3.20)$$

Here, $\mathcal{M}_{ij}^{\mathcal{G}_b^{1:\ell}(\kappa_c)}$ is the set of non-critical devices that have at least one vertex adjacent to vertex v_{ij} in $\mathcal{G}_b^{1:\ell}(\kappa_c)$. Once probability $\hat{\mathbb{P}}^{(t+1)}_{(1)}[T_B^{1:\ell} \leq Q-1]$ is calculated for all vertices in $\mathcal{G}_b^{1:\ell}(\kappa_c)$, the algorithm chooses vertex v_{ij}^* with the maximum probability as:

$$v_{ij}^* = \arg \max_{v_{ij} \in \mathcal{G}_b^{1:\ell}(\kappa_c)} \{\hat{\mathbb{P}}^{(t+1)}_{(1)}[T_B^{1:\ell} \leq Q-1]\}. \quad (3.21)$$

After adding vertex v_{ij}^* to κ_b (i.e., $\kappa_b = \{v_{ij}^*\}$), the algorithm extracts the subgraph $\mathcal{G}_b^{1:\ell}(\kappa_c \cup \kappa_b)$ of vertices in $\mathcal{G}_b^{1:\ell}(\kappa_c)$ that are adjacent to all the vertices in $(\kappa_c \cup \kappa_b)$. At Step 2, the algorithm selects another vertex v_{mn} from subgraph $\mathcal{G}_b^{1:\ell}(\kappa_c \cup \kappa_b)$ and adds it to κ_b. Note that the new set of potentially targeted non-critical devices after Step 2 is $\{U_i \cup U_m \cup \mathcal{M}_{mn}^{\mathcal{G}_b^{1:\ell}(\kappa_c \cup \kappa_b)}\}$, which is a subset of $\{U_i \cup \mathcal{M}_{ij}^{\mathcal{G}_b^{1:\ell}(\kappa_c)}\}$. Consequently, the new probability $\hat{\mathbb{P}}^{(t+1)}_{(2)}[T_B^{1:\ell} \leq Q-1]$ due to the stepwise reduction in the number of targeted non-critical devices can be computed as:

$$\hat{\mathbb{P}}^{(t+1)}_{(2)}[T_B^{1:\ell} \leq Q-1] = \prod_{U_o \in \{U_i \cup U_m \cup \mathcal{M}_{mn}^{\mathcal{G}_b^{1:\ell}(\kappa_c \cup \kappa_b)}\}} \mathbb{P}[T_{W_o^{1:\ell}} \leq Q] \times \prod_{U_o \in \{\mathcal{B}^{1:\ell}(t) \setminus (U_i \cup U_m \cup \mathcal{M}_{mn}^{\mathcal{G}_b^{1:\ell}(\kappa_c \cup \kappa_b)})\}} \mathbb{P}[T_{W_o^{1:\ell}} \leq Q-1]. \quad (3.22)$$

Similar to Step 1, once probability $\hat{\mathbb{P}}^{(t+1)}_{(2)}[T_B^{1:\ell} \leq Q-1]$ is calculated for all vertices in the subgraph $\mathcal{G}_b^{1:\ell}(\kappa_c \cup \kappa_b)$, the algorithm chooses vertex v_{mn}^* with the maximum probability as:

$$v_{mn}^* = \arg \max_{v_{mn} \in \mathcal{G}_b^{1:\ell}(\kappa_c \cup \kappa_b)} \{\hat{\mathbb{P}}^{(t+1)}_{(2)}[T_B^{1:\ell} \leq Q-1]\}. \quad (3.23)$$

After adding new vertex v_{mn}^* to κ_b (i.e., $\kappa_b = \{\kappa_b, v_{mn}^*\}$), the algorithm repeats the vertex search process until no further vertex in $\mathcal{G}_b^{1:\ell}$ is adjacent to all the vertices in $(\kappa_c \cup \kappa_b)$. The

Algorithm 2: Heuristic Packet Selection Algorithm over a Given Window ω_ℓ

```
Extract FSM $\mathbf{F}^{1:\ell}$ corresponding to a given window $\omega_\ell$;
Construct $\mathcal{G}^{1:\ell}(\mathcal{V}, \mathcal{E})$ according to the extracted FSM $\mathbf{F}^{1:\ell}$;
Partition $\mathcal{G}^{1:\ell}$ into $\mathcal{G}_c^{1:\ell}$ and $\mathcal{G}_b^{1:\ell}$ according to the devices in critical set $\mathcal{C}^{1:\ell}$ and
non-critical set $\mathcal{B}^{1:\ell}$;
Initialize $\kappa_c = \varnothing$ and $\kappa_b = \varnothing$;
while $\mathcal{G}_c^{1:\ell} \neq \varnothing$ do
    Compute $A^{1:\ell}(t+1) - A^{1:\ell}(t), \forall v_{ij} \in \mathcal{G}_c^{1:\ell}(\kappa_c)$ using (3.16) or (3.18);
    Select $v_{ij}^* = \arg\min_{v_{ij} \in \mathcal{G}_c^{1:\ell}(\kappa_c)} \{A^{1:\ell}(t+1) - A^{1:\ell}(t)\}$;
    Set $\kappa_c \leftarrow \kappa_c \cup v_{ij}^*$;
    Update subgraph $\mathcal{G}_c^{1:\ell}(\kappa_c)$ and $\mathcal{G}_b^{1:\ell}(\kappa_c)$;
end
while $\mathcal{G}_b^{1:\ell} \neq \varnothing$ do
    Compute $\hat{\mathbb{P}}^{(t+1)}[T_B^{1:\ell} \leq Q-1], \forall v_{ij} \in \mathcal{G}_b^{1:\ell}(\kappa_c \cup \kappa_b)$ using (3.20) or (3.22);
    Select $v_{ij}^* = \arg\max_{v_{ij} \in \mathcal{G}_b^{1:\ell}(\kappa_c \cup \kappa_b)} \{\hat{\mathbb{P}}^{(t+1)}[T_B^{1:\ell} \leq Q-1]\}$;
    Set $\kappa_b \leftarrow \kappa_b \cup v_{ij}^*$;
    Update subgraph $\mathcal{G}_b^{1:\ell}(\kappa_c \cup \kappa_b)$;
end
Set $\kappa \leftarrow \kappa_c \cup \kappa_b$.
```

final maximal clique κ over a given window ω_ℓ is the union of κ_c and κ_b (i.e., $\kappa = \kappa_c \cup \kappa_b$). The proposed heuristic algorithm is summarized in Algorithm 2.

Remark 5. *The complexity of the proposed heuristic packet selection algorithm is $\mathcal{O}(M^2N)$ since it requires weight computations for the $\mathcal{O}(MN)$ vertices in each step and a maximal clique can have at most M vertices. Using this heuristic algorithm, the complexity of the EW-IDNC algorithm is $\mathcal{O}(M^2NL)$ since it can perform the heuristic algorithm at most L times over L windows. Moreover, using this heuristic algorithm, the complexity of the NOW-IDNC algorithm is $\mathcal{O}(M^2N)$ since it performs the heuristic algorithm once over the smallest feasible window.*

3.8 Simulation Results for a Real Video Sequence

In this section, we first discuss the scalable video test sequence used in the simulation and then present the performance of different algorithms for that video sequence.

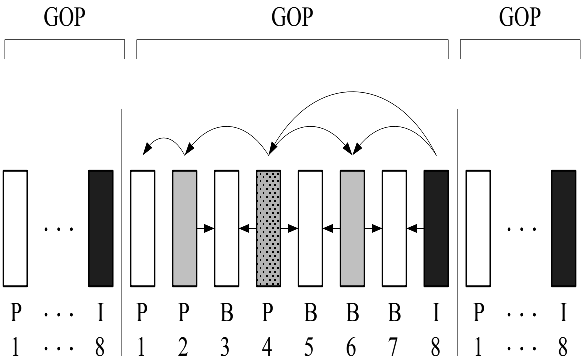


Figure 3.4: A closed GOP with 4 layers and 8 frames (a sequence of I, P and B frames).

3.8.1 Scalable Video Test Sequence

We now describe the H.264/SVC video test sequence used in this chapter. We consider a standard video sequence, Soccer [100]. This video sequence is in common intermediate format (CIF, i.e., 352×288) and has 300 frames with 30 frames per second. We encode the video sequence using the JSVM 9.19.14 version of H.264/SVC codec [16,101] while considering the GOP size of 8 frames and temporal scalability of SVC. As a result, there are 38 GOPs for the test sequence. Each GOP consists of a sequence of I, P and B frames that are encoded into four video layers as shown in Figure 3.4. The frames belonging to the same video layer are represented by the identical shade and the more important video layers are represented by the darker shades. In fact, the GOP in Figure 3.4 is a closed GOP, in which the decoding of the frames inside the GOP is independent of frames outside the GOP [44]. Based on the figure, we can see that a device can decode $1, 2, 4$ or 8 frames upon receiving first 1, 2, 3 or 4 video layers, respectively. Therefore, nominal temporal resolution of 3.75, 7.5, 15 or 30 frames per second is experienced by a viewer depending on the number of decoded video layers.

To assign the information bits to packets, we consider the maximum transmission unit (MTU) of 1500 bytes as the size of a packet. We use 100 bytes for header information and remaining 1400 bytes for video data. 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. 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 α bit per second and a packet length is 1500×8 bits, the allowable number of transmissions Θ for a GOP is fixed. We can conclude that

$\Theta = \frac{8\alpha}{1500 \times 8 \times 30}$.

3.8.2 Simulation Results

We present the simulation results comparing the performance of our proposed EW-IDNC and NOW-IDNC algorithms (using the heuristic packet selection algorithm described in Section 3.7) with the following algorithms.

- Expanding window RLNC (EW-RLNC) algorithm [43, 44] that uses RLNC strategies to encode the packets in different windows while taking into account the decoding order of video layers and the hard deadline. The encoding and decoding processes of EW-RLNC algorithm are described in the appendix in Section 3.10.

- Maximum clique (Max-Clique) algorithm [52] that uses IDNC strategies to service a large number of devices with any new packet in each transmission while ignoring the decoding order of video layers and the hard deadline.

- Interrelated priority encoding (IPE) algorithm [9] that uses IDNC strategies and balances between the number of transmissions required for delivering the base layer and the number of transmissions required for delivering all video layers. However, IPE algorithm ignores the hard deadline in making coding decisions.

Figures 3.5 and 3.6 show the percentage of mean decoded video layers and the percentage of minimum decoded video layers performance of different algorithms for different deadlines Θ (for $M = 15, \epsilon = 0.2$) and different numbers of devices M (for $\Theta = 25, \epsilon = 0.2$). In the case of average erasure probability $\epsilon = 0.2$, the erasure probabilities of different devices are in the range $[0.05, 0.35]$. We adopt a wide range $[0.05, 0.35]$ of channel erasure probabilities to represent different levels of physical channel conditions (e.g., fading, shadowing, etc.) experienced by different devices. Here, a large erasure probability models an extremely poor channel condition of a device which is located at the edge of the coverage area of the base station and experiences deep fading, interference and shadowing.

We choose 6 values for threshold λ from $[0.2, 0.95]$ with step size of 0.15. This results in 6 points on each trade-off curve of EW-IDNC and EW-RLNC algorithms such that $\lambda = 0.2$ and $\lambda = 0.95$ correspond to the top point and the bottom point, respectively. Moreover, we use

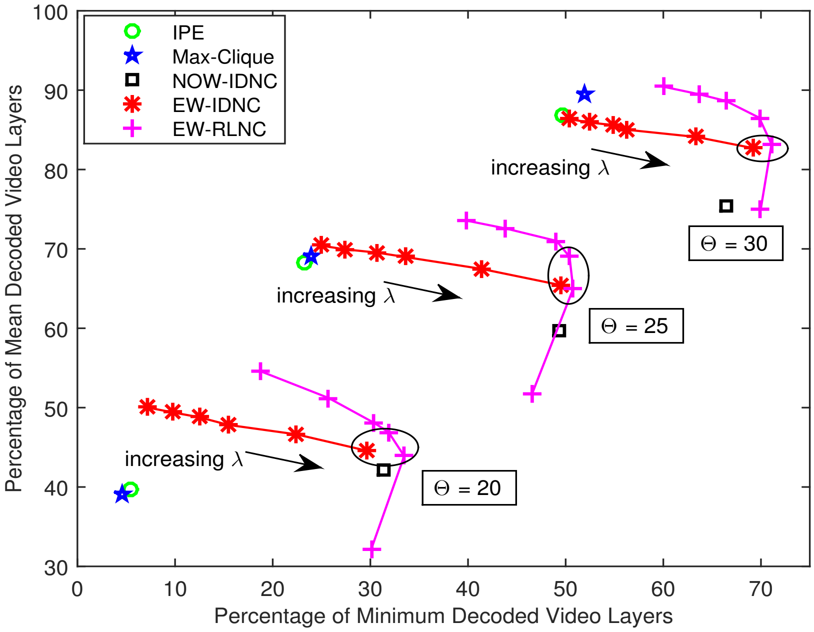


Figure 3.5: Percentage of mean decoded video layers versus percentage of minimum decoded video layers for different deadlines Θ

ellipses to represent efficient operating points (i.e., thresholds λ which are defined in Section 3.4.3) on the trade-off curves. As expected from EW-IDNC and EW-RLNC algorithms, the minimum decoded video layers over all devices increases with the increase of threshold λ at the expense of reducing the mean decoded video layers over all devices. In general, given a small threshold λ, the design criterion is satisfied for a large number of video layers in each transmission, which results in a large coding window and a low level of priority to the lower video layers. Consequently, several devices may decode a large number of video layers, while other devices may decode only the first video layer before the deadline.

Figures 3.5 and 3.6 show that expanding window RLNC (EW-RLNC) algorithm performs poorly for large threshold values λ (required for maximizing the minimum number of decoded video layers over all devices). This is due to transmitting a large number of coded packets from a small coding window to obtain high decoding probabilities of the first video layer at all devices and meet a large threshold λ for the first video layer. Note that EW-RLNC algorithm explicitly determines the number of coded packets from each window at the beginning of the Θ transmissions, which results in a large number of coded packets from the first window to meet a large threshold λ for the first video layer. On the other hand, the proposed EW-IDNC

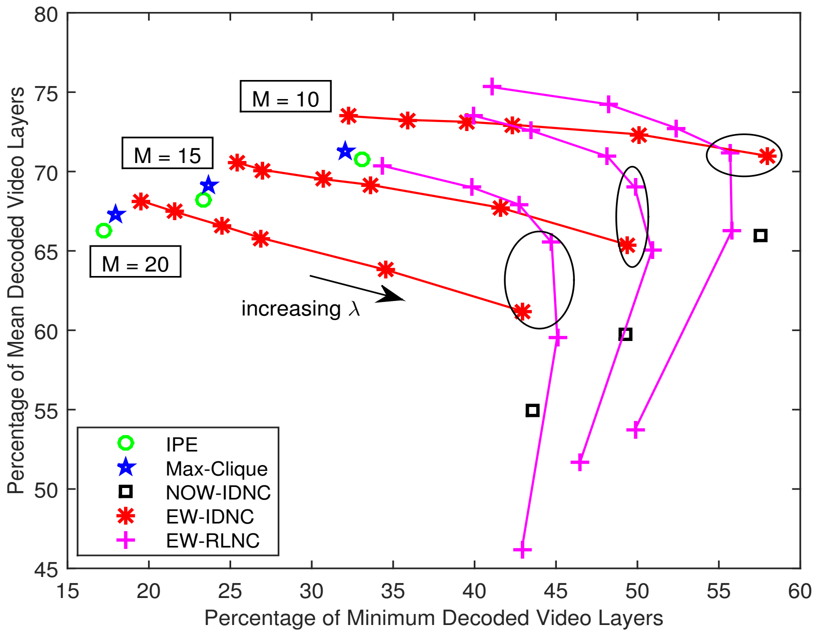


Figure 3.6: Percentage of mean decoded video layers versus percentage of minimum decoded video layers for different number of devices M

algorithm uses feedback to exploit the packet reception status at the devices and determines an efficient coding window in each time slot. As a result, a large threshold λ value for EW-IDNC algorithm provides a high level of prioritization to the first video layer in each transmission while adjusting the coding window based on the past packet receptions. From the figures, it is observed that EW-RLNC outperforms EW-IDNC when a small threshold value λ is chosen for maximizing the mean decoded video layers. However, in this chapter, we adopt a maxmin policy and address the problem of maximizing the minimum number of decoded video layers, which requires a large threshold value λ.

Our proposed EW-IDNC algorithm achieves similar performances compared to the EW-RLNC algorithm in terms of the minimum and the mean decoded video layers. In fact, both algorithms guarantee a high probability of completing the broadcast of a lower video layer (using threshold λ) before expanding the window over the successor video layers. To increase the minimum decoded video layers while respecting the mean decoded video layers, an efficient threshold λ for the EW-IDNC algorithm is around 0.95 and an efficient threshold λ for the EW-RLNC algorithm is around 0.65. Our proposed NOW-IDNC algorithm achieves a similar performance compared to EW-IDNC and EW-RLNC algorithms in terms of the

Table 3.3: Summary of Figure 3.5. The efficient threshold values λ and the corresponding percentages of minimum and mean decoded video layers for EW-IDNC and EW-RLNC algorithms in different deadlines Θ.

	EW-IDNC			EW-RLNC		
Deadline Θ	*Threshold* λ	*Minimum*	*Mean*	*Threshold* λ	*Minimum*	*Mean*
20	0.95	29%	44%	0.65	31%	47%
25	0.95	49%	65%	0.65	50%	69%
30	0.95	69%	82%	0.65	70%	86%

minimum decoded video layers. However, the NOW-IDNC algorithm performs poorly in terms of the mean decoded video layers due to always selecting a packet combination over a single video layer.

From these figures, we can observe that EW-RLNC outperforms EW-IDNC when a small threshold value λ is chosen for maximizing the mean decoded video layers. However, in this chapter, we adopt a maxmin policy and address the problem of maximizing the minimum number of decoded video layers, which requires a large threshold value λ. Indeed, with a large threshold λ, EW-IDNC outperforms EW-RLNC for maximizing the minimum number of decoded video layers.

Tables 3.3 and 3.4 summarize Figures 3.5 and 3.6, respectively listing the efficient threshold values λ and the corresponding percentages of minimum and mean decoded video layers for EW-IDNC and EW-RLNC algorithms in different scenarios. From these tables, we can see that the performance degradation in terms of the minimum and the mean decoded video layers are around 1% and 3%, respectively for EW-IDNC algorithm compared to EW-RLNC algorithm given both algorithms use the efficient threshold values λ. This comparable performance is achieved by the EW-IDNC algorithm while preserving the benefits of IDNC strategies, namely low decoding delay, simple XOR encoding and decoding operations, and low coefficient reporting overhead.

As expected, Max-Clique and IPE algorithms perform poorly compared to our proposed EW-IDNC and NOW-IDNC algorithms in terms of the minimum decoded video layers. Both Max-Clique and IPE algorithms make coding decisions across all video layers and thus, do not address the hard deadline for the most important video layer. As a result, several devices may receive packets from the higher video layers, which cannot be used for decoding those video layers if a packet in a lower video layer is missing at the end of the deadline.

Table 3.4: Summary of Figure 3.6. The efficient threshold values λ and the corresponding percentages of minimum and mean decoded video layers for EW-IDNC and EW-RLNC algorithms in different number of devices M.

	EW-IDNC			EW-RLNC		
No. of devices M	*Threshold λ*	*Minimum*	*Mean*	*Threshold λ*	*Minimum*	*Mean*
10	0.95	57%	71%	0.65	55%	72%
15	0.95	49%	65%	0.65	50%	69%
20	0.95	42%	61%	0.65	44%	65%

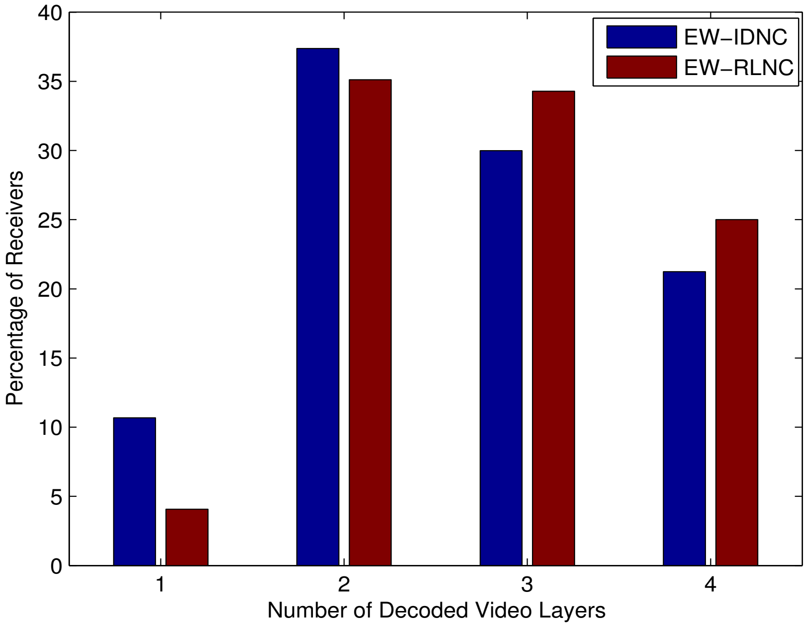


Figure 3.7: Histogram showing the percentage of devices that successfully decode one, two, three and four video layers before the deadline

Figure 3.7 shows the histogram obtained by EW-IDNC algorithm (using $\lambda = 0.95$) and EW-RLNC algorithm (using $\lambda = 0.65$) for $\Theta = 25, M = 15, \epsilon = 0.2$. This histogram illustrates the percentage of devices that successfully decode one, two, three and four video layers before the deadline. From this histogram, we can see that most of the devices decode three or four video layers out of four video layers in a GOP. Moreover, the percentage of devices that decode the first four video layers in EW-RLNC algorithm is slightly higher compared to that in EW-IDNC algorithm. Note that we use another standard video sequence *Foreman* in the simulations and observe the similar results as in the case of Soccer. However, we do not include those results in this chapter due to brevity.

3.9 Conclusion

In this chapter, we developed an efficient, yet computationally simple, IDNC framework for real-time scalable video broadcast over wireless networks. In particular, we derived an upper bound on the probability that the completion time of all devices meet the deadline. Using this probability with other guidelines, we designed EW-IDNC and NOW-IDNC algorithms that provide a high level of prioritization to the most important video layer before considering additional video layers in coding decisions. We used a real scalable video sequence in the simulation and showed that our proposed IDNC algorithms improve the received video quality compared to the existing IDNC algorithms and achieve a similar performance compared to the EW-RLNC algorithm.

3.10 Appendix A

Expanding Window Random Linear Network Coding

We follow the work in [44] and consider a deterministic approach, where the number of coded packets from each window is explicitly determined at the beginning of the Θ transmissions. The sender broadcasts these coded packets in Θ transmissions without receiving any feedback. Let us assume that θ_ℓ coded packets are generated (and thus transmitted) from the packets in the ℓ-th window ω_ℓ. Then $\Sigma_{\ell=1}^{L}\theta_\ell = \Theta$ and $\mathbf{z} = [\theta_1, \theta_2, ..., \theta_L]$ is an EW-RLNC transmission policy. Given a fixed number of allowable transmissions Θ, all possible transmission policies can be defined as all combinations of the number of coded packets from each window. Now, we describe the process of selecting a transmission policy as follows.

We use $\mathbf{n} = [n_1, n_2, ..., n_L]$ to denote the number of packets from different layers in a GOP. For a given transmission policy $\mathbf{z}$, we denote the probability that device U_i with erasure probability ϵ_i can decode the packets of layer ℓ (and all the packets of its lower layers) by $\mathbb{P}_i^\ell(\mathbf{n}, \mathbf{z})$. This probability can be computed using expression (1) in [44]. Now we extend this probability to M devices and compute the probability that M devices can decode the packets of layer ℓ (and all the packets of its lower layers) as follows:

$$\mathbb{P}^\ell(\mathbf{n}, \mathbf{z}) = \prod_{U_i \in \mathcal{M}} \mathbb{P}_i^\ell(\mathbf{n}, \mathbf{z}). \tag{3.24}$$

Given transmission policy $\mathbf{z}$, the probability in (3.24) is computed for each of L video layers. Furthermore, we consider all possible transmission policies and compute probability $\mathbb{P}^{\ell}(\mathbf{n}, \mathbf{z}), \forall \ell \in [1, ..., L]$, for each transmission policy. Finally, we select the transmission policy $\mathbf{z}$ among all transmission policies that satisfies condition $\mathbb{P}^{\ell}(\mathbf{n}, \mathbf{z}) \geq \lambda$ for the largest number of ℓ successive video layers (i.e., satisfies condition for the largest ℓ-th video layer and of course all its lower layers). Here, condition $\mathbb{P}^{\ell}(\mathbf{n}, \mathbf{z}) \geq \lambda$ is adopted following the same approach as in our proposed EW-IDNC algorithm. The details of decoding a video layer based on the number of received packets from different windows can be found in [44].

Chapter 4

Content Aware Network Codes in D2D Networks

4.1 Overview

The rapid growth of high quality content in addition to the limited radio resources creates a number of challenges for the communication and networking communities to meet the quality of service requirements of wireless users. To solve these problems, D2D communications is widely considered as a promising paradigm wherein devices exchange their received packets with others using a short-range wireless technology, e.g., IEEE 802.11 adhoc mode [23–25]. In such D2D networks, multiple devices are allowed to transmit concurrently given there is no destructive collision of a transmission by others. This results in an increased possibility of meeting the hard deadline for the video packets without fully consuming the cellular resources.

The network coded D2D communications have recently drawn a significant attention to take advantage of both network coding and devices' cooperation [32,35,65,86,87]. Such joint solution offers an increased throughput, delay reduction and traffic reduction of the cellular networks. On the other hand, the video streaming community advocates for taking into account the unequal importance of video packets and the hard deadline to design efficient technologies [11, 13, 14]. This chapter bridges the gap between these three approaches (i.e., network coding, D2D communications and video coding), and facilitates the development of a unified framework that improves video quality, throughput, delay and coverage zones of wireless networks.

In particular, we aim to design an efficient IDNC framework that minimizes the mean video distortion before the deadline in a D2D network. For 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 are as follows:

- We introduce a cooperation aware IDNC graph that represents both coding and transmission conflicts of a D2D network with one common transmission channel. In fact, this graph representation has to account for the instant decodability constraint, coverage zones of different devices, potential collisions over the common channel from simultaneous transmissions, and the constraint that each device cannot transmit and receive concurrently.

- 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 mean video distortion at the end of the deadline. By considering the properties of the problem formulation, we further design a low-complexity maximal independent set selection heuristic with suboptimal performance.

- Simulation results over real video sequences show that our proposed IDNC algorithms improve the received video quality compared to the existing IDNC algorithms that consider either a fully connected D2D network for reducing video distortion [56] or a partially connected D2D network for serving a large number of devices with any new packet [26].

The rest of this chapter is organized as follows. We describe the system model in Section 4.2. Section 4.3 defines the novel IDNC graph. We formulate the minimum video distortion problem into an MDP framework in Section 4.4 and design a simple heuristic in Section 4.5. Section 4.6 describes the calculations for the importance of individual video packet. Simulation results are presented in Section 4.7. Finally, Section 4.8 concludes the chapter.

4.2 System Model

We consider a wireless network with a set of M devices $\mathcal{M} = \{U_1, ..., U_M\}$.[*] Each device in $\mathcal{M}$ is interested in receiving the same set of N source packets $\mathcal{N} = \{P_1, ..., P_N\}$. 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. Due to erasures in long-range wireless channels, a subset of devices from $\mathcal{M}$ receive each broadcast packet. We assume that at least one device receives each broadcast packet.

The second phase starts after N time slots (referred to as the *D2D phase*), in which the devices cooperate with each other to recover their missing packets using local area channels. There is a limit on the number of allowable time slots Θ used in the D2D phase as the deadline for delivering N packets expires after Θ D2D time slots. This deadline constraint arises from the minimum delivery delay requirement in real-time video streaming applications. At any D2D time slot $t \in [1, 2, ..., \Theta]$, we can compute the number of remaining time slots for delivering N packets as, $Q(t) = \Theta - t + 1$. 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) $\mathbf{Y} = [y_{i,k}], \forall (U_i, U_k) \in \mathcal{M}$, such that:

$$y_{i,k} = \begin{cases} 1 - \epsilon_{i,k} & \text{if } U_i \text{ is directly connected to } U_k, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

$$y_{i,i} = 1, \quad \forall U_i \in \mathcal{M}. \tag{4.2}$$

[*] Throughout this chapter, we use calligraphic letters to denote sets and their corresponding capital letters to denote the cardinalities of these sets. Let $\mathcal{M}$ be a set. Then, $M = |\mathcal{M}|$.

Here, (4.2) is a conventional notation of the self transmission, but does not have a physical meaning or use. Further, a packet transmission from device U_i to device U_k is subject to an independent Bernoulli erasure with probability $\epsilon_{i,k}$. Therefore, an upper bound on the channel erasure probability is close to 1. As a result, an erasure probability as high as 0.99 is theoretically possible but is not practical. Therefore, in the simulation section of this chapter, we consider the erasure probabilities in the rage $[0.01, 0.3]$ to represent various physical channel conditions of communication networks. While an erasure probability 0.01 represents a good channel condition, an erasure probability 0.3 represents a harsh channel condition due to the long distance, interference, low transmission power, shadowing and deep fading.

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 in erasure probability, to another channel connecting another pair of devices. In fact, a device $U_i \in \mathcal{M}$ is directly connected to a subset of devices in $\mathcal{M}$ depending on the location of the device in the network.

Example 7. *An example of SCM with $M = 4$ devices is given as follows:*

$$\mathbf{Y} = \begin{pmatrix} 1 & 0.84 & 0 & 0 \\ 0.84 & 1 & 0.75 & 0 \\ 0 & 0.75 & 1 & 0.91 \\ 0 & 0 & 0.91 & 1 \end{pmatrix}. \tag{4.3}$$

The SCM in (4.3) represents a line network shown in Fig. 4.1. In this example, device U_1 is not directly connected to device U_3 and thus, $y_{1,3} = 0$. Moreover, device U_1 is directly connected to device U_2 with packet reception probability $y_{1,2} = 1 - \epsilon_{1,2} = 0.84$.

Definition 11. *(Coverage Zone) The coverage zone of transmitting device U_i (denoted by $\mathcal{Y}_i$) is defined as the set of neighboring devices that are directly connected to it using local area wireless channels. In other words, $\mathcal{Y}_i = \{U_k \mid y_{i,k} \neq 0\}$.*

Definition 12. *(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 U_i and U_r of device U_k transmit simultaneously, their transmissions*

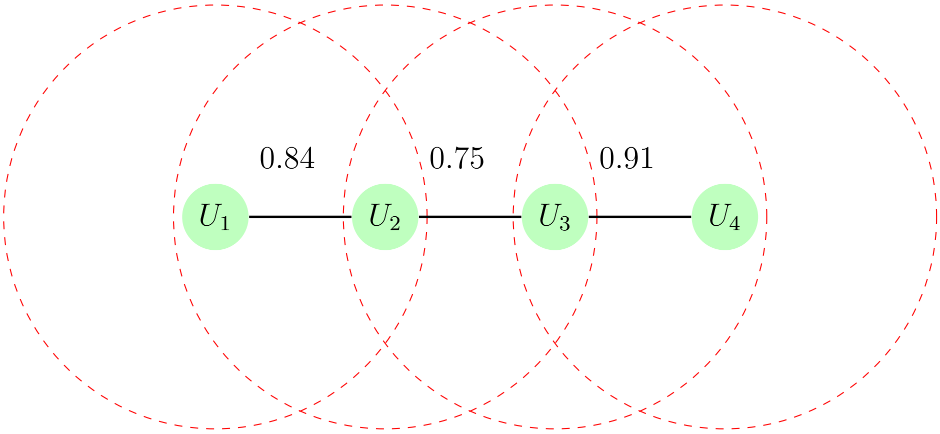


Figure 4.1: A line network corresponding to SCM in (4.3). In this figure, the dotted red circles illustrate the coverage zones of individual devices and a solid line between two devices represents a channel connecting these two devices.

will collide at device U_k and it 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 $M \times N$ *feedback status matrix (FSM)* $\mathbf{F} = [f_{k,l}]$, $\forall U_k \in \mathcal{M}, P_l \in \mathcal{N}$, such that:

$$f_{k,l} = \begin{cases} 0 & \text{if packet } P_l \text{ is received by device } U_k, \\ 1 & \text{if packet } P_l \text{ is missing at device } U_k. \end{cases} \tag{4.4}$$

Example 8. *An example of FSM with $M = 4$ devices and $N = 3$ packets is given as follows:*

$$\mathbf{F} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}. \tag{4.5}$$

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

1. The *Has set* ($\mathcal{H}_k$) of device U_k is defined as the set of packets that are successfully received by device U_k. In (4.5), the Has set of device U_1 is $\mathcal{H}_1 = \{P_3\}$.

2. The *Wants set* ($\mathcal{W}_k$) of device U_k is defined as the set of packets that are missing at

device U_k. In other words, $\mathcal{W}_k = \mathcal{N} \setminus \mathcal{H}_k$. In (4.5), the Wants set of device U_1 is $\mathcal{W}_1 = \{P_1, P_2\}$.

The set of devices having *non-empty Wants sets* is denoted by $\mathcal{M}_w$. This set can be defined as: $\mathcal{M}_w = \{U_k \mid \mathcal{W}_k \neq \varnothing\}$. At any given time slot t, a device U_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., $W_k \geq Q, \forall U_k \in \mathcal{C}$).
- The *non-critical set* of devices ($\mathcal{B}$) 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., $W_k < Q, \forall U_k \in \mathcal{B}$).

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

Definition 13. *(Instantly Decodable Packet) A transmitted packet is instantly decodable for device U_k if it contains exactly one source packet from $\mathcal{W}_k$.*

Definition 14. *(Targeted Device) Device U_k is targeted by transmitting device U_i with packet P_l at time slot t when device U_k belongs to the coverage zone of a single transmitting device U_i and will immediately decode packet P_l upon receiving the transmitted packet from device U_i.*

Definition 15. *(Individual Completion Time) At any time slot t, individual completion time of device U_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 U_k for W_k missing packets can be $T_{W_k} = W_k, W_k + 1, ...$ depending on the number of time slots in which this device is targeted with a new packet (i.e., satisfies Definition 14) and the channel erasures experienced by this device in those transmissions.

Definition 16. *(Completion Time of All Non-critical Devices) At any time slot t, individual completion times of all non-critical devices (denoted by T_B) is the total number of time slots required to deliver all the missing packets to all non-critical devices in $\mathcal{B}$.*

4.2.1 Centralized Protocol for Implementing the System

In this chapter, we adopt a centralized approach to solve the maximum video quality problem in D2D networks for the following advantages [8]. First, it satisfies and executes reliably the service provider policy of maximizing the mean video quality over all devices. Second, it can optimally solve the maximum video quality problem using the high computational capabilities of the central station. Third, the centralized implementation requires low processing capabilities at the devices. Finally, it is adaptive to the mobility of devices. On the other hand, a fully distributed approach can be adopted to perform the decision making process at the devices and the base station separately. However, such distributed approach suffers from high processing requirements at the devices, high sensitivity to devices mobility and high security risks [8]. Therefore, we now describe the possible implementation processes of the centralized IDNC system, where a central station forms the SCM $\mathbf{Y}$ and the FSM $\mathbf{F}$, and coordinates the global decision making process in each time slot.

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 $\mathcal{O}(M-1)$ 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. With respect to the stringent deadline for delivering N video packets, the locations of the devices in a network are assumed to be static. However, the devices' locations can change from a set of packets delivery to another set, in which case the coverage zones are determined again using M^2 bits. Therefore, for N video packets delivery, the communication overhead is M^2 bits.

Note that this chapter considers an idealistic Hello message of 1 bit. In practice, a transmitted packet includes several overhead bits such as control bits, an ID of the source, a tag for the control packet and redundancy bits for the integrity of data. Since the size of the Hello message has no impact on the network coding solution, we consider a simple 1 bit message. In fact, the Hello messages, regardless of whether network coding is used or not, are needed to be exchanged to form a D2D communication network and finding the optimal size for Hello messages belongs to the study of D2D network design and analysis.

Packet Reception Probability

In this chapter, 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. Such test messages can be repeated a number of times in order to make a more accurate estimation of the erasure probability. 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 $M^2\lceil \log_2 100 \rceil = 7M^2$ bits. Using this information, the central station forms the SCM $\mathbf{Y}$. Note that the frequency of updating the channel erasure probabilities depends on the available network resources and the speed at which the fading gain of the channel changes. For example, the fading gain of the channels fluctuate for mobile devices and, consequently, the frequent update on the channel status is required to reduce the channel estimation error.

FSM 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 FSM $\mathbf{F}$ in each time slot.

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 while using the indices of individual packets. In fact, a packet combination can be formed by XORing $\mathcal{O}(N)$ 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 at most $\frac{M}{2}$ transmitting devices since a device cannot receive and transmit simultaneously. The overall communication overhead to inform at most $\frac{M}{2}$ transmitting devices about their packet combinations is $\mathcal{O}(MN)$ bits, which is negligible compared to the typical size of a packet in wireless networks.

4.2.2 Importance of an Individual Packet

The importance of a 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 [15]. To compute the importance of packet P_l, we follow a similar approach as in [14, 15] 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 U_k at time slot t as:

$$D_k^{(t)} = \sum_{P_l \in \mathcal{W}_k} \delta_{k,l} \tag{4.6}$$

where $\delta_{k,l}$ is the importance of missing packet P_l at device U_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 4.6.

$$\begin{array}{cccc} \begin{array}{|c|}\hline 0\\\hline 1\\\hline\end{array} & \begin{array}{|c|}\hline 1\\\hline 0\\\hline 0\\\hline\end{array} & \begin{array}{|c|c|}\hline 0 & 1\\\hline 0 & 0\\\hline 1 & 0\\\hline\end{array} & \begin{array}{|c|}\hline 0\\\hline 0\\\hline\end{array} \\ \\ \mathbf{F}_1 & \mathbf{F}_2 & \mathbf{F}_3 & \mathbf{F}_4 \end{array}$$

Figure 4.2: Four LSMs for four devices corresponding to SCM in (4.3) and FSM in (4.5)

4.3 Cooperation aware IDNC Graph

In this section, we define a cooperation aware IDNC (C-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.

4.3.1 Vertex Set

To define the vertex set $\mathcal{V}$ of the C-IDNC graph $\mathcal{G}$, given FSM $\mathbf{F}$ at time slot t, we form an $Y_i \times H_i$ *local status matrix (LSM)* $\mathbf{F}_i = [f_{k,l}]$, $\forall U_k \in \mathcal{Y}_i, P_l \in \mathcal{H}_i$, for a device $U_i \in \mathcal{M}$ such that:

$$f_{k,l} = \begin{cases} 0 & \text{if packet } P_l \text{ is received by device } U_k, \\ 1 & \text{if packet } P_l \text{ is missing at device } U_k. \end{cases} \tag{4.7}$$

Note that the rows in LSM $\mathbf{F}_i$ represent the devices which are in the coverage zone of device U_i and the columns in LSM $\mathbf{F}_i$ represent the packets in the Has set of device U_i which are used for forming a transmitted packet from device U_i. Fig. 4.2 shows four LSMs for four devices corresponding to SCM in (4.3) and FSM in (4.5).

We generate a vertex for a missing packet in each LSM at C-IDNC graph $\mathcal{G}$. In fact, for each LSM $\mathbf{F}_i, \forall U_i \in \mathcal{M}$, a vertex $v_{i,kl}$ is generated for a packet $P_l \in \{\mathcal{H}_i \cap \mathcal{W}_k\}, \forall U_k \in \mathcal{Y}_i$.[†] In other words, a vertex is generated for a missing packet of another device in $\mathcal{Y}_i$, which

[†] Note that vertex $v_{i,kl}$ represents a transmission from device $U_i \in \mathcal{M}$ to a neighboring device $U_k \in \mathcal{Y}_i$ with packet P_l.

also belongs to the Has set $\mathcal{H}_i$ of potential transmitting device U_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 C-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.

4.3.2 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:

- **C1**: $P_l \neq P_n$ and $U_k = U_m$. In other words, two vertices are induced by different missing packets P_l and P_n at the same device U_k.
- **C2**: $U_k \neq U_m$ 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 U_k and U_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 the XOR combination of $P_l \oplus P_n$.

4.3.3 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**: $U_i \neq U_r$ and $U_k = U_m \in \{\mathcal{Y}_i \cap \mathcal{Y}_r\}$. In other words, two vertices representing the transmissions from two different devices U_i and U_r to the same device U_k in the coverage zones of both transmitting devices U_i and U_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**: $U_i \neq U_r$ and $U_k \neq U_m$ but $U_k \in \{\mathcal{Y}_i \cap \mathcal{Y}_r\}$ or $U_m \in \{\mathcal{Y}_i \cap \mathcal{Y}_r\}$. In other words, two vertices representing the transmissions from two different devices U_i and U_r to two different devices U_k and U_m, but at least one of these two devices U_k and U_m is in the coverage zones of both transmitting devices U_i and U_r. This prohibits transmission

from device U_r to device U_m in the case of transmission from device U_i to device U_k, and vice versa.

- **C5**: $U_i \neq U_r$ but $U_i = U_m$ or $U_r = U_k$. In other words, two vertices representing the transmissions from two different devices U_i and U_r, but at least one of these two devices U_i and U_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.

4.3.4 Maximal Independent Sets

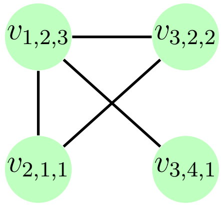
With this graph representation, we can define all feasible coding and transmission conflict-free decisions by the set of all maximal independent sets in C-IDNC graph $\mathcal{G}$.

Definition 17. *(Independent Set) An independent set or a stable set in a graph is a set of pairwise non-adjacent vertices.*

Definition 18. *(Maximal Independent Set) A maximal independent set (denoted by κ) 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 [102].*

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

Example 9. *The new C-IDNC graph $\mathcal{G}$ corresponding to SCM in* (4.3) *and FSM in* (4.5) *is shown in Fig. 4.3. An example of a maximal independent set of this graph is $\kappa = \{v_{2,1,1}, v_{3,4,1}\}$. Here, the set of transmitting devices is $\mathcal{Z}(\kappa) = \{U_2, U_3\}$ and the set of targeted devices is $\mathcal{X}(\kappa) = \{U_1, U_4\}$.*



$$\begin{aligned}\kappa_1 &= \{v_{1,2,3}\}\\ \kappa_2 &= \{v_{2,1,1}, v_{3,4,1}\}\\ \kappa_3 &= \{v_{3,4,1}, v_{3,2,2}\}\end{aligned}$$

Figure 4.3: Cooperation aware IDNC graph corresponding to SCM in (4.3) and FSM in (4.5).

4.4 Minimum Video Distortion Problem Formulation

We now discuss the characteristics of the minimum video distortion problem and demonstrate 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 FSMs 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 state 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. Consequently, 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 Θ used in the D2D phase, over which the decisions are made. The MDP problem is a finite horizon problem with Θ time slots.

2. *State Space* $\mathcal{S}$: States are defined by all possibilities of FSM $\mathbf{F}$ that may occur during the D2D phase. FSM 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, $\mathbf{h}(s) = [H_1(s), ..., H_M(s)]$ and $\mathbf{w}(s) = [W_1(s), ..., W_M(s)]$. The state at the starting of the D2D phase is denoted by s_a and its Has and Wants vectors are denoted by $\mathbf{h}(s_a) = [H_1(s_a), ..., H_M(s_a)]$ and $\mathbf{w}(s_a) = [W_1(s_a), ..., W_M(s_a)]$.

   Given FSM $\mathbf{F}$ is an $M \times N$ binary matrix, the size of the state space is $|\mathcal{S}| = \mathcal{O}(2^{MN})$. 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 $|\mathcal{S}| = 2^{MN-(\Sigma_{U_i \in \mathcal{M}} H_i(s_a))}$.

3. *Action Space* $\mathcal{A}(s)$: The action space for each state s consists of the set of all possible maximal independent sets in C-IDNC graph $\mathcal{G}(s)$. The size of the action space for a given state $\mathbf{F}(s)$ is $|\mathcal{A}(s)| = \mathcal{O}(3^{|\mathcal{V}|/3})$ [102], where $|\mathcal{V}|$ is the size of the vertex set $\mathcal{V}$ in graph $\mathcal{G}(s)$.

4. *State-Action Transition Probability* $\mathbb{P}_a(s, \acute{s})$: The state-action transition probability $\mathbb{P}_a(s, \acute{s})$ for an action $a = \kappa(s)$ can be defined based on the possibilities of the variations in FSM $\mathbf{F}(s)$ 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 $\mathcal{S}(s, a) = \{\acute{s} | \mathbb{P}_a(s, \acute{s}) > 0\}$. To define $\mathbb{P}_a(s, \acute{s})$, we first introduce the following two sets:

$$\mathcal{T} = \{U_k | U_k \in \mathcal{X}(\kappa), W_k(\acute{s}) = W_k(s) - 1\} \tag{4.8}$$

$$\tilde{\mathcal{T}} = \{U_k | U_k \in \mathcal{X}(\kappa), W_k(\acute{s}) = W_k(s)\} \tag{4.9}$$

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 $\mathbb{P}_a(s, \acute{s})$ as follows:

$$\mathbb{P}_a(s, \acute{s}) = \prod_{U_k \in \mathcal{T} : v_{i,kl} \in \kappa(s)} (1 - \epsilon_{i,k}) \times \prod_{U_k \in \tilde{\mathcal{T}} : v_{i,kl} \in \kappa(s)} (\epsilon_{i,k}) \tag{4.10}$$

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 $U_k \in \mathcal{M}_w(s)$ is defined as the expected video distortion reduction at device U_k at the successor state s'. We can calculate the expected reward of action $a = \kappa(s)$ on each targeted device $U_k \in \mathcal{X}(a)$ 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 $U_k \in \{\mathcal{M}_w(s) \setminus \mathcal{X}(a)\}$ as $\bar{r}_k(s, a | U_k \in \mathcal{M}_w(s) \setminus \mathcal{X}(a)) = 0$. 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:

$$\bar{r}(s, a) = \sum_{U_k \in \mathcal{M}_w(s)} \bar{r}_k(s, a) = \sum_{U_k \in \mathcal{X}(a) : v_{i,kl} \in \kappa(s)} \delta_{k,l}(1 - \epsilon_{i,k}). \tag{4.11}$$

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 π. It can be recursively expressed as [103]:

$$V_\pi(s) = \bar{r}(s, a) + \sum_{s' \in \mathcal{S}(s,a)} \mathbb{P}_a(s, \acute{s}) V_\pi(s'), \qquad \forall s \in \mathcal{S}. \tag{4.12}$$

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 defined as [103]:

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}(s)} \{V_\pi(s)\}, \quad \forall s \in \mathcal{S}. \tag{4.13}$$

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 in the order of $\mathcal{O}(|\mathcal{S}|^2|\mathcal{A}|)$. Based on the sizes of state space $\mathcal{S}$ and action space $\mathcal{A}(s)$ 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.

4.5 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.[‡]

- To prioritize the critical devices, we partition the C-IDNC graph $\mathcal{G}$ into critical graph

[‡] 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.

$\mathcal{G}_c$ and non-critical graph $\mathcal{G}_b$. The critical graph $\mathcal{G}_c$ includes the vertices representing transmissions from all devices to the critical devices. Similarly, the non-critical graph $\mathcal{G}_b$ 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 κ_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 smaller numbers of missing packets. Consequently, we select a non-critical maximal independent set κ_b^* over non-critical graph $\mathcal{G}_b$ that increases the probability of delivering all the missing packets to all non-critical devices before the deadline. However, κ_b^* is selected without violating the independent set constraint (thus, prohibiting coding and transmission conflicts) for the targeted critical devices in κ_c^*.

Remark 6. *The proposed heuristic in this chapter is different from the one in Chapter 3 due to the difference in the application requirements. In fact, Chapter 3 studies layered video delivery applications wherein all packets comprising of a layer need to be successfully decoded for any of these packets to be usable by the application layer. On the contrary, this chapter considers content delivery applications wherein packets have unequal importance and each received packet contributes to the overall content quality of the application layer.*

4.5.1 Maximal Independent Set Selection Algorithm over Critical Graph

In this subsection, we select a critical maximal independent set κ_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 κ_c and $D_k^{(t+1)}(\kappa_c)$ as the expected individual video distortion of critical device $U_k \in \mathcal{C}(t)$ at time slot $t+1$ due to selecting κ_c. This can be expressed as:

$$D_k^{(t+1)}(\kappa_c) = \begin{cases} D_k^{(t)} & \text{if } U_k \in \mathcal{C}(t) \setminus \mathcal{X}_c(\kappa_c), \\ D_k^{(t)} - \delta_{k,l}(1-\epsilon_{i,k}) & \text{if } U_k \in \mathcal{X}_c(\kappa_c) : v_{i,kl} \in \kappa_c \end{cases} \tag{4.14}$$

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 $t+1$. We now express the expected sum video distortion of all critical devices after time slot t as:

$$\sum_{U_k \in \mathcal{C}(t)} \mathbb{E}[D_k^{(t+1)}(\kappa_c)] = \sum_{U_k \in \{\mathcal{C}(t) \setminus \mathcal{X}_c(\kappa_c)\}} D_k^{(t)} + \sum_{U_k \in \mathcal{X}_c(\kappa_c)} D_k^{(t)} - \delta_{k,l}(1-\epsilon_{i,k}). \tag{4.15}$$

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

$$\begin{aligned} \kappa_c^* &= \arg\min_{\kappa_c \in \mathcal{G}_c} \{ \sum_{U_k \in \{\mathcal{C}(t) \setminus \mathcal{X}_c(\kappa_c)\}} D_k^{(t)} + \sum_{U_k \in \mathcal{X}_c(\kappa_c)} D_k^{(t)} - \delta_{k,l}(1-\epsilon_{i,k}) \} \\ &= \arg\max_{\kappa_c \in \mathcal{G}_c} \{ \sum_{U_k \in \mathcal{X}_c(\kappa_c)} \delta_{k,l}(1-\epsilon_{i,k}) \}. \end{aligned} \tag{4.16}$$

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 chapter, we use the Bron-Kerbosch algorithm to find κ_c^* among all maximal independent sets in $\mathcal{G}_c$ [104]. In the following two subsections, 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 κ_b^*.

4.5.2 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 FSM. 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 U_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 U_k being equal to $W_k + x, x \in [0, 1, ..., Q - W_k]$ can be expressed using negative binomial distribution as:

$$\mathbb{P}[T_{W_k} = W_k + x] = \binom{W_k + x - 1}{x} (\bar{\epsilon}_k)^x (1 - \bar{\epsilon}_k)^{W_k}, \tag{4.17}$$

where, $\bar{\epsilon}_k$ is the average of the channel erasure probabilities connecting device U_k to other devices. In other words, $\bar{\epsilon}_k = \frac{\sum_{U_i \in \mathcal{I}} \epsilon_{i,k}}{|\mathcal{I}|}$, where $\mathcal{I} = \{U_i | y_{i,k} \neq 0, U_i \neq U_k\}$. This average erasure probability represents that device U_k can receive its missing packets from any other neighboring device in the remaining time slots. Note that the consideration of the average erasure probability leads to an approximate equation (4.17) with the benefits of computational simplicity.

Consequently, the probability that the individual completion time T_{W_k} of the non-critical device U_k is less than or equal to the remaining Q time slots can be expressed as:

$$\mathbb{P}[T_{W_k} \leq Q] = \sum_{x=0}^{Q-W_k} \mathbb{P}[T_{W_k} = W_k + x]. \tag{4.18}$$

We now consider a scenario with a set of non-critical devices $\mathcal{B}$ 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 the 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 may limit 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 completion time of all non-critical devices in $\mathcal{B}(t)$ are less than or equal to the remaining Q time slots (denoted by $\hat{\mathbb{P}}^{(t)}[T_B \leq Q]$) as:

$$\hat{\mathbb{P}}^{(t)}[T_B \leq Q] = \prod_{U_k \in \mathcal{B}(t)} \sum_{x=0}^{Q-W_k} \mathbb{P}[T_{W_k} = W_k + x]. \tag{4.19}$$

In the following subsection, we use expression (4.19) as a metric of selecting a non-critical maximal independent set in each time slot.

4.5.3 Maximal Independent Set Selection Algorithm over Non-critical Graph

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

Let us define $\mathcal{X}_b(\kappa_b)$ as the set of targeted non-critical devices in κ_b and $W_k^{(t+1)}(\kappa_b)$ as the expected number of missing packets at a non-critical device $U_k \in \mathcal{B}(t)$ at time slot $t+1$ due to selecting κ_b. This can be expressed as:

$$W_k^{(t+1)}(\kappa_b) = \begin{cases} W_k^{(t)} & \text{if } U_k \in \mathcal{B}(t) \setminus \mathcal{X}_b(\kappa_b), \\ (W_k^{(t)} - 1)(1 - \epsilon_{i,k}) + (W_k^{(t)})(\epsilon_{i,k}) & \text{if } U_k \in \mathcal{X}_b(\kappa_b) : v_{i,kl} \in \kappa_b \end{cases} \tag{4.20}$$

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 κ_b selection at time slot t, let $\hat{\mathbb{P}}^{(t+1)}[T_B \leq Q - 1]$ be the resulting upper bound on the probability that individual completion times of all non-critical devices in $\mathcal{B}(t)$, starting from the successor time slot $t+1$, are less than or equal to the remaining $Q-1$ time slots. We can express probability $\hat{\mathbb{P}}^{(t+1)}[T_B \leq Q-1]$ as:

$$\begin{aligned}\hat{\mathbb{P}}^{(t+1)}[T_B \leq Q-1] = \prod_{U_k \in \mathcal{X}_b(\kappa_b)} &(\mathbb{P}[T_{W_k-1} \leq Q-1].(1-\epsilon_{i,k}) + \mathbb{P}[T_{W_k} \leq Q-1].(\epsilon_{i,k})) \\ &\times \prod_{U_k \in \mathcal{B} \setminus \mathcal{X}_b(\kappa_b)} \mathbb{P}[T_{W_k} \leq Q-1] \end{aligned} \tag{4.21}$$

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 $\hat{\mathbb{P}}^{(t+1)}[T_B \leq Q-1]$ as a non-critical maximal independent set κ_b^* selection problem over non-critical subgraph $\mathcal{G}_b(\kappa_c^*)$ such that:

$$\begin{aligned}\kappa_b^* &= \arg \max_{\kappa_b \in \mathcal{G}_b(\kappa_c^*)} \left\{ \hat{\mathbb{P}}^{(t+1)}[T_B \leq Q-1] \right\} \\ &= \arg \max_{\kappa_b \in \mathcal{G}_b(\kappa_c^*)} \{ \prod_{U_k \in \mathcal{X}_b(\kappa_b)} (\mathbb{P}[T_{W_k-1} \leq Q-1].(1-\epsilon_{i,k}) + \mathbb{P}[T_{W_k} \leq Q-1].(\epsilon_{i,k})) \\ &\qquad \times \prod_{U_k \in \mathcal{B} \setminus \mathcal{X}_b(\kappa_b)} \mathbb{P}[T_{W_k} \leq Q-1] \} \end{aligned} \tag{4.22}$$

In other words, the problem of maximizing probability $\hat{\mathbb{P}}^{(t+1)}[T_B \leq Q-1]$ is equivalent to finding all maximal independent sets in the non-critical subgraph $\mathcal{G}_b(\kappa_c^*)$, and selecting the maximal independent set among them that results in the maximum probability $\hat{\mathbb{P}}^{(t+1)}[T_B \leq Q-1]$. Similar to Section 4.5.1, we use the Bron-Kerbosch algorithm to find κ_b^* among all maximal independent sets in $\mathcal{G}_b(\kappa_c^*)$. The computational complexity of using Bron-Kerbosch algorithm for a graph $\mathcal{G}$ with $|\mathcal{V}|$ vertices is $\mathcal{O}(3^{\frac{|\mathcal{V}|}{3}})$, where the number of vertices is $\mathcal{O}(M^2N)$ in C-IDNC graph. We see that the complexity of Bron-Kerbosch algorithm grows quickly with the increase in the number of devices M in a network. The algorithm is still applicable to moderate sized social networks (rather than public networks) that only involve trusted and friendly users. For a large number of devices M, a maximal independent set can be

Algorithm 3: Two-Stage Maximal Independent Set (TS-MIS) Selection Algorithm

```
**Construct** C-IDNC graph $\mathcal{G}$ according to all LSMs $\mathbf{F}_i, \forall U_i \in \mathcal{M}$;
Partition $\mathcal{G}$ into $\mathcal{G}_c$ and $\mathcal{G}_b$ according to the critical and the non-critical devices;
Initialize $\kappa_c^* = \varnothing$ and $\kappa_b^* = \varnothing$;
if $\mathcal{G}_c \neq \varnothing$ then
    Select $\kappa_c^* = \arg\max_{\kappa_c \in \mathcal{G}_c} \left\{\sum_{U_k \in \mathcal{X}_c} \delta_{k,l}(1-\epsilon_{i,k})\right\}$ following (4.16) ;
end
Update subgraph $\mathcal{G}_b(\kappa_c^*)$;
if $\mathcal{G}_b(\kappa_c^*) \neq \varnothing$ then
    Select $\kappa_b^* = \arg\ \max_{\kappa_b \in \mathcal{G}_b(\kappa_c^*)} \left\{\hat{\mathbb{P}}^{(t+1)}[T_B \leq Q-1]\right\}$ following (4.22) ;
end
Set $\kappa^* \leftarrow \kappa_c^* \cup \kappa_b^*$;
```

selected based on a greedy vertex search approach proposed in [93].

The final maximal independent set κ^* is the union of two maximal independent sets κ_c^* and κ_b^* (i.e., $\kappa^* = \{\kappa_c^* \cup \kappa_b^*\}$). The set of transmitting devices are determined using the indices of all the vertices in κ^*. Each of the selected transmitting devices forms a coded packet by XORing the source packets identified by the vertices in κ^* representing transmission from that device. The proposed two-stage maximal independent set (TS-MIS) selection algorithm is summarized in Algorithm 3.

4.6 Calculations of Packet Importance for a Real Video Sequence

In this section, we first discuss the H.264/SVC video test sequence used in this chapter and then provide details about the calculations for individual packet importance. We use a standard video sequence, *Soccer* [100]. This sequence is in common intermediate format (CIF, i.e., 352 × 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 [16, 101].[§] Moreover, we encode the video sequence considering a temporal dependence among the video frames so that several complete frames (i.e., complete pictures) can be dropped and a basic video quality can still be recovered. Such temporal encoding is often used in practice for video

[§] Note that our proposed IDNC framework is general and can be applied to a single layer H.264/AVC video sequence considered in [15, 56].

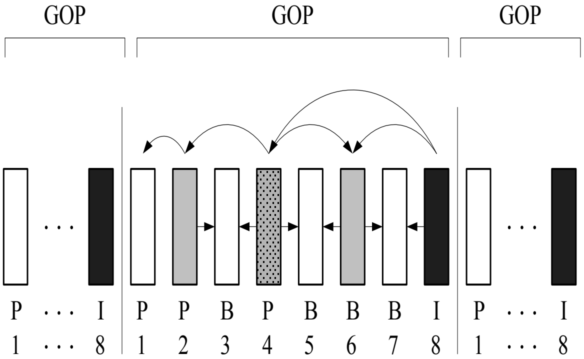


Figure 4.4: A closed GOP with 4 layers and 8 frames (a sequence of I, P and B frames).

frame rate reduction in networks with limited transmission capacity [13]. We consider the size of each group of pictures (GOP) is 8 frames, which results in 38 GOPs for the video sequence. As shown in Fig. 4.4, each GOP consists of a sequence of I, P and B frames that are encoded into four video layers. Further, the frames belonging to the same video layer are represented by the identical shade and the more important video layers are represented by the darker shades.

We use 1500 bytes as the packet length. 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 σ bytes, the required number of packets for this frame and layer can be calculated as $\lceil \frac{\sigma}{1400} \rceil$. Here, the ceiling function $\lceil . \rceil$ 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 λ bits per second and a packet length is 1500×8 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 chapter, 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 [44], we obtain α_{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,

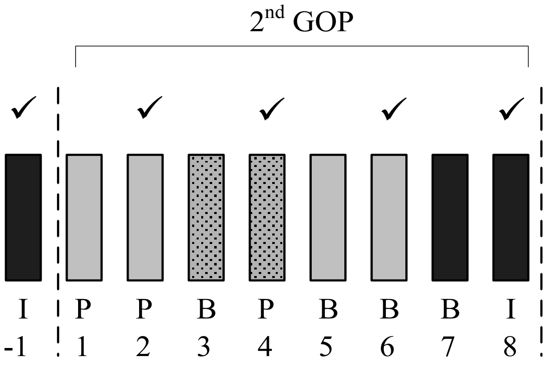


Figure 4.5: The nearest decoded frames are used to conceal the loss of undecoded frames.

if the first ℓ layers of four video layers are docodable ($0 \leq \ell \leq 4$).[¶] 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 average PSNR of the second GOP can be calculated as:

$$\bar{\alpha}_2 = \frac{\sum_{f_i \in \mathcal{R}} \alpha_{f_i,f_i} + \sum_{f_i \notin \mathcal{R}} \alpha_{f_i,f_j}}{8} \tag{4.23}$$

where, $\mathcal{R}$ is the set of frames of the decodable layers of the second GOP. The drop in the average PSNR value represents that a fraction of 8 frames in a GOP is decoded and these decoded frames (i.e., recovered pictures) are displayed in place of the missing frames (i.e., dropped pictures).

Example 10. *Let us consider the GOP shown in Fig. 4.4. 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. 4.5 and the resulting average PSNR can be computed as:*

$$\bar{\alpha}_2 = \frac{\alpha_{f_1,f_2} + \alpha_{f_2,f_2} + \alpha_{f_3,f_4} + \alpha_{f_4,f_4} + \alpha_{f_5,f_6} + \alpha_{f_6,f_6} + \alpha_{f_7,f_8} + \alpha_{f_8,f_8}}{8} \tag{4.24}$$

Remark 7. *(PSNR without Error) The average PSNR of the encoded Soccer sequence over*

[¶] Note that the ℓ-th layer of a scalable video can be decoded only if all packets in the first ℓ layers are received before the deadline.

38 *GOPs is* 35.64 *decibel (dB) if there is no error in the sequence.*

4.7 Simulation Results

In this section, we present the mean PSNR performance of different algorithms in various scenarios. The mean PSNR is calculated by taking average of the received PSNR at all M devices at the end of the deadline. We compare the performance of the BIA that solves the formulated MDP problem and the proposed TS-MIS algorithm to the following algorithms:

- 'Fully Connected Distortion (FCD)' algorithm [56] 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 [26] that considers a partially connected network and uses IDNC to serve the maximum number of devices with any new packet in each time 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 [88] for a fully connected D2D network and in [52] for a PMP network.

We first consider a line network with $M = 4$ devices described in (4.3) and encode four video layers of *Soccer* video sequence into four different packets, i.e., $N = 4$. As discussed earlier, the modelling and computational complexities of the BIA scale with the size of the state space $|\mathcal{S}|$, which is $\mathcal{O}(2^{16})$ even for $M = N = 4$. Moreover, as discussed in Section 4.2, 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 reflect the erasures in long-range wireless channels.

Fig. 4.6 shows the mean PSNR achieved by different algorithms against the different number of allowable D2D time slots Θ (i.e., different deadlines). From this figure, we can see

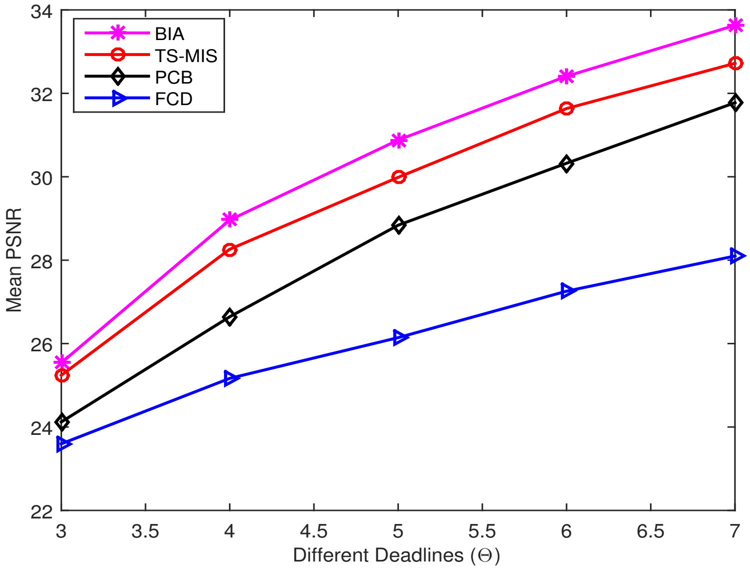


Figure 4.6: Mean PSNR versus different deadlines Θ.

that our proposed BIA and TS-MIS algorithms quickly increase the received PSNR at the devices with the increase in the deadline. Indeed, both BIA and TS-MIS algorithms use the new C-IDNC graph to make 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 without the hard deadline consideration.

Fig. 4.7 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 after 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 U_2 and U_3 experience a slightly better video quality compared to devices U_1 and U_4 since these are the intermediate devices in the line network shown in Fig. 4.1.

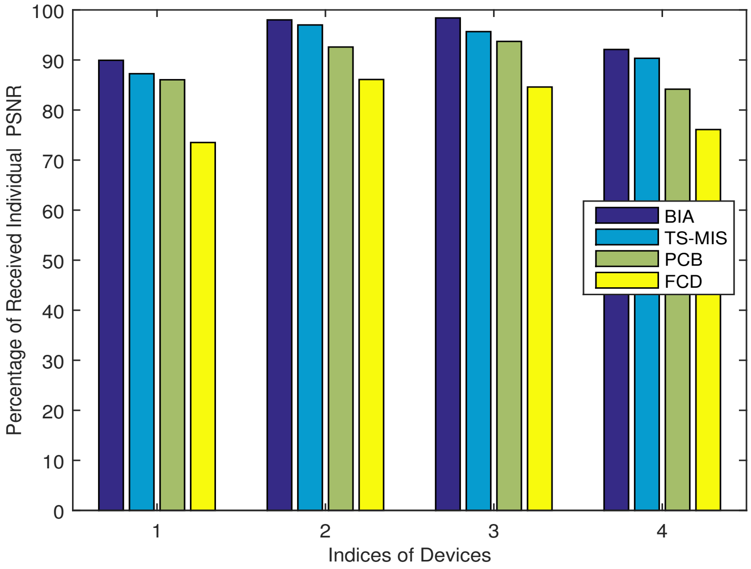


Figure 4.7: Histogram showing the percentage of received PSNR at individual devices before the deadline corresponding to the line network in Fig. 4.1.

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 only. We use the *Soccer* video sequence discussed in Section 4.6, where the packet length is 1500 bytes and each video layer is encoded into multiple packets. We compute the average connectivity index in the network as $\bar{y} = \frac{\sum_{(U_i,U_k)} \lceil y_{i,k} \rceil}{M \times M}$, which represents the average number of direct connections from a device to other devices. In the case of a fully connected network, the average connectivity index is $\bar{y} = 1$. As an example of a general network topology, an SCM with $M = 12$ devices, average connectivity index

$\bar{y} = 0.38$ and range of packet reception probabilities $[0.8, 0.96]$ is given as follows:

$$\mathbf{Y} = \begin{pmatrix} 1 & 0.85 & 0 & 0.89 & 0.82 & 0 & 0 & 0 & 0 & 0.88 & 0.96 & 0 \\ 0.85 & 1 & 0 & 0 & 0.85 & 0 & 0 & 0.87 & 0 & 0 & 0.91 & 0 \\ 0 & 0 & 1 & 0.93 & 0 & 0 & 0 & 0.88 & 0.87 & 0 & 0 & 0.87 \\ 0.89 & 0 & 0.93 & 1 & 0 & 0 & 0 & 0.95 & 0 & 0.81 & 0.88 & 0.89 \\ 0.82 & 0.85 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0.90 & 0.86 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.90 & 1 & 0 & 0.94 & 0 & 0 & 0 \\ 0 & 0.87 & 0.88 & 0.95 & 0 & 0.86 & 0 & 1 & 0 & 0 & 0.88 & 0 \\ 0 & 0 & 0.87 & 0 & 0 & 0 & 0.94 & 0 & 1 & 0 & 0 & 0.84 \\ 0.88 & 0 & 0 & 0.81 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0.96 & 0.91 & 0 & 0.88 & 0 & 0 & 0 & 0.88 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0.87 & 0.89 & 0 & 0 & 0 & 0 & 0.84 & 0 & 0 & 1 \end{pmatrix} \tag{4.25}$$

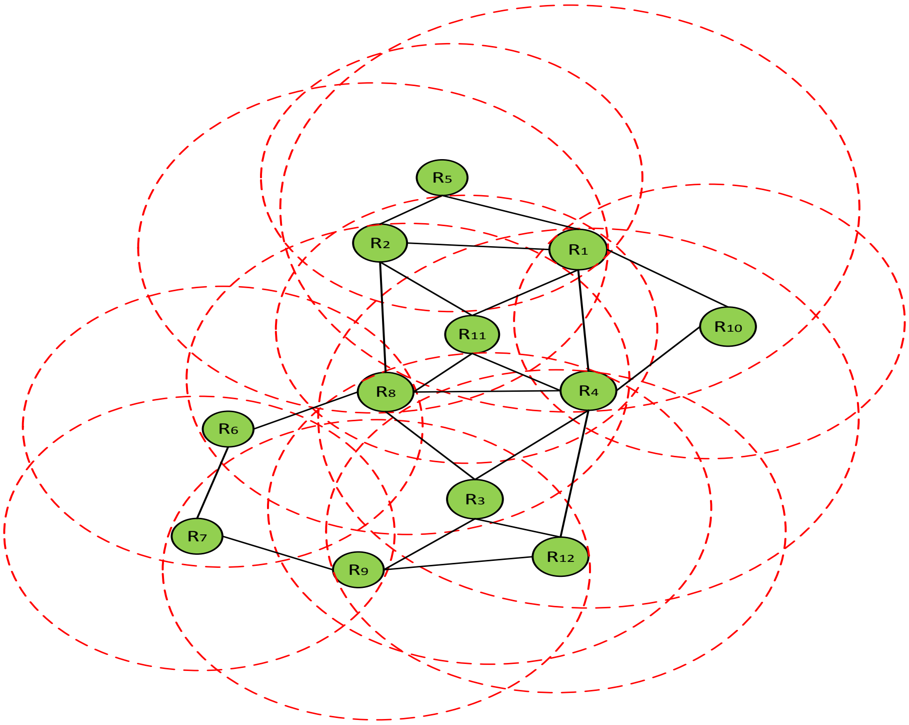


Figure 4.8: A network topology corresponding to SCM in (4.25)

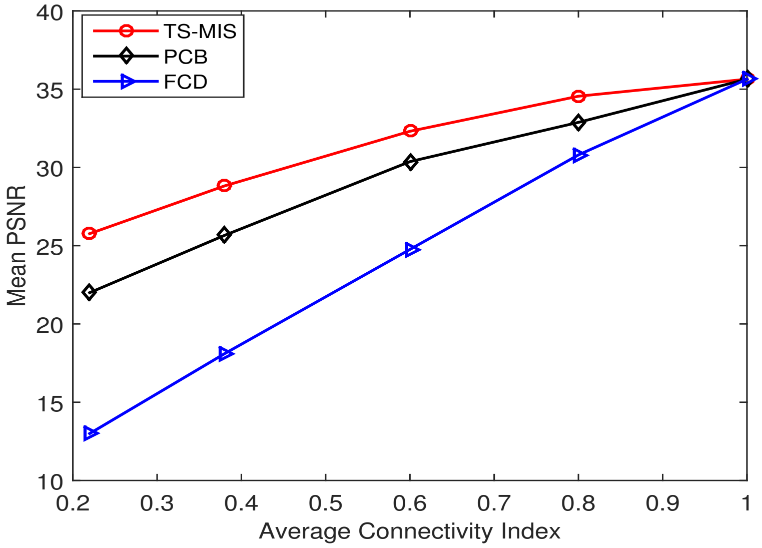


Figure 4.9: Mean PSNR versus different average connectivity indices $\bar{y}$.

The network topology corresponding to SCM in (4.25) is shown in Fig. 4.8. In this figure, the dotted red circles illustrate the coverage zones of individual devices and a solid line between two devices represents a channel connecting these two devices. For example, device U_2 is in the coverage zones of devices U_1, U_5 U_8 and U_{11} and thus, it is connected to devices U_1, U_5 U_8 and U_{11} with packet reception probabilities $y_{1,2} = 0.85$, $y_{5,2} = 0.85$, $y_{8,2} = 0.87$ and $y_{11,2} = 0.91$, respectively. Throughout this section, for a given number of devices M, value of average connectivity index $\bar{y}$ and range of packet reception probabilities of all channels, we generate an SCM $\mathbf{Y}$ in similar fashion and characterize an arbitrary network topology.

Fig. 4.9 shows the mean PSNR achieved by different algorithms against different average connectivity indices $\bar{y}$ (for $M = 12$ devices, $\Theta = 18$ D2D time slots and range of channels' packet reception probabilities $[0.8, 0.96]$). From this figure, we can see that our proposed TS-MIS algorithm substantially outperforms the FCD algorithm, except in the case of a fully connected network, i.e., $\bar{y} = 1$, when both algorithms use a large number of time slots $\Theta = 18$ to deliver the highest video quality 35.64 dB to all devices. As expected, FCD algorithm performs poorly in low average connectivity indices due to always selecting a single transmitting device. On the other hand, our proposed TS-MIS algorithm selects multiple transmitting devices by exploiting the partial connections among devices. From this figure, we can also see that the performance of the PCB algorithm considerably deviates

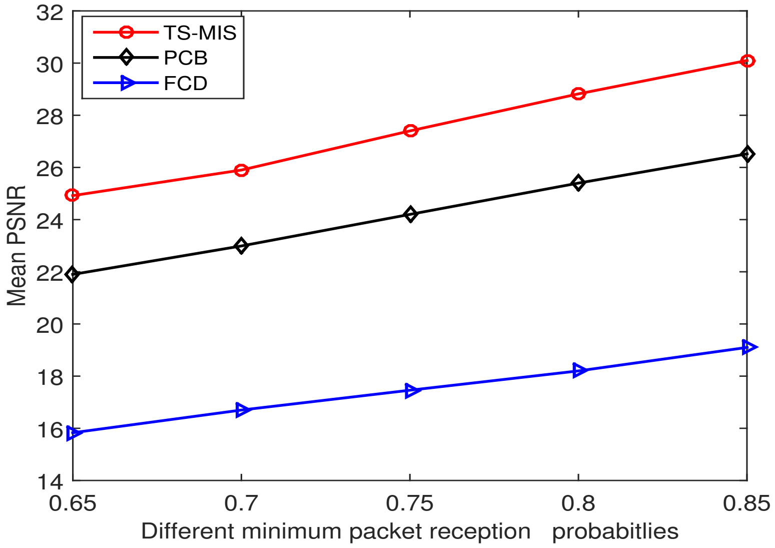


Figure 4.10: Mean PSNR versus different minimum packet reception probabilities of channels

from the TS-MIS algorithm since PCB algorithm does not address the hard deadline for the high importance video packets. Note that our proposed TS-MIS algorithm adopts a decision that does 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 it prioritizes the critical devices that have the number of missing packets greater than or equal to the number of remaining transmissions before the deadline.

Fig. 4.10 shows the mean PSNR achieved by different algorithms against different minimum packet reception probabilities of channels while always setting the maximum packet reception probability equal to 0.96 (for $M = 12$ devices, $\bar{y} = 0.38$ average connectivity index, $\Theta = 18$ D2D time slots). Such different ranges of packet reception probabilities represent

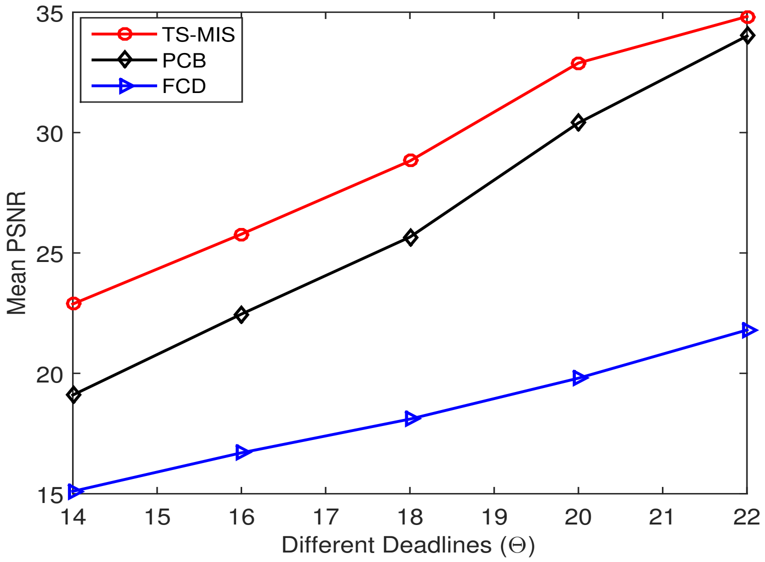


Figure 4.11: Mean PSNR versus different number of allowable time slots Θ

different levels of physical channel conditions experienced by devices. As expected, the performance of all algorithms improve with the increase in the packet reception probabilities of channels. In fact, in good channel conditions, the devices have a high possibility of successfully receiving most of the transmitted packets and therefore, most of the packets in a GOP before the deadline. This results in a low frame loss rate at individual devices. In other words, a few lost frames are replaced with the decoded frames to conceal the errors in the video sequence.

Fig. 4.11 shows the mean PSNR achieved by different algorithms against different deadlines Θ (for $\bar{y} = 0.38$ average connectivity index, $M = 12$ devices and range of packet reception probabilities $[0.8, 0.96]$). As expected, the performance of all algorithm improves with the increase in the deadline. Moreover, our proposed TS-MIS algorithm outperforms both FCD and PCB algorithms in all scenarios. In fact, our proposed TS-MIS algorithm makes an efficient decision by taking into account the unequal importance of video packets, hard deadline, erasures of wireless channels, coding and transmission conflicts.

Fig. 4.12 shows the mean PSNR achieved by different algorithms against different number of devices M (for $\bar{y} = 0.38$ average connectivity index, $\Theta = 18$ D2D time slots and range of packet reception probabilities $[0.8, 0.96]$). As expected, the mean PSNR of all algorithms decrease with the increase in the number of devices for a fixed deadline. Moreover, the

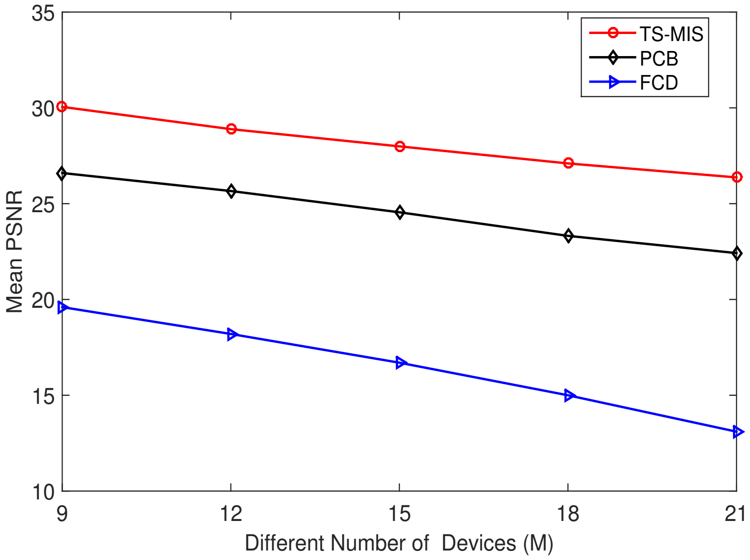


Figure 4.12: Mean PSNR versus different number of devices M

FCD algorithm performs poorly for a large number of devices due to always selecting a single transmitting device. Note that we have used another video sequence *Foreman* in the simulations and observed the similar results as in the case of *Soccer*.

4.8 Conclusion

In this chapter, we developed an efficient IDNC framework for distributing a real-time video sequence between a group of cooperative devices in a partially connected D2D network. In particular, we introduced a novel C-IDNC graph that represents all feasible coding and transmission conflict-free decisions in one unified framework. Using the new C-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 real video sequences showed that our proposed IDNC algorithms improve the received video quality compared to existing IDNC algorithms in terms of mean PSNR (dB).

Chapter 5

Packet Order Aware Network Codes in Heterogeneous Networks

5.1 Overview

In this chapter, we consider a heterogeneous network with coexistence of cellular and D2D networks [28–30], wherein devices are equipped with two wireless interfaces allowing simultaneous transmission and reception of packets. In particular, one interface communicates with the base station using a long-range technology, e.g., LTE, and the other interface communicates with near-by devices using a short-range technology, e.g., IEEE 802.11 ad-hoc mode. Given that cellular and local area channels operate concurrently using different parts of the spectrum, such system substantially improves the throughput, delay and robustness of the networks [24, 25]. Furthermore, for such heterogeneous networks, we emphasize on the significance of selecting transmitted packets for the base station and transmitting device with the awareness of underlying network coding operations.

As the data traffic of such coded heterogeneous networks, we consider time-critical and order-constrained applications requiring quick and reliable in-order decoding of packets [8,9]. Such order constraint affects the achievable throughput by using both network coding and heterogeneous network architecture. In fact, this chapter bridges the gap between the three different features (i.e., network coding, heterogeneous network architecture and packets' order constraint), and develops an unified framework that provides reliable in-order delivery of packets.

In particular, we are interested in designing an efficient IDNC framework that provides reliable in-order decoding of packets at the wireless devices in a heterogeneous network. For such scenarios, IDNC framework needs to take into account the order constraint of packets, erasures of wireless channels and dual interfaces of wireless devices in making decisions. In this context, our main contributions are as follows:

- We first introduce the delivery delay as a measure of degradation compared to the optimal in-order packet delivery to devices. We then define a dual interface IDNC graph, where a vertex represents a combination of transmitting device and missing packet, and an edge represents a coding or transmission conflict decision. In other words, this graph represents all feasible coding and transmission conflicts-free decisions.
- We formulate the minimum delivery delay problem as a maximum weight independent set selection problem over the graph, in which the weight of a vertex represents the expected delivery delay. Given the computational hardness of finding the optimal solution, we further propose a delivery delay reduction heuristic based on a greedy vertex search.
- Simulation results demonstrate that the proposed IDNC algorithm effectively reduces the delivery delay compared to the existing network coding algorithms that consider either dual interface scenarios for reducing the number of required transmissions [25] or in-order packet delivery for single interface scenarios [63].

The rest of this chapter is organized as follows: In Section 5.2, the system model is presented. The minimum delivery delay problem is formulated in Section 5.3. Section 5.4 introduces the dual interface IDNC graph. The optimal and heuristic delivery delay are investigated in Section 5.5. Simulation results are presented in Section 5.6 and the chapter is concluded in Section 5.7.

5.2 System Model and Delay Metrics

5.2.1 Notations

We use the following notations throughout this chapter. Calligraphic letters are used to represent sets and their corresponding capital letters are used to represent the cardinalities of these set. Let $\mathcal{N}$ be a set. Then, N denotes the cardinality of $\mathcal{N}$, i.e., $|\mathcal{N}| = N$. Further, $\mathcal{P}(\mathcal{N})$ represents the power set of $\mathcal{N}$. The set denoted by $\mathcal{M} \times \mathcal{N}$ represents the Cartesian product of the two sets $\mathcal{M}$ and $\mathcal{N}$.

5.2.2 System Model

We consider a network model in which M geographically close devices are interested in receiving N packets. Let $\mathcal{M} = \{U_1, ..., U_M\}$ denote the set of devices wherein each device $U_i \in \mathcal{M}$ initially holds a subset of packets from $\mathcal{N} = \{P_1, ..., P_N\}$. Such prior received packets model the side information of devices or received packets from the base station in previous broadcast sessions [9, 25]. Moreover, we consider order constrained applications for which packets can only be delivered and used at the application layer sequentially in order. In other words, packet P_j is delivered to the application layer of device U_i only if packets $P_1, \cdots, P_j$ are decoded. Out-of-order decoded packets are stored in the devices' buffers, but cannot be delivered to the application layer.

We consider that all devices are equipped with two wireless interfaces. In other words, devices are able to use long-range cellular and short-range local area channels concurrently for transmission and reception. We herein assume that all devices are in the transmission range of each other forming a fully connected cooperative network. Hence, to avoid interference, a single device is allowed to transmit in the D2D communications spectrum in each time slot. Therefore, in addition to an XORed packet transmission from the base station, one of the M devices simultaneously transmits an XORed packet using a different part of the spectrum. Such protocol may result in receiving either a single or two packets by each device at a time slot t.

Each transmitted packet from device U_i to device U_k is subject to an independent Bernoulli erasure with probability $\epsilon_{i,k}$. For notational simplicity, we denote the base station by U_0 with $\epsilon_{0,k}$ representing its channel erasure probability to device U_k. The channels are independent, but not necessarily identical in terms of erasure probability and their statistics are assumed to be known to the base station.

Similar to Chapters 3 and 4, in this chapter, we consider a centralized decision making system to solve the in-order packet delivery problem in a heterogeneous network for the following advantages. First, a centralized system satisfies and executes reliably the service providers policy of quick in-order packet delivery to all devices. Second, it can optimally solve the quick in-order packet delivery problem using the high computational capabilities of the base station. Third, the centralized implementation has low processing requirements at the devices. Finally, it is adaptive to the devices' mobility as the base station can deploy additional resources to be adaptive to the variations in the network [31].

At each time slot t, the base station selects an XOR packet combination to be transmitted by itself, a transmitting device and an XOR packet combination for the transmitting device. The decision is made using the information about the diversity of lost and received packets at the devices as well as the channels' erasure probabilities. Such information assemblage is accomplished by the collection of two feedback bits per device (i.e., one bit for each interface) after each time slot. After the reception of the feedback bits, the base station updates the information about the packet reception status of all devices and stores it in an $M \times N$ *feedback status matrix* (FSM) $\mathbf{F} = [f_{k,l}],\ \forall\, (U_k, P_l) \in \mathcal{M} \times \mathcal{N}$, such that:

$$f_{k,l} = \begin{cases} 0 & \text{if packet } P_l \text{ is decoded by device } U_k, \\ 1 & \text{if packet } P_l \text{ is missing at device } U_k. \end{cases} \tag{5.1}$$

Example 11. *An example of FSM for a network composed of $M = 2$ devices and $N = 4$ packets is given as follows:*

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}. \tag{5.2}$$

Based on the reception status of the packets at a given time slot, the following three sets are attributed to each device U_k:

- The *Has* set $\mathcal{H}_k$ is defined as the set of packets successfully decoded by device U_k. In Example 11, the Has set of device U_1 is $\mathcal{H}_1 = \{P_2, P_4\}$.

- The *Wants* set $\mathcal{W}_k = \mathcal{N} \setminus \mathcal{H}_k$ is defined as the set of missing packets at device U_k. In Example 11, the Wants set of device U_1 is $\mathcal{W}_1 = \{P_1, P_3\}$.

- The *Delivered* set $\mathcal{L}_k \subseteq \mathcal{H}_k$ is defined as the set of packets delivered to the application layer of device U_k. This set includes all the preceding packets of the first missing packet. In Example 11, the Delivered set of devices U_1 and U_2 are $\mathcal{L}_1 = \varnothing$ and $\mathcal{L}_2 = \{P_1, P_2\}$, respectively. Therefore, the undelivered set of devices U_1 and U_2 are $\mathcal{N} \setminus \mathcal{L}_1 = \{P_1, P_2, P_3, P_4\}$ and $\mathcal{N} \setminus \mathcal{L}_2 = \{P_3, P_4\}$, respectively.

Throughout this chapter, the notation $W_k^e \in \mathcal{W}_k$ denotes the index of the e-th missing packet of device U_k. In Example 11, $W_2^1 = 3$ refers to the first missing packet of device U_2, i.e., packet P_3, and $W_2^2 = 4$ refers to the second missing packet of device U_2, i.e., packet P_4.

5.2.3 Delay Metrics

In this subsection, we introduce the relevant definitions used throughout this chapter. We first define instantly decodable transmissions as follows:

Definition 19 (Instantly Decodable Transmission). *A transmitted packet combination is instantly decodable for device U_k if it contains exactly one packet from its Wants set $\mathcal{W}_k$ [9, 52, 63, 93].*

We now introduce different delay metrics of IDNC-enabled networks. The completion time is widely regarded as an appropriate metric to quantify the throughput of IDNC-enabled systems. Note that the completion time metric is inversely proportional to throughput and can be converted into throughput with the usage of bandwidth and packet size. The completion time metric, denoted by τ, is defined as follows:

Definition 20 (Completion Time). *The completion time τ is defined as the number of time slots required until all M devices recover all N packets [93].*

Example 12. *Consider the FSM $\mathbf{F}$ in Example 11 and assume erasure-free transmissions. Let $\mathcal{S}$ be the schedule of transmissions in which:*

1. *The base station and device U_2 send packet P_3 and packet P_1, respectively, in the first time slot.*
2. *The base station or device U_1 broadcasts packet P_4 in the second time slot.*

The evolution of the FSM after each time slot is given by:

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \underset{t=1}{\Rightarrow} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \underset{t=2}{\Rightarrow} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{5.3}$$

The schedule $\mathcal{S}$ requires two time slots to complete the reception of all packets by all devices. Therefore, the completion time is $\tau = 2$.

Despite its direct impact on the throughput, the completion time metric is not suitable for order-constrained applications since it only considers the minimum number of time slots regardless of the order of transmitted packets. For example, the reversed transmissions in Example 12 yields the same completion time but different patterns of in-order delivered packets. A more appropriate metric for order-constrained applications is the delivery time, which can be defined as follows:

Definition 21 (Individual Delivery Time)**.** *The individual delivery time T_k of device U_k increases by one unit for each undelivered packet in each transmission. In other words, the individual delivery time of device U_k increases by $|\mathcal{N} \setminus \mathcal{L}_k| = N - W_k^1 + 1$ units for $|\mathcal{N} \setminus \mathcal{L}_k|$ undelivered packets in every time slot before recovering all N packets.*

Definition 22 (Overall Delivery Time)**.** *The overall delivery time T is the summation of the individual delivery times of all devices over all the time slots until the reception of all N packets by all M devices.*

Example 13. *To illustrate the delivery time metric, consider the evolution of FSM in* (5.3) *using the schedule $\mathcal{S}$ described in Example 12. After the first time slot, the only undelivered packet is packet P_4 of device U_2. Such schedule results in the overall delivery time $T = 1$. Let $\mathcal{S}'$ be the schedule in which the order of the transmissions are reversed. The evolution of the FSM after each time slot is given by:*

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \underset{t=1}{\Rightarrow} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \underset{t=2}{\Rightarrow} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{5.4}$$

After the first time slot, the undelivered packets of device U_1 are P_1, P_2, P_3, P_4 and the undelivered packets of device U_2 are P_3, P_4. Such schedule results in the overall delivery time $T = 4 + 2 = 6$. Note that both schedules have the same completion time $\tau = 2$ but experience very different delivery times.

5.3 Delivery Delay Problem Formulation

The delivery time minimization problem is formulated in [63] as a stochastic shortest path (SSP) problem. The authors take advantage of the characteristics of the complex SSP formulation to draw guidelines for a heuristic IDNC algorithm. In [92], the authors reduce the delivery time by controlling the delivery delay defined as the degradation as compared to the minimum achievable delivery time. Such delivery delay-based approach is shown to outperform the SSP-based heuristic of [63]. As shown in [63, 92], the delivery time minimization problem is computationally intractable even for single interface networks. In fact, the dynamic nature of channel realizations and the dependence of the optimal coding policy on them makes the optimization problem anti-causal. In addition, considering a complete schedule of transmissions to find the optimal schedule requires a search over all feasible coding opportunities and has an exponential growth of the computational complexity with the number of devices and packets. To develop a computationally tractable solution, a popular approach is to approximate the delivery time minimization problem by an online optimization problem involving the delivery delay. This chapter extends the delivery delay metric [92] of single interface networks to the dual interface networks and proposes a graphical solution to minimize such quantity.

5.3.1 Delivery Delay Metric

The authors in [92] define the delivery delay for single interface networks as a measure of ordered packet delivery degradation as compared to the minimum achievable delivery time. In fact, the delivery delay integrates the packets' order of each device by incurring additional delay generated from undelivered packets even though the transmission is instantly decodable for that device. In other words, assuming $\overline{T}_k$ is the minimum achievable delivery time for device U_k, the delivery delay D_k is defined so as to satisfy the following equality in erasure-free transmissions [92]:

$$T_k = \overline{T}_k + D_k. \tag{5.5}$$

Note that the minimum delivery time $\overline{T}_k$ depends solely on the FSM and is achieved using a schedule that contains only in-order instantly decodable packets for device U_k. For example, given $\mathcal{W}_k = \mathcal{N}$, the minimum delivery time is $\overline{T}_k = \dfrac{N(N-1)}{2}$, when device U_k receives in-order missing packets in every transmission [92]. From expression (5.5), it can be stated that the delivery delay is the additional delay experienced by a device in the transmissions in which it does not receive in-order missing packets.

We now extend the definition of the instantaneous delivery delay to the dual interface networks by introducing a theorem. The instantaneous delivery delay refers to the increase in the delivery delay at each time slot. In fact, the delivery delay defined in (5.5) is the sum of the instantaneous delivery delay over all time slots until the completion time. However, in the rest of the chapter, the term instantaneous is dropped and both quantities are denoted by D_k for device U_k.

Let us consider that the packet combination ξ is transmitted by the base station and the packet combination $\xi' \neq \xi$ is transmitted by another device in a given time slot. The notation $\xi \cap \mathcal{W}_k = \circledast$ represents the case where the packet combination does not include exactly one packet that is missing at the device, i.e., $|\xi \cap \mathcal{W}_k| \neq 1$. Similarly, the notation $\xi' \neq \xi$ represents the case where both packet combinations are not intended to the same device with the same packet, i.e., $(\xi, \xi') \cap \mathcal{W}_k = (W_k^e, W_k^{e'})$ implies that $W_k^e \neq W_k^{e'}$. Here, $e, e' > 1$ and, thus, $W_k^e \neq W_k^{e'} \neq W_k^1$.

Theorem 1. *The delivery delay $D_k(\xi, \xi')$ of device U_k increases after transmission of the packet combination ξ by the base station and the packet combination $\xi' \neq \xi$ by another device in a given time slot by the following quantity:*

$$D_k(\xi,\xi') = \begin{cases} W_k^{e'} - W_k^2 & \text{if } (\xi,\xi') \cap \mathcal{W}_k = (W_k^1, W_k^{e'}) \\ W_k^{e} - W_k^2 & \text{if } (\xi,\xi') \cap \mathcal{W}_k = (W_k^e, W_k^{1}) \\ W_k^{e} + W_k^{e'} - 2W_k^1 & \text{if } (\xi,\xi') \cap \mathcal{W}_k = (W_k^e, W_k^{e'}) \\ N - W_k^2 + 1 & \text{if } (\xi,\xi') \cap \mathcal{W}_k = (W_k^1, \circledast) \\ N - W_k^2 + 1 & \text{if } (\xi,\xi') \cap \mathcal{W}_k = (\circledast, W_k^1) \\ W_k^{e} + N - 2W_k^1 + 1 & \text{if } (\xi,\xi') \cap \mathcal{W}_k = (W_k^e, \circledast) \\ W_k^{e'} + N - 2W_k^1 + 1 & \text{if } (\xi,\xi') \cap \mathcal{W}_k = (\circledast, W_k^{e'}) \\ 2(N - W_k^1 + 1) & \text{if } (\xi,\xi') \cap \mathcal{W}_k = (\circledast, \circledast) \end{cases} \tag{5.6}$$

Proof. To prove this theorem, concurrent transmissions from the base station and one of the M devices are first decomposed into sequential transmissions. The delivery delay incurred by each transmission is derived and the overall delivery delay incurred by two transmissions in the current time slot is computed by summing individual delays. Using a similar decomposition of the delivery time as in [92], the delivery delay defined in expression (5.6) is shown to satisfy the delivery time-delay expression in (5.5). A complete proof can be found in the appendix in Section 5.8. ■

Example 14. *To further clarify the delivery delay metric of Theorem 1, consider the following FSM with $M = 2$ devices and $N = 6$ packets:*

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}. \tag{5.7}$$

The delivery delay increase is demonstrated for the following five scenarios after the current time slot:

- *Device U_1 decodes packets P_1 and P_3, which corresponds to the first two cases in* (5.6) ⇒ *The device obtains its* 1*-st and* 2*-nd missing packets. Therefore, there is no additional delivery delay, i.e.,* $(W_1^2 - W_1^2) = 0$.

- *Device U_1 decodes packets P_3 and P_6, which corresponds to the third case in* (5.6) ⇒ *The device obtains its* 2*-nd and* 4*-th missing packets. Therefore, the delivery delay increase is* 7*, i.e.,* $(W_1^2 + W_1^4 - 2W_1^1) = (3 + 6 - 2) = 7$.

- *Device U_1 decodes only packet P_1, which corresponds to the fourth and fifth cases in* (5.6) ⇒ *The device is served with its* 1*-st missing packet. Therefore, the delivery delay increase is* 4*, i.e.,* $(N - W_1^2 + 1) = (6 - 3 + 1) = 4$.

- *Device U_1 decodes only packet P_6, which corresponds to the sixth and seventh cases in* (5.6) ⇒ *The device is served with its* 4*-th missing packet. The delivery delay increase is* 11*, i.e.,* $(W_1^4 + N - 2W_1^1 + 1) = (6 + 6 - 2 + 1) = 11$.

- *Device U_1 does not decode any new packet, which corresponds to the last case in* (5.6) ⇒ *The delivery delay increase is* 12*, i.e.,* $2(N - W_1^1 + 1) = 2(6 - 1 + 1) = 12$.

5.3.2 Problem Formulation

In this subsection, the transmitting device and packet combinations are chosen so as to minimize the expected increase of the instantaneous delivery delay at each time slot. The following lemma approximates the expected instantaneous delivery delay experienced by device U_k when the base station transmits packet combination ξ and device U_i transmits another packet combination ξ'.

Lemma 1. *The expected delivery delay increase $D_k^i(\xi, \xi')$ of device U_k after the transmissions of packet combinations ξ and $\xi' \neq \xi$ by the base station and device U_i, respectively, can be approximated by the following expression:*

$$\mathbb{E}[D_k^i(\xi,\xi')] \approx \tag{5.8}$$
$$\begin{cases} (\epsilon_{0,k}+\epsilon_{i,k})(N+1)+W_k^e(1-\epsilon_{0,k})+W_k^{e'}(1-\epsilon_{i,k})-2W_k^1 & \text{if } (\xi,\xi')\cap\mathcal{W}_k=(W_k^e,W_k^{e'}) \\ \epsilon_{0,k}(N+1)+W_k^e(1-\epsilon_{0,k})+N-2W_k^1+1 & \text{if } (\xi,\xi')\cap\mathcal{W}_k=(W_k^e,\circledast) \\ \epsilon_{i,k}(N+1)+W_k^{e'}(1-\epsilon_{i,k})+N-2W_k^1+1 & \text{if } (\xi,\xi')\cap\mathcal{W}_k=(\circledast,W_k^{e'}) \\ 2(N-W_k^1+1) & \text{if } (\xi,\xi')\cap\mathcal{W}_k=(\circledast,\circledast) \end{cases}$$

Proof. This lemma is demonstrated by computing the expected value of the delivery delay given in Theorem 1 with respect to the erasure probability distributions of the channels. Afterward, the expected delivery expression is simplified by considering an upper bound on the delivery delay. The complete proof can be found in the appendix in Section 5.9. ■

The above lemma approximates the instantaneous delivery delay to trade-off some accuracy in calculation for much more computational simplicity in designing efficient packet and device selection algorithm. In fact, the approximation is exact if none of the instantly decodable packets is the first missing packet. Otherwise, the approximation is an upper bound on the expected delivery delay. We now introduce a theorem to solve the problem of minimizing the expected delivery delay defined in Lemma 1.

Theorem 2. *The delivery delay reduction problem can be formulated as the problem of finding the optimal transmitting device and transmitted packet combinations so as to maximize* *the following objective function:*

$$\max_{\substack{U_i \in \mathcal{M} \\ \xi \in \mathcal{P}(\mathcal{N}) \\ \xi' \in \mathcal{P}(\mathcal{H}_i)}} \sum_{U_k \in \mathcal{X}(\xi)} (1-\epsilon_{0,k})(N - W_k^e + 1) + \sum_{U_k \in \mathcal{X}(\xi')} (1-\epsilon_{i,k})(N - W_k^{e'} + 1), \tag{5.9}$$

where $\mathcal{X}(\xi)$ and $\mathcal{X}(\xi')$ are the set of targeted devices with instantly decodable packets by the packet combinations ξ and ξ', respectively.

Proof. The delivery delay minimization problem is formulated as a joint optimization problem over the set of transmitting devices and packet combinations. Using the expression of the expected delivery delay derived in Lemma 1, the problem is simplified to a couple of independent optimization problems of transmitted packet selection. The complete proof of this theorem can be found in the appendix in Section 5.10. ■

5.4 Dual Interface IDNC graph

In this chapter, we suggest using a graphical method to find the optimal solution of the delivery delay minimization problem in (5.9). Therefore, we now introduce a dual interface IDNC (DI-IDNC) graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ to represent both coding and transmission conflict-free opportunities in one unified framework. The representation of all conflicts in one graph is suggested in [105] for distributed storage networks with a single interface, i.e., all transmissions are performed by the base stations. The new graph model extends the work of [105] by allowing both the base station and another device to transmit concurrently in each time slot. Afterward, the optimization problem in (5.9) is shown to be equivalent to a maximum weight independent set selection problem over the DI-IDNC graph $\mathcal{G}$.

5.4.1 Vertex Generation

Each vertex in the set of vertices $\mathcal{V}$ of the DI-IDNC graph $\mathcal{G}$ represents a triplet of a transmitting device, a targeted device, and a missing packet of the targeted device. This subsection describes the generation of vertices and the following subsection defines the connectivity conditions between these vertices.

In the DI-IDNC graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, vertices are generated according to the following two criteria:

- **C1**: The base station U_0 is one of the two senders. Given that the base station possesses all packets, a vertex $v_{0,k,l}$ is generated for each missing packet $P_l \in \mathcal{W}_k$ of each device $U_k \in \mathcal{M}$.
- **C2**: Each device $U_i \in \mathcal{M}$ is a potential transmitting device as one of the M devices is another sender. In contrast with the base station, device U_i possesses a subset of all packets, i.e., $\mathcal{H}_i \subseteq \mathcal{N}$. Therefore, for a potential transmitting device U_i, a vertex $v_{i,k,l}$ is generated for each missing packet $P_l \in \{\mathcal{H}_i \cap \mathcal{W}_k\}$ of another device $U_k \in \mathcal{M}$.

Note that a missing packet of a device generates multiple vertices according to the number of potential transmitting devices. All such vertices representing the same missing packet by different transmitting devices are included in the graph since it is yet to determine which transmitting device will serve this packet to the targeted device. However, once the targeted device is served with that packet, all such vertices are removed from the graph as discussed in the following subsection.

5.4.2 Edge Generation

Having generated the set of vertices in the DI-IDNC graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, a pair of vertices are set adjacent by an edge due to either a coding conflict or a transmission conflict. While the former represents the scenario in which the transmission is not instantly decodable, the latter expresses the fact that a single device is allowed to transmit a packet combination that is different from the transmitted packet combination of the base station.

Coding Conflicts

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

- **C1**: $U_i = U_r$ and $U_k = U_m$ and $P_l \neq P_n$. In other words, two vertices represent transmissions from the same sender to the same device with different missing packets P_l and P_n.

- **C2**: $U_i = U_r$ and $U_k \neq U_m$ and $(P_l, P_n) \notin \mathcal{H}_m \times \mathcal{H}_k$. In other words, two vertices represent transmissions from the same sender to devices U_k and U_m requiring 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 $P_l \oplus P_n$.

Transmission Conflicts

Two vertices $v_{i,k,l}$ and $v_{r,m,n}$ are set adjacent by an edge due to a transmission conflict if one of the following two conditions holds:

- **C3**: $U_i = U_0$ and $U_r \in \mathcal{M}$, and $U_k = U_m$ and $P_l = P_n$. In other words, a vertex representing the transmission from the base station U_0 to device U_k with packet P_l is set adjacent to all vertices representing the transmissions from all other devices to the same device U_k with the same packet P_l. This prevents both base station and transmitting device to target a device with the same packet.

- **C4**: $U_i \neq U_r$ and $(U_i, U_r) \in \mathcal{M}$. In other words, a vertex representing a transmission from a device is set adjacent to all other vertices representing transmissions from all other devices. This prevents selecting more than one transmitting device in each time slot.

5.4.3 Maximal Independent Set

Definition 23 (Maximal Independent Set)**.** *A maximal independent set cannot be extended to include one more vertex without violating the pairwise non-adjacent vertices constraint [102].*

Definition 24 (Maximum Weight Independent Set)**.** *In a weighted graph with different vertices' weights, the maximum weight independent set is the maximal independent set with the largest sum of its vertices' weights.*

Example 15. *The DI-IDNC graph $\mathcal{G}$ corresponding to the FSM in Example 11 is shown in Figure 5.1. A maximal independent set of this graph is $\kappa = \{v_{0,1,3}, v_{0,2,3}, v_{2,1,1}\}$, in which the base station U_0 transmits packet P_3 and device U_2 transmits packet P_1. With this maximal*

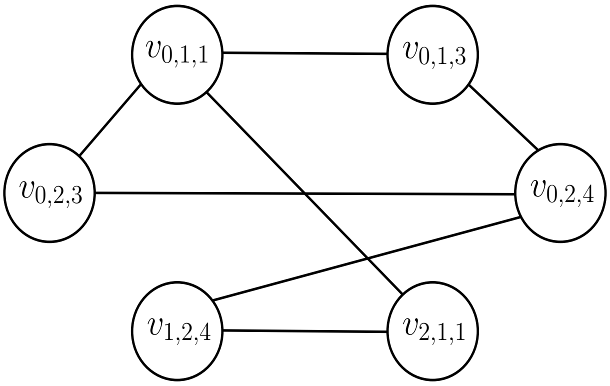


Figure 5.1: The dual interface IDNC graph corresponding to FSM in Example 11.

independent set, device U_1 is targeted with two new packets P_1 and P_3 and device U_2 is targeted with one new packet P_3.

5.4.4 Graph Construction Complexity

This subsection describes the complexity of constructing the DI-IDNC graph $\mathcal{G}$ for a given FSM in a time slot. Note that $\mathcal{O}(MN)$ vertices are generated for each potential transmitting device representing the different packets in the Has set of that device, which are missing at the other devices. In the dual interface system, there are M potential transmitting devices and a base station. As a result, the number of vertices in the graph is $\mathcal{O}(M^2N)$.

To build the adjacency matrix of the graph, the algorithm needs to evaluate the adjacency conditions **C1, C2, C3** and **C4** for each pair of vertices and determine whether these vertices should be connected with an edge. For a graph with $\mathcal{O}(M^2N)$ vertices, the number of pairs can be defined as $\binom{M^2N}{2}$ and can be expressed as $\mathcal{O}(M^4N^2)$. For each pair of vertices, the algorithm needs a constant number of operations to check four conflict conditions and thus, the number of operations is $\mathcal{O}(1)$. With these results, the total complexity of DI-IDNC graph construction is $\mathcal{O}(M^4N^2)$.

5.5 Proposed Solution

This section suggests reformulating the delivery delay optimization problem in (5.9) as a maximum weight independent set selection problem over the DI-IDNC graph. Given the computational complexity of finding the maximum weight independent set, this section further proposes an efficient heuristic approach to greedily select a maximal independent set.

5.5.1 Minimum Delivery Delay Solution

The DI-IDNC graph allows to represent all feasible decisions of transmitting devices and transmitted packets as illustrated in the following lemma:

Lemma 2. *All feasible coding and transmission conflict-free opportunities are defined by the set of all maximal independent sets in DI-IDNC graph.*

Proof. The proof of this lemma is omitted here as it follows the similar steps as in [105]. Whereas the authors in [105] consider a single interface, the proposed DI-IDNC graph allows concurrent transmissions from the base station and one of the devices. Applying the results of [105] to each sub-graph corresponding to a single transmitting device allows to conclude that all feasible coding opportunities are represented by the set of maximal independent sets in each sub-graph. Finally, from the transmission conflict conditions, it can readily be seen that only a single device is allowed to transmit in addition to the base station, which concludes the proof. ■

We attribute the following three properties to a maximal independent set κ of the DI-IDNC graph $\mathcal{G}$:

- A single transmitting device in addition to the base station can belong to a maximal independent set κ.
- A device can have at most two vertices, i.e., two missing packets, in a maximal independent set κ representing transmissions from the base station and a transmitting device.
- The transmitting device and the base station form the coded packets by XORing the source packets identified by the vertices in maximal independent set κ representing transmissions from that device and the base station, respectively.

The following proposition uses the DI-IDNC graph to characterize the solution of the delivery delay optimization problem in (5.9).

Proposition 1. *The optimal packet combinations of the base station and a transmitting device that minimize the expected delivery delay is the one corresponding to the maximum weight independent set in the DI-IDNC graph wherein the weight of a vertex $v_{i,k,l}$ is defined by:*

$$w(v_{i,k,l}) = (1 - \epsilon_{i,k})(N - l + 1), \tag{5.10}$$

where l represents the index of packet P_l corresponding to vertex $v_{i,k,l}$.

Proof. Let $\mathcal{I}$ be the set of maximal independent sets in the DI-IDNC graph $\mathcal{G}$. Given the results of Lemma 2, the delivery delay reduction problem can be formulated as a maximal independent set selection problem in the DI-IDNC graph as follows:

$$\begin{aligned}
&\max_{\substack{U_i \in \mathcal{M} \\ \xi \in \mathcal{P}(\mathcal{N}) \\ \xi' \in \mathcal{P}(\mathcal{H}_i)}} \sum_{U_k \in \mathcal{X}(\xi)} (1 - \epsilon_{0,k})(N - W_k^e + 1) + \sum_{U_k \in \mathcal{X}(\xi')} (1 - \epsilon_{i,k})(N - W_k^{e'} + 1) \\
&= \max_{\kappa \in \mathcal{I}} \sum_{v_{i,k,l} \in \kappa} (1 - \epsilon_{i,k})(N - l + 1) \\
&= \max_{\kappa \in \mathcal{I}} \sum_{v_{i,k,l} \in \kappa} w(v_{i,k,l}).
\end{aligned} \tag{5.11}$$

Hence, the problem of selecting a transmitting device and the packet combinations for the base station and the selected transmitting device that reduce the expected delivery delay is equivalent to the maximum weight independent set selection problem over the DI-IDNC graph. ■

According to Proposition 1, the minimum delivery delay problem in a dual interface scenario is equivalent to finding all the maximal independent sets in the DI-IDNC graph and choosing the maximum weight independent set among them. Therefore, an efficient maximum weight independent set selection algorithm, i.e., Bron-Kerbosch algorithm [104], can be employed to discover the optimal solution. However, the complexity of these optimal solvers grows exponentially with the increase in the number of devices M and packets N. For example, the complexity of the Bron-Kerbosch algorithm for a graph of V vertices is in the order of $\mathcal{O}(3^{\frac{V}{3}})$, wherein the number of vertices is $\mathcal{O}(M^2N)$ in the DI-IDNC graph. To reduce the computational complexity, the next subsection proposes a greedy maximal

Algorithm 4: Delivery Delay Reduction Heuristic

```
Construct DI-IDNC graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ ;
Set maximal independent set $\kappa = \varnothing$;
Set $\mathcal{G}(\kappa) \leftarrow \mathcal{G}$ ;
while $\mathcal{G}(\kappa) \neq \varnothing$ do
    Compute weight $\psi(v_{i,k,l}), \forall v_{i,k,l} \in \mathcal{G}(\kappa)$ using (5.13);
    Select $v^*_{i,k,l} = \arg\max_{v_{i,k,l} \in \mathcal{G}(\kappa)} \{\psi(v_{i,k,l})\}$;
    Set $\kappa \leftarrow \kappa \cup v^*_{i,k,l}$;
    Extract new sub-graph $\mathcal{G}(\kappa)$;
end
```

independent set selection algorithm with sub-optimal performance.

5.5.2 Delay Reduction Heuristic

This subsection proposes a simple delivery delay reduction heuristic that selects a maximal independent set using a greedy vertex search in the DI-IDNC graph. The heuristic first constructs the DI-IDNC graph $\mathcal{G}$ and defines $\pi_{ikl,rmn}$ as the non-adjacency indicator of vertices $v_{i,k,l}$ and $v_{r,m,n}$ in $\mathcal{G}$ such that: $\pi_{ikl,rmn} = 1$, if $v_{i,k,l}$ is not connected to $v_{r,m,n}$, and $\pi_{ikl,rmn} = 0$, otherwise. It then defines the non-adjacency weight Ω_{ikl} of vertex $v_{i,k,l}$ as:

$$\Omega_{ikl} = \sum_{v_{r,m,n} \in \mathcal{G}} \pi_{ikl,rmn} w(v_{r,m,n}), \tag{5.12}$$

where $w(v_{r,m,n})$ is defined according to (5.10). This heuristic finally defines the weight of vertex $v_{i,k,l}$ as:

$$\psi(v_{i,k,l}) = w(v_{i,k,l}) \times \Omega_{ikl} = \{(1 - \epsilon_{i,k})(N - l + 1)\} \, \Omega_{ikl}. \tag{5.13}$$

Having defined the vertices' weights, the maximal independent set κ is initialized to the vertex $v^*_{i,k,l}$ with the maximum weight such as:

$$v^*_{i,k,l} = \arg \max_{v_{i,k,l} \in \mathcal{G}} \{\psi(v_{i,k,l})\}. \tag{5.14}$$

After adding vertex $v^*_{i,k,l}$ to κ, i.e., $\kappa = \{v^*_{i,k,l}\}$, the heuristic extracts the sub-graph $\mathcal{G}(\kappa)$ consisting of vertices in $\mathcal{G}$ that are non-adjacent to vertex $v^*_{i,k,l}$. At the next step, the heuristic repeats the vertex search process over the sub-graph $\mathcal{G}(\kappa)$ and adds a new vertex to κ. The process terminates when there is no more vertex in the graph $\mathcal{G}$ that are non-adjacent to all the vertices in κ (i.e., when κ becomes a full maximal independent set). The steps of the maximal independent set selection heuristic are summarized in Algorithm 4.

By following a similar analysis of [93, 105], it can be inferred that the overall complexity of greedily selecting a maximal independent set in the DI-IDNC graph is in the order of $\mathcal{O}(M^3N)$. In fact, at a given time slot, a maximal independent set can have $\mathcal{O}(2M)$ vertices as each of M devices can be served with at most two new packets using two interfaces. In every vertex search step, the algorithm can search over $\mathcal{O}(M^2N)$ vertices. Therefore, the complexity of greedily selecting a maximal independent set over the DI-IDNC graph is $\mathcal{O}(M^3N)$.

5.6 Simulation Results

This section presents the performance of the proposed solution using both the Bron-Kerbosch algorithm as a solver for the maximum weighted independent set selection problem and the proposed heuristic in Algorithm 4 as a low-complexity solution. The simulation results show the average of the delivery delays of all M devices until the complete recovery of the N packets. The performance of both algorithms is compared to the following algorithms:

- '*Single-IDNC*' algorithm, which uses a single interface to deliver IDNC packets in-order to all devices [92].
- *'Uncoded Broadcast'* scheme, which uses two interfaces simultaneously. However, it selects an uncoded packet for the base station and another uncoded packet for a transmitting device that are missing at the largest number of devices.
- *Dual-IDNC* algorithm, which uses two interfaces and IDNC to reduce the completion time [25].
- *Dual-RLNC* algorithm, which uses two interfaces and RLNC to reduce the completion time [25].

First, consider a small social network with $M = 5$ devices and $N = 10$ order constrained packets due to the high computational complexity of Bron-Kerbosch algorithm. The simu-

Table 5.1: Mean delivery delay and completion time performances of different algorithms for a network composed of $M = 5$ devices and $N = 10$ packets

Algorithm	*Delivery delay*	*Completion time*
Single IDNC	59.75	10.12
Uncoded Broadcast	55.44	8.23
Dual RLNC	48.46	4.80
Dual IDNC	31.56	5.28
Proposed Heuristic	17.92	5.46
Bron-Kerbosch	15.69	5.11

lations assume that each device initially possesses between 40% and 50% of the N packets. As discussed in Section 5.2, such initially possessed packets represent the side information of the devices or the received packets from the base station in previous broadcast sessions.

Table 5.1 summarizes the delivery delay and the completion time achieved by the aforementioned algorithms for a network composed of $M = 5$ devices, $N = 10$ packets, average cellular channel erasure probability $\bar{\epsilon_0} = 0.3$, average local area channel erasure probability $\bar{\epsilon} = 0.15$. When the average cellular channel erasure probability is $\bar{\epsilon_0} = 0.3$, the erasure probabilities of different cellular channels are in the range $[0.2, 0.4]$. Moreover, when the average local area channel erasure probability is $\bar{\epsilon} = 0.15$, the erasure probabilities of different local area channels are in the range $[0.1, 0.2]$. Such variations of the channel erasure probabilities model different conditions of the physical channels.

From Table 5.1, it can be readily seen that the proposed low-complexity heuristic achieves a comparable delivery delay and completion time performances as the high-complexity Bron-Kerbosch algorithm. Indeed, both algorithms use the new DI-IDNC graph and take into account the order constraint of packets and the channel erasure probabilities in making coding and transmission decisions. While the Bron-Kerbosch algorithm is applicable for small to moderate sized networks that involve a small number of users, e.g., small social networks, the proposed heuristic is applicable for large scale networks, e.g., public networks that involve a large number of users. Moreover, the Bron-Kerbosch algorithm and the proposed heuristic achieve a significant delivery delay reduction as compared to the other four schemes. In particular, the table shows that the delay performance of the dual IDNC algorithm considerably deviates from the Bron-Kerbosch and proposed heuristic algorithms due to its order insensitivity and channel oblivious nature. Unlike the immediately decodable packet transmissions of IDNC-based algorithms, the dual RLNC algorithm requires devices collecting a sufficient number of independent coded packets to start the decoding process. Such packet decoding delay also incurs a high delivery delay in the dual RLNC algorithm. As expected, the delivery delay performance of the single-IDNC and uncoded broadcast schemes significantly deviate from the proposed algorithms due to using a single interface and uncoded transmissions, respectively.

Finally, the table shows that the Bron-kerbosch and proposed heuristic algorithms maintain an acceptable completion time degradation as compared to the throughput-optimal dual RLNC algorithm. Moreover, the proposed heuristic algorithm performs almost similar to the dual IDNC algorithm, which is particularly designed for reducing the completion time in IDNC-enabled systems.

Having demonstrated the performance of both the Bron-Kerbosch and proposed heuristic algorithms for a small network, the remaining of this section considers more general network scenarios and illustrates only the performance of the proposed heuristic algorithm. Figure 5.2 shows the mean delivery delay and the completion time achieved by different algorithms against different number of devices M for $N = 15$ packets, average cellular erasure probability $\bar{\epsilon_0} = 0.3$, average local area erasure probability $\bar{\epsilon} = 0.15$. From Figure 5.2, it can be observed that the proposed heuristic algorithm provides a significant delivery delay reduction as compared to the other four schemes by using the new DI-IDNC graph and the properties of the minimum delivery delay formulation, i.e., two wireless interfaces of devices, order constraint of packets and packet reception probabilities.

On the other hand, the dual-IDNC algorithm uses two wireless interfaces but performs poorly in terms of delivery delay due to ignoring the order of packets. Moreover, as expected, the dual-RLNC algorithm suffers from high delivery delay due to requiring a sufficient number of independent coded packets at devices before decoding all coded packets simultaneously. The uncoded broadcast scheme always transmits uncoded packets from the base station and a transmitting device. Such network coding oblivious scheme serves fewer devices in each time slot as compared to network coding aware schemes as it does not take advantage of the diversity of received and lost packets at devices in making decisions. Finally, the single-IDNC algorithm takes into account the order of packets while using a single interface for packet reception. Therefore, a device is served with at most one packet in each time slot.

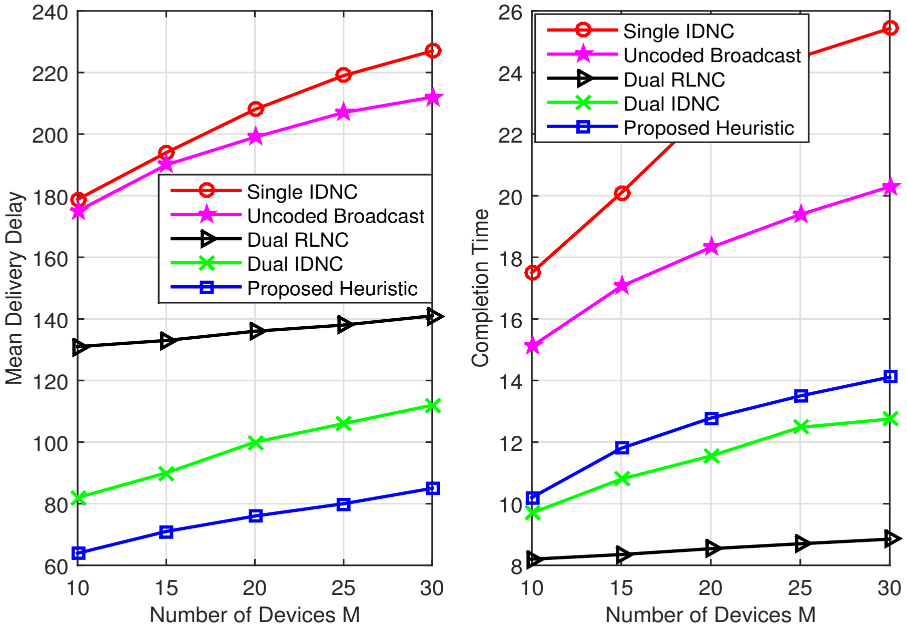


Figure 5.2: Mean delivery delay and completion time versus different number of devices M.

In consistent with the results of Table 5.1, Figure 5.2 demonstrates that the proposed heuristic algorithm closely follows the completion time performance of the dual-RLNC and dual-IDNC algorithms. This good performance of the heuristic algorithm results from using the new DI-IDNC graph that systematically defines all feasible coding and transmission conflict-free decisions. As expected, the completion time performance of both single-IDNC and uncoded broadcast schemes significantly deviates from that of the benchmark dual-RLNC algorithm.

Figure 5.3 shows the mean delivery delay and the completion time achieved by different algorithms against different number of packets N for $M = 15$ devices, average cellular erasure probability $\bar{\epsilon}_0 = 0.3$, average local area erasure probability $\bar{\epsilon} = 0.15$. From this figure, it can be easily observed that as the number of packets increases, the delivery delay performance gap between the proposed heuristic algorithm and the other four algorithms increases. In fact, in dual-IDNC and dual-RLNC algorithms, the devices decode missing packets in an arbitrary order and thus, a large number of undelivered packets in every time slot incur a substantial quantity of delivery delay increase. In single-IDNC algorithm, each device decodes at most one missing packet in each time slot and experiences a large delivery delay from not receiving a packet on another interface. In this figure, the completion time

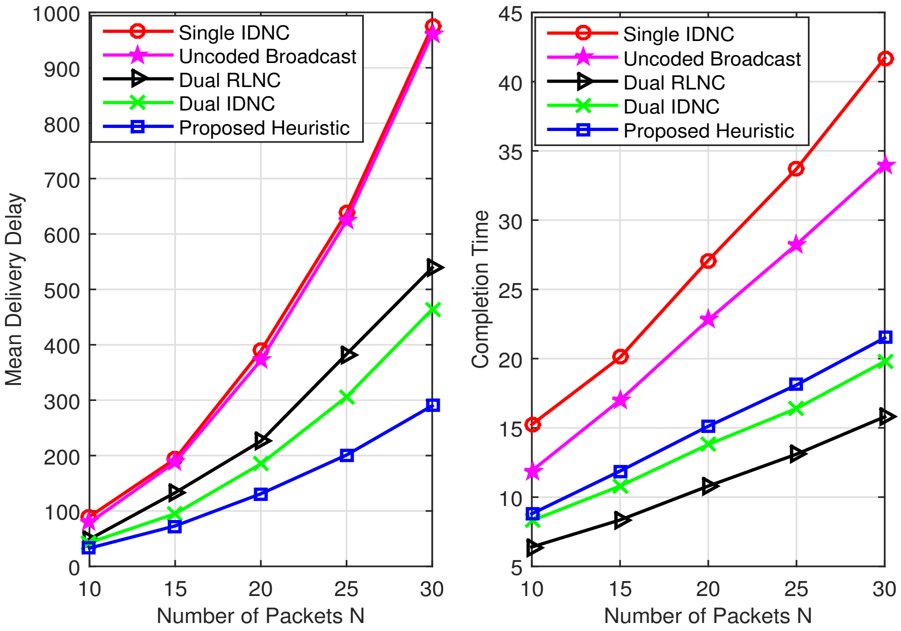


Figure 5.3: Mean delivery delay and completion time versus different number of packets N.

performance of different algorithms are consistent to those in Figure 5.2.

Figure 5.4 demonstrates the mean delivery delay and the completion time achieved by the different algorithms against different average cellular and local area erasure probabilities $\bar{\epsilon}_0$ and $\bar{\epsilon}$ for $M = N = 15$. Such variations in channel erasure probabilities model different conditions of physical channels. In all scenarios, the erasures of local area channels are considered to be significantly lesser compared to the erasures of cellular channels due to the short distances among devices. As expected, the completion time and delivery delay of all algorithms increase with the degradation of channel qualities.

5.7 Conclusion

This chapter develops an efficient framework for order-constrained applications that utilizes dual wireless interfaces and network coding to provide quick and reliable in-order packet delivery to the devices. The delivery delay is introduced as a performance metric that not only represents the degradation to the optimal coding strategy but also reflects the packets' order. To address the delivery delay minimization problem, the dual interface IDNC graph is constructed and the problem is formulated as a maximum weight independent set selection

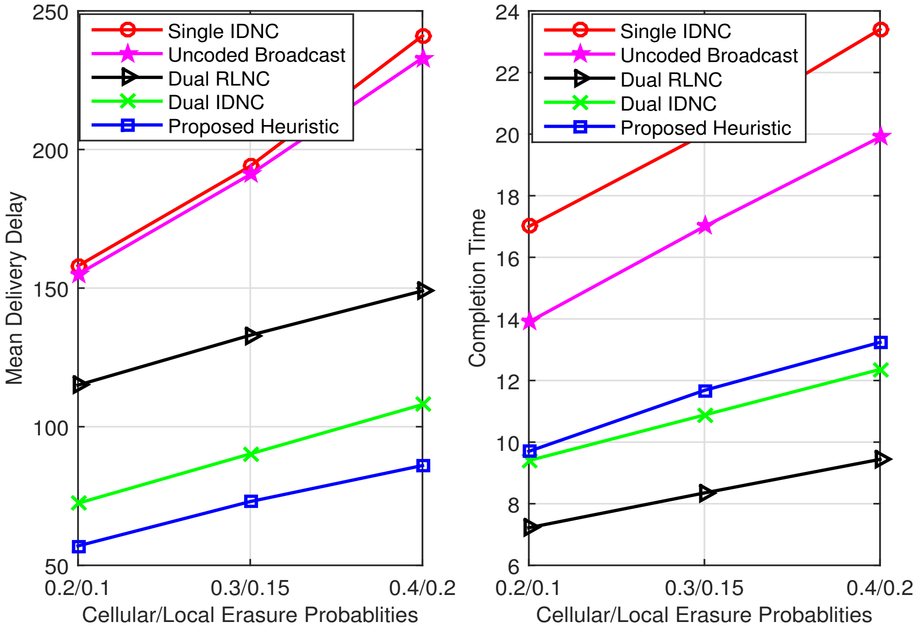


Figure 5.4: Mean delivery delay and completion time versus different average cellular and local area channel erasure probabilities $\bar{\epsilon_0}$ and $\bar{\epsilon}$.

problem over this graph. Given the NP-hardness of finding the maximum weight independent set, a simple heuristic is suggested to select a maximal independent set based on a greedy vertex search. Simulation results show that the proposed IDNC algorithm effectively reduces the delivery delay as compared to the existing network coding algorithms while maintaining a tolerable completion time degradation compared to the RLNC algorithm.

5.8 Appendix A

Proof of Theorem 1

To demonstrate this theorem, concurrent transmissions from the base station and a device are first decomposed into sequential transmissions. The delivery delay incurred by each transmission is derived and the overall delivery delay incurred by two transmissions in a time slot is computed by summing individual delays. Using a similar decomposition of the delivery time as in [92], the delivery delay defined in expression (5.6) is shown to satisfy the delivery time-delay expression in (5.5).

In [92], the authors show that the delivery delay increase $D_k(\xi)$ of device U_k after the reception of packet combination ξ that satisfies the constraint (5.5) is given by the following quantity:

$$D_k(\xi) = \begin{cases} W_k^e - W_k^1 & \text{if } \xi \cap \mathcal{W}_k = W_k^e \\ N - W_k^1 + 1 & \text{otherwise} \end{cases} \tag{5.15}$$

To derive a similar expression for the dual interface systems, let ξ and ξ' be the distinct transmitted packets in two interfaces. The concurrent transmissions are considered as two consecutive transmissions in which a device first receives the first missing packet and then receives the second missing packet. Further, let $W_k^e(1)$ and $W_k^e(2)$ be the e-th missing packet before and after the first transmission. According to the instant decodability property and the reception status of packet combinations ξ and ξ', the following nine scenarios can be distinguished:

1. The packet combination ξ is instantly decodable and includes the first missing packet of device U_k, i.e., $\xi \cap \mathcal{W}_k = W_k^1(1)$. According to (5.15), such first transmission does not increase the delay. Depending on the second transmission ξ', the overall delivery delay is given by:

   a. The packet combination ξ' is instantly decodable and includes the first missing packet of device U_k, i.e., $\xi' \cap \mathcal{W}_k = W_k^1(2)$. The delivery delay increase is 0 in this scenario.

   b. The packet combination ξ' is instantly decodable and includes the e'-th missing packet with $e' > 1$, i.e., $\xi' \cap \mathcal{W}_k = W_k^{e'}$. According to expression (5.15), the delivery delay generated by the second transmission is $W_k^{e'} - W_k^1(2)$. Note that, given the first transmission includes the first missing packet of device U_k, $W_k^2(1) = W_k^1(2)$. Therefore, the overall delivery delay generated by both transmissions is $W_k^{e'} - W_k^2$.

   c. The packet combination ξ' does not bring new useful information to device U_k. According to expression (5.15), the delivery delay incurred from the second transmission is $N - W_k^1(2) + 1$. Using the equality of $W_k^2(1) = W_k^1(2)$, the overall delivery delay is $N - W_k^2 + 1$.

2. The packet combination ξ is instantly decodable and includes the e-th missing packet with $e > 1$. Depending on the second transmission ξ', the overall delivery delay is given by:

   a. The packet combination ξ' is instantly decodable and includes the first missing packet of device U_k, i.e., $\xi' \cap \mathcal{W}_k = W_k^1(2)$. This scenario is similar to Case 1)b. Therefore, the overall delivery delay is $W_k^e - W_k^2$.

   b. The packet combination ξ' is instantly decodable and includes the e'-th missing packet with $e' > 1$, i.e., $\xi' \cap \mathcal{W}_k = W_k^{e'}$. According to (5.15), the delivery delay incurred by the first and the second transmissions are $W_k^e - W_k^1(1)$ and $W_k^{e'} - W_k^1(2)$, respectively. Since neither of the transmissions provides the device with its first missing packet, $W_k^1(1) = W_k^1(2)$. Therefore, the overall delivery delay is $W_k^e + W_k^{e'} - 2W_k^1$.

   c. The packet combination ξ' does not bring new useful information to device U_k. The delivery delay incurred by the first and the second transmissions are $W_k^e - W_k^1(1)$ and $N - W_k^1(2) + 1$, respectively. As a result, the overall delivery delay is $W_k^e + N - 2W_k^1 + 1$.

3. The packet combination ξ does not bring new useful information to device U_k. Depending on the second transmission ξ', the overall delivery delay is given by:

   a. The packet combination ξ' is instantly decodable and includes the first missing packet of device U_k. This is similar to Case 1)c. Hence, the overall delivery delay is $N - W_k^2 + 1$.

   b. The packet combination ξ' is instantly decodable and includes the e'-th missing packet with $e' > 1$, i.e., $\xi' \cap \mathcal{W}_k = W_k^{e'}$. This scenario is similar to Case 2)c. Therefore, the overall delivery delay is $W_k^{e'} + N - 2W_k^1 + 1$.

   c. The packet combination ξ' does not bring new useful information to device U_k. Both transmissions incur a delay of $N - W_k^1 + 1$ separately. Therefore, the overall delivery delay is $2(N - W_k^1 + 1)$.

Remark 8. *The first and the second transmissions serve a device with the first missing packets in consecutive transmissions settings can be interpreted as two transmissions serve* *the device with the first and the second missing packets in concurrent transmissions settings. In other words, if $(\xi, \xi') \cap \mathcal{W}_k = (W_k^1(1), W_k^1(2))$, then $(\xi, \xi') \cap \mathcal{W}_k = (W_k^1(1), W_k^2(1))$. Therefore, the delay incurred by two consecutive transmissions can be unified to obtain the overall delay from two concurrent transmissions.*

The delivery delay $D_k(\xi, \xi')$ of device U_k increases after transmissions of packet combination ξ by the base station and packet combination $\xi' \neq \xi$ by another device by the following quantity:

$$D_k(\xi, \xi') = \begin{cases} W_k^{e'} - W_k^2 & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (W_k^1, W_k^{e'}) \\ W_k^e - W_k^2 & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (W_k^e, W_k^1) \\ W_k^e + W_k^{e'} - 2W_k^1 & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (W_k^e, W_k^{e'}) \\ N - W_k^2 + 1 & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (W_k^1, \circledast) \\ N - W_k^2 + 1 & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (\circledast, W_k^1) \\ W_k^e + N - 2W_k^1 + 1 & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (W_k^e, \circledast) \\ W_k^{e'} + N - 2W_k^1 + 1 & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (\circledast, W_k^{e'}) \\ 2(N - W_k^1 + 1) & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (\circledast, \circledast) \end{cases} \tag{5.16}$$

Finally, following similar steps as in [92], it can be shown that the delivery delay defined in (5.16) satisfies the delivery time-delay relationship in expression (5.5).

5.9 Appendix B

Proof of Lemma 1

To demonstrate the lemma, the expected delivery delay of a device U_k after the transmissions of packet combinations ξ and $\xi' \neq \xi$ by the base station and device U_i, respectively, are derived using the delivery delay expression in Theorem 1. Note that transmitted packets are subject to Bernoulli random erasure with probability $\epsilon_{0,k}$ from the base station to device U_k and probability $\epsilon_{i,k}$ from device U_i to device U_k. The expected delivery delay $\mathbb{E}[D_k^i(\xi, \xi')]$ of device U_k after the transmissions of packet combinations ξ and $\xi' \neq \xi$ by the base station and device U_i, respectively, can be expressed as:

$$\begin{aligned}\mathbb{E}[D_k^i(\xi,\xi')] =&(1-\epsilon_{0,k})(1-\epsilon_{i,k})D_k^i(\xi,\xi') + (1-\epsilon_{0,k})\epsilon_{i,k}D_k^i(\xi,\varnothing)\\ &+\epsilon_{0,k}(1-\epsilon_{i,k})D_k^i(\varnothing,\xi') + \epsilon_{0,k}\epsilon_{i,k}D_k^i(\varnothing,\varnothing)\end{aligned} \tag{5.17}$$

Here, expression (5.17) includes four cases corresponding to four combinations of the packet reception and loss probabilities in two interfaces. Now, using the delivery delay expression defined in Theorem 1, the expected delivery delay increment in each scenario can be expressed as follows:

- If $(\xi,\xi')\cap\mathcal{W}_k = (W_k^1, W_k^{e'})$, then

$$\mathbb{E}[D_k^i(\xi,\xi')] = (\epsilon_{0,k}+\epsilon_{i,k})(N+1) + W_k^2(1-\epsilon_{0,k}) + W_k^{e'}(1-\epsilon_{i,k}) - 2\epsilon_{0,k}W_k^1 \tag{5.18}$$

- If $(\xi,\xi')\cap\mathcal{W}_k = (W_k^e, W_k^1)$, then

$$\mathbb{E}[D_k^i(\xi,\xi')] = (\epsilon_{0,k}+\epsilon_{i,k})(N+1) + W_k^e(1-\epsilon_{0,k}) + W_k^2(1-\epsilon_{i,k}) - 2\epsilon_{i,k}W_k^1 \tag{5.19}$$

- If $(\xi,\xi')\cap\mathcal{W}_k = (W_k^e, W_k^{e'})$, then

$$\mathbb{E}[D_k^i(\xi,\xi')] = (\epsilon_{0,k}+\epsilon_{i,k})(N+1) + W_k^e(1-\epsilon_{0,k}) + W_k^{e'}(1-\epsilon_{i,k}) - 2W_k^1 \tag{5.20}$$

- If $(\xi,\xi')\cap\mathcal{W}_k = (W_k^1, \circledast)$, then

$$\mathbb{E}[D_k^i(\xi,\xi')] = \epsilon_{0,k}(N+1) - W_k^2(1-\epsilon_{0,k}) + N - 2\epsilon_{0,k}W_k^1 + 1 \tag{5.21}$$

- If $(\xi,\xi')\cap\mathcal{W}_k = (\circledast, W_k^1)$, then

$$\mathbb{E}[D_k^i(\xi,\xi')] = \epsilon_{i,k}(N+1) - W_k^2(1-\epsilon_{i,k}) + N - 2\epsilon_{i,k}W_k^1 + 1 \tag{5.22}$$

- If $(\xi,\xi')\cap\mathcal{W}_k = (W_k^e, \circledast)$, then

$$\mathbb{E}[D_k^i(\xi,\xi')] = \epsilon_{0,k}(N+1) + W_k^e(1-\epsilon_{0,k}) + N - 2W_k^1 + 1 \tag{5.23}$$

- If $(\xi, \xi') \cap \mathcal{W}_k = (\circledast, W_k^{e'})$, then

$$\mathbb{E}[D_k^i(\xi, \xi')] = \epsilon_{i,k}(N+1) + W_k^{e'}(1-\epsilon_{i,k}) + N - 2W_k^1 + 1 \tag{5.24}$$

- If $(\xi, \xi') \cap \mathcal{W}_k = (\circledast, \circledast)$, then

$$\mathbb{E}[D_k^i(\xi, \xi')] = 2(N - W_k^1 + 1). \tag{5.25}$$

The expected delivery delay is simplified by considering the following four cases of expression (5.6) for the delivery delay increment:

$$D_k(\xi, \xi') \approx \begin{cases} W_k^e + W_k^{e'} - 2W_k^1 & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (W_k^e, W_k^{e'}) \\ W_k^e + N - 2W_k^1 + 1 & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (W_k^e, \circledast) \\ W_k^{e'} + N - 2W_k^1 + 1 & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (\circledast, W_k^{e'}) \\ 2(N - W_k^1 + 1) & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (\circledast, \circledast) \end{cases} \tag{5.26}$$

The simplification is exact if none of the instantly decodable packets at device U_k is its first missing packet. Otherwise, the simplification is an upper bound on the expected delivery delay. Here, we trade-off some accuracy in calculation for much more analytical simplicity. Using the expected delivery delay expressions for erasure scenarios and considering the simplified delivery delay in (5.26), finally, the expected delivery delay $\mathbb{E}[D_k^i(\xi, \xi')]$ of device U_k after the transmissions of packet combinations ξ and $\xi' \neq \xi$ by the base station and device U_i, respectively, can be given by the following expression:

$$\mathbb{E}[D_k^i(\xi, \xi')] = \tag{5.27}$$

$$\begin{cases} (\epsilon_{0,k} + \epsilon_{i,k})(N+1) + W_k^e(1-\epsilon_{0,k}) + W_k^{e'}(1-\epsilon_{i,k}) - 2W_k^1 & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (W_k^e, W_k^{e'}) \\ \epsilon_{0,k}(N+1) + W_k^e(1-\epsilon_{0,k}) + N - 2W_k^1 + 1 & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (W_k^e, \circledast) \\ \epsilon_{i,k}(N+1) + W_k^{e'}(1-\epsilon_{i,k}) + N - 2W_k^1 + 1 & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (\circledast, W_k^{e'}) \\ 2(N - W_k^1 + 1) & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (\circledast, \circledast) \end{cases}$$

5.10 Appendix C

Proof of Theorem 2

Let the transmissions be composed of packet combinations ξ and $\xi' \neq \xi$ by the base station and device U_i, respectively, at time slot t. The delivery delay minimization problem can be formulated as the problem of finding the optimal transmitting device and transmitted packet combinations that minimize the following objective function:

$$\min_{\substack{U_i \in \mathcal{M} \\ \xi \in \mathcal{P}(\mathcal{N}) \\ \xi' \in \mathcal{P}(\mathcal{H}_i)}} \sum_{U_k \in \mathcal{M}} \mathbb{E}[D_k^i(\xi, \xi')]. \tag{5.28}$$

For ease of notation, the expected delivery delay of device U_k, given in Lemma 1, is denoted by the following expression:

$$\mathbb{E}[D_k^i(\xi, \xi')] = \begin{cases} \mathbb{E}[D_k^i(W_k^e, W_k^{e'})] & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (W_k^e, W_k^{e'}) \\ \mathbb{E}[D_k^i(W_k^e, \circledast)] & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (W_k^e, \circledast) \\ \mathbb{E}[D_k^i(\circledast, W_k^{e'})] & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (\circledast, W_k^{e'}) \\ \mathbb{E}[D_k^i(\circledast, \circledast)] & \text{if } (\xi, \xi') \cap \mathcal{W}_k = (\circledast, \circledast) \end{cases}$$

Let $\mathcal{X}(\xi)$ and $\mathcal{X}(\xi')$ be the sets of targeted devices with instantly decodable packets by packet combinations ξ and ξ', respectively. The delivery delay minimization problem can be expressed as:

$$\begin{aligned} \min_{\substack{U_i \in \mathcal{M} \\ \xi \in \mathcal{P}(\mathcal{N}) \\ \xi' \in \mathcal{P}(\mathcal{H}_i)}} & \sum_{U_k \in \mathcal{M} \backslash (\mathcal{X}(\xi) \cap \mathcal{X}(\xi'))} \mathbb{E}[D_k^i(\circledast, \circledast)] + \sum_{U_k \in \mathcal{X}(\xi) \backslash \mathcal{X}(\xi')} \mathbb{E}[D_k^i(W_k^e, \circledast)] + \sum_{U_k \in \mathcal{X}(\xi') \backslash \mathcal{X}(\xi)} \mathbb{E}[D_k^i(\circledast, W_k^{e'})] \\ & + \sum_{U_k \in \mathcal{X}(\xi) \cap \mathcal{X}(\xi')} \mathbb{E}[D_k^i(W_k^e, W_k^{e'})]. \end{aligned} \tag{5.29}$$

In other words, expression (5.29) minimizes the expected delivery delays of all devices belonging to four different sets. Note that these four sets represent all feasible scenarios that any device can experience from ξ and ξ' transmissions. For example, the second set $\{\mathcal{X}(\xi) \backslash \mathcal{X}(\xi')\}$ of the expression includes the devices that are targeted with a packet from the base station only. The delivery delay minimization problem in (5.29) can be equivalently expressed as the maximization problem as follows:

$$\max_{\substack{U_i \in \mathcal{M} \\ \xi \in \mathcal{P}(\mathcal{N}) \\ \xi' \in \mathcal{P}(\mathcal{H}_i)}} \sum_{U_k \in \mathcal{X}(\xi) \backslash \mathcal{X}(\xi')} \mathbb{E}[D_k^i(\circledast, \circledast)] - \mathbb{E}[D_k^i(W_k^e, \circledast)] + \sum_{U_k \in \mathcal{X}(\xi') \backslash \mathcal{X}(\xi)} \mathbb{E}[D_k^i(\circledast, \circledast)] - \mathbb{E}[D_k^i(\circledast, W_k^{e'})]$$
$$+ \sum_{U_k \in \mathcal{X}(\xi) \cap \mathcal{X}(\xi')} \mathbb{E}[D_k^i(\circledast, \circledast)] - \mathbb{E}[D_k^i(W_k^e, W_k^{e'})] \qquad (5.30)$$

The above maximization problem represents that the devices, who incur a large delay when they are not targeted with specific packets, need to be targeted with those packets. According to the expression of the expected delivery delay in Lemma 1, the following equality holds:

$$\mathbb{E}[D_k^i(W_k^e, W_k^{e'})] = \mathbb{E}[D_k^i(W_k^e, \circledast)] + \mathbb{E}[D_k^i(\circledast, W_k^{e'})] - \mathbb{E}[D_k^i(\circledast, \circledast)] \qquad (5.31)$$

Therefore, using the equality expression in (5.31), the last two terms in (5.30) can be expressed as:

$$\begin{aligned} \mathbb{E}[D_k^i(\circledast, \circledast)] - \mathbb{E}[D_k^i(W_k^e, W_k^{e'})] =& \mathbb{E}[D_k^i(\circledast, \circledast)] - \mathbb{E}[D_k^i(W_k^e, \circledast)] \\ &+ \mathbb{E}[D_k^i(\circledast, \circledast)] - \mathbb{E}[D_k^i(\circledast, W_k^{e'})] \end{aligned} \qquad (5.32)$$

Substituting the results of (5.32) in expression (5.30), the delivery delay minimization problem can be formulated as the problem of finding the optimal transmitting device and transmitted packet combinations so as to maximize the following objective function:

$$\begin{aligned} &\max_{\substack{U_i \in \mathcal{M} \\ \xi \in \mathcal{P}(\mathcal{N}) \\ \xi' \in \mathcal{P}(\mathcal{H}_i)}} \sum_{U_k \in \mathcal{X}(\xi)} \mathbb{E}[D_k^i(\circledast, \circledast)] - \mathbb{E}[D_k^i(W_k^e, \circledast)] + \sum_{U_k \in \mathcal{X}(\xi')} \mathbb{E}[D_k^i(\circledast, \circledast)] - \mathbb{E}[D_k^i(\circledast, W_k^{e'})] \\ &= \max_{\substack{U_i \in \mathcal{M} \\ \xi \in \mathcal{P}(\mathcal{N}) \\ \xi' \in \mathcal{P}(\mathcal{H}_i)}} \sum_{U_k \in \mathcal{X}(\xi)} (1 - \epsilon_{0,k})(N - W_k^e + 1) + \sum_{U_k \in \mathcal{X}(\xi')} (1 - \epsilon_{i,k})(N - W_k^{e'} + 1). \end{aligned} \qquad (5.33)$$

Chapter 6

Completion Time Aware Network Codes in D2D Networks

6.1 Overview

Popular applications in wireless devices require high bandwidth and transmission energy from the base station. To address such an increasing throughput demand, cooperation among wireless devices has widely been considered as a promising solution [23, 24, 106]. In such scenarios, a group of near-by devices who are within the transmission range of each other form a D2D network. Therefore, to avoid the interference caused by multiple transmissions, a single device is allowed to transmit in each time slot. Moreover, the network coded D2D communications that takes advantages of both network coding and devices' cooperation are proven to further enhance the transmission efficiency of D2D networks [56,66, 88,89]. However, all these works consider an upper layer view of the network and abstract its physical channel conditions into simple erasure channel models. Such abstraction simplifies the analysis, design and optimization of IDNC systems at the expense of limited exploitation of the channel capacities.

In this chapter, we are interested in time-critical applications, in which each decoded packet brings new information and is immediately used at the application layer irrespective of its order. For such time-critical applications, we address the problem of minimizing the overall completion time required for recovering all missing packets at the devices using IDNC and devices' cooperation. Further, wireless channels are dynamic with heterogeneous capacities, which requires intelligent selection of the transmission rate by the transmitting device. Indeed, a packet transmission with a high rate will take a shorter time but will be only successfully received by the devices with high channel capacities. On the contrary, a packet transmission with a lower rate will be received by more devices but will take a longer time. Therefore, a balance among these conflicting effects is necessary to obtain the minimum completion time. Subsequently, this chapter bridges the gap between the three different features (i.e., transmission rate aware network coding, D2D networks and time-critical applications), and develops an unified framework that reduces the completion time while allowing progressive decoding of packets.

In particular, we aim to design an efficient rate-aware IDNC framework that minimizes the completion time in a fully connected D2D network. Our main contributions can be summarized as follows:

- We introduce a rate aware IDNC graph to represent all feasible rate and coding decisions for all potential transmitting devices in one unified framework. Each maximal clique of this graph represents a transmitting device, a transmission rate and an XOR packet combination.

- Using the rate aware graph and the properties of the optimal schedule, we design a completion time reduction heuristic that jointly selects a transmitting device, a transmission rate and an XOR packet combination. Moreover, this heuristic strikes a balance between the transmission rate and the number of targeted devices with a new packet in each transmission.

- Simulation results show that our proposed IDNC algorithm reduces the completion time compared to the rate oblivious network coding for D2D networks [65, 66].

The rest of the chapter is organized as follows. The system model is described in Section 6.2. The rate aware IDNC graph is introduced in Section 6.3. The minimum overall completion time problem is formulated in Section 6.4. Using the rate aware graph and the problem formulation, we design a heuristic in Section 6.5. Simulation results are presented in Section 6.6. We conclude the chapter in Section 6.7.

6.2 System Model and Parameters

In this section, we first describe the network and physical layer models and then characterize the packet reception status of the devices. We also provide several important definitions that will be used in the remaining of this chapter.

6.2.1 Network and Physical Layer Models

We consider a set of devices $\mathcal{M} = \{U_1, ..., U_M\}$, where each device $U_i \in \mathcal{M}$ initially possesses a subset of source packets from $\mathcal{N} = \{P_1, ..., P_N\}$ (e.g., side information or received from the central station in the previous broadcasting [88,96]). We further consider that a packet from $\mathcal{N}$ is possessed by at least one device in $\mathcal{M}$.[*] All M devices are interested in receiving all N packets. To recover the missing packets, the devices cooperate with each other by exchanging XOR coded packets. Each packet in $\mathcal{N}$ consists of B bits and thus, an XOR coded packet is also B bits. We consider a fully connected D2D communications network such that the devices are within transmission range of each other. As simultaneous transmissions create interference at the devices, a single device transmits at any given transmission (referred to as *a transmitting device*). We use time index t to represent the starting time of the t-th D2D transmission. Therefore, $t = 0$ refers to the beginning of the D2D communications phase.

Let $h_{i,k}$ be the complex channel gain from device U_i to device U_k. We assume that $h_{i,k}$ remains constant during a D2D transmission and changes independently from a transmission to the other. Let Q_i be the transmit power of a device U_i fixed to a nominal value. Moreover, the transmit powers of different devices are not necessarily identical. The channel capacity $C_{i,k}$ from device U_i to device U_k can be expressed as:

$$C_{i,k} = \text{Bandwidth} \times \log_2 \left(1 + \frac{Q_i |h_{i,k}|^2}{\sigma^2}\right), \tag{6.1}$$

where σ^2 denotes the Gaussian noise variance. The channel capacities of all pairs of devices can be stored in an $M \times M$ *capacity status matrix (CSM)* $\mathbf{C} = [C_{i,k}], \ \forall(U_i, U_k)$. Note that $C_{i,i} = 0, \forall(U_i, U_i)$ since a device U_i cannot transmit to itself. The information of the channel capacities can be assembled by using control signals over dedicated channels, but at

[*] Throughout this chapter, we use calligraphic letters to denote sets and their corresponding capital letters to denote the cardinalities of these sets (e.g., $M = |\mathcal{M}|$).

the expense of additional overhead in the network. Therefore, the proposed solution is more suitable for medium sized social networks with a moderate number of devices rather than large public networks.

Example 16. *An example of CSM with $M = 3$ devices is:*

$$\mathbf{C} = \begin{pmatrix} 0 & 5.3 & 6.5 \\ 6.1 & 0 & 5.9 \\ 8.5 & 7.7 & 0 \end{pmatrix}. \tag{6.2}$$

For simplicity, the CSM of (6.2) *assumes unit bandwidth, which can be easily replaced by another appropriate constant scaling factor. Now, $C_{3,1} = 8.5$ represents that the channel capacity from transmitting device U_3 to device U_1 is 8.5. Note that this CSM is not symmetric because of the difference in the transmit powers of different devices and the different levels of interference experienced by each of them. In other words, $C_{3,1} \neq C_{1,3}$.*

Definition 25. *(Absolute Transmission Time) The absolute transmission time in the t-th D2D transmission is defined as the time required for sending a packet of size B using an adopted transmission rate $r(t)$ at the transmitting device. In other words, the absolute transmission time in the t-th D2D transmission is $\frac{B}{r(t)}$.*

In this chapter, we assume that any transmitting device U_i can adjust its modulation scheme to adopt any transmission rate $r(t)$. Moreover, the t-th D2D transmission from device U_i will be successful at device U_k if the adopted transmission rate $r(t)$ is smaller than or equal to the channel capacity from device U_i to device U_k (i.e., $r(t) \leq C_{i,k}$). For $r(t) > C_{i,k}$, device U_k will not be able to receive the transmission from device U_i.

6.2.2 Packet Reception Status

At any given time index t, two sets of packets can be attributed to each device U_k.

- The *Has* set $\mathcal{H}_k(t)$ is defined as the set of packets successfully received by device U_k.
- The *Wants* set $\mathcal{W}_k(t)$ is defined as the set of missing packets at device U_k. In other words, $\mathcal{W}_k(t) = \mathcal{N} \setminus \mathcal{H}_k(t)$.

The set of devices having *non-empty Wants sets* at time index t is denoted by $\mathcal{M}_w(t)$. In other words, $\mathcal{M}_w(t) = \{U_k \in \mathcal{M} | \mathcal{W}_k(t) \neq \varnothing\}$.

Example 17. *Let us consider a network with three Has sets* $\mathcal{H}_1 = \{P_1\}$*,* $\mathcal{H}_2 = \varnothing$*,* $\mathcal{H}_3 = \{P_1, P_2\}$ *. Therefore, three Wants sets are:* $\mathcal{W}_1 = \{P_2\}$*,* $\mathcal{W}_2 = \{P_1, P_2\}$*,* $\mathcal{W}_3 = \varnothing$*.*

We now provide several important definitions that will be used throughout this chapter.

Definition 26. *(Instantly Decodable Transmission) A transmission from device* U_i *is instantly decodable for device* U_k *if it contains exactly one source packet from* $\mathcal{W}_k$ *and the adopted transmission rate* $r(t)$ *is smaller than or equal to the channel capacity* $C_{i,k}$*.*

Definition 27. *(Individual Completion Time) The individual completion time* T_k *of device* U_k *is the total transmission time required in the D2D communications phase until it recovers all* N *packets.*

Definition 28. *(Overall Completion Time) The overall completion time* T *is the total transmission time required in the D2D communications phase until all* M *devices recover all* N *packets. In other words,* $T = \max_{U_k \in \mathcal{M}}\{T_k\}$*.*

Definition 29. *(D2D Transmission Schedule) A D2D transmission schedule* $\mathcal{S} = \{\kappa(t)\}, \forall t \in \{1, ..., |\mathcal{S}|\}$ *is defined as a collection of* $\kappa(t)$ *representing the transmitting devices, packet combinations and transmission rates until all* M *devices recover all* N *packets.* $\mathbf{S}$ *is then defined as the set of all possible D2D transmission schedules, i.e.,* $\mathcal{S} \in \mathbf{S}$*.*

Due to the instant decodability constraint, IDNC may not be able to target all devices with a new packet in each transmission. Further, in D2D networks, a transmitting device certainly cannot benefit from its own transmission. Therefore, the devices who do not receive a new packet from a transmission experience an increase in their individual completion time, which can be defined as follows:

Definition 30. *(Time Delay) In the* t*-th D2D transmission, device* U_k *with non-empty Wants set experiences time delay of* $\frac{B}{r(t)}$ *seconds when the transmission with rate* $r(t)$ *is not instantly decodable for that device [95].*

Definition 31. *(Accumulative Time Delay) Given a D2D transmission schedule* $\mathcal{S}$*, accumulative time delay* $\mathcal{D}_k(\mathcal{S})$ *of device* U_k *is defined as the sum of time delays experienced by device* U_k *in the D2D communications phase until it recovers all* N *packets.*

6.3 Rate Aware IDNC Graph

In this section, we introduce a rate aware IDNC graph that represents all feasible rate and coding decisions for all M potential transmitting devices in one unified framework. At any given time index t, this graph allows us to systematically select a transmitting device, a packet combination and a transmission rate so that the transmission is instantly decodable to a subset or all of other devices. The representation of rate and coding decisions in one graph was suggested in [95] for PMP networks, where a central transmission unit possesses all packets and always transmits packets to devices. However, our proposed graph for a D2D network is different from the one in [95] as any device can be chosen as the transmitting device at any time index t. Moreover, the chosen device may possess only a subset of N packets, which can be used to form a coded packet by that device.

6.3.1 Vertex Generation

We first focus on generating the vertices for a potential transmitting device U_i in the graph $\mathcal{G}$. Note that device U_i can form a packet combination using its previously received packets. Therefore, one or more vertices are generated in graph $\mathcal{G}$ from each single missing packet of other devices in $\mathcal{M}_w$, if such packet belongs to the Has set $\mathcal{H}_i$ of device U_i.

Definition 32. *The set of transmission rates $\mathcal{R}_i(t)$ for a potential transmitting device U_i includes the channel capacities from device U_i to the devices having at least one missing packet that belongs to the Has set $\mathcal{H}_i$ of device U_i. In other words, $\mathcal{R}_i(t) = \{C_{i,k} | (\mathcal{H}_i \cap \mathcal{W}_k) \neq \varnothing, \forall U_k \in \mathcal{M}\}$.*

Definition 33. *The set of feasible transmission rates $\mathcal{R}_{i,k}(t)$ from device U_i to device U_k includes a subset of transmission rates from $\mathcal{R}_i(t)$, which are less than or equal to channel capacity $C_{i,k}$. In other words, $\mathcal{R}_{i,k}(t) = \{r \in \mathcal{R}_i(t) | r \leq C_{i,k}\}$.*

A device U_k will receive a transmission from device U_i if the adopted transmission rate $r(t) \in \mathcal{R}_{i,k}(t)$. Therefore, $|\mathcal{R}_{i,k}(t)|$ vertices are generated for each missing packet $P_l \in \{\mathcal{H}_i \cap \mathcal{W}_k\}, \forall U_k \in \mathcal{M}$. In other words, a vertex v^i_{rkl} is generated for each feasible association of a transmitting device U_i, a transmission rate $r \in \mathcal{R}_{i,k}(t)$ and a missing packet $P_l \in \{\mathcal{H}_i \cap \mathcal{W}_k\}$ of a device $U_k \in \mathcal{M}$. Similarly, the vertices for M potential transmitting devices are generated in graph $\mathcal{G}$.

Example 18. *To illustrate the vertex generation, let us consider the system in Example 17 and the CSM in* (6.2)*. We focus on generating the vertices only for the potential transmitting device U_3 in the graph $\mathcal{G}$. Note that Wants sets of devices U_1 and U_2 are $\mathcal{W}_1 = \{P_2\}$ and $\mathcal{W}_2 = \{P_1, P_2\}$, respectively. Moreover, these missing packets belong to Has set $\mathcal{H}_3 = \{P_1, P_2\}$ of device U_3. Here, the set of transmission rates for device U_3 is $\mathcal{R}_3(t) = \{8.5, 7.7\}$, which includes the channel capacities from device U_3 to devices U_1 and U_2 as $(\mathcal{H}_3 \cap \mathcal{W}_1) \neq \varnothing$ and $(\mathcal{H}_3 \cap \mathcal{W}_2) \neq \varnothing$. With this result, the set of feasible transmission rates from device U_3 to device U_1 is $\mathcal{R}_{3,1}(t) = \{8.5, 7.7\}$. Therefore, packet P_2 of device U_1 generates two vertices such as $v^3_{8.5,1,2}, v^3_{7.7,1,2}$.*[†] *Similarly, the set of feasible transmission rates from device U_3 to device U_2 is $\mathcal{R}_{3,2}(t) = \{7.7\}$. Therefore, each of packets P_1 and P_2 of device U_2 generates a single vertex such as $v^3_{7.7,2,1}$ and $v^3_{7.7,2,2}$. To conclude, the vertices for potential transmitting device U_3 in graph $\mathcal{G}$ are $\{v^3_{8.5,1,2}, v^3_{7.7,1,2}, v^3_{7.7,2,1}, v^3_{7.7,2,2}\}$.*

6.3.2 Edge Generation

Unlike Chapters 4 and 5, this chapter considers a single sender and, therefore, it is more convenient to represent coding opportunities in the graph. Therefore, two vertices v^i_{rkl} and $v^{i'}_{r'k'l'}$ are adjacent by an edge in graph $\mathcal{G}$, if all the following three conditions are true:

- **C1** : $i = i'$. This represents that all adjacent vertices in the graph correspond to the same transmitting device.

- **C2** : $r = r'$. This represents that all adjacent vertices in the graph have the same transmission rate.

- **C3** : $(P_l = P_{l'})$ OR $(P_{l'} \in \mathcal{H}_k$ and $P_l \in \mathcal{H}_{k'})$. This represents that the packet combination is immediately decodable for both devices U_k and $U_{k'}$. Indeed, $(P_l = P_{l'})$ represents that the same packet is requested by two different devices U_k and $U_{k'}$. Moreover, $(P_{l'} \in \mathcal{H}_k$ and $P_l \in \mathcal{H}_{k'})$ represents the immediate decodability of the packet combination $P_l \oplus P_{l'}$ for both devices U_k and $U_{k'}$ as the missing packet of one device belongs to the Has set of the other device.

[†] Vertex $v^3_{8.5,1,2}$ represents transmitting device U_3, transmission rate 8.5, and device U_1 targeted with packet P_2.

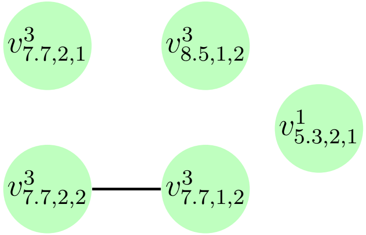


Figure 6.1: The rate aware IDNC graph $\mathcal{G}$ corresponding to system in Example 17 and CSM in (6.2).

6.3.3 Maximal Cliques

With the rate aware IDNC graph representation, a maximal clique κ in graph $\mathcal{G}$ satisfies the following three criterion:

- All the vertices in maximal clique κ belong to a single transmitting device.
- The transmission rate $r(t)$ identified by the vertices in maximal clique κ is no larger than the channel capacities of all the devices having vertices in maximal clique κ. In other words, $r(t) \leq C_{i,k}; \forall U_k | v^i_{rkl} \in \kappa$.
- All the devices having vertices in maximal clique κ can decode a new packet from the transmission.

The selected transmitting device generates a coded packet by XORing all the source packets identified by the vertices of maximal clique κ in graph $\mathcal{G}$. It also adopts the transmission rate corresponding to the vertices of maximal clique κ. Moreover, each device can have at most one vertex in a maximal clique and the selection of a maximal clique κ means the selection of a set of *targeted devices.*

Example 19. *The rate aware IDNC graph $\mathcal{G}$ corresponding to system in Example 17 and CSM in* (6.2) *is shown in Fig. 6.1. An example of a maximal clique in graph $\mathcal{G}$ is $\kappa = \{v^3_{7.7,2,2}, v^3_{7.7,1,2}\}$.*

6.4 Minimum Completion Time Formulation

In this section, we aim to find the optimal schedule $\mathcal{S}^*$ that minimizes the overall completion time T in D2D communications phase. The optimal schedule that minimizes the overall completion time can be expressed in terms of individual completion times of all devices such as:

$$
\begin{aligned}
\mathcal{S}^* &= \arg\min_{\mathcal{S}\in\mathbf{S}}\{T(\mathcal{S})\} \\
&= \arg\min_{\mathcal{S}\in\mathbf{S}}\left\{\max_{U_k\in\mathcal{M}}\{T_k(\mathcal{S})\}\right\}
\end{aligned} \tag{6.3}
$$

We now define the optimal schedule $\mathcal{S}^*$ in expression (6.3) using the following theorem:

Theorem 3. *The minimum overall completion time problem in a fully connected D2D communications network can be formulated as a transmission schedule selection problem such that:*

$$
\mathcal{S}^* = \arg\min_{\mathcal{S}\in\mathbf{S}}\left\{\max_{U_k\in\mathcal{M}}\left\{\frac{B.W_k(0)}{\tilde{R}_k(\mathcal{S})} + \mathcal{D}_k(\mathcal{S})\right\}\right\}, \tag{6.4}
$$

where B is the packet size, $W_k(0)$ is the initial Wants set size of device U_k at time index $t = 0$, $\mathcal{D}_k(\mathcal{S})$ is the accumulative time delay of device U_k in schedule $\mathcal{S}$ and $\tilde{R}_k(\mathcal{S})$ is the harmonic mean of the transmission rates of time indices that are instantly decodable for device U_k in schedule $\mathcal{S}$.

Proof. To prove this theorem, the individual completion time of a device is first expressed as a sum of the instantly decodable transmissions time and the delay time. Then, the time of all instantly decodable transmissions is obtained by using the harmonic mean of the rates of the transmissions that are instantly decodable for that device. The complete proof of this theorem can be found in the appendix in Section 6.8. ■

The optimal schedule that minimizes the overall completion time in a fully connected D2D communications network is the solution of the optimization problem in (6.4). Moreover, the optimal schedule needs to exploit the heterogeneity of devices' channel capacities and the interdependence of devices' packet reception. In fact, the decision at the current time index is dependent on the future coding situations. With this result, we infer that finding the optimal schedule $\mathcal{S}^*$ is computationally complex, which was also shown in [66, 93] without dynamic rate adaptation using a stochastic shortest path framework. To reduce the computational complexity, in the next section, we design a heuristic algorithm that efficiently reduces the overall completion time.

6.5 Completion Time Reduction Heuristic

In this section, we design a computationally simple heuristic algorithm that selects a transmitting device, a packet combination and a transmission rate at a given time index t without going through all future possibilities of Has sets and rate aware coding decisions. Moreover, this heuristic algorithm approximates the minimum completion time problem in (6.4) by an online completion time reduction problem.

6.5.1 Characterizing the Completion Time Reduction Problem

At a given time index t, we compute a lower bound on the individual completion times of all devices. This lower bound is computed separately for each device and does not require to exploit the interdependence of devices' packet reception. In fact, we use this lower bound as a metric to map the transmission schedule selection problem in (6.4) into an online maximal clique selection problem.

Corollary 1. *A lower bound on individual completion time $\bar{T}_k(t)$ of device $U_k \in \mathcal{M}_w$ in a given time index t can be approximated as:*

$$\bar{T}_k(t) \approx \frac{B.W_k(0)}{\tilde{C}_k} + \mathcal{D}_k(t), \tag{6.5}$$

where $\mathcal{D}_k(t)$ is the accumulative time delay experienced by device U_k until time index t and $\tilde{C}_k$ is the harmonic mean of the channel capacities from all other devices $\{\mathcal{M} \setminus U_k\}$ to device U_k.

Proof. The expression in Corollary 1 is the same expression in Theorem 3, except $\mathcal{D}_k(\mathcal{S})$ and $\tilde{R}_k(\mathcal{S})$ of Theorem 3 is replaced by $\mathcal{D}_k(t)$ and $\tilde{C}_k$, respectively. In the best case scenario, all transmissions starting from time index t will be instantly decodable for device U_k and it will experience no further time delay. With such scenario, $\mathcal{D}_k(\mathcal{S}) = \mathcal{D}_k(t)$. Furthermore, device U_k can receive a missing packet from any other device in a D2D transmission until it recovers all N packets. Therefore, we replace $\tilde{R}_k(\mathcal{S})$ by $\tilde{C}_k$, where $\tilde{C}_k$ is the harmonic mean of the channel capacities from all other devices to device U_k. This is an approximation as $\tilde{C}_k$ will be exactly equal to $\tilde{R}_k(\mathcal{S})$ if device U_k receives an equal number of packets from other devices with the rates equal to the channel capacities. ■

Finally, the following proposition describes an online maximum weight clique selection problem over graph $\mathcal{G}$ that efficiently reduces the overall completion time in D2D communications phase.

Proposition 2. *To find an efficient solution with a low computational complexity, the completion time reduction problem can be solved heuristically by an online maximum weight clique selection in the rate aware IDNC graph $\mathcal{G}$ wherein the weight of a vertex v^i_{rkl} is given by:*

$$\psi(v^i_{rkl}) = 2^{M_w - d_k + 1} \bar{T}_k(t) \left(\frac{r}{B}\right), \tag{6.6}$$

where M_w is the number of devices with non-empty Wants sets at time index t, d_k is the order of the vertex's corresponding device U_k in the group that arranges all devices in $\mathcal{M}_w(t)$ in non-increasing order of lower bound on individual completion times and r is the rate corresponding to vertex v^i_{rkl}.

Proof. The proof of this proposition can be found in the appendix in Section 6.9. ■

It is well known that finding the maximum weight clique in a graph is NP-hard [102] and thus, solving Proposition 2 is also NP-hard. In the next sub-section, we greedily select a maximal clique using the vertices' weights defined in (6.6).

6.5.2 Greedy Maximal Clique Selection Algorithm

We now describe a maximal clique selection algorithm following a similar greedy vertex search approach as in [66, 93], but using the priority of vertices defined in Proposition 2. To define the vertices' weights in this heuristic, we first define $\pi_{rkl,r'k'l'}$ as the adjacency indicator of vertices v^i_{rkl} and $v^{i'}_{r'k'l'}$ in graph $\mathcal{G}$ such that:

$$\pi_{rkl,r'k'l'} = \begin{cases} 1 & \text{if } v^i_{rkl} \text{ is adjacent to } v^{i'}_{r'k'l'} \text{ in } \mathcal{G}, \\ 0 & \text{otherwise.} \end{cases} \tag{6.7}$$

Algorithm 5: Proposed RAC-IDNC Algorithm

```
1. Construct rate aware graph $\mathcal{G}$.
   Initialize maximal clique $\kappa = \varnothing$.
   Set $\mathcal{G}(\kappa) \leftarrow \mathcal{G}$.
2. While $\mathcal{G}(\kappa) \neq \varnothing$ do
   Compute weight $w(v^i_{rkl}), \forall v^i_{rkl} \in \mathcal{G}(\kappa)$ using (6.9).
   Select $v^{i^*}_{rkl} = \arg\max_{v^i_{rkl} \in \mathcal{G}(\kappa)} \{w(v^i_{rkl})\}$.
   Set $\kappa \leftarrow \kappa \cup v^{i^*}_{rkl}$.
   Extract new subgraph $\mathcal{G}(\kappa)$.
   end while
```

We then define the weighted degree Ω^i_{rkl} of vertex v^i_{rkl} as:

$$\Omega^i_{rkl} = \sum_{v^{i'}_{r'k'l'} \in \mathcal{G}} \pi_{rkl,r'k'l'} \;\; \psi(v^{i'}_{r'k'l'}), \tag{6.8}$$

where $\psi(v^{i'}_{r'k'l'})$ is the priority of vertex $v^{i'}_{r'k'l'}$ defined in (6.6). We finally define the weight of vertex v^i_{rkl} in this heuristic as:

$$w(v^i_{rkl}) = \psi(v^i_{rkl})\Omega^i_{rkl} = 2^{M_w - d_k + 1}\bar{T}_k(t)\left(\frac{r}{B}\right)\Omega^i_{rkl}. \tag{6.9}$$

At Step 1, the algorithm selects the vertex $v^{i^*}_{rkl}$ that has the maximum weight $w(v^i_{rkl})$ and adds it to maximal clique κ (i.e., $\kappa = \{v^{i^*}_{rkl}\}$). Then, the algorithm extracts the subgraph $\mathcal{G}(\kappa)$, which consists of vertices in graph $\mathcal{G}$ that are adjacent to vertex $v^{i^*}_{rkl}$. At Step 2, the algorithm selects a new maximum weight vertex $v^{i^*}_{rmn}$ from subgraph $\mathcal{G}(\kappa)$. This process is repeated until no further vertex is adjacent to all the vertices in maximal clique κ. We refer to this heuristic as *rate aware cooperative IDNC* (RAC-IDNC) algorithm and summarized in Algorithm 5. Using a similar analysis of [93], it can be shown that the overall complexity of selecting a maximal clique using our proposed RAC-IDNC algorithm is $\mathcal{O}(M^3N)$.

6.6 Simulation Results

In this section, we present the performance of our proposed RAC-IDNC algorithm in a D2D network. We consider a square plane with area of 2500 meter^2, in which the devices

Table 6.1: Physical layer parameters

Bandwidth	1 MHz
Channel model	Rayleigh fading
Channel estimation	Perfect
Path loss model	$148 + 40\log_{10}(\text{distance[km]})$
Shadowing variance	4 dB
Transmit power	[-42.5, -47.5] dBm/Hz
Noise power	-174 dBm/Hz

are uniformly distributed. Table 6.1 summarizes the physical layer parameters used in the simulations. We present the results comparing the performance of our proposed RAC-IDNC algorithm to the following algorithms:

- The *uncoded broadcast* in which a random device transmits an uncoded packet from its Has set that is missing at the largest number of other devices. Moreover, this scheme uses the minimum channel capacity from the transmitting device to all other devices as the transmission rate.

- *Cooperative-RLNC:* This RLNC algorithm selects the device with the highest total knowledge space rank as the transmitting device in a D2D transmission [65]. The selected device encodes all packets in its knowledge space using random coefficient from a large Galois field. However, this algorithm ignores the dynamic transmission rates. Therefore, for the transmission to be successfully received by all other devices, the minimum channel capacity from the transmitting device to all other devices is adopted as the transmission rate.

- *Cooperative-IDNC:* This IDNC algorithm jointly selects a transmitting device and an XOR packet combination to reduce the completion time in each D2D transmission [66]. However, this algorithm ignores the dynamic transmission rates. Therefore, for the transmission to be successfully received by all the targeted devices, the minimum channel capacity from the transmitting device to all targeted devices is adopted as the transmission rate.

As discussed in Section 6.2.1, at the beginning of the D2D communications phase, each device holds between 45% and 55% of N packets in all scenarios. These percentages of initial

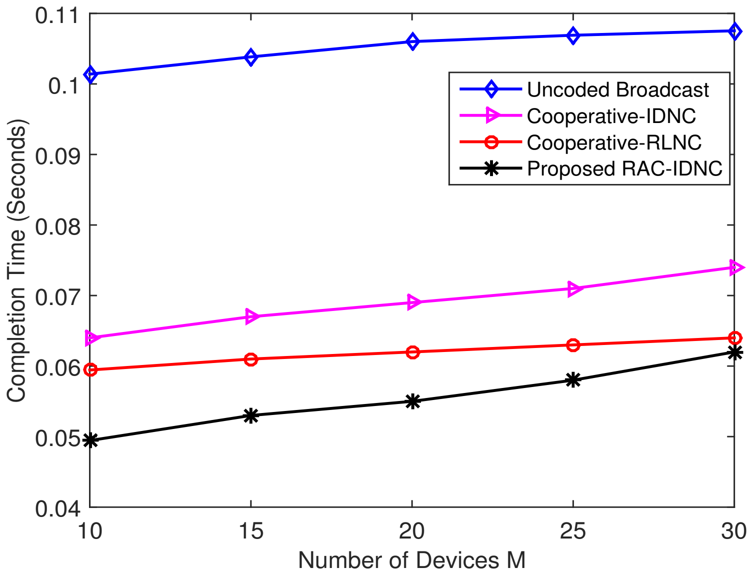


Figure 6.2: Completion time versus different number of devices

received packets are adopted arbitrarily to model the side information or received packets from the central transmission unit in previous broadcast sessions.

Fig. 6.2 shows the completion time achieved by different algorithms against different number of devices M (for $N = 20$ packets, a packet's size $B = 80$ kilobits $= 10$ kilobytes). From this figure, we can see that our proposed RAC-IDNC algorithm outperforms both cooperative-IDNC and cooperative-RLNC algorithms due to using the rate aware IDNC graph and taking into account dynamic transmission rates and lower bound on the individual completion times in making decisions. Moreover, both cooperative-IDNC and cooperative-RLNC algorithms blindly adopt the minimum channel capacity from the transmitting device to a large number of other devices as the transmission rate. This is a low transmission rate as it is the minimum of a set of heterogeneous channel capacities. As expected, the uncoded broadcast scheme performs poorly due to ignoring both network coding and dynamic rate adaptation.

Fig. 6.3 shows the completion time achieved by different algorithms against different number of packets N (for $M = 20$ devices, a packet's size $B = 80$ kilobits $= 10$ kilobytes). From this figure, we can observe that our proposed RAC-IDNC algorithm outperforms both cooperative-IDNC and cooperative-RLNC algorithms as it balances between the number of targeted devices and the transmission rate. Finally, Fig. 6.4 shows the completion time

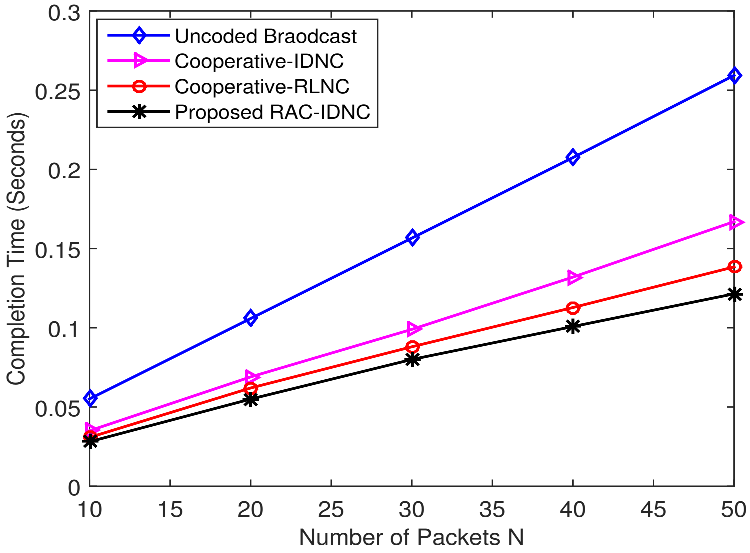


Figure 6.3: Completion time versus different number of packets

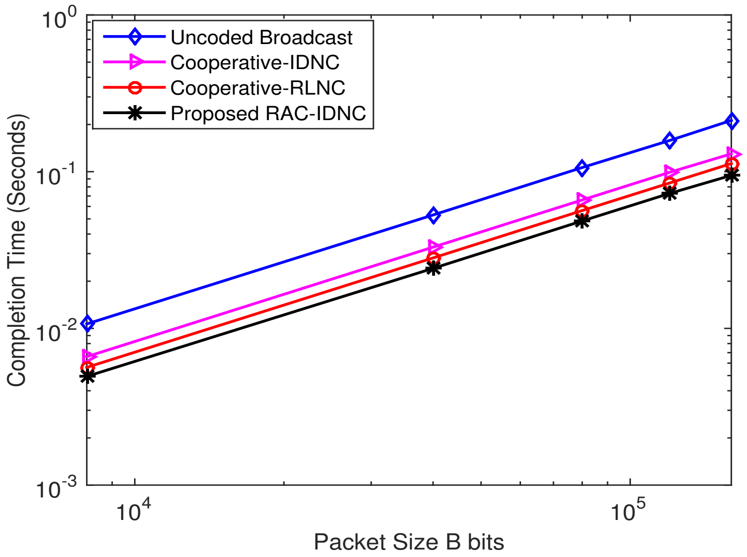


Figure 6.4: Completion time versus different packet's sizes

achieved by different algorithms against different packet's sizes B (for $M = 20$ devices, $N = 20$ packets). According to Definition 25, the absolute transmission time required for each transmission is linear with the size of a packet. Moreover, according to expression (6.4), the completion time is also linear with the size of a packet. This property holds for all algorithms in Fig. 6.4.

6.7 Conclusion

In this chapter, we developed an efficient rate-aware IDNC framework that reduces the overall completion time in a fully connected D2D network. In particular, we introduced a rate aware IDNC graph to represent all feasible rate and coding decisions for all potential transmitting devices in one unified framework. Since finding the minimum completion time schedule was computationally complex, we designed a completion time reduction heuristic that balances between the transmission rate and the number of targeted devices. Simulation results showed that our proposed RAC-IDNC algorithm reduces the overall completion time compared to the conventional network coding algorithms.

6.8 Appendix A

Proof of Theorem 3

The optimal schedule that minimizes the overall completion time can be expressed in terms of individual completion times of all devices as follows:

$$\begin{aligned} \mathcal{S}^* &= \arg\min_{\mathcal{S}\in\mathbf{S}} \{T(\mathcal{S})\} \\ &= \arg\min_{\mathcal{S}\in\mathbf{S}} \left\{ \max_{U_k\in\mathcal{M}} \{T_k(\mathcal{S})\} \right\} \end{aligned} \tag{6.10}$$

To prove Theorem 3, we need to express individual completion time T_k of device U_k in terms of packet size B, initial Wants set size $W_k(0)$, harmonic mean $\tilde{R}_k(\mathcal{S})$ and accumulative time delay $\mathcal{D}_k(\mathcal{S})$ such as:

$$T_k(\mathcal{S}) = \frac{B.W_k(0)}{\tilde{R}_k(\mathcal{S})} + \mathcal{D}_k(\mathcal{S}). \tag{6.11}$$

For a given transmission schedule $\mathcal{S}$, a transmission at time index $t \in \mathcal{S}$ can be either instantly decodable or not instantly decodable for device $U_k \in \mathcal{M}$. With these two cases, individual completion time of device U_k can be defined as:

$$T_k(\mathcal{S}) = \sum_{t \in \alpha_k(\mathcal{S})} \frac{B}{r(t)} + \sum_{t \in \beta_k(\mathcal{S})} \frac{B}{r(t)}, \tag{6.12}$$

where $\alpha_k(\mathcal{S})$ and $\beta_k(\mathcal{S})$ are the sets of time indices that are instantly decodable and not instantly decodable, respectively for device U_k.

We now further elaborate the expression in (6.12). Device U_k needs to recover $W_k(0)$ missing packets in the D2D phase and thus, needs to experience $W_k(0)$ instantly decodable transmissions in schedule $\mathcal{S}$. Since set $\alpha_k(\mathcal{S})$ includes all instantly decodable transmissions for device U_k in schedule $\mathcal{S}$ and one packer is decoded per instantly decodable transmission, we can set $W_k(0) = |\alpha_k(\mathcal{S})|$. With this result, the first term in (6.12) can be expressed as:

$$\sum_{t \in \alpha_k(\mathcal{S})} \frac{B}{r(t)} = B \sum_{t \in \alpha_k(\mathcal{S})} \frac{1}{r(t)} = \frac{B.W_k(0)}{\tilde{R}_k(\mathcal{S})}, \tag{6.13}$$

where, $\tilde{R}_k(\mathcal{S})$ is the harmonic mean of the transmission rates of time indices in $\alpha_k(\mathcal{S})$.

On the other hand, device U_k having non-empty Wants set experiences time delay increase in each time index $t \in \beta_k(\mathcal{S})$. From Definition 30, device U_k experiences time delay of $\frac{B}{r(t)}$ in time index $t \in \beta_k(\mathcal{S})$. With this result, using Definition 31, the accumulative time delay experienced by device U_k over all transmissions in $\beta_k(\mathcal{S})$ is:

$$\sum_{t \in \beta_k(\mathcal{S})} \frac{B}{r(t)} = \mathcal{D}_k(\mathcal{S}). \tag{6.14}$$

Using the results of (6.13) and (6.14) in expression (6.12), individual completion time of device U_k in schedule $\mathcal{S}$ can be expressed as:

$$T_k(\mathcal{S}) = \frac{B.W_k(0)}{\tilde{R}_k(\mathcal{S})} + \mathcal{D}_k(\mathcal{S}). \tag{6.15}$$

6.9 Appendix B

Proof of Proposition 2

To prove the proposition, we first show that the term $2^{M_w - d_k + 1}\bar{T}_k(t)$ satisfies the property that the maximum individual completion time lower bound controls the overall completion time lower bound (i.e., $\bar{T}(t) = \max_{U_k \in \mathcal{M}_w}\{\bar{T}_k(t)\}$). We then show that the term $\left(\frac{r}{B}\right)$ provides a balance between the transmission rate and the number of targeted devices in each D2D transmission.

To begin with, we arrange all devices in $\mathcal{M}_w(t)$ in non-increasing order of lower bound on individual completion times. In other words, $\{U_1, U_2, ..., U_{M_w}\}$ such that $\bar{T}_1(t) \geq \bar{T}_2(t) \geq ... \geq \bar{T}_{M_w}(t)$. We, for now, assign all vertices induced by missing packets of device U_k with a weight $2^{M_w - d_k + 1}\bar{T}_k(t)$, where d_k is the order of device U_k in the organized group. Let us consider M_w vertices from M_w devices such that each vertex of them is corresponding to each device in $\mathcal{M}_w(t)$. We show that the vertex corresponding to the first device (having the maximum individual completion time lower bound) in the organized group has larger weight than the summation of weights of other $M_w - 1$ vertices.

$$\begin{aligned}
&2^{M_w - d_2 + 1}\bar{T}_2(t) + ... + 2^{M_w - d_{M_w} + 1}\bar{T}_{M_w}(t) \\
&\leq 2^{M_w - d_2 + 1}\bar{T}_2(t) + ... + 2^{M_w - d_{M_w} + 1}\bar{T}_2(t) \\
&= \bar{T}_2(t)[2^{M_w - d_2 + 1} + ... + 2^{M_w - d_{M_w} + 1}] \\
&\leq \bar{T}_1(t)[2^{M_w - d_2 + 1} + ... + 2^{M_w - d_{M_w} + 1}] \\
&< \bar{T}_1(t)2^{M_w - d_1 + 1}
\end{aligned} \tag{6.16}$$

Such vertices' weights results in the device with the maximum individual completion time lower bound is included and targeted in the maximum weight clique. With a similar analysis, we can show that a device has larger weight than the summation of all weights of the devices having smaller individual completion times lower bound than that device. In other words, the devices are included in the maximum weight clique following the descending order of individual completion times.

To exploit dynamic rate adaptation, we first note that the device with the maximum individual completion time lower bound can be served with different transmission rates representing different vertices in the graph. A greedy approach is minimizing the absolute transmission time $\frac{B}{r}$ required for the current transmission to that device, which is equivalent to targeting that device with the highest possible transmission rate $\frac{r}{B}$. However, including a vertex representing the highest transmission rate in a maximal clique may mean that no other vertex of other devices can be included. Indeed, with the selection of the highest rate vertex in the maximal clique, other vertices of other devices may not meet the same transmission rate requirement of condition C2 in Section 6.3.2 and will not be connected to that vertex via edges. Therefore, term $(\frac{r}{B})$ is associated with the weight of each vertex to achieve a balance between the transmission rate and the number of targeted devices. This results in the weight of vertex v^i_{rkl} becomes $\psi(v^i_{rkl}) = 2^{M_w-d_k+1}\bar{T}_k(t)(\frac{r}{B})$. In this case, a maximum weight clique includes a set of vertices having high lower bound on individual completion times and a high transmission rate. This result concludes the proof of Proposition 2.

Chapter 7

Conclusion, Observation and Future Directions

In this chapter, we conclude the thesis by discussing the contributions, observations and future research directions.

7.1 Thesis Conclusion

In this thesis, we were inspired by the need of reliable communication for time-critical applications in lossy wireless networks where packet reception delay plays a crucial role in end users' experience. Therefore, we focused on low-delay and simple IDNC, and studied practical problems regarding analysis, design and optimization of IDNC under various applications requirements and network configurations. In particular, we answered the following four questions:

1. How can we design an efficient IDNC scheme that maximizes the minimum number of decoded video layers over all devices before the deadline in a PMP network?

2. How can we design an efficient IDNC scheme that minimizes the mean video distortion before the deadline in a D2D network?

3. How can we design an efficient IDNC scheme that minimizes the in-order delivery delay in a heterogeneous network with coexistence of cellular and D2D networks?

4. How can we design an efficient IDNC scheme that exploits the physical layer rate adaptation to minimize the completion time in a D2D network?

To address the aforementioned four questions, here is a summary of our contributions:

7.1.1 Layered Video Aware Network Codes in PMP Networks

In this study, we developed a simple IDNC framework that efficiently broadcasts a scalable video to a set of devices over wireless PMP networks. In particular, we derived an upper bound on the probability that the completion time of all devices meet the deadline. Using this probability with other guidelines, we designed two prioritized algorithms, namely the EW-IDNC algorithm and the NOW-IDNC algorithm. These algorithms provide a high level of priority to the most important video layer, namely the base layer, before considering additional video layers, namely the enhancement layers, in coding decisions. Simulation results showed that our proposed IDNC algorithms improve the received video quality compared to the existing IDNC algorithms and achieve a comparable performance to the EW-RLNC algorithm.

7.1.2 Content Aware Network Codes in D2D Networks

In this study, we developed an efficient IDNC framework that distributes a real-time video sequence to a group of cooperative devices in a D2D network. In such scenarios, the coding conflicts occur to serve multiple devices with an immediately decodable packet and the transmission conflicts occur from simultaneous transmissions of multiple devices. Our proposed solution avoids coding and transmission conflicts by systematically constructing a graph, and meets the hard deadline for high importance video packets in lossy D2D networks by incorporating these factors into the vertices' weights. Simulation results over real video sequences showed that our proposed IDNC algorithms provide an appreciable video quality gain compared to the existing IDNC algorithms.

7.1.3 Packet Order Aware Network Codes in Heterogeneous Networks

In this study, we developed an efficient IDNC framework that utilizes dual wireless interfaces and network coding to provide reliable in-order packet delivery to the devices. To address the minimum in-order delivery delay problem, we constructed the dual interface IDNC graph and formulated the problem as a maximum weight independent set selection problem over the graph. The vertices of the graph are weighted by considering dual interfaces of individual devices, in-order delivery requirements of packets and lossy channel conditions. Simulation results showed that our proposed IDNC algorithm effectively reduces the delivery delay as compared to the existing network coding algorithms while maintaining a tolerable completion time degradation compared to the RLNC algorithm.

7.1.4 Completion Time Aware Network Codes in D2D Networks

In this study, we proposed an efficient IDNC framework that exploits the physical layer rate adaptation and cooperation among devices to minimize the completion time in a D2D network. In particular, we first introduced a rate aware IDNC graph that defines all feasible rate and coding decisions for all potential transmitting devices in one unified framework. Using this graph and the properties of the optimal schedule, we designed a completion time reduction heuristic that jointly selects a transmitting device, a transmission rate and an XOR packet combination. In fact, this heuristic strikes a balance between the transmission rate and the number of targeted devices with a new packet in each transmission. Simulation results demonstrated that our proposed IDNC algorithm reduces the completion time compared to the rate oblivious network coding algorithms.

7.2 Main Observations

In this thesis, we analyzed, designed and optimized IDNC algorithms to solve four practical problems under different application requirements and network configurations. The main observations of this thesis are summarized as follows:

- IDNC is attractive to delay-sensitive applications for its ability to immediately decode the received packets while achieving a comparable throughput performance to the RLNC for moderate number of devices in a network.

- IDNC needs to be designed based on the application requirements to exploit its full potential for a specific application.
- IDNC achieves the best performance when it bridges the gap between network coding, applications characteristics and network configurations in making decisions.
- A graph based IDNC algorithm systematically finds the solution of a given problem for the following properties. First, the graph theory provides a structure with exact representation of coding decisions, conflict free transmissions and efficient objective representation on the graph. Second, it contains theorems which facilitate the analysis of a problem. Third, it contains combinatorial techniques for the calculation of quantitative aspects of a problem. Finally, it offers numerous heuristic techniques for managing large graphs representing complex problems.
- Although finding the optimal solution of a given problem in IDNC-enabled systems is NP-hard, low-complexity heuristics are shown to perform well and improve the quality of service compared to various uncoded and coded schemes.
- IDNC is suitable for moderate sized social networks with a moderate number of friendly and trusted devices due to its resource consumption and additional delay in the feedback collection process.
- A D2D communication technology provides a better throughput since it allows multiple communication between devices that are not in the interference range of each other (spatial reuse). However, in D2D communication technology, the participating devices may possess only a small subset of all original packets while helping their neighbours to obtain the video. In such cases, a centralized technology (wherein all devices communicate with only the central base station) can be preferred over a D2D technology to expedite the video delivery process.

  In fact, the benefits of the cellular and D2D technologies depend on the initial Has and Wants sets of all devices, number of devices in the network, number of packets, channel erasure probabilities, number of transmitting devices and their geographic locations. Therefore, these parameters need to be carefully considered in the selection of a suitable technology.

- Given the decision maker can estimate the instantaneous channel quality, IDNC performs better when it intelligently adopts the transmission rate as compared to the rate oblivious IDNC solutions.

Overall, the observation is that when the system parameters are efficiently designed, analyzed and optimized, IDNC performs well in a wide range of applications and networks. We believe that the main observations of this thesis can be useful for optimizing system protocols and bridging the gap between the theory of IDNC and its practical applications. The implementation of the proposed IDNC solutions in real test bed are necessary to demonstrate their advantage in practical settings and facilitate the inclusion of IDNC into the next generation technologies.

7.3 Future Directions

The work in this thesis can be extended in three different directions namely, the physical layer rate awareness, the intermittent feedback scenarios and joint routing and network coding. In the next three subsections, we provide details of the potential extension in each direction.

7.3.1 Rate Aware Network Codes

In this thesis, particularly in Chapters 3, 4 and 5, we studied the problems of reducing the video distortion and in-order delivery delay with an abstraction of the physical layer conditions, e.g., fading, shadowing, into simple erasure channels. Moreover, we assumed that all transmissions from the base station or the transmitting devices have the same physical layer rate and therefore occupy a fixed duration of time. Such considerations simplified the analysis, design and optimization of IDNC-enabled systems at the expense of the approximation of the channel characteristics.

From physical layer perspective, users in wireless networks experience heterogeneous capacities of their physical channels. As a result, the users need to be served at various rates by the base station or transmitting devices in order to deliver the transmitted packets.

However, a packet transmission is successful at a user only if the adopted transmission rate is lower than or equal to the channel capacity of the device. In fact, such dynamic rate adaptation determines not only the ability of different users to successfully receive a transmission, but also the time duration required for delivering the packet. In other words, the rate adaptation affects the delivered video quality to the devices before the deadline and the in-order decoding of the packets at individual devices studied in Chapters 3, 4 and 5.

Similar to the study in Chapter 6, the upper-layer IDNC frameworks of Chapters 3, 4 and 5 can be modified to integrate the capacities of the physical channels and intelligently adopt the transmission rates. In fact, it is beneficial to identify the performance gain (delivery delay and video quality) of our proposed IDNC frameworks when the joint optimization over packet combinations, transmitting devices and transmission rates is performed. We believe that the further investigation of cross layer optimization is necessary to understand the full benefits of network coding in practical systems.

7.3.2 Intermittent Feedback Aware Network Codes

In this thesis, we studied the problems of reducing the video distortion and in-order delivery delay in different network configurations. These studies considered that the packet reception status of all devices is accurately updated at the central decision maker after each transmission using dedicated feedback channels and error protection codes. This idealistic consideration simplified the analysis and optimization of IDNC-enabled systems and provided an upper bound on the throughput, delay reduction, and video quality.

Even though a high level of protection for feedback packets is employed in wireless networks, unavoidable occasions of deep fading and shadowing can still expose them to loss events. Further, to reduce the energy consumption in sensors and robots, the generation of feedback packets after each transmission is not an attractive option. For a network with a large number of devices, the frequent generation of feedback packets consumes significant amount of the scarce radio resources. In some networks, due to the long distance between the sender and the devices, a large round trip time requires for the feedback arrival and, therefore, instant feedback collection after each transmission might not be practical. Therefore, in all these scenarios, the decision maker needs to select packet combinations for several transmissions at each feedback reception instant. In such intermittent feedback scenarios, the reception status of the previously transmitted but unacknowledged packets are uncertain (lost or received) at the decision maker. In other words, the decision maker needs to make blind decisions in each transmission.

The sole difference between the perfect and intermittent feedback systems is the uncertainty of packet reception status introduced by unheard feedback events. Therefore, if we can find a good estimate of the packet recaption status in the uncertain cases, we can apply the IDNC frameworks developed for perfect feedback in this thesis. Note that, in the stochastic partially observable domain, the best estimation of the packet reception status is the one representing the maximum likelihood state, i.e., the state that represents the highest probability in all the states. We believe that the findings of this thesis can be used to optimize algorithms and protocols, and bridge the gap between the network coding theory and the practical implementations.

7.3.3 Joint Routing and Network Coding

Another possible future research direction is to study joint routing and network coding algorithm similar to [107], and design an IDNC solution that enhances throughput, delay and video quality in D2D communication networks.

Bibliography

[1] T. S. Rappaport *et al.*, *Wireless communications: principles and practice.* Prentice Hall PTR New Jersey, 1996, vol. 2.

[2] M. van Der Schaar *et al.*, "Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms," *IEEE wireless Communications*, vol. 12, no. 4, pp. 50–58, 2005.

[3] O. Tickoo and B. Sikdar, "On the impact of ieee 802.11 mac on traffic characteristics," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 2, pp. 189–203, 2003.

[4] N. Aboutorab, P. Sadeghi, and S. Sorour, "Enabling a tradeoff between completion time and decoding delay in instantly decodable network coded systems," *IEEE Trans. Commun.*, vol. 62, no. 4, pp. 1296 –1309, 2014.

[5] S. Sorour and S. Valaee, "Minimum broadcast decoding delay for generalized instantly decodable network coding," in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2010, pp. 1–5.

[6] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 74–93, 2001.

[7] G. Joshi, Y. Kochman, and G. Wornell, "Throughput-smoothness tradeoffs in multicasting of an ordered packet stream," 2014. [Online]. Available: http://arxiv.org/abs/1405.3697

[8] A. Fu, P. Sadeghi, and M. Médard, "Dynamic rate adaptation for improved throughput and delay in wireless network coded broadcast," *IEEE Trans. Netw.*, pp. 1–14, 2013.

[9] S. Wang, C. Gong, X. Wang, and M. Liang, "Instantly decodable network coding schemes for in-order progressive retransmission," *IEEE Commun. Lett.*, vol. 17, no. 6, pp. 1069–1072, 2013.

[10] CISCO, "Cisco visual networking index: Global mobile data traffic forecast update, 20142019," 2014. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.pdf

[11] G. Van der Auwera, P. T. David, and M. Reisslein, "Traffic and quality characterization of single-layer video streams encoded with the h. 264/mpeg-4 advanced video coding standard and scalable video coding extension," *IEEE Transactions on Broadcasting*, vol. 54, no. 3, pp. 698–718, 2008.

[12] K. Rao, D. N. Kim, and J. J. Hwang, "Video coding standards," *The Netherlands: Springer*, 2014.

[13] P. Seeling, M. Reisslein, and B. Kulapala, "Network performance evaluation using frame size and quality traces of single-layer and two-layer video: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 6, no. 3, pp. 58–78, 2004.

[14] D. Nguyen, T. Nguyen, and X. Yang, "Multimedia wireless transmission with network coding," in *IEEE Packet Video*, 2007, pp. 326–335.

[15] 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.

[16] 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.

[17] A. Tassi, I. Chatzigeorgiou, and D. Vukobratovic, "Resource-allocation frameworks for network-coded layered multimedia multicast services," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 2, pp. 141–155, 2015.

[18] J.-R. Ohm, "Advances in scalable video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 42–56, 2005.

[19] I. Gopal and J. Jaffe, “Point-to-multipoint communication over broadcast links,” *IEEE Transactions on Communications*, vol. 32, no. 9, pp. 1034–1044, 1984.

[20] L. Hai, H. Wang, and J. Wang, “Instantly decodable network coding for multiple unicast retransmissions in wireless point-to-multipoint networks,” 2015.

[21] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, “Link-level measurements from an 802.11 b mesh network,” in *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, 2004, pp. 121–132.

[22] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan, “Measurement-based characterization of 802.11 in a hotspot setting,” in *Proceedings of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis*, 2005, pp. 5–10.

[23] L. Song, D. Niyato, Z. Han, and E. Hossain, *Wireless Device-to-Device Communications and Networks.* Cambridge University Press, 2015.

[24] P. Pahlevani, M. Hundeboll, M. V. Pedersen, D. Lucani, H. Charaf, F. Fitzek, H. Bagheri, and M. Katz, “Novel concepts for device-to-device communication using network coding,” *IEEE Commun. Mag.*, vol. 52, no. 4, pp. 32–39, 2014.

[25] Y. Keshtkarjahromi, H. Seferoglu, R. Ansari, and A. Khokhar, “Network coding for cooperative mobile devices with multiple interfaces,” in *IEEE Military Communications Conference (MILCOM’ 2015), Tampa, Florida, USA*, 2015, pp. 1–5.

[26] 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.

[27] 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.

[28] A. Damnjanovic, J. Montojo, Y. Wei, T. Ji, T. Luo, M. Vajapeyam, T. Yoo, O. Song, and D. Malladi, “A survey on 3gpp heterogeneous networks,” *IEEE Wireless Communications*, vol. 18, no. 3, pp. 10–21, 2011.

[29] D. Lopez-Perez, I. Guvenc, G. De la Roche, M. Kountouris, T. Q. Quek, and J. Zhang, "Enhanced intercell interference coordination challenges in heterogeneous networks," *IEEE Wireless Communications*, vol. 18, no. 3, pp. 22–30, 2011.

[30] Y. Moreno, R. Pastor-Satorras, and A. Vespignani, "Epidemic outbreaks in complex heterogeneous networks," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 26, no. 4, pp. 521–529, 2002.

[31] 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.

[32] H. Khamfroush, P. Pahlevani, 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.

[33] 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.

[34] N. J. Hernandez Marcano, 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.

[35] 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.

[36] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on information theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[37] S.-Y. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE transactions on information theory*, vol. 49, no. 2, pp. 371–381, 2003.

[38] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking (TON)*, vol. 11, no. 5, pp. 782–795, 2003.

[39] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413 –4430, 2006.

[40] 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.

[41] M. V. Pedersen, J. Heide, P. Vingelmann, and F. H. Fitzek, "Network coding over the 2 32-5 prime field," in *2013 IEEE International Conference on Communications (ICC)*, 2013, pp. 2922–2927.

[42] N. Thomos and P. Frossard, "Toward one symbol network coding vectors," *IEEE Commun. Lett.*, vol. 16, no. 11, pp. 1860–1863, 2012.

[43] 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.

[44] 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

[45] J. H. Sørensen, P. Popovski, and J. Østergaard, "Uep lt codes with intermediate feedback." *IEEE Commun. Lett.*, vol. 17, no. 8, pp. 1636–1639, 2013.

[46] D. E. Lucani, M. V. Pedersen, J. Heide, and F. H. Fitzek, "Fulcrum network codes: A code for fluid allocation of complexity," 2014. [Online]. Available: http://arxiv.org/abs/1404.6620

[47] A. Talari and N. Rahnavard, "On the intermediate symbol recovery rate of rateless codes," *IEEE Trans. Commun.*, vol. 60, no. 5, pp. 1237–1242, 2012.

[48] ——, "Lt-af codes: Lt codes with alternating feedback," in *IEEE International Symposium on Information Theory Proceedings (ISIT)*, 2013, pp. 2646–2650.

[49] A. Kamra, V. Misra, J. Feldman, and D. Rubenstein, "Growth codes: Maximizing sensor network data persistence," in *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, 2006, pp. 255–266.

[50] L. Hai, H. Wang, and J. Wang, "Instantly decodable network coding for multiple unicast retransmissions in wireless point-to-multipoint networks," *IEEE Trans. Vehi. Tech.*, 2015.

[51] S. Sorour, A. Douik, S. Valaee, T. Y. Al-Naffouri, and M.-S. Alouini, "Partially blind instantly decodable network codes for lossy feedback environment," *IEEE Trans. Wireless Commun.*, vol. 13, no. 9, pp. 4871–4883, 2014.

[52] 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.

[53] 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.

[54] 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.

[55] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: practical wireless network coding," in *ACM SIGCOMM Comput. Commun. Review*, vol. 36, no. 4, 2006, pp. 243–254.

[56] 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.

[57] 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.

[58] S. Y. El Rouayheb, M. A. R. Chaudhry, and A. Sprintson, "On the minimum number of transmissions in single-hop wireless coding networks," in *IEEE Information Theory Workshop (ITW)*, 2007, pp. 120–125.

[59] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, and A. Markopoulou, "Microcast: cooperative video streaming on smartphones," in *Proc. of ACM 10th international conference on Mobile systems, applications, and services (MobiSys' 2012), Lake District, United Kingdom*, 2012, pp. 57–70.

[60] M. S. Karim, P. Sadeghi, S. Sorour, and N. Aboutorab, "Instantly decodable network coding for real-time scalable video broadcast over wireless networks," *EURASIP Journal on Advances in Signal Processing*, pp. 1–24, 2016.

[61] G. Fodor, E. Dahlman, G. Mildh, S. Parkvall, N. Reider, G. Miklós, and Z. Turányi, "Design aspects of network assisted device-to-device communications," *IEEE Communications Magazine*, vol. 50, no. 3, pp. 170–177, 2012.

[62] M. S. Karim, S. Sorour, and P. Sadeghi, "Network coding for video distortion reduction in device-to-device communications," *IEEE Trans. Vehicular Tech.*, pp. 1–15, 2016.

[63] M. S. Karim, P. Sadeghi, N. Aboutorab, and S. Sorour, "In order packet delivery in instantly decodable network coded systems over wireless broadcast," in *IEEE International Symposium on Network Coding (NetCod' 2015), Sydney, Australia*, 2015, pp. 11–15.

[64] M. S. Karim, A. Douik, P. Sadeghi, and S. Sorour, "On using dual interfaces with network coding for delivery delay reduction," *IEEE Trans. Wireless Com. (Under Review).*

[65] 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.

[66] N. Aboutorab and P. Sadeghi, "Instantly decodable network coding for completion time or decoding delay reduction in cooperative data exchange systems," *IEEE Trans. Vehi. Tech.*, 2015.

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

[68] E. Magli, M. Wang, P. Frossard, and A. Markopoulou, "Network coding meets multimedia: A review," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1195–1212, 2013.

[69] H. Seferoglu, L. Keller, B. Cici, A. Le, and A. Markopoulou, "Cooperative video streaming on smartphones," in *49th Annual Allerton Conference on Communication, Control, and Computing*, 2011, pp. 220–227.

[70] 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.

[71] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," *Physical Communication*, vol. 1, no. 1, pp. 3–20, 2008.

[72] A. Eryilmaz, A. Ozdaglar, M. Médard, and E. Ahmed, "On the delay and throughput gains of coding in unreliable networks," *IEEE Transactions on Information Theory*, vol. 54, no. 12, pp. 5511–5524, 2008.

[73] D. E. Lucani, M. Médard, and M. Stojanovic, "Broadcasting in time-division duplexing: A random linear network coding approach," in *2009 Workshop on Network Coding, Theory, and Applications*, 2009, pp. 62–67.

[74] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, *Trading structure for randomness in wireless opportunistic routing*, 2007, vol. 37, no. 4.

[75] X. Wang and M. T. Orchard, "Design of superposition coded modulation for unequal error protection," in *IEEE International Conference on Communications, ICC*, vol. 2, 2001, pp. 412–416.

[76] K. P. Boyle, P. J. Lin, and C. Y. Christopher, "Rateless unequal error protection codes for the additive white gaussian noise channel," in *IEEE Military Communications Conference, MILCOM*, 2009, pp. 1–6.

[77] S. Sorour and S. Valaee, "Completion delay minimization for instantly decodable network codes," *IEEE/ACM Trans. Networking*, vol. 23, no. 5, pp. 1553–1567, 2015.

[78] P. Sadeghi, R. Shams, and D. Traskov, "An optimal adaptive network coding scheme for minimizing decoding delay in broadcast erasure channels," *EURASIP J. on Wireless Commun. and Netw.*, pp. 1–14, 2010.

[79] P. A. Chou and Y. Wu, "Network coding for the internet and wireless networks," *IEEE Signal Processing Magazine*, vol. 24, no. 5, p. 77, 2007.

[80] T. Ho and D. Lun, *Network coding: an introduction.* Cambridge University Press, 2008.

[81] J.-P. Wagner, J. Chakareski, and P. Frossard, "Streaming of scalable video from multiple servers using rateless codes," in *IEEE International Conference on Multimedia and Expo*, 2006, pp. 1501–1504.

[82] D. Vukobratović, V. Stanković, D. Sejdinović, L. Stanković, and Z. Xiong, "Scalable video multicast using expanding window fountain codes," *IEEE Trans. Multimedia*, vol. 11, no. 6, pp. 1094–1104, 2009.

[83] D. Sejdinovic, D. Vukobratovic, A. Doufexi, V. Senk, and R. Piechocki, "Expanding window fountain codes for unequal error protection," *IEEE Trans. Commun.*, vol. 57, no. 9, pp. 2510–2516, 2009.

[84] N. Thomos, J. Chakareski, and P. Frossard, "Prioritized distributed video delivery with randomized network coding," *IEEE Trans. Multimedia*, vol. 13, no. 4, pp. 776–787, 2011.

[85] 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.

[86] S. El Rouayheb, A. Sprintson, and P. Sadeghi, "On coding for cooperative data exchange," in *IEEE Information Theory Workshop (ITW)*, 2010, pp. 1–5.

[87] 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.

[88] 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 Proceedings (ISIT)*, 2013, pp. 3095–3099.

[89] M. S. Karim, N. Aboutorab, 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.

[90] 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.

[91] 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.

[92] A. Douik, M. S. Karim, P. Sadeghi, and S. Sorour, "Delivery time reduction for order-constrained applications using binary network codes," in *IEEE Wireless Communications and Networking Conference (WCNC' 2016), Doha, Qatar*, 2015.

[93] S. Sorour and S. Valaee, "Completion delay minimization for instantly decodable network codes," *IEEE/ACM Transactions on Networking*, vol. 23, no. 5, pp. 1553–1567, 2015.

[94] A. Douik, S. Sorour, M.-S. Alouini, and T. Y. Al-Naffouri, "Completion time reduction in instantly decodable network coding through decoding delay control," in *IEEE Global Communications Conference (GLOBECOM)*, 2014, pp. 5008–5013.

[95] A. Douik, S. Sorour, T. Y. Al-Naffouri, and M.-S. Alouini, "Rate aware instantly decodable network codes," in *IEEE Global Telecommunications Conference (GLOBECOM) Workshop*, 2015, pp. 1–6.

[96] X. Wang, C. Yuen, and Y. Xu, "Coding-based data broadcasting for time-critical applications with rate adaptation," *IEEE Trans. Vehicular Tech.*, vol. 63, no. 5, pp. 2429–2442, 2014.

[97] 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.

[98] M. Esmaeilzadeh and P. Sadeghi, "Optimizing completion delay in network coded systems over tdd erasure channels with memory," in *2012 International Symposium on Communications and Information Technologies (ISCIT)*, 2012, pp. 883–888.

[99] M. R. Garey, *Computers and intractability.* Freeman New York, 1979.

[100] "Test video sequences (retrieved june 2014)." [Online]. Available: ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/

[101] "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-reference-software.html

[102] D. B. West *et al.*, *Introduction to graph theory.* Prentice hall Upper Saddle River, 2001, vol. 2.

[103] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming.* John Wiley & Sons, 2009, vol. 414.

[104] 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.

[105] A. A. Al-Habob, S. Sorour, N. Aboutorab, and P. Sadeghi, "Conflict free network coding for distributed storage networks," in *Proc. of IEEE International Conference on Communications (ICC' 2015), London, United Kingdom*, 2015, pp. 5517–5522.

[106] 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.

[107] S. Sengupta, S. Rayanchu, and S. Banerjee, “An analysis of wireless network coding for unicast sessions: The case for coding-aware routing,” in *26th IEEE international conference on computer communications*, 2007, pp. 1028–1036.