Optimization of ARQ Allocation for Ultra-Reliable and Low-Latency Communication in Multi-Hop Networks

by

JAYA GOEL

Submitted

in fulfillment of the requirements of the Degree of Doctor of Philosophy

to the

BHARTI SCHOOL OF TELECOMMUNICATION TECHNOLOGY AND MANAGEMENT

INDIAN INSTITUTE OF TECHNOLOGY DELHI, INDIA

OCTOBER, 2022

Dedicated

To

My Beloved

Parents and Sister -

who always inspire me to dream big!

Bharti School of Telecommunication and Management
Indian Institute of Technology Delhi
Hauz Khas, New Delhi, India-110016

Certificate

This is to certify that the thesis entitled **Optimization of ARQ Allocation for Ultra-Reliable and Low-Latency Communication in Multi-Hop Networks**, submitted by **Jaya (aka Jaya Goel)** to the Bharti School of Telecommunication and Management, Indian Institute of Technology Delhi, for the award of the degree of **Doctor of Philosophy** has been carried out under my supervision. The work contained in this thesis have not been submitted either in part or in full to any other university or institute for the award of any degree.

Prof. Harshan Jagadeesh
Department of Electrical Engineering,
Indian Institute of Technology Delhi
Hauz Khas, New Delhi, India - 110016.

OCTOBER 2022

Acknowledgment

First and foremost thanks to the almighty God for giving me the wisdom, determination and patience in exploring the emerging world of wireless networks.

I would like to express my deep and sincere gratitude to my supervisor, Prof. Harshan Jagadeesh for giving me the opportunity to work under his guidance. His vision, sincerity, kindness and motivation have been a source of inspiration to me. He always gave me the freedom to think and explore new research ideas. He has not only taught me the right way to present my research but also, taught me the importance and values of a good research. His assistance during my research enabled me to complete the research successfully. It would never be possible without his guidance and support. I am truly grateful to his support for all these years.

I would like to thank the Ministry of Education, Government of India for providing financial support during my Ph.D. (2018-2022). I gratefully acknowledge the travel and work support from contingency fund from IIT Delhi, Indigenous 5G Test Bed project DoT (from Ministry of Communication), and CSIR, which have given me the wonderful opportunity to participate in international conferences to present my work across the globe.

I would like to extend my heartfelt thanks to all my labmates. Specially, I thank my senior Prateek Chakaraborty and Kamal Biswas for their incredible support at my workplace. Also, I would like to thank my hostel friends Anjali Gupta and Amita Giri with whom I had some long thoughtful discussion about life and its purpose, especially during the evening walk. Also, I would like to extend my thanks to my friends Namrata Tiwari, Shravya, Kunal, Muskan Ahuja, Vishal Srivastava, Mayank Gupta, Aadil Hussain and Diksha Garg. I feel extremely lucky and privileged to have so many wonderful friends.

Most importantly, I shall be forever grateful to my parents Sh. Prem Chand and Smt. Meena Rani for supporting me with their heart and soul to pursue my academic career and have patiently nurtured me throughout my life. I would like to thank my sister Neha Goel for her support and affection. Special thanks to my uncle Sh. Sitaram Goel, who inspired me to dream big. I also wish to extend a special thanks to my lifelong friend Nilaksh Bansal for his constant encouragement.

At last, I would like to pay my sincere gratitude to one and all who have directly or indirectly helped me throughout this journey.

Jaya

Abstract

Multi-hop networks comprising Line-of-Sight (LOS) dominated wireless channels have found attractive applications in wireless networks, such as in vehicular networks involving unmanned aerial vehicles. These applications are known to demand both high-reliability and low-latency features on communication links. While Automatic-Repeat-Requests (ARQs) or Hybrid-ARQs (HARQ) based decode-and-forward (DF) relaying strategies provide the high-reliability feature, they are known to incur delays owing to multiple re-transmissions of packets. To address the trade-off between the reliability and the low-latency features of ARQ-based or HARQ-based DF strategies, in this thesis , we impose a sum constraint on the total number of ARQs across the nodes as it captures an upper bound on end-to-end delay for packet transmissions. Subsequently, we solve the problem of optimally distributing the total number of ARQs across the nodes such that the end-to-end packet drop probability (PDP) is minimized under various non-cooperative, semi-cumulative, cluster-based, and fully-cumulative strategies. For each of the proposed strategies, we provide a set of necessary and sufficient conditions on the optimal ARQ distribution and use these conditions to synthesize low-complexity algorithms to obtain near-optimal solutions. In addition to the PDP metric, we also analyze the packet delay profiles to compare the latency of all the strategies and show that the packets reach the destination within the deadline with an overwhelming probability.

Contents

List of Figures

List of Tables

Part I

Prerequisites

Chapter 1

Introduction

1.1 Motivation

Multi-hop networks have found promising applications in wireless communication as they enhance the reliability of communication between wireless devices that are either outside of each other's coverage area or unable to establish a communication link due to signal-blockage difficulties [1–3]. While the problem of designing efficient and reliable protocols for multi-hop networks have been traditional topics of interest, recent advances in the field of vehicle-to-vehicle/infrastructure (V2X) communication, Industrial-IoT (IIoT) etc. have given rise to new requirements such as high-reliability and low-latency on the protocols [4, 5]. Example use-cases include autonomous V2X communication in which strict deadlines are imposed on the round-trip delay between the vehicles and the infrastructure, and automation tasks by deploying massive low-power IoT devices in factory settings and so on, after which either the messages become stale or the deadline violation may result in catastrophic consequences [6, 7]. Other well-known use-cases of high-reliability and low-latency in multi-hop networks are networks of Unmanned

Aerial Vehicles (UAVs) [8–10], in which power-limited UAVs either act as relays in coordinating the movement of autonomous vehicles, or act as airborne base-stations [11–14]. Thus, owing to increasing use-cases for achieving high-reliability and low-latency in multi-hop networks, the problem of designing efficient, reliable and importantly *low-latency* protocols, is of utmost importance in the context of next-generation networks.

To provide end-to-end reliability over such a multi-hop network, several protocols have been studied. Some well known protocols include amplify-and-forward (AF) protocols and decode-and-forward (DF) protocols [15], to name a few. While the AF protocols are known for facilitating low-complexity processing at the intermediate relays, they are also known to boost the effective noise at the destination. On the other hand, DF protocols are known for providing high end-to-end reliability provided the messages are encoded using a strong *code* of sufficiently large block-length to mitigate the effect of fading at each link. Although the DF protocols with codes of large block-lengths ensure high reliability, the messages reach the destination with a significant delay. Another variant of DF protocols that are known to provide high reliability are the automatic-repeat-request (ARQ) based DF protocols [16], wherein each relay is allowed to retransmit the packet when the next node in the chain is unable to decode the packet. In comparison with DF strategies based on fixed block-length codes, DF strategies based on ARQs introduce a variable delay on the packets by capitalizing on the stochastic nature of the fading channels, however, also contributing additional delay due to the use of ACK/NACK in the reverse channel from the receiver to the transmitter. In addition to ARQs, Hybrid-ARQ (HARQ) is a promising technique wherein, on every retransmission, the receiver node combines the latest packet along with its previous copies to decode the information. Among many variants of HARQ schemes in practical systems, a popular scheme is Chase-Combining HARQ (CC-HARQ), wherein each retransmission block is identical to the original code block, and all the received blocks are combined using the maximum-ratio combining technique at the receiver [17].

Towards handling the low-latency feature, the end-to-end delay in an N-hop network employing an ARQ or CC-HARQ based DF strategy depends on: (i) the packet size and the processing delay for encoding and decoding operations at each node, and (ii) the transmission delay for re-transmission of packets including the use of ACK/NACK at each link. Suppose that the processing time at each hop is τ_p seconds (which includes packet encoding and decoding time), the delay incurred for packet transmission at each hop is τ_d seconds (which includes the propagation delay and the time-frame of the packet), and the delay incurred because of NACK overhead is τ_{NACK} (which is the time taken for the transmitter to receive the NACK). Given the stochastic nature of the wireless channel at each link, the total number of packet re-transmissions before the packet reaches the destination is a random variable, denoted by n, and as a result, the end-to-end delay between the source and the destination is upper bounded by $n \times (\tau_p + \tau_d + \tau_{NACK})$ seconds. In particular, when $\tau_{NACK} << \tau_p + \tau_d$, the end-to-end delay can be approximated as $n \times (\tau_p + \tau_d)$ seconds. Thus, when the packet size and the decoding protocol at each node are established, and when the deadline on end-to-end delay (denoted by τ_{total}) is known, we may impose an upper bound on n, provided by $q_{sum} = \lfloor \frac{\tau_{total}}{\tau_p + \tau_d} \rfloor$. This implies that q_{sum} captures the maximum number of re-transmissions that can be tolerated over the multi-hop network in order to respect the deadline on the delay. In the event when τ_{NACK} is not negligible compared to $\tau_p + \tau_d$, we have the option of either using $q_{sum} = \lfloor \frac{\tau_{total}}{\tau_p + \tau_d} \rfloor$ or $q_{sum} = \lfloor \frac{\tau_{total}}{\tau_p + \tau_d + \tau_{NACK}} \rfloor$. In the former case, while a larger fraction of packets reach the destination due to higher q_{sum}, a non-zero fraction of the packets that reach the destination may arrive after the deadline, thereby violating the latency constraint. However, in the latter case, although a smaller fraction of the packets reach the destination due to lower q_{sum}, all of them arrive within the deadline. Thus, with either options for deciding q_{sum}, performance degrades as τ_{NACK} increases. Henceforth, we use $q_{sum} = \lfloor \frac{\tau_{total}}{\tau_p + \tau_d} \rfloor$ assuming that the frame structure and the resources for ACK/NACK communication support the condition $\tau_{NACK} << \tau_p + \tau_d$. Once q_{sum} captures the latency-constraint, the subsequent task is to handle the reliability metric by distributing the q_{sum} ARQs across the nodes. Thus, q_{sum}

number of re-transmissions captures the worst-case delay that can be tolerated over the multi-hop network. However, the way in which q_{sum} ARQs are distributed across the N links will dictate the end-to-end reliability of the DF strategy. This aspect is particularly important when the reliability of the underlying links are heterogeneous. Motivated by the above arguments, we present the problem statement and the research objectives of this thesis in the next section.

1.2 Problem Statement and Research Objectives

To ensure high reliability in multi-hop networks, we consider ARQ or CC-HARQ based DF strategies, wherein each intermediate relay is allotted an appropriate number of ARQs. This implies that a relay node will use its allotted number of ARQs until the next node is able to successfully decode the packet. However, if the next node is unable to decode the packet within the given number of attempts, then the packet is said to be dropped in the network. As the packet can be dropped at any link in the network, our reliability metric is the packet-drop-probability (PDP), that is the fraction of packets that do not reach the destination. Since the multi-hop networks of our interest are composed of wireless links characterized by arbitrary LOS components, the PDP of such networks is dependent on (i) the LOS components of the links, (ii) the number of ARQs allotted to the links, (iii) the underlying signal-to-noise-ratio (SNR) values, and importantly, (iv) the specific protocol used to implement the ARQ or CC-HARQ strategy. One way to guarantee high reliability in multi-hop networks is to optimize the ARQ allocation of each hop locally [18]. However, since we are interested in optimizing the PDP with upper bounds on end-to-end delay, we take the approach of jointly optimizing the ARQ distribution across the links by placing a sum constraint on the total number of ARQs. Towards that direction, we ask: "*When using an ARQ or CC-HARQ based DF strategy, how to allocate ARQs at each intermediate node such that its PDP is minimized under the sum constraint on the total number of ARQs in the network?*". Formally, the problem statement is presented in Problem 1.1.

Problem 1.1. *For an arbitrary N-hop network with a given LOS vector, and a given SNR, solve*

$$q_1^*, q_2^*, \ldots q_N^* = \arg \underset{q_1, q_2, \ldots q_N}{\mathit{min}} \; PDP$$

$$\textit{subject to } q_k \geq 1, q_k \in \mathbb{Z}_+, \; q_1 + q_2 + \ldots + q_N \; = \; q_{sum}.$$

Since there is a sum constraint on $\sum_{k=1}^{N} q_k = q_{sum}$, it is straightforward to note that the search space for determining the optimal distribution is bounded. With large values of q_{sum} and N, it is not feasible to implement exhaustive search to solve optimization problem. Identifying this limitation of exhaustive search, we obtain analytical results on the structure of the objective function and the underlying constraints so that the optimization problem can be solved with lower complexity than that of exhaustive search.

Furthermore, we also ask: "Given that the packets are designed to reach the destination satisfying a given reliability within τ_{total} time units, what are the delay profiles on the packets under various ARQ or CC-HARQ based DF strategies?" Towards handling the above questions, we present the contributions of this thesis in the next section.

1.3 Summary of the Contributions of this Thesis

The contributions of this thesis are as follows:

1. We propose a non-cooperative ARQ based DF strategy, wherein each node only knows the number of ARQs allotted to itself; it does not know the ARQs assigned to the other nodes in the network. With such constraints, we formulate an optimization problem of distributing an appropriate number of ARQs at each link such that the PDP is minimized for a given q_{sum}. We show that the optimization problem is non-linear with non-negative

integer constraints on the solution and therefore, it is extremely challenging to solve it. Towards finding the solution of the optimization problem, we derive sufficient and necessary conditions on the optimal ARQ distribution, and then propose a low-complexity algorithm. Through simulation results, we show that the proposed algorithm is amenable to implementation in practice, and also provides the optimal solution of the problem. We also present simulation results on the delay profiles of the packets by comparing the non-cooperative ARQ strategy with a standard baseline that does not require the use of ACK/NACK. The simulation results show that the ARQ strategy assists in reducing the average delay on the packets since the idea of asking for re-transmissions outweigh the delay-overhead introduced by NACK. We highlight that the non-cooperative scheme is referred to as the non-cumulative scheme interchangeably because by the virtue of non-cooperation, the residual ARQs at any hop do not get added to its next hop during the packet transmission.

2. After allocating an appropriate number of ARQs at each node, we identify that there may be unused ARQs at some relays owing to the stochastic nature of the wireless channels. Therefore, without violating the sum constraint, we propose a variety of cooperative ARQ strategies wherein the unused ARQs of one node can be used by other nodes with negligible increase in the communication-overhead. We refer to such strategies as the fully-cumulative ARQ schemes, cluster based cooperative ARQ schemes, semi-cumulative ARQ schemes, and cluster based semi-cumulative ARQ schemes. Subsequently, we characterize the PDP expressions of each of the above schemes, and then solve the problem of allocating an appropriate number of ARQs to the relay nodes so as to minimize the PDP under the sum constraint on the total number of ARQs. Through extensive analysis and simulation results, we show that the non-cooperative ARQ strategy and the fully-cumulative ARQ strategy respectively offer the worst and the best PDP performance among the schemes, whereas the

class of semi-cumulative ARQ schemes trade-off PDP with the communication-overhead between these two extreme classes of ARQ based DF strategies. Overall, in contrast to the idea of optimizing the reliability of each link in a hop-by-hop manner, our approach jointly optimizes the ARQ allocation across the links because of the constraints on end-to-end delay.

3. We propose CC-HARQ based strategies for multi-hop networks with slow-fading channels (denoted as CC-HARQ-SF), wherein the channels are assumed to be static over the allotted attempts at each link. Under this scenarios, we propose two types of strategies, namely: non-cumulative strategy and the fully-cumulative strategy. For the non-cumulative strategy, we derive a closed-form expression on PDP and formulate an optimization problem of minimizing the PDP for a given q_{sum}. We show that the optimization problem is non-tractable as it contains Marcum-Q function of first-order. Towards obtaining near-optimal ARQ distributions, we propose a tight approximation on the first-order Marcum-Q functions, and then present non-trivial theoretical results for synthesizing a low-complexity algorithm. Through extensive simulations, we show that our analysis on the near-optimal ARQ distribution gives us the desired results with affordable complexity. For the fully-cumulative strategy, we provide theoretical results on the optimal ARQ distribution in closed-form.

4. We propose CC-HARQ based strategies for fast-fading scenarios (denoted as CC-HARQ-FF), wherein the channels are statistically independent across allotted attempts at each link. Similar to CC-HARQ-SF strategies, under this scenarios, we propose two types of strategies, namely: non-cumulative strategy and the fully-cumulative strategy. For the non-cumulative strategy, we derive a closed-form expression on PDP and formulate an optimization problem of minimizing the PDP for a given q_{sum}. We show that the optimization problem is non-tractable as it contains Marcum-Q function of higher-order.

Towards obtaining near-optimal ARQ distributions, we propose a tight approximation on the higher-order Marcum-Q functions, and then present non-trivial theoretical results for synthesizing a low-complexity algorithm. Similar to CC-HARQ-SF strategies, using extensive simulations, we show that our analysis on the near-optimal ARQ distribution gives us the desired results with manageable complexity. For the fully-cumulative strategy, we present theoretical results on the optimal ARQ distribution in closed-form.

5. For each of our strategies, we have presented a detailed analysis on end-to-end delay by considering the following metrics: (i) average end-to-end delay, (ii) packet deadline violation (PDV), which is defined by the number of packets reaching the destination after the given deadline, and (iii) delay profile, which represents the percentage of packets reaching the destination at a certain time for a given deadline. By using the aforementioned delay-metrics, we have provided valuable insights on the merits and demerits of our strategies in achieving high-reliability with bounded constraints on end-to-end delay.

1.4 Literature Review

In [8], the authors studied different relaying strategies involving UAVs with single multi-hop links and multiple dual-hop links. They optimized the placement of UAVs for different channel models by considering both LOS and non-LOS channel models. In [10], the authors considered low-latency and high reliability in the control and non-payload communications (CNPC) links of UAVs, wherein they achieved ultra-reliability and low-latency in terms of the available range of the CNPC links between UAVs and a ground control station. In [9], the authors investigated the average achievable data rates of information (which requires ultra-reliable and low-latency communication) exchanged between a ground station and the UAVs by using Gaussian-Chebyshev quadrature (GCQ). In [17], authors have considered CC-HARQ schemes wherein they minimized the PDP under a total average transmit power constraint and minimized the average transmit power under a fixed PDP constraint. In [19], authors have studied the fundamental performance limits and linear dispersion code design for the MIMO-ARQ slow fading channel. There are other contributions in the literature [20]- [22], that use different forms of HARQ strategies. In [23], the authors investigated packet-drop-probability for energy harvesting nodes in a multi-hop network using ARQs and Hybrid-ARQs. In [24], the authors considered two classes of ARQs protocols, namely cooperative ARQ and non-cooperative ARQ protocols, to ensure reliable data collection in energy-harvesting wireless sensor networks. In the class of cooperative ARQ, neighboring nodes help each other and take advantage of the broadcasting nature of the uplink channel by using spatial diversity. In [25], the authors have analyzed the end-to-end performance of a cluster-based multi-hop network by using an ARQ cooperative diversity scheme. They considered multiple relays in each hop to obtain cooperative diversity. In [26], the authors addressed the ultra-reliability feature under the constraint of strict latency by using a cooperative ARQ scheme. They varied the reserved time for re-transmissions to optimize the latency feature. In a nutshell, none of the above contributions have addressed optimal allocation of ARQs with a sum constraint on the total number of ARQs over a multi-hop network.

1.5 Organization of the Thesis

This thesis is divided into five main parts where Part-I contains Chapter 1 and Chapter 2 that cover the state of art, motivation, problem statement, research objectives, literature review, contributions and prerequisites for this thesis. Part-II contains Chapter 3 and Chapter 4 that cover the ARQ based non-cooperative and the variants of cooperative strategy for achieving high reliability and low-latency communication over a multi-hop network. Furthermore, Part-III includes Chapter 5 and Chapter 6 that deal with the variants of semi-cumulative strategy for achieving high reliability and low-latency communication over a multi-hop network. After that Part-IV includes Chapter 7 and Chapter 8 that cover the HARQ based strategies for both slow-fading and fast-fading scenarios. In Part-V, we discuss about the conclusions of this thesis and some directions for the future work.

Different parts of the results presented in this thesis have already appeared in the proceedings of several conferences and journals. The list of publications based on this thesis can be found in **Appendix-I**.

Chapter 2

An Overview of Multi-Hop Network under Consideration

2.1 Introduction

In this chapter, we introduce the broad picture of our system model, which is used throughout this thesis. Consider an N-hop network, as shown in Fig. 2.1, which includes a source node, a destination node, and a set of $N-1$ relays. We assume that the information bits from the source node are aggregated in the form of packets, and these packets are communicated to the destination in a multi-hop fashion using the $N-1$ intermediate relays. In other words, the multi-hop network consists of N wireless links, wherein the first link corresponds to the channel between the source node and relay R_1, the second link corresponds to the channel between R_1 and R_2, and similarly, the N-th link corresponds to the channel between relay R_{N-1} and the destination node. We assume that the channel between any two successive nodes is characterized by Rician fading with a quasi-static time-interval of L channel uses. In particular, the complex

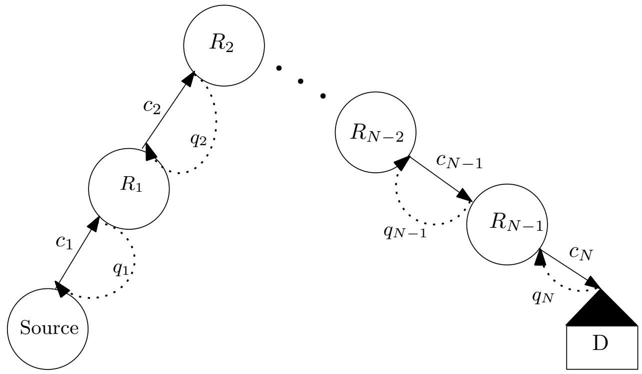

Figure 2.1: Depiction of LOS dominated N-hop network, where $0 \leq c_k \leq 1$ represents the LOS component of the k-th link, and $q_k \in \mathbb{Z}_+$ denotes the number of ARQs allotted to the transmitter of the k-th link, for $1 \leq k \leq N$.

baseband channel of the k-th link, for $1 \leq k \leq N$, is modeled by

$$h_k = \sqrt{\frac{c_k}{2}}(1 + \iota) + \sqrt{\frac{(1 - c_k)}{2}} g_k,$$

where $0 \leq c_k \leq 1$ captures the LOS component, and $1 - c_k$ is the Non-LOS (NLOS) component such that g_k is distributed as $\mathcal{CN}(0, 1)$. In this signal model, the LOS component c_k is a deterministic quantity, thereby ensuring the equality $\mathbb{E}[|h_k|^2] = 1$ irrespective of the value of c_k. As a special case, $c_k = 0$ and $c_k = 1$ capture the well-known Rayleigh and Gaussian channels, respectively, whereas the intermediate values capture different degrees of Rician fading channels. Assuming that the LOS components of the N links can be potentially different, henceforth, throughout this thesis, we use the vector $\mathbf{c} = [c_1, c_2, \ldots, c_N]$ to highlight the LOS components of the N-hop network.

Let $\mathcal{C} \subset \mathbb{C}^L$ denote the channel code employed at the source node of rate R bits per channel use, i.e., $R = \frac{1}{L}\log(|\mathcal{C}|)$. Let $\mathbf{x} \in \mathcal{C}$ denote a codeword (henceforth referred to as packets) transmitted by the source node such that $\frac{1}{L}\mathbb{E}[|\mathbf{x}|^2] = 1$, where the expectation is taken over $\mathcal{C}$. When $\mathbf{x}$ is transmitted over the k-th link, for $1 \leq k \leq N$, the corresponding received symbols after L channel uses is given by $\mathbf{y}_k = h_k\mathbf{x} + \mathbf{w}_k \in \mathbb{C}^L$, where $\mathbf{w}_k$ is the additive white Gaussian noise (AWGN) at the receiver of the k-th link, distributed as $\mathcal{CN}(0, \sigma^2\mathbf{I}_L)$. We assume that the receiver of the k-th link has perfect knowledge of h_k, whereas the transmitter of the k-th link does not have the knowledge of h_k. Since the channel realization h_k is random, and the realization remains constant for L channel uses, the instantaneous mutual information of the k-th link may not support the transmission rate. Therefore, the corresponding relay node will be unable to correctly decode the packet when the mutual information offered by the channel is less than R. The probability of such an outage event is given by [1]

$$P_k = \text{Pr}\Big(R > \log_2(1 + |h_k|^2\gamma)\Big) = \text{F}\left(\frac{2^R - 1}{\gamma}\right), \tag{2.1}$$

where $\gamma = \frac{1}{\sigma^2}$ is the average signal-to-noise-ratio (SNR) of the k-th link, $\text{F}(x)$ is the cumulative distribution function of $|h_k|^2$, defined as

$$\text{F}\left(\frac{2^R - 1}{\gamma}\right) = 1 - Q_1\left(\sqrt{\frac{2c_k}{(1-c_k)}}, \sqrt{\frac{2(2^R - 1)}{\gamma(1-c_k)}}\right),$$

such that $Q_1(\cdot,\cdot)$ is the first-order Marcum-Q function [27]. Henceforth, for the given system model, we propose variants of ARQ and CC-HARQ based DF strategies in the next section to achieve both ultra-reliability and low-latency communication. In the next chapter, we discuss the ARQ based non-cooperative multi-hop model and we address the problem of obtaining optimal ARQ distribution for a given sum constraint q_{sum}.

[1] We note that P_k in (2.1) is computed by comparing R with $\log_2(1 + |h_k|^2\Theta)$. However, this comparison may not be accurate in our model since $\log_2(1 + |h_k|^2\Theta)$ is achievable for asymptotic block-lengths. In order to obtain non-asymptotic expressions on P_k, we must replace $\log_2(1 + |h_k|^2\Theta)$ by the non-asymptotic rate given in [28]. Upon incorporating this change, we observe that the non-asymptotic expression for P_k [28, Theorem 2] is a decreasing function of the LOS component c_k, and this is the same behaviour of (2.1) with respect to c_k.

Part II

ARQ Schemes for Achieving URLLC over Multi-Hop Networks

Chapter 3

Non-Cooperative ARQ Schemes in Multi-Hop Networks

3.1 Introduction

In this chapter, we address the problem of obtaining the optimal ARQ distribution for a non-cooperative multi-hop network [1]. The main contributions of this chapter are as follows:

1) We propose a new problem on distributing the number of re-transmissions across multiple relay nodes in a LOS dominated multi-hop network so as to support low-latency and ultra-reliability features on the underlying packets (See Section 3.2). In particular, given a multi-hop network with potentially distinct LOS components of the links, we formulate an optimization problem of minimizing the PDP under the constraint that the sum of the ARQs across all the links in the network is bounded. First, we show that this optimization problem involves a non-linear objective function with non-negative integer-constraints on the solution. Because of the sum

[1] Part of the results presented in this chapter are available in publications [29, 30]

constraint on the total number of ARQs, we show that the size of the search space is bounded. However, we also show that computing the optimal distribution of ARQs through exhaustive search is not feasible to implement in practice.

2) At high signal-to-noise-ratio (SNR) values, we observe that the set of necessary and sufficient conditions simplifies to a set of linear equations relating the number of re-transmissions allotted to the N links. Using this special case, we convert the problem of computing the optimal distribution of re-transmissions to an equivalent problem of solving a system of linear equations in $\mathbb{R}^N$, and to another problem of searching a distribution of re-transmissions in the integer search-space that are nearest to the real solution. Through this approach, we show that the search space for finding the optimal distribution of re-transmission can be significantly reduced when compared to the exhaustive search method. Although our approach of formulating an equivalent problem is motivated by high SNR approximation of the necessary and sufficient conditions, we show through simulation results that our algorithm continues to generate a small list size at low and moderate SNR values (See Section 3.4). Furthermore, we highlight that our algorithm scales well with the number of hops, and importantly provides substantial reduction in complexity when the bound on the total number of ARQs allotted across the relay nodes increases. Towards solving this problem with low complexity methods, we prove a set of necessary and sufficient conditions on the optimal solution of the optimization problem (See Section 3.3).

3.2 Signal Model

Consider an N-hop network, as shown in Fig. 3.1, which includes a source node, a destination node, and a set of $N-1$ relays. We assume that the channel between any two successive nodes is characterized by Rician fading with a quasi-static time-interval of L channel uses. Along the

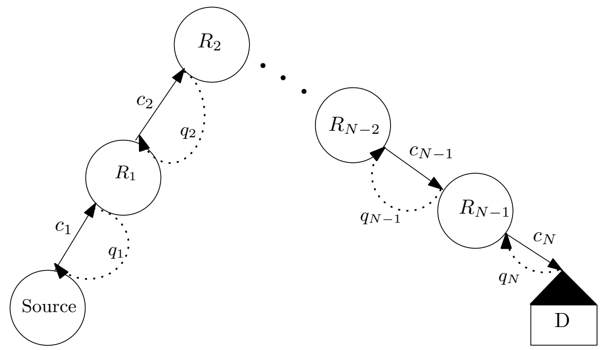

Figure 3.1: Depiction of LOS dominated N-hop network, where $0 \leq c_k \leq 1$ represents the LOS component of the k-th link, and $q_k \in \mathbb{Z}_+$ denotes the number of ARQs allotted to the transmitter of the k-th link, for $1 \leq k \leq N$.

similar lines of Section 2.1 (from Chapter 2), the probability of such an outage event is given by

$$P_k = \mathbf{Pr}\Big(R > \log_2(1 + |h_k|^2\gamma)\Big) = \mathrm{F}\left(\frac{2^R - 1}{\gamma}\right), \tag{3.1}$$

where $\gamma = \frac{1}{\sigma^2}$ is the average signal-to-noise-ratio (SNR) of the k-th link. In this N-hop network model, we assume that communication between any two successive nodes follows the ARQ protocol, i.e., a transmitter node gets an ACK or NACK from the next node in the chain indicating the success or failure of the transmission, respectively. Upon receiving a NACK, the transmitter re-transmits the packet. Let q_k be the maximum number of attempts given to the transmitter of the k-th link. Consolidating the number of attempts given to each link, the ARQ distribution of the multi-hop network is represented by the vector $\mathbf{q} = [q_1, q_2, \ldots, q_N]$. Since we are addressing low-latency applications, we impose the constraint $\sum_{i=1}^{N} q_i = q_{sum}$, for some $q_{sum} \in \mathbb{Z}_+$, which captures an upper bound on the end-to-end delay on the packets.

Note that if a node fails to deliver the packet to the next node within the allotted number of attempts, then the packet is said to be dropped in the network. As the packet can be dropped in any of the links, the packet-drop-probability (PDP) of the N-hop network is given by

$$p_d = \sum_{k=1}^{N} P_k^{q_k} \left(\prod_{i=1}^{k-1} (1 - P_i^{q_i}) \right). \tag{3.2}$$

When calculating the above expression, we have assumed that the channel realization h_k takes independent realizations across the number of attempts, and the number of ARQs assigned to a transmitter is not known to the other nodes in the network [2] .

3.2.1 Formulation of Optimization Problem

For a multi-hop network with LOS vector $\mathbf{c}$ and SNR $\gamma = \frac{1}{\sigma^2}$, we are interested in computing the ARQ distribution $\mathbf{q}$ which minimizes the PDP expression in (3.2) under the constraint that $\sum_{k=1}^{N} q_k = q_{sum}$, for a given $q_{sum} \in \mathbb{Z}_+$. We present this problem formulation as Problem 3.1, as shown below. Henceforth, throughout this chapter, we refer to the solution of Problem 3.1 as the optimal ARQ distribution. We highlight that Problem 3.1 is a non-linear optimization problem with non-negative integer constraints on the solution. Since there is a sum constraint on $\sum_{k=1}^{N} q_k = q_{sum}$, it is straightforward to note that the search space for determining the optimal distribution is bounded. In particular, it can be shown that the number of candidates in the search space is $\binom{q_{sum}-1}{N-1}$. Therefore, with large values of q_{sum} and N, it is not feasible to implement

[2] From the expression for p_d in (3.2), we note that P_k in (3.1) is computed by comparing R with $\log_2(1+|h_k|^2\Theta)$. However, this comparison may not be accurate in our model since $\log_2(1+|h_k|^2\Theta)$ is achievable for asymptotic block-lengths. In order to obtain non-asymptotic expressions on P_k, we must replace $\log_2(1+|h_k|^2\Theta)$ by the non-asymptotic rate given in [28]. Upon incorporating this change, we observe that the non-asymptotic expression for P_k [28, Theorem 2] is a decreasing function of the LOS component c_k, and this is the same behaviour of (3.1) with respect to c_k. Furthermore, since the ARQ distribution problem is solved by minimizing the PDP for a given $\mathbf{c}$, R and SNR, the terms $P_1, P_2, \ldots, P_N$ are fed to the optimization problem as fixed inputs. As a result, any algorithm presented to solve Problem 3.1 will hold good when using the non-asymptotic expression for P_k in place of (3.1). Due to these reasons, we continue to use (3.1) when describing our solutions to the optimization problems.

exhaustive search to solve Problem 3.1. Identifying this limitation of exhaustive search, we obtain analytical results on the structure of the objective function and the underlying constraints so that Problem 3.1 can be solved with lower complexity than that of exhaustive search.

Problem 3.1. *For a multi-hop network with a given LOS vector* $\mathbf{c}$*, and a given SNR* $\gamma = \frac{1}{\sigma^2}$*, solve*

$$\begin{aligned} q_1^*, q_2^*, \ldots q_N^* &= \arg \min_{q_1, q_2, \ldots q_N} p_d \\ \textit{subject to } q_k &\geq 1, \\ q_k &\in \mathbb{Z}_+, \\ q_1 + q_2 + \ldots + q_N &= q_{sum}. \end{aligned}$$

3.3 Sufficient and Necessary Conditions on the Optimal ARQ Distribution

In this section, we present some insights on the expression of PDP, which in turn will be useful in solving Problem 3.1.

Definition 3.1. *Let* $\pi : \mathbb{R}^N \rightarrow \mathbb{R}^N$ *denote a permutation operator on an* N*-dimensional Euclidean space. For an* N*-hop network with LOS vector* $\mathbf{c}$*, we define an equivalent multi-hop network with LOS vector* $\bar{\mathbf{c}} = \pi(\mathbf{c})$*, wherein the wireless channel of the* k*-th link experiences the LOS component* $\bar{\mathbf{c}}(k)$*, where* $\bar{\mathbf{c}}(k)$ *denotes the* k*-th component of the vector* $\bar{\mathbf{c}}$*.*

Theorem 3.1. *The PDP of an* N*-hop network with LOS vector* $\mathbf{c}$ *and ARQ distribution* $\mathbf{q}$ *is equal to the PDP of an* N*-hop network with LOS vector* $\pi(\mathbf{c})$ *and ARQ distribution* $\pi(\mathbf{q})$*, where*

π is any permutation operator on $\mathbb{R}^N$.

Proof. We will prove this theorem using the method of induction. For $N = 2$, the PDP can be written as

$$p_d = P_1^{q_1} + P_2^{q_2}(1 - P_1^{q_1}) = P_1^{q_1} + P_2^{q_2} - P_1^{q_1}P_2^{q_2}. \tag{3.3}$$

By swapping c_1 and c_2, and also q_1 and q_2, we obtain

$$p_d' = P_2^{q_2} + P_1^{q_1}(1 - P_2^{q_2}) = P_2^{q_2} + P_1^{q_1} - P_2^{q_2}P_1^{q_1}. \tag{3.4}$$

From (3.3) and (3.4), it is clear that $p_d = p_d'$. Thus, the statement of the theorem is proved for $N = 2$.

Assume that for $N = k$, swapping any two links will not change the PDP. For $N = k + 1$, we want to prove that swapping any two links will not change the PDP. The PDP expression in such a case can be written as

$$p_d = P_1^{q_1} + P_2^{q_2}(1 - P_1^{q_1}) + \ldots + P_k^{q_k}\Big[\prod_{j=1}^{k-1}(1 - P_j^{q_j})\Big] + \; P_{k+1}^{q_{k+1}}\Big[\prod_{j=1}^{k}(1 - P_j^{q_j})\Big].$$

By taking $(1 - P_1^{q_1})$ common from the second term onward, we can rewrite the above expression as

$$p_d = P_1^{q_1} + (1 - P_1^{q_1})\Bigg[P_2^{q_2} + P_3^{q_3}(1 - P_2^{q_2}) + \ldots + \; P_{k+1}^{q_{k+1}}(1 - P_2^{q_2})\ldots(1 - P_k^{q_k})\Bigg]. \tag{3.5}$$

It can be seen from (3.5) that the k terms in the square bracket constitute the PDP of a k-hop network with the LOS vector $[c_2, c_3, \ldots, c_{k+1}]$ and the ARQ distribution $[q_2, q_3, \ldots, q_{k+1}]$. Therefore, by hypothesis of induction, swapping any two links within the set $\{c_2, c_3, \ldots, c_{k+1}\}$ will not change the PDP. It now remains to show that swapping the link with LOS component c_1 with any of the links in the set $\{c_2, c_3, \ldots, c_{k+1}\}$ will not change the PDP. For illustrative purposes, we will show that swapping c_1 with c_{k+1} does not change the PDP, although this approach can be applied to swap c_1 with any of the links in the set $\{c_2, c_3, \ldots, c_{k+1}\}$. Towards swapping c_1 with c_{k+1}, let us first swap c_2 and c_{k+1} using (3.5), to obtain

$$p_d = P_1^{q_1} + (1 - P_1^{q_1})\left[P_{k+1}^{q_{k+1}} + P_3^{q_3}\left(1 - P_{k+1}^{q_{k+1}}\right) + \ldots + \; P_2^{q_2}(1 - P_{k+1}^{q_{k+1}})\left(\prod_{j=3}^{k}(1 - P_j^{q_j})\right)\right].$$

Note that this manipulation does not change the PDP due to the induction step. We further rewrite p_d as

$$p_d = P_1^{q_1} + P_{k+1}^{q_{k+1}}(1 - P_1^{q_1}) + (1 - P_1^{q_1})\left[P_3^{q_3}(1 - P_{k+1}^{q_{k+1}}) + \ldots + P_2^{q_2}(1 - P_{k+1}^{q_{k+1}}) \left(\prod_{j=3}^{k}(1 - P_j^{q_j})\right)\right].$$

By swapping the links with LOS components c_1 and c_{k+1} in the above expression, we obtain

$$p_d' = \; P_{k+1}^{q_{k+1}} + P_1^{q_1}(1 - P_{k+1}^{q_{k+1}}) + (1 - P_{k+1}^{q_{k+1}})\left[P_3^{q_3}(1 - P_1^{q_1}) + \ldots + P_2^{q_2}(1 - P_1^{q_1}) \left(\prod_{j=3}^{k}(1 - P_j^{q_j})\right)\right].$$

It is straightforward to observe that $p_d = p_d'$. Therefore, for $N = k + 1$, we have shown that swapping any two links will not change the PDP. Finally, since it is well known that a permutation π can be realized through a sequence of swaps, it follows that the PDP of an N-hop network with LOS vector $\mathbf{c}$ and ARQ distribution $\mathbf{q}$ is same as that of the PDP of the network with LOS vector $\pi(\mathbf{c})$ and ARQ distribution $\pi(\mathbf{q})$. $\square$

The following theorem shows that a link with higher LOS component must not be given more ARQs than the link with lower LOS component.

Theorem 3.2. *With the LOS vector* $\mathbf{c}$*, let the SNR be such that* $P_k < \frac{1}{2}, \forall\, k$*. Then the optimal ARQ distribution* $\mathbf{q}$ *satisfies the property that whenever* $c_i \geq c_j$*, we have* $q_i \leq q_j \;\forall\, i, j$*.*

Proof. To highlight c_i and c_j, we rewrite $\mathbf{c}$ as $[c_1, c_2, \ldots, c_i, \ldots, c_j, \ldots, c_{N-1}, c_N]$ such that $j > i$. Suppose that $c_j > c_i$, and q_i and q_j respectively denote the number of ARQs given to the i-th link and the j-th link. Furthermore, let us assume that $q_i = q_j = q$. Suppose that we have an additional ARQ with us, and the problem is whether to allot that additional ARQ to the i-th link or the j-link such that the PDP is minimized. Towards solving this problem, let us consider an equivalent multi-hop network with LOS vector $\mathbf{c}' = [c_1, c_2, \ldots, c_{N-1}, \ldots, c_N, \ldots, c_i, c_j]$, wherein $\mathbf{c}'$ is obtained from $\mathbf{c}$ by swapping c_i with c_{N-1} and c_j with c_N. From Theorem 3.1, we know that the PDP of the multi-hop networks with the LOS vectors $\mathbf{c}$ and $\mathbf{c}'$ are identical. Furthermore, the PDP of the N-hop network with LOS vector $\mathbf{c}'$, is written as

$$p_d = P_1^{q_1} + P_2^{q_2}(1 - P_1^{q_1}) + \ldots + \left(P_i^{q_i} + P_j^{q_j}(1 - P_i^{q_i})\right) \prod_{k \in [N] \setminus \{i,j\}} (1 - P_k^{q_k}).$$

Note that $P_i^{q_i}$ and $P_j^{q_j}$ appear only in the last term of the above expression. Since the question of allocating the additional ARQ is dependent only on the expression $P_i^{q_i} + P_j^{q_j}(1 - P_i^{q_i})$, we henceforth do not use the entire expression for PDP. Additionally, since $q_i = q_j = q$, we obtain one of the following expressions when allocating the additional ARQ,

$$\begin{aligned} A &= P_i^{q+1} + P_j^{q}(1 - P_i^{q+1}), \\ B &= P_i^{q} + P_j^{q+1}(1 - P_i^{q}). \end{aligned}$$

Since $c_i < c_j$, we know that $P_i > P_j$. To prove the statement of the theorem, we have to show that $A < B$. As $0 < P_i, P_j < 1$, it is clear that $P_i^{q+1} < P_i^q$ and $P_j^{q+1} < P_j^q$. Furthermore, $A - B$ can be calculated as

$$A - B = P_i^q(P_i - 1) + P_j^q(1 - P_j) + P_j^q P_i^q (P_j - P_i). \tag{3.6}$$

Note that the first and the third term in above equation are negative, whereas the second term is positive. Therefore, if the absolute value of the first term is greater than the absolute value of the second term, then $A - B < 0$. In the rest of the proof, we show that $P_i^q(1 - P_i) > P_j^q(1 - P_j)$, for any $q \geq 1$. With $q = 1$, the above equation can be written as $P_i(1 - P_i) > P_j(1 - P_j)$. It is straightforward to prove that the above inequality holds if $P_i + P_j < 1$. Thus, the statement of the theorem is proved for $q = 1$. Now, since $P_i > P_j$, note that $\frac{P_i^q}{P_j^q}$ increases as q increases, and therefore, for any $q \in \mathbb{Z}_+$, we have the inequality

$$\frac{P_i^q}{P_j^q}\frac{(1 - P_i)}{(1 - P_j)} > \frac{P_i}{P_j}\frac{(1 - P_i)}{(1 - P_j)} > 1. \tag{3.7}$$

This implies that the magnitude of the first term of (3.6) is greater than the magnitude of the second term, and therefore, we have $A - B < 0$. This completes the proof. □

In the following definition, we formally introduce the search space for the optimal ARQ distribution as highlighted in Problem 3.1.

Definition 3.2. *The search space for the optimal ARQ distribution is denoted by* $\mathbb{S} = \{\mathbf{q} \in \mathbb{Z}_+^N \mid \sum_{i=1}^N q_i = q_{sum} \;\&\; q_i \geq 1 \;\forall\, i\}$.

For a given point $\mathbf{q} \in \mathbb{S}$, we define its neighbors in the following definition.

Definition 3.3. *For a given* $\mathbf{q} \in \mathbb{S}$*, the set of its neighbors is defined as* $\mathcal{D}(\mathbf{q}) = \{\bar{\mathbf{q}} \in \mathbb{S} \mid d(\mathbf{q}, \tilde{\mathbf{q}}) = 2\}$*, where* $d(\mathbf{q}, \bar{\mathbf{q}})$ *denotes the number of disagreements between* $\mathbf{q}$ *and* $\bar{\mathbf{q}}$.

Note that for a given $\mathbf{q} \in \mathbb{S}$, we have $|\mathcal{D}(\mathbf{q})| \leq 2\binom{N}{2}$. In the next definition, we formally introduce a local minima of the space $\mathbb{S}$ by evaluating the PDP of the multi-hop network over the points in $\mathbb{S}$.

Definition 3.4. *An ARQ distribution* $\mathbf{q}^* \in \mathbb{S}$ *is said to be a local minima of* $\mathbb{S}$*, if it satisfies the condition* $p_d(\mathbf{q}^*) \leq p_d(\mathbf{q})$*, for every* $\mathbf{q} \in \mathcal{D}(\mathbf{q}^*)$*, where* $p_d(\mathbf{q}^*)$ *and* $p_d(\mathbf{q})$ *represent the PDP evaluated at the distributions* $\mathbf{q}^*$ *and* $\mathbf{q}$*, respectively.*

Using the above definition, we derive a set of necessary and sufficient conditions on the local minima in the following theorem.

Theorem 3.3. *For an N-hop network with LOS vector $\mathbf{c}$, the ARQ distribution $\mathbf{q}^* = [q_1^*, q_2^*, \ldots, q_N^*]$ is a local minima if and only if q_i^* and q_j^* for $i \neq j$ satisfy the following bounds*

$$\frac{q_i^*}{(q_j^*-1)} \geq \frac{1}{(q_j^*-1)\log P_i} \log\left(\frac{C_{q_i^*-1}}{C_{q_j^*-2}}\right) + \frac{\log P_j}{\log P_i}, \tag{3.8}$$

$$\frac{q_i^*-1}{q_j^*} \leq \frac{1}{q_j^* \log P_i} \log\left(\frac{D_{q_i^*-1}}{D_{q_j^*}}\right) + \frac{\log P_j}{\log P_i}, \tag{3.9}$$

where $C_{q_i^*-1} = \sum_{r=0}^{q_1^*-1} P_i^r$, $C_{q_j^*-2} = \sum_{k=0}^{q_j^*-2} P_j^k$, $D_{q_2^*} = \sum_{k=0}^{q_j^*} P_j^k$ *and* $D_{q_i^*-1} = \sum_{r=0}^{q_i^*-1} P_i^r$.

Proof. From Definition 3.3, it is clear that a neighbor of $\mathbf{q}^*$ in the search space $\mathbb{S}$ differs in two positions with respect to $\mathbf{q}^*$. Let us consider two neighbors of $\mathbf{q}^*$ that differ in the i-th and j-th index, where $i \neq j$. Such neighbors are of the form $\tilde{\mathbf{q}}_+ = [q_1^*, q_2^*, \ldots, q_i^* + 1, \ldots, q_j^* - 1, \ldots, q_N^*]$ and $\tilde{\mathbf{q}}_- = [q_1^*, q_2^*, \ldots, q_i^* - 1, \ldots, q_j^* + 1, \ldots, q_N^*]$ provided $q_i^* - 1 \geq 1$ and $q_j^* - 1 \geq 1$. From Theorem 3.1, instead of considering the multi-hop network with LOS vector $\mathbf{c} = [c_1, c_2, \ldots, c_i, \ldots, c_j, \ldots, c_{N-1}, c_N]$, we consider a permuted version of it with the LOS vector $\mathbf{c} = [c_1, c_2, \ldots, c_{N-1}, \ldots, c_N, \ldots, c_i, c_j]$, wherein the i-th link is swapped with $(N-1)$-th link, and the j-th link is swapped with N-th link. Correspondingly, the local minima and its two neighbors are respectively of the form $\mathbf{q}^* = [q_1^*, q_2^*, \ldots, q_{N-1}^*, \ldots, q_N^*, \ldots, q_i^*, q_j^*]$, $\tilde{\mathbf{q}}_+ = [q_1^*, q_2^*, \ldots, q_{N-1}^*, \ldots, q_N^*, \ldots, q_i^*+1, q_j^*-1]$ and $\tilde{\mathbf{q}}_- = [q_1^*, q_2^*, \ldots, q_{N-1}^*, \ldots, q_N^*, \ldots, q_i^*-1, q_j^*+1]$. From the definition of local minima, we have the inequalities

$$p_d(\mathbf{q}^*) \leq p_d(\tilde{\mathbf{q}}_+), \text{ and } p_d(\mathbf{q}^*) \leq p_d(\tilde{\mathbf{q}}_-), \tag{3.10}$$

where $p_d(\mathbf{q}^*)$, $p_d(\tilde{\mathbf{q}}_+)$ and $p_d(\tilde{\mathbf{q}}_-)$ represent the PDP evaluated at the distributions $\mathbf{q}^*$, $\tilde{\mathbf{q}}_+$, and $\tilde{\mathbf{q}}_-$, respectively. Due to the structure of the PDP and the fact that $\tilde{\mathbf{q}}_+$ and $\tilde{\mathbf{q}}_1$ differ only in the last two positions, it can be shown that the inequalities in (3.10) are equivalent to

$$P_i^{q_i^*} + P_j^{q_j^*}\left(1 - P_i^{q_i^*}\right) \leq P_i^{q_i^*+1} + P_j^{q_j^*-1}\left(1 - P_i^{q_j^*+1}\right), \tag{3.11}$$

$$P_i^{q_i^*} + P_j^{q_j^*}\left(1 - P_i^{q_j^*}\right) \leq P_i^{q_i^*-1} + P_j^{q_j^*+1}\left(1 - P_i^{q_j^*-1}\right), \tag{3.12}$$

respectively. First, let us proceed with (3.11) to derive a necessary and sufficient condition on q_i^* and q_j^*. After modifications, the inequality in (3.11) can be rewritten as

$$P_i^{q_i^*}(1-P_i) \; + \; P_j^{q_j^*}(1-P_i^{q_i^*}) - P_j^{q_j^*-1}(1-P_i^{q_i^*+1}) \leq 0.$$

We can further rewrite it as

$$(1-P_i)\left(P_i^{q_i^*} + P_j^{q_j^*}\left(\sum_{r=0}^{q_i^*-1} P_i^r\right) - P_j^{q_j^*-1}\left(\sum_{k=0}^{q_i} P_i^k\right)\right) \leq 0,$$

using the following standard equality,

$$(1-P_i^n) = (1-P_i)(1+P_i+P_i^2+\ldots+P_i^{n-1}). \tag{3.13}$$

Since $(1-P_i) \geq 0$ is always true, this implies that

$$P_i^{q_i^*} + P_j^{q_j^*}\left(\sum_{r=0}^{q_i^*-1} P_i^r\right) - P_j^{q_j^*-1}\left(\sum_{k=0}^{q_i^*} P_i^k\right) \leq 0.$$

Furthermore, we can rewrite the above inequality as

$$P_i^{q_i^*}\left(1-P_j^{q_j^*-1}\right) - \left(\sum_{r=0}^{q_i^*-1} P_i^r\right)\left(P_j^{q_j^*-1}(1-P_j)\right) \leq 0$$

Expanding $\left(1-P_j^{q_j^*-1}\right)$ and also using the fact that $(1-P_j) \geq 0$, we can write the above inequality as

$$P_i^{q_i^*}\left(\sum_{k=0}^{q_j^*-2} P_j^k\right) - \left(\sum_{r=0}^{q_i^*-1} P_i^r\right) P_j^{q_j^*-1} \leq 0.$$

This further implies that

$$\frac{P_i^{q_i^*}}{P_j^{q_j^*-1}} \leq \left(\frac{\sum_{r=0}^{q_i^*-1} P_i^r}{\sum_{k=0}^{q_j^*-2} P_j^k}\right). \tag{3.14}$$

With $C_{q_i^*-1} \triangleq \left(\sum_{r=0}^{q_i^*-1} P_i^r\right)$, and $C_{q_j^*-2} \triangleq \left(\sum_{k=0}^{q_j^*-2} P_j^k\right)$, we have,

$$\frac{P_i^{q_i^*}}{P_j^{q_j^*-1}} \leq \frac{C_{q_i^*-1}}{C_{q_j^*-2}}.$$

By taking logarithm on both sides, and subsequently rearranging the terms, we get

$$\frac{q_i^*}{(q_j^*-1)} \geq \frac{1}{(q_j^*-1)\log P_i}\log\left(\frac{C_{q_i^*-1}}{C_{q_j^*-2}}\right) + \frac{\log P_j}{\log P_i}. \tag{3.15}$$

This completes the proof for the first necessary condition. Although the second necessary condition can be proved along the same lines using (3.12), we omit the proof due to lack of space in this thesis. We highlight that the two conditions in (3.8) and (3.9) are also sufficient since the bounds are obtained by rearranging the terms in the condition on local minima. □

Corollary 3.1. *At high SNR, i.e., when P_k is negligible for each k, we have*

$$\frac{q_i^*}{(q_j^*-1)} \geq \frac{\log P_j}{\log P_i} + \epsilon_{i,j}^{(1)}, \tag{3.16}$$

$$\frac{q_i^*-1}{q_j^*} \leq \frac{\log P_j}{\log P_i} + \epsilon_{i,j}^{(2)}, \tag{3.17}$$

where $|\epsilon_{i,j}^{(1)}|$ and $|\epsilon_{i,j}^{(2)}|$ are small numbers.

Proof. When P_i and P_j are negligible, the first terms of the right hand side of both (3.8) and (3.9) are negligible because $\log\left(\frac{C_{q_i^*-1}}{C_{q_j^*-2}}\right) \approx 0$ and $\log\left(\frac{D_{q_i^*-1}}{D_{q_j^*}}\right) \approx 0$, and $\log(\frac{1}{P_i}) >> 0$. However, we note that these terms may either be positive or negative depending on the values of $q_i, q_j, P_i,$ and P_j. Therefore, by considering the polarity of these values, we bound the absolute values of $\epsilon_{i,j}^{(1)}$ and $\epsilon_{i,j}^{(2)}$ in the statement of the corollary. □

Based on the necessary and sufficient conditions derived in Theorem 3.3, we are ready to synthesize a low complexity algorithm to solve Problem 3.1.

3.4 Low-Complexity List-Decoding Algorithm

From Corollary 3.1, it is straightforward to note that at high SNR values, the necessary and sufficient conditions on the local minima satisfy the bounds in (3.16) and (3.17), for every pair

i, j such that $i \neq j$. We immediately notice that the following inequality also holds

$$\frac{q_i^* - 1}{q_j^*} < \frac{q_i^*}{q_j^*} < \frac{q_i^*}{q_j^* - 1}. \tag{3.18}$$

Using (3.16), (3.17), and the strict inequality constraints in (3.18), we propose a method to choose the ARQ distribution $\mathbf{q}$ in the following proposition.

proposition 3.1. *If the ARQ distribution* $\mathbf{q}$ *is chosen such that* $\frac{q_i}{q_j} = \frac{\log P_j}{\log P_i}$, *for* $i \neq j$, *then* $\mathbf{q}$ *is a local minima of the search space at high SNR.*

Proof. By choosing $\mathbf{q}$ such that $\frac{q_i}{q_j} = \frac{\log P_j}{\log P_i}$ for $i \neq j$ ensures that the sufficient conditions in (3.16) and (3.17) are trivially satisfied when $\epsilon_{i,j}^{(1)} < 0$ and $\epsilon_{i,j}^{(2)} > 0$. However, when $\epsilon_{i,j}^{(1)} > 0$ and $\epsilon_{i,j}^{(2)} < 0$, the sufficient conditions in (3.16) and (3.17) continue to satisfy, provided the SNR is sufficiently large to bound $|\epsilon_{i,j}^{(1)}| < \frac{q_i^*}{q_j^*-1} - \frac{q_i^*}{q_j^*}$ and $|\epsilon_{i,j}^{(2)}| < \frac{q_i^*}{q_j^*} - \frac{q_i^*-1}{q_j^*}$. $\square$

Based on the results in Proposition 3.1, we formulate Problem 3.2 as a means of solving Problem 3.1 at high SNR. However, from Problem 3.2, it is straightforward to note that a solution is not guaranteed since the ratio $\frac{\log P_j}{\log P_i}$, which is computed based on the LOS components and the SNR, need not be in $\mathbb{Q}$. Therefore, we propose to solve Problem 3.2 without the integer constraints, i.e., to find an ARQ distribution $\mathbf{q} \in \mathbb{R}^N$ satisfying the constraints $\frac{q_i}{q_j} = \frac{\log P_j}{\log P_i}$, for all i, j such that $i \neq j$, and $\sum_{k=1}^{N} q_k = q_{sum}$.

Problem 3.2. *For a given $\{P_1, P_2, \ldots, P_N\}$, find $q_1, q_2, \ldots q_N$ such that*

$$\begin{aligned} \frac{q_i}{q_j} &= \frac{\log P_j}{\log P_i}, \ \forall\, i, j \text{ such that } i \neq j, \\ q_k &\geq 1, \ \forall\, k, \\ q_k &\in \mathbb{Z}_+, \ \forall k, \\ q_1 + q_2 + \ldots + q_N &= q_{sum}. \end{aligned}$$

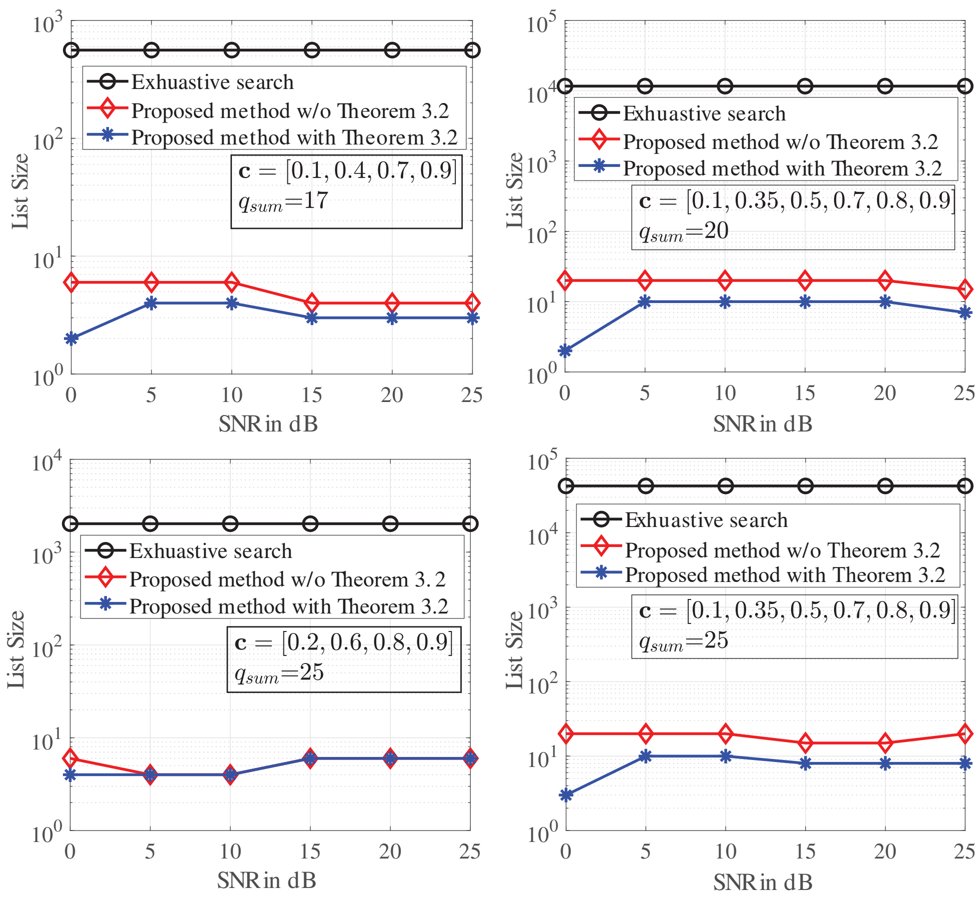

Figure 3.2: List size for $N = 4$ and $N = 6$ at $R = 1$.

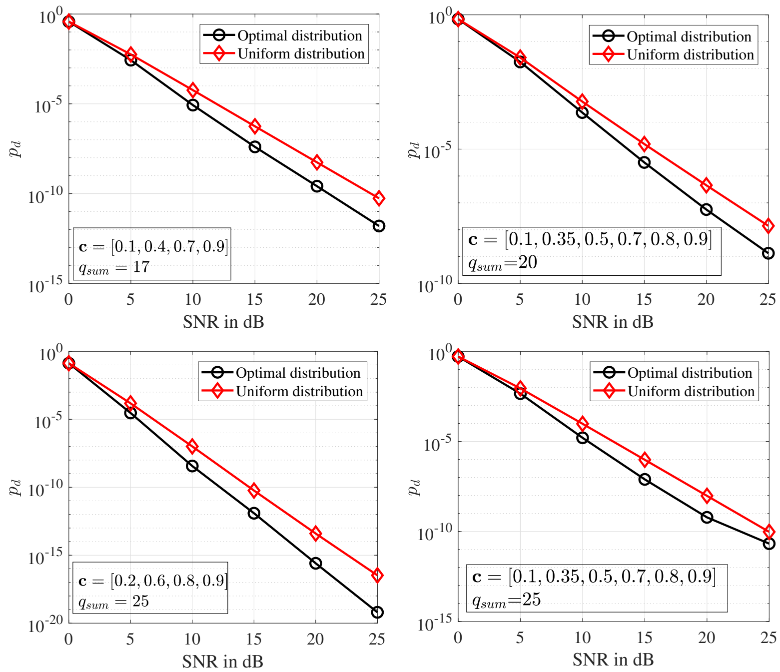

Figure 3.3: PDP for $N = 4$ and $N = 6$ at $R = 1$. With uniform distribution, each link is first allotted $\lfloor \frac{q_{sum}}{N} \rfloor$ ARQs, whereas the remaining ARQs are equally shared by the first q_{sum} mod N links.

3.4.1 Towards Solving Problem 3.2 without Integer Constraints

Towards solving Problem 3.2 without the integer constraints, we define $d_{i,j} \triangleq \frac{\log P_j}{\log P_i}$ for $i \neq j$. With that, the task of solving Problem 3.2 in $\mathbb{R}^N$ can be viewed as the task of solving the system of linear equations: $\mathbf{A}\mathbf{q}_{real} = \mathbf{b}$, where

$$\mathbf{A} = \begin{bmatrix} 1 & -d_{1,2} & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & -d_{2,3} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \dots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \dots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \dots & 1 & -d_{N-1,N} \\ 1 & 1 & 1 & \dots & \dots & 1 & 1 \end{bmatrix} \in \mathbb{R}^{N \times N},$$

$\mathbf{q}_{real} = [q_1, q_2, \dots, q_N]^T$ and $\mathbf{b} = [0, 0, \dots, 0, q_{sum}]^T$. Subsequently, a solution in $\mathbb{R}^N$ can be obtained as

$$\mathbf{q}_{real} = \mathbf{A}^{-1}\mathbf{b}, \tag{3.19}$$

as long as $\mathbf{A}$ is full rank. Although $\mathbf{q}_{real}$ in (3.19) satisfies the first and the last constraints of Problem 3.2, it cannot be used in the framework of multi-hop network since its components need not belong to $\mathbb{Z}_+$. In order to force the solution to lie in $\mathbb{Z}_+$, in the next section, we provide an algorithm that searches for ARQ distributions in $\mathbb{S}$ that are nearest to $\mathbf{q}_{real}$.

Remark 3.1. *It is possible to prove by contradiction that $\mathbf{q}_{real}$ cannot have any negative components since $d_{i,j}$ is strictly non-negative for all i, j. If at least one component of $\mathbf{q}_{real}$ is negative, it implies that every component of $\mathbf{q}_{real}$ is negative, and therefore, the sum constraint corresponding to the last row of $\mathbf{A}\mathbf{q}_{real} = \mathbf{b}$ will not be satisfied.*

3.4.2 List Generation using the Non-Integer Solution

Our approach, as presented in Algorithm 1, is to search for integer solutions in $\mathbb{S}$ that are nearest to $\mathbf{q}_{real}$. In particular, using $\mathbf{q}_{real}$, we obtain an ARQ distribution, denoted by $\tilde{\mathbf{q}} = [\tilde{q}_1, \tilde{q}_2, \dots, \tilde{q}_N] \in \mathbb{Z}^N$, by ceiling every component of $\mathbf{q}_{real}$, i.e., $\tilde{\mathbf{q}} = \lceil \mathbf{q}_{real} \rceil$. Since $\tilde{\mathbf{q}}$ may have zeros in some positions, we provide a brute-force correction by converting those zeros to ones. Subsequently, we compute $\sum_{i=1}^{N}\tilde{q}_i$, to verify the sum constraint. Due to the ceiling operation on each component, $\sum_{i=1}^{N}\tilde{q}_i$ is expected to exceed the sum constraint. Let E denote $(\sum_{i=1}^{N}\tilde{q}_i) - q_{sum}$. To identify the candidates in $\mathbb{S}$, we choose E positions in $\tilde{\mathbf{q}}$ and subtract one ARQ from each of these positions to make sure that the sum constraint is satisfied. Although, at most $\binom{N}{E}$ vectors in $\mathbb{S}$ can be generated this way, some of the combinations may not be valid due to the results of Theorem 3.2. Thus, we create a list of ARQ distributions in $\mathbb{S}$ (denoted by $\mathcal{L} \subset \mathbb{S}$) from $\mathbf{q}_{real}$. Finally, we compute the PDP of every ARQ distribution in $\mathcal{L}$, and then choose the one which minimizes the PDP. An illustrative example of our approach is given in Fig. 3.4 for a 2-hop network.

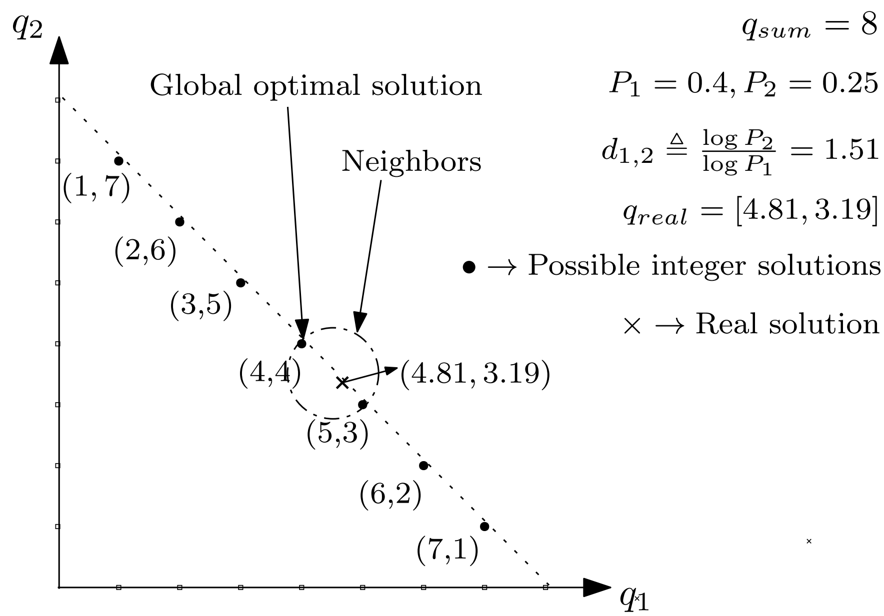

Figure 3.4: An illustrative example with $N = 2$: The LOS components and the SNR of the two links are such that $P_1 = 0.4$ and $P_2 = 0.25$. With $q_{sum} = 8$, our approach generates a list consisting 2 ARQ distributions, whereas the size of the search space is 7.

Algorithm 1 List Generation Based Algorithm

Require: $\mathbf{A}$, $\mathbf{b}$, q_{sum}, $\mathbf{c} = [c_1, c_2, \dots, c_N]$

Ensure: $\mathcal{L} \subset \mathbb{S}$ - List of ARQ distributions in $\mathbb{S}$.

Compute $\mathbf{q}_{real} = \mathbf{A}^{-1}\mathbf{b}$.
Compute $\tilde{\mathbf{q}} = \lceil \mathbf{q}_{real} \rceil$.
for $i = 1 : N$ **do**
  if $\tilde{q}_i = 0$ **then**
    $\tilde{q}_i = \tilde{q}_i + 1$
  end if
end for
Compute $E = \left(\sum_{i=1}^{N} \tilde{q}_i \right) - q_{sum}$
$\mathcal{L} = \{\mathbf{q} \in \mathbb{S} \mid d(\mathbf{q}, \tilde{\mathbf{q}}) = E, q_j \not> q_i$ for $c_i < c_j\}$.

3.5 Complexity Analysis and Simulation Results

As highlighted in Section 3.2, the computational complexity for solving Problem 3.1 through exhaustive search is $\binom{q_{sum}-1}{N-1}$. In contrast, we have used the results from Theorem 3.3, to first solve a relaxed version of Problem 3.2 in $\mathbb{R}^n$ (instead of $\mathbb{Z}^n$), and then search for candidates in $\mathbb{S}$ that are nearest to $\mathbf{q}_{real}$. Thus, the computational complexity of our method is dominated by the complexity of solving the system of linear equations, and that of the algorithm used to generate the list of candidates in $\mathbb{S}$. While the complexity for the former case is $O(N^3)$, the complexity of generating the list is at most $\binom{N}{E}$, where E is the excess number of ARQs after the ceiling operation.

To showcase the difference between the size of the list and that of the search space, we plot them in Fig. 3.2 for several instantiations of multi-hop networks with $N = 4$ and $N = 6$. In particular, we plot the list size both with and without incorporating the results of Theorem 3.2.

For the former case, when subtracting one ARQ from all possible $\binom{N}{E}$ positions from $\tilde{\mathbf{q}}$, we discard those ARQ distributions which follow the rule $q_i > q_j$ whenever $c_i > c_j$. As a result, we observe that the list size shortens remarkably after incorporating the rule of Theorem 3.2. Based on the simulation results, we observe that the ARQ distribution which minimizes the PDP from the list $\mathcal{L}$ matches the result of exhaustive search, thereby confirming that our list indeed encapsulates the optimal ARQ distribution of the underlying problem. Although we used high SNR results of Corollary 3.1 to synthesize the list-decoding method, we observe that the size of the list reduces significantly at low and medium range of SNR values as well. We attribute this behavior to the fact that the parameters $\epsilon_{i,j}^{(1)}$ and $\epsilon_{i,j}^{(2)}$, for $i \neq j$, satisfied $\epsilon_{i,j}^{(1)} < 0$ and $\epsilon_{i,j}^{(2)} > 0$, which in turn ensured that $\mathbf{q}_{real}$ satisfied the sufficient conditions of local minima. Finally, for the parameters considered in Fig. 3.2, we also plot the corresponding PDP in Fig. 3.3 so as to highlight the suboptimality of uniform ARQ distribution in LOS dominated multi-hop networks.

3.5.1 Simulation Results on Delay Analysis

It is well known that ARQ based strategies introduce additional delay on the packets owing to the use of ACK/NACK. As a result, introducing ARQ strategies to support low-latency applications may seem to contradict the motivation. However, we observe that these strategies provide benefits in the average delay because: (i) re-transmissions are asked only if required thereby avoiding the use of fixed block transmissions, and (ii) the loss in delay due to ACK/NACK is usually negligible compared to the transmission time for the packet. For instance, in the LTE FDD frame structure, the number of physical resource blocks (PRBs) used by PUCCH in the uplink to carry ARQs is at most eight per subframe [33], and this is a small portion of the total number of PRBs used in the downlink. To showcase these results, we present simulation results on the end-to-end delay for the following two schemes: (i) The non-cooperative ARQ strategy, wherein every relay is allotted ARQs by satisfying the sum constraint $\sum_{i=1}^{N} q_i = q_{sum}$. (ii) A repetition strategy, wherein multiple copies of the packet are transmitted on each link without having to wait for ACK/NACK. To keep the focus on ACK/NACK, in this scheme, the transmitter of the i-th hop sends all the q_i packets one after the other, so that the receiver can decode using all the packets. Using a 4-hop network to compare the above strategies, we measure the delay on every received packet at the destination, and then a build a delay profile by computing a probability mass function (PMF) on the delay. For the non-cooperative ARQ strategy, we use the total delay on a packet as $qT + q_{NACK}T_{NACK}$, where q is the total number of transmissions across all the nodes to reach the destination, T is the time taken for the packet transmission each time, q_{NACK} is the number of times NACK is sent in the network, and T_{NACK} is the additional delay introduced by NACK. Since q and q_{NACK} are random variables, the end-to-end delay on each packet is a random variable. In contrast, for the repetition scheme, the total time taken by the packet to reach the destination is a deterministic quantity, and this is equal to $q_{sum}T$. To generate the PMF, we use $T_{NACK} = 0.1T$ as a representative value since the NACK overhead is usually a small fraction of T. To justify 10% ARQ overhead for simulations, we have used the LTE FDD frame structure for 1.4 MHz, wherein the ratio of the number of PRBs used for PUCCH in the uplink to the number of PRBs used in the downlink is $\frac{8}{84} = 0.09$ [33]. We have also used $q_{sum} = 8$ by distributing two ARQs to each node in the 4-hop network. At the top of Fig. 3.5, we plot the PMF on the delay for various parameters of the non-cooperative ARQ strategy. The plots show that a majority of the packets reach the destination before the deadline of $8T$ time units. However, in the repetition strategy, all the packets reach the destination at $8T$ time units. These results confirm that the average delay offered by the ARQ strategy is much smaller compared to the repetition strategy, and this advantage is applicable despite using NACK. From the plots in Fig. 3.5, it is observed that the average delay on the packet is higher for low SNR values, and this behaviour is attributed to increased number of NACK when the SNR is low. Additional set of plots on the average delay is also presented at the bottom of Fig. 3.5. While we have pointed advantages in terms of average delay in Fig. 3.5, we also observe that the repetition strategy provides higher reliability to the packet; this is because each relay node decodes the packet using multiple copies instead of just one copy. As a result, the PDP of the repetition strategy is much lower compared to the ARQ strategy, as also shown in Fig. 3.5. Since the repetition strategy has the advantage to decode using multiple copies of the packet, we note that its comparison with the ARQ based strategy may not be fair. As a result, we present additional simulation results using a Hybrid ARQ strategy, which is a variant of ARQ strategy. In this scheme, unlike the ARQ based strategy, each node combines the received packet with the copy of previous failed attempts, and then decodes the packet. In Fig. 3.5, we plot both the delay profile and the PDP of the Hybrid ARQ scheme. The plots show that the PDP of the Hybrid ARQ matches that of the repetition method along with substantial reduction in the average delay on the packets. Overall, the simulation results on the delay profile confirm that ARQ based strategy does help in reducing the average delay despite making use of NACK.

In the next chapter, we address a range of cooperative ARQ strategies to achieve ultra-reliability and low-latency communication over a multi-hop network.

3.6 Summary

In this chapter, we have addressed a new framework to reliably communicate low-latency packets over a non-cooperative multi-hop network dominated by line-of-sight channels. We have specifically considered the question of how to distribute a given number of ARQs across the relay nodes in an ARQ based decode-and-forward relaying protocol such that the packet-drop-probability is minimized. To facilitate solving this problem with low-complexity methods, we have derived necessary and sufficient conditions on the optimal distribution of ARQs, and have subsequently used these conditions to propose a list-based enumeration algorithm. Simulation results confirm that the generated list is substantially shorter than that of exhaustive search, thereby rendering our algorithm amenable to implementation in practice.

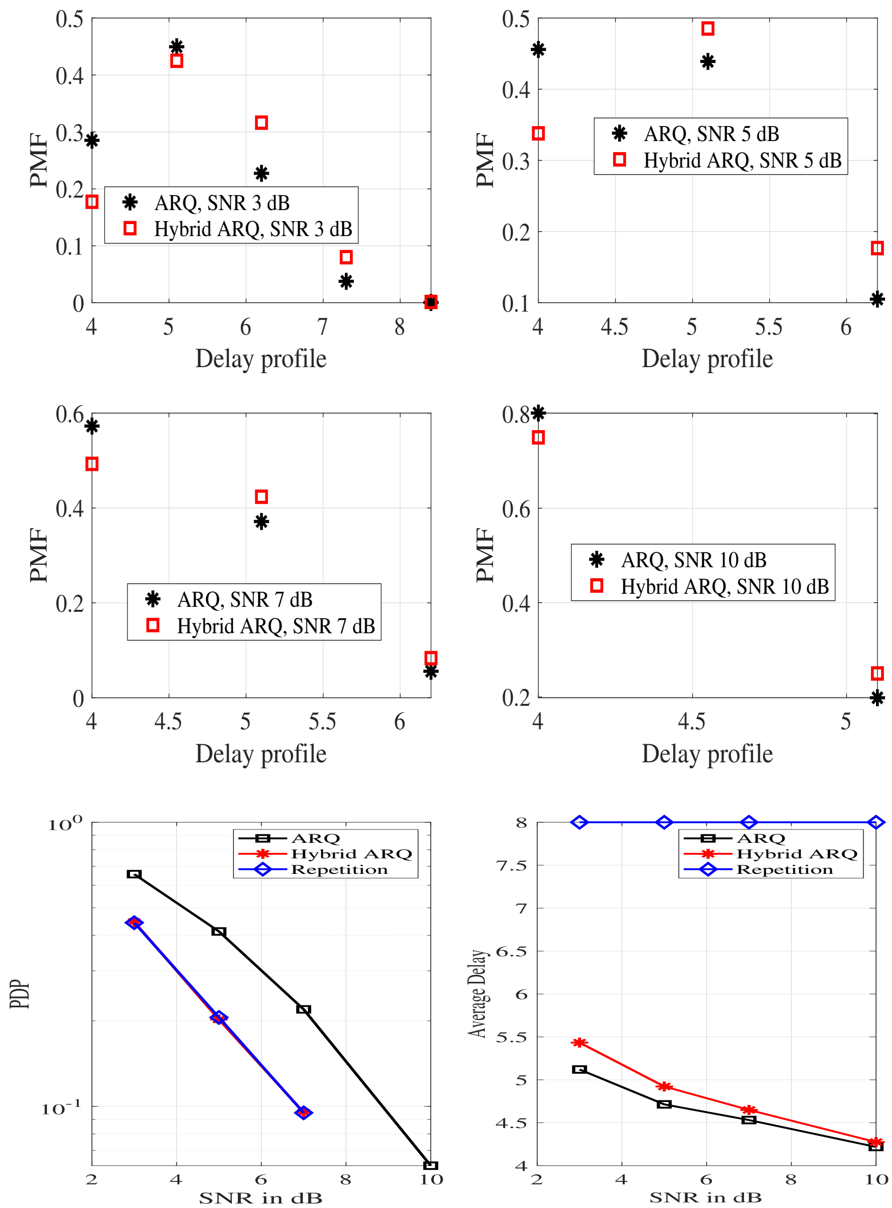

Figure 3.5: Simulation results using a 4-hop network with LOS vector $\mathbf{c} = [0.7,\ 0.3,\ 0.1,\ 0.5]$, $q_{sum} = 8$ at $R = 1$ and various values of SNR. At the top: Delay profile of the non-cooperative ARQ strategy and a hybrid ARQ strategy when $T = 1$ microsecond is the time taken for packet transmission, and $T_{NACK} = 0.1$ microsecond is the time taken for NACK. At the bottom: plots on the average delay and PDP of the non-cooperative ARQ along with that of a hybrid ARQ scheme and a repetition strategy.

Chapter 4

Cooperative ARQ Schemes in Multi-Hop Networks

4.1 Introduction

In the previous chapter, we discussed the non-cooperative multi-hop model wherein we assumed that every node has the knowledge of the ARQs allotted to themselves only. After allocating an appropriate number of ARQs at each hop, we identified that there may be unused ARQs at some hops because of the stochastic nature of the wireless channels. To circumvent this limitation, in this chapter, we propose the following ARQ protocols for a multi-hop network:

- Fully-cumulative strategies, wherein every node is assigned a maximum number of ARQs for packet re-transmission. In addition, every node has the knowledge of ARQs allotted to the node from which it receives the packet, and also residual number of ARQs unused by all the preceding nodes before it.

- Cluster-based cooperative strategy, wherein every node is assigned a maximum number of ARQs for packet re-transmission. In addition, some nodes have the knowledge of the

ARQs allotted to the node from which they receive the packet.

We show that by using the above protocols, the unused ARQs of the preceding node may be used by the next node in the chain and thereby providing an opportunity to reduce the end-to-end PDP [1]. In the next section, we discuss about the fully-cumulative strategy over a multi-hop network.

4.2 Fully-Cumulative ARQ Scheme

Consider the N-hop network which contains a source node, a destination node, and a set of $N-1$ intermediate relay nodes. The underlying assumptions for the system model in this section are similar to the non-cooperative model (as discussed in Chapter 3), wherein the LOS vector $\mathbf{c} = [c_1, c_2, \ldots, c_N]$ is used to represent the LOS components of the N-hop network, and $\mathbf{q} = [q_1, q_2, \ldots, q_N]$ is used to represent the corresponding ARQ distribution. In contrast to the non-cooperative model, we assume that every node has the knowledge of the number of ARQs allotted to its preceding node. This way, the unused number of ARQs of the preceding node can be used by the next node in the chain. Furthermore, we also assume that the packet structure contains a dedicated portion, referred to the *counter*, in order to carry the residual number of ARQs unused by all the preceding nodes in the network. We highlight that the use of the counter in the packet structure is necessary since a given relay node cannot overhear the residual ARQs from the upstream nodes other than the immediately preceding one. Thus, every node not only uses the residual ARQs allotted to the preceding node, but also uses the residual ARQs of all the nodes that forwarded the packet before it, thereby providing scope for minimizing the end-to-end PDP. Henceforth, throughout this work, we refer to this scheme as the fully-cumulative ARQ scheme. To explain the fully-cumulative ARQ model, suppose that the initial ARQ distribution of the multi-hop network is represented by $\mathbf{q} = [q_1, q_2, \ldots, q_N]$, where q_k is the number of ARQs

[1] Part of the results presented in this chapter are available in publication [30]

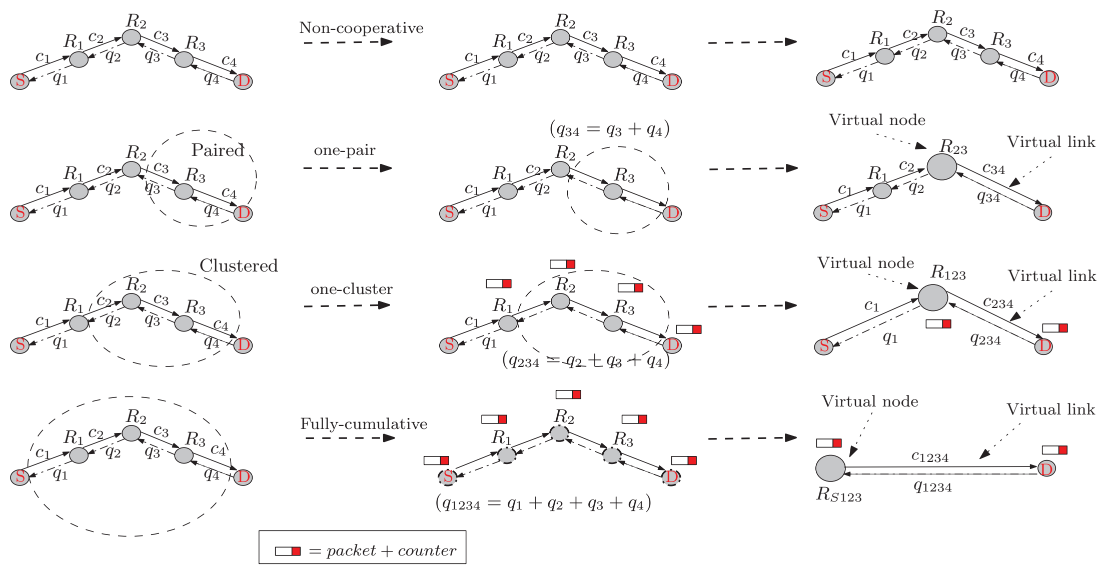

Figure 4.1: An illustrative example of 4-hop network employing (i) the non-cooperative strategy, (ii) the one-pair based cooperative scheme, (iii) the one-cluster based cooperative scheme, and (iv) the fully-cumulative scheme.

allotted to the k-th link. On the first link, suppose that $q_1^{'}$ out of q_1 ARQs are used to successfully transmit the packet to the next node, and as a result, $q_1^{''} = q_1 - q_1^{'}$ ARQs are unused. Since the next node in the chain has the knowledge of q_1 as well as the number of attempts made by the preceding node for successful transmission of the packet, it can use $q_2 + q_1^{''}$ ARQs to transmit the packet to the next node. Furthermore, in order to help the next node in the chain to make use of any residual number of ARQs, the second node encodes the number $q_2 + q_1^{''}$ in a dedicated portion of the packet, referred to as the counter, and then starts transmitting the packet to the next node. If the second node uses $q_2^{'}$ attempts, then the third node (which is the next node in the chain) can use $q_3 + q_2 + q_1^{''} - q_2^{'}$ number of attempts. Similarly, the third node encodes the number $q_3 + q_2 + q_1^{''} - q_2^{'}$ in the counter, and then starts transmitting the packet to the next node. This way, each intermediate relay node gains additional ARQs for packet transmission by using

- the knowledge of the number of ARQs allotted to itself,
- the number encoded in the counter portion of the packet, and
- the number of unsuccessful attempts made by the preceding node.

An example for the fully-cumulative ARQ scheme is given in Fig. 4.1, wherein all the nodes of a 4-hop network use the counter in the packet to learn the residual number of ARQs unused by the preceding nodes in the network. In this scheme, note that as the preceding node may have more number of ARQs than allotted to it (because of additional ARQs accumulated from its preceding node), the next node in the chain does not know the number of attempts that would be made by its preceding node. Therefore, a given node, upon sending a NACK to its preceding node, will wait for the packet only for τ amount of time. If it does not receive the packet within τ units of time, then it implies that the preceding node has exhausted the allotted number of ARQs, and therefore, the packet is said to be dropped in the network. It is intuitive that this process of cumulatively adding the unused number of ARQs at each hop will reduce the PDP in comparison with the non-cooperative strategy without changing the sum constraint on latency $q_{\text{sum}} = \sum_{i=1}^{N} q_i$.

In the following theorem, we show that for a given q_{sum}, the strategy of transferring q_{sum} ARQs to the source node minimizes the PDP of the fully-cumulative ARQ scheme.

Theorem 4.1. *For an N-hop network with LOS vector $\mathbf{c} = [c_1, c_2, \ldots, c_N]$, if the N nodes implement the fully-cumulative ARQ scheme, then the optimal ARQ distribution that minimizes the PDP is given by $\mathbf{q}^* = [q_{sum}, 0, \ldots, 0]$, where $q_{sum} = \sum_{j=1}^{N} q_j$.*

Proof. First, we prove the result for $N = 2$. With ARQ distribution $\mathbf{q} = [q_1, q_2]$, the PDP of the fully-cumulative ARQ scheme is given by

$$PDP_1 = p_1^{q_1} - (1 - p_1) \left(\sum_{j=1}^{q_1} p_1^{q_1 - j} p_2^{q_2 + j - 1} \right),$$

where the running index j captures the residual ARQs unused by the first link. Now, let us transfer one ARQ from the second link to the first link assuming that $q_2 > 1$. Therefore, with the ARQ distribution $\mathbf{q} = [q_1 + 1, q_2 - 1]$, the PDP is

$$PDP_2 = p_1^{q_1+1} - (1 - p_1) \left(\sum_{j=1}^{q_1+1} p_1^{q_1+1-j} p_2^{q_2-1+j-1} \right).$$

In the rest of the proof under $N = 2$, we will show that $PDP_2 < PDP_1$. After simple manipulations, the expression for $PDP_2 - PDP_1$ can be written as

$$\begin{aligned} PDP_2 - PDP_1 &= p_1^{q_1+1} - p_1^{q_1} + (1 - p_1)p_1^{q_1} p_2^{q_2-1} \\ &= p_1^{q_1} \left(p_1 + (1 - p_1)p_2^{q_2-1} - 1 \right). \end{aligned}$$

In the above expression, since $0 < p_1 < 1$, and $0 < p_2^{q_2-1} \leq 1$, the term $p_1 + (1 - p_1)p_2^{q_2-1}$ is a convex combination of 1 and $p_2^{q_2-1}$, and therefore, $p_1 + (1 - p_1)p_2^{q_2-1} \leq 1$. As a consequence, we have $PDP_2 - PDP_1 \leq 0$, wherein the equality holds only if $q_2 = 1$. Since this result is true for any q_1 and $q_2 \geq 1$, we can recursively apply the logic of transferring one ARQ from the second link to the first link to show that the ARQ distribution $[q_{sum}, 0]$ is the optimal ARQ distribution of the fully-cumulative cooperative ARQ scheme. This completes the proof of the theorem for $N = 2$.

Using the hypothesis step of induction, we assume that the statement of the theorem is true for $N = k$. To prove the result for $N = k + 1$, let $\mathbf{q} = [q_1, q_2, \ldots, q_{k+1}]$ be the ARQ distribution across the $k + 1$ nodes. Since the nodes apply the fully-cumulative ARQ scheme, we treat the set of the transmitters of the k links including 2nd link, 3rd link, ..., $(k + 1)$-th link as a single virtual node, henceforth referred to as R_v. With that, we have a virtual two-hop network involving the source, node R_v, and the destination, wherein the source and node-v are allotted q_1 and $q_{sum,v} = \sum_{i=2}^{k+1} q_i$ number of re-transmissions, respectively. Since R_v is a virtual node, the manner in which $q_{sum,v}$ ARQs is internally distributed among the k constituent nodes dictates the probability with which the packet is dropped by the virtual node. With a total of $q_{sum,v}$ ARQs allotted to the virtual node, let $PDP_v(q_{sum,v})$ denote the probability that the packet is dropped at the virtual node. Thus, since the source node and node R_v also follow the fully-cumulative cooperative ARQ scheme, the PDP of this two-hop virtual network is given by

$$PDP = p_1^{q_1} - (1 - p_1) \left(\sum_{j=1}^{q_1} p_1^{q_1 - j} PDP_v(q_{sum,v} + j - 1) \right). \tag{4.1}$$

For a fixed p_1 and q_1, the PDP expression (4.1) can be minimized by minimizing $PDP_v(q_{q_{sum,v}} + j - 1)$ for each $1 \leq i \leq q_1$. To help this cause, from the induction step, we know that $PDP_v(q_{q_{sum,v}} + j - 1)$ can be minimized by allocating $q_{sum,v} + j - 1$ ARQs to the transmitter of the second link, and zero ARQs to the other links in the downstream. In the context of the N-hop network, this leaves us with the question of minimizing PDP in (4.1) over the ARQ distributions of the form $[q_1, q_{sum} - q_1, 0, \ldots, 0]$ such that $1 \leq q_1 \leq q_{qsum}$. Finally, towards proving the statement of the theorem, we transfer one ARQ from the virtual node to the source node. As a result, the updated expression of PDP, denoted by PDP_u, is

$$p_1^{q_1+1} - (1 - p_1) \left(\sum_{j=1}^{q_1+1} p_1^{q_1+1-j} PDP_v(q_{sum,v} - 1 + j - 1) \right). \tag{4.2}$$

Similar to the proof for the statement of this theorem for $N = 2$, we will show that $PDP_u < PDP$. Towards that direction, the term $PDP_u - PDP$ can be written (after some simple manipulations) as

$$\begin{aligned} & p_1^{q_1+1} - p_1^{q_1} + (1 - p_1) p_1^{q_1} PDP_v(q_{sum,v} - 1) \\ = \; & p_1^{q_1} \left(p_1 + (1 - p_1) PDP_v(q_{sum,v} - 1) - 1 \right). \end{aligned}$$

Note that $PDP_v(q_{sum,v} - 1) < 1$. Furthermore, since $p_1 < 1$, the term $p_1 + (1 - p_1) PDP_v(q_{sum,v} - 1)$ is a convex combination of 1 and $PDP_v(q_{sum,v} - 1)$. Therefore, we have the inequality $p_1 + (1 - p_1) PDP_v(q_{sum,v}) < 1$, and this in turn, implies that $PDP_u - PDP < 0$. Finally, since this inequality can be proved for any q_1, we can show through an iterative process that the optimal ARQ distribution for the N-hop case is $[q_{sum}, 0, 0, \ldots, 0]$. This completes the proof. $\square$

Although the above result is applicable for any $N > 1$, it is important to note that the overhead of conveying the residual number of ARQs in the packet is not required when $N = 2$. This is because the second node can learn the number of residual ARQs of the first node by counting the number of unsuccessful attempts.

4.3 Cluster based ARQ Schemes

From the previous section, the fully-cumulative ARQ scheme is not applicable when some nodes in the network do not have the knowledge of the ARQs allotted to their preceding nodes. In such low-latency applications, only a proper subset of nodes in the network may cooperate to use the residual number of ARQs from the preceding nodes, whereas the rest of the nodes may only use the number of ARQs allotted to them. As a result, it is important to derive the end-to-end PDP in such scenarios, and then solve the problem of computing the ARQ distribution that minimizes the PDP of the network. Henceforth, throughout this chapter, we refer to such strategies as the cluster-based ARQ schemes.

In the next two sections, we consider two variants of the cluster-based ARQ schemes: (i) the pair-wise cooperative ARQ scheme, wherein only disjoint pairs of nodes in the path apply the cooperative ARQ method among themselves, and (ii) the cluster-wise cooperative ARQ scheme, wherein disjoint clusters, made up of more than two consecutive nodes, are formed to apply the fully-cumulative ARQ scheme among themselves.

4.3.1 Pair-wise Cooperative Scheme

Instead of incorporating the fully-cumulative strategy, we introduce cooperation among a subset of nodes such that the end-to-end PDP of the network can be reduced with respect to the non-cooperative case. To start with a simple version of this idea, we pair the k-th and $(k+1)$-th link, for $1 \leq k \leq N-1$, which are allotted q_k and q_{k+1} number of ARQs, respectively. This implies that the transmitter of the $(k+1)$-th link has the knowledge of q_k, and as a result, the residual ARQs unused by the preceding node can be used by the transmitter of the $(k+1)$-th link. However, apart from these two nodes, the rest of the nodes in the network do not implement cooperation. With such a simple idea, it is interesting to understand the characteristics of the optimal ARQ distribution that minimizes the PDP of the strategy, and also answer the question of how to compute the optimal ARQ distribution. Furthermore, this idea of pairing two nodes can also be generalized to create at most $\lfloor \frac{N}{2} \rfloor$ pairs such that cooperation among the nodes outside the pairs is forbidden. Towards that direction, we define a pair-wise non-cooperative scheme.

Definition 4.1. *The N-hop network is said to employ a pair-wise cooperative scheme defined by the set $\mathcal{P}$, for $\mathcal{P} \subset [N]$, where $[N] = \{1, 2, \ldots, N\}$, if it satisfies the following properties:*

- *For each $a \in \mathcal{P}$, we have $a+1 \notin \mathcal{P}$ and $a-1 \notin \mathcal{P}$ provided $a+1 \in [N]$ and $a-1 \in [N]$, respectively.*
- *For each $a \in \mathcal{P}$, the transmitter of the $(a+1)$-th link has the knowledge of both q_a and q_{a+1}.*

From the above definition, it is clear that for a pair-wise cooperative scheme with $\mathcal{P}$, we have $|\mathcal{P}| \leq \lfloor \frac{N}{2} \rfloor$. In the following theorem, we characterize the optimal ARQ distribution when a pair-wise cooperative scheme defined by $\mathcal{P}$ is employed.

Theorem 4.2. *For an N-hop network with LOS vector $\mathbf{c} = [c_1, c_2, \ldots, c_N]$, let $\mathbf{q} = [q_1, q_2, \ldots, q_N]$ be the corresponding ARQ distribution assuming that the nodes implement the non-cooperative* *scheme. Now, if the N nodes implement a pair-wise cooperative scheme defined by the set $\mathcal{P}$, then conditioned on the ARQ distribution $\mathbf{q}$, the optimal ARQ distribution $\mathbf{q}^* = [q_1^*, q_2^*, \ldots, q_N^*]$ that minimizes the PDP of this scheme is of the form $q_a^* = q_a + q_{a+1}$ and $q_{a+1}^* = 0$, for each $a \in \mathcal{P}$. However, for $a \notin \mathcal{P}$ such that $a - 1 \notin \mathcal{P}$, then we have $q_a^* = q_a$.*

Proof. First, we prove this theorem by considering the case of a pair-wise cooperative scheme defined by the set $\mathcal{P}$ such that $|\mathcal{P}| = 1$. This implies that out of the N nodes, two successive nodes, say R_j and R_{j+1}, for some $1 \leq j \leq N-2$, have been paired to improve the PDP of the network. This further implies that node R_{j+1} has the knowledge of q_j as well as q_{j+1}, and therefore the residual number of ARQs unused by R_j can be used by R_{j+1}. By combining the nodes R_j and R_{j+1}, we form a virtual node $R_{j,j+1}$ that is connected to node R_{j-1} and the node R_{j+2} on either side. Note that this gives rise to a virtual $(N-1)$-hop network wherein the number of ARQs allocated to the virtual node $R_{j,j+1}$ is $q_j + q_{j+1}$. With this formulation, the question of how do these two nodes internally distribute $q_j + q_{j+1}$ ARQs among them is of interest in this theorem. From first principles, the PDP of this virtual $(N-1)$-hop network is given by

$$P_1^{q_1} + P_2^{q_2}(1 - P_1^{q_1}) + \ldots + P_{j-1}^{q_{j-1}}\Big[\prod_{i=1}^{j-2}(1 - P_i^{q_i})\Big] + \ PDP_v(q_j + q_{j+1})\ldots$$
$$\Big[\prod_{i=1}^{j-1}(1 - P_i^{q_i})\Big] + \ P_N^{q_N}\Big[\prod_{i=1, i\neq j, i\neq j+1}^{N-1}(1 - P_i^{q_i})\Big](1 - PDP_v(q_j + q_{j+1})),$$

where $PDP_v(q_j + q_{j+1})$ represents the probability that the packet is dropped at the virtual node after using a total of $q_j + q_{j+1}$ number of ARQs between them. By using the swapping results of Theorem 3.1 on this virtual $(N-1)$-hop network, we can exchange the position of the virtual node and the node R_{N-1} to obtain the same PDP, given by

$$P_1^{q_1} + P_2^{q_2}(1 - P_1^{q_1}) + P_{j-1}^{q_{j-1}}\Big[\prod_{i=1}^{j-2}(1 - P_i^{q_i})\Big] + \ P_{N-1}^{q_{N-1}}\Big[\prod_{i=1}^{j-1}(1 - P_i^{q_i})\Big] + \dots$$
$$+ PDP_v(q_j + q_{j+1})\Big[\prod_{i=1, i\neq j, i\neq j+1}^{N-1}(1 - P_i^{q_i})\Big].$$

Since the number of ARQs allocated to the other nodes in the network is fixed, we would need to find a way to internally distribute the ARQs among the paired nodes. It is to be noted that conditioned on $q_1, q_2, \dots, q_{j-1}, q_{j+2}, \dots, q_N$ (which does not include q_j and q_{j+1}), minimizing the above expression of PDP is same as minimizing $PDP_v(q_j + q_{j+1})$, which is the packet drop probability of the virtual node. However, since $PDP_v(q_j + q_{j+1})$ is the PDP of a two-hop network internally formed between node-j and node-$(j+1)$, from Theorem 4.1, we state that the optimal distribution to internally distribute q_j and q_{j+1} ARQs so as to minimize the $PDP_v(q_j + q_{j+1})$ is to provide all the q_{j+1} ARQs to the relay R_j, and then to keep zero ARQs at relay R_{j+1}. This further implies that the optimal ARQ distribution between node-j and node-$(j+1)$ conditioned on the ARQ $q_1, q_2, \dots, q_{j-1}, q_{j+2}, \dots, q_N$ is as followed by the statement of the theorem for $|\mathcal{P}| = 1$.

In general, when $|\mathcal{P}| > 1$, the statement of the theorem can be proved using contradiction. Suppose that the optimal ARQ distribution is such that there exists at least one pair of nodes, R_j and R_{j+1} that satisfy the ARQ relation $q_j \neq 0$ and $q_{j+1} \neq 0$. Using the swapping results, we can exchange the virtual node formed by R_j and R_{j+1} with node R_{N-1}, and then obtain the PDP as

$$p_d \ = \ P_1^{q_1} + P_2^{q_2}(1 - P_1^{q_1}) + \dots + \ PDP_v(q_j + q_{j+1})\Big[\prod_{i=1}^{N-1}(1 - P_i^{q_i})\Big].$$

Given that the ARQs for the rest of the nodes are fixed, the above PDP can be further minimized by minimizing $PDP_v(q_j + q_{j+1})$ by transferring all the ARQs of node R_{j+1} to R_j. This implies that we can synthesize an ARQ distribution that yields PDP smaller than the optimal ARQ distribution, which in turn is a contradiction. This completes the proof. □

For a multi-hop network implementing a pair-wise cooperative scheme defined by $\mathcal{P}$ such that $|\mathcal{P}| = \gamma$, let $N_{\gamma p}$ denote the number of links in the virtual multi-hop network by forming virtual nodes after combining the paired nodes. Furthermore, let the ARQ distribution on the corresponding virtual $N_{\gamma p}$-hop network be denoted by $\bar{q}_1, \bar{q}_2, \ldots, \bar{q}_{N_{\gamma p}}$, wherein $\bar{q}_k$ is the number of ARQs allotted to the k-th node in the virtual $N_{\gamma p}$-hop network, and where $N_{\gamma p} = (N - |\mathcal{P}|)$. The nodes in this $N_{\gamma p}$-hop network can be partitioned into two groups, as $[N_{\gamma p}] = \mathcal{V} \cup \mathcal{V}^c$, wherein $\mathcal{V}$ denotes the set of virtual nodes formed by combining successive pairs of nodes in the original network. For this setup, a formulation of the optimization problem to solve the ARQ distribution of the $N_{\gamma p}$-hop network is given in Problem 4.1, where $p_d(\mathcal{P})$ represents the PDP of the virtual γp-hop network obtained by using the set $\mathcal{P}$.

After solving Problem 4.1, we use $\bar{q}_1^*, \bar{q}_2^*, \ldots, \bar{q}_{N_{\gamma p}}^*$ to obtain the ARQ distribution for the original N-hop network by providing $\bar{q}_k^*$ ARQs to the first node of the virtual node whenever $k \in \mathcal{V}$, and zero ARQs to the other node in the virtual node. However, when $k \in \mathcal{V}^c$, we assign $\bar{q}_k^*$ ARQs to the corresponding physical node in the network. In the next section, we discuss the idea of cluster-wise cooperative ARQ scheme.

Problem 4.1. *For the virtual $N_{\gamma p}$-hop network, where $N_{\gamma p} = (N - |\mathcal{P}|)$, solve*

$$\bar{q}_1^*, \bar{q}_2^*, \ldots \bar{q}_{N_{\gamma p}}^* = \arg \min_{\bar{q}_1, \bar{q}_2, \ldots \bar{q}_L} p_d(\mathcal{P})$$

subject to

$$\begin{aligned} \bar{q}_k &\in \mathbb{Z}_+, \\ \bar{q}_k &\geq 1, \ \text{if } k \in \mathcal{V}^c \\ \bar{q}_k &\geq 2, \ \text{if } k \in \mathcal{V} \\ \bar{q}_1 + \bar{q}_2 + \ldots + \bar{q}_{N_{\gamma p}} &= q_{sum}. \end{aligned}$$

4.3.2 Cluster based Cooperative Scheme

In this section, we discuss a cluster-wise cooperative scheme wherein the cooperation is among a set of more than two successive nodes akin to the fully-cumulative scheme. As a result, there will be additional communication-overhead since the cooperating nodes also forward the residual ARQs in the packet, in addition to knowing the number of ARQs allotted to its preceding node. Since we consider more than two successive nodes for cooperation, we refer to this method as the cluster-wise cooperative scheme. In this approach, let t represent the cluster size i.e., the number of links in the cluster, and let γ represent the number of such clusters in the network. The network model of this cluster-wise cooperative model is as exemplified in Fig. 4.1. Note that the cluster-wise cooperative scheme reduces to pair-wise cooperative scheme when $t = 2$. The following definition formally introduces a cluster-wise cooperative model with cluster-size t.

Definition 4.2. *The N-hop network is said to employ a cluster-wise cooperative scheme defined by the cluster set $\mathcal{C}_t$, for $\mathcal{C}_t \subset [N]$, if it satisfies the following properties:*

- *For each $a \in \mathcal{C}_t$, we have $a + i \notin \mathcal{C}_t$ for $1 \leq i \leq t - 1$, and $a - i \notin \mathcal{C}_t$ for $1 \leq i \leq t - 1$.*
- *For each $a \in \mathcal{C}_t$, the transmitter of the $(a + i)$-th link has the knowledge of q_{a+i-1} for each $1 \leq i \leq t - 1$.*
- *For each $a \in \mathcal{C}_t$, the transmitter of the $(a + i)$-th link with $0 \leq i < t - 1$ uses a counter in the packet structure in order to convey the residual ARQs.*

In the following theorem, we characterize the optimal ARQ distribution when a cluster-wise cooperative scheme, defined by $\mathcal{C}_t$, is employed.

Theorem 4.3. *For an N-hop network, let $\mathbf{q} = [q_1, q_2, \ldots, q_N]$ be the ARQ distribution on the nodes. If the N nodes implement a cluster-wise cooperative scheme defined by the set $\mathcal{C}_t$, then conditioned on the ARQ distribution $\mathbf{q}$, the optimal ARQ distribution $\mathbf{q}^* = [q_1^*, q_2^*, \ldots, q_N^*]$ that* *minimizes the PDP is of the form* $q_a^* = \sum_{i=0}^{t-1} q_i$ *and* $q_{a+i}^* = 0$*, for* $1 \leq i \leq t-1$*. However, for* $a \notin \mathcal{C}_t$ *such that* $a - i \notin \mathcal{C}_t$*, for* $1 \leq i \leq t-1$*, then we have* $q_a^* = q_a$*.*

Proof. The statement of this theorem can be proved along the similar lines of Theorem 4.2. However, instead of forming a virtual node by combining two nodes, we form a virtual node by combining a set of t successive nodes that form a cluster. □

For a multi-hop network implementing a cluster-wise cooperative scheme defined by the set $\mathcal{C}_t$, let the ARQ distribution of the corresponding virtual $N_{\gamma c}$-hop network be denoted by $\bar{q}_1, \bar{q}_2, \ldots, \bar{q}_{N_{\gamma c}}$, where $N_{\gamma c} = (N + |\mathcal{C}_t|(1-t))$, and $\bar{q}_k$ is the number of ARQs allotted to the transmitter of the k-th link in the virtual $N_{\gamma c}$-hop network. Furthermore, the nodes in this $N_{\gamma c}$-hop network can be partitioned into two groups $[N_{\gamma c}] = \mathcal{V} \cup \mathcal{V}^c$, wherein $\mathcal{V}$ denotes the set of virtual nodes formed by combining t successive nodes in the network, and $\mathcal{V}^c$ denotes the set of physical nodes that do not participate in cooperation. A formulation of the optimization problem to solve the ARQ distribution of the $N_{\gamma c}$-hop network is given in Problem 4.2, wherein $p_d(\mathcal{C}_t)$ represents the PDP of the $N_{\gamma c}$-hop network defined by the set $\mathcal{C}_t$.

After solving Problem 4.2, we use $\bar{q}_1^*, \bar{q}_2^*, \ldots \bar{q}_{N_{\gamma c}}^*$ to obtain the ARQ distribution for the original N-hop network by providing $\bar{q}_k^*$ ARQs to the first node of the virtual node whenever $k \in \mathcal{V}$. On the other hand, when $k \in \mathcal{V}^c$, we provide $\bar{q}_k^*$ number of ARQs to the corresponding physical node.

Note that our approach of solving the problem for both pair-wise cooperative and cluster-wise cooperative schemes are similar. However, their implementations are different since the cluster-wise cooperative scheme necessitates sharing the residual number of ARQs among the nodes in the cluster through the counter embedded in the packet. On the other hand, the idea of pair-wise cooperative scheme does not require additional overheads in the packet.

Problem 4.2. *For the virtual $N_{\gamma c}$-hop network, where $N_{\gamma c} = N + |\mathcal{C}_t|(1-t)$, solve*

$$\bar{q}_1^*, \bar{q}_2^*, \ldots \bar{q}_{N_{\gamma c}}^* = \arg \underset{\bar{q}_1, \bar{q}_2, \ldots \bar{q}_{N_{\gamma c}}}{min} \; p_d(\mathcal{C}_t)$$

subject to

$$\begin{aligned} \bar{q}_k &\in \mathbb{Z}_+, \\ \bar{q}_k &\geq 1, \; \textit{if } k \in \mathcal{V}^c \\ \bar{q}_k &\geq t, \; \textit{if } k \in \mathcal{V} \\ \bar{q}_1 + \bar{q}_2 + \ldots + \bar{q}_{N_{\gamma c}} &= q_{sum}. \end{aligned}$$

4.3.3 Algorithm for the Cluster based Cooperative ARQ Scheme

In this section, we present an algorithm to compute a near-optimal ARQ distribution for the cluster-wise cooperative scheme. As a special case, when $t = 2$, this algorithm is also applicable for the pair-wise cooperative scheme. Similar to our algorithm for the non-cooperative scheme (as discussed in Chapter 3), our approach is to generate a list of ARQ distributions that are contenders for the optimal ARQ distribution.

Our algorithm solves Problem 4.2, wherein the original N-hop network has been reduced to a virtual $N_{\gamma c}$-hop network comprising L_v virtual nodes and L_p physical nodes. Due to the presence of virtual nodes, we immediately notice that the probability that a packet is dropped at a virtual node is coupled with the number of ARQs in a manner different from that of the physical nodes. Therefore, when attempting to compute the list of ARQ distributions for the $N_{\gamma c}$-hop network, first we fix the ARQ numbers on the virtual nodes with the constraint that (i) $q_k \geq t$ for each $k \in \mathcal{V}$, and (ii) $\sum_{k \in \mathcal{V}} q_k \leq q_{sum} - L_p$. Subsequently, for each combination of ARQs given to the virtual nodes, we obtain a list of ARQ distributions on the physical nodes using the low-complexity algorithm proposed for the non-cooperative case. A detailed description of our approach to generate a list of ARQ distribution for the cooperative scheme is given in Algorithm 2. For the sake of description, our algorithm assumes that the first L_p links of the network do not cooperate, while the rest of the links are grouped into γ clusters each containing t links.

Out of the q_{sum} ARQs, we assign a total of q_c ARQs to the L_v virtual nodes, where $tL_v \leq q_c \leq q_{sum} - L_p$ (see line 2). This is because a minimum of one ARQ must be allotted to each of the physical nodes, and a minimum of t ARQs must be allotted to each of the virtual nodes. Towards finding the ARQ distribution for the L_p physical nodes, we define $d_{\alpha,\beta} \triangleq \frac{\log P_\beta}{\log P_\alpha}$ for $\alpha \neq \beta$ and $\alpha, \beta \in \mathcal{V}^c$. With that, the task of obtaining ARQ distribution with integer constraints can be approached by first solving the system of linear equations $\mathbf{A}_p \mathbf{q}_{p,real} = \mathbf{b}_p$ in $\mathbb{R}^{L_p}$, where

$$\mathbf{A}_p = \begin{bmatrix} 1 & -d_{1,2} & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & -d_{2,3} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \dots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \dots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \dots & 1 & -d_{L_p-1,L_p} \\ 1 & 1 & 1 & \dots & \dots & 1 & 1 \end{bmatrix},$$

$\mathbf{q}_{p,real} = [q_1, q_2, \dots, q_{L_p}]^T$ and $\mathbf{b}_p = [0, 0, \dots, 0, (q_{sum} - q_c)]^T$, where q_c is the total number of ARQs allotted to the L_v virtual nodes. Subsequently, for the physical nodes, an ARQ solution in $\mathbb{R}^{L_p}$ can be obtained as $\mathbf{q}_{p,real} = \mathbf{A}_p^{-1} \mathbf{b}_p$, as long as $\mathbf{A}_p$ is full rank (see line 4). Using $\mathbf{q}_{p,real}$, we use techniques similar to the non-cooperative case to force a solution in the search space satisfying the sum constraint. Specifically, as shown from line 5 of Algorithm 2, each component of $\mathbf{q}_{p,real}$ is forced to be a positive integer using the ceiling operation, and then unnecessary number of ARQs are removed from E_p locations such that the sum constraint on the ARQs on the physical nodes is satisfied. By subtracting one ARQ from each of the E_p locations, we create a list of ARQ distributions on the L_p physical nodes, denoted by $\mathcal{L}_p^{'}$ (see line 12).

Algorithm 2 List Generation Algorithm for the cluster-based cooperative ARQ Strategy

Require: $\mathbf{A}_p$, $\mathbf{b}_p$, q_{sum}, $\mathbf{c} = [c_1, c_2, \ldots, c_{L_p}]$, t, L_v, L_p

Ensure: $\mathcal{L}_{final} \subset \mathbb{S}$ - List of ARQ distributions in $\mathbb{S}$.

$\mathcal{L}_{final} = \phi$

for $q_c = tL_v : (q_{sum} - L_p)$ **do**

  Assign $\mathbf{b}_p = [0, 0, \ldots, q_{sum} - q_c]$.

  Compute $\mathbf{q}_{p,real} = \mathbf{A}_p^{-1}\mathbf{b}_p$.

  Compute $\tilde{\mathbf{q}}_p = \lceil \mathbf{q}_{p,real} \rceil$.

  for $j = 1 : L_p$ **do**

    if $\tilde{q}_{p,j} = 0$ **then**

      $\tilde{q}_{p,j} = \tilde{q}_{p,j} + 1$

    end if

  end for

  Compute $E_p = \left(\sum_{l=1}^{L_p} \tilde{q}_{p,l}\right) - (q_{sum} - q_c)$.

  $\mathcal{L}_p^{'} = \{\mathbf{q_p} \in \mathbb{Z}_+^{L_p} \mid \sum_{\alpha \in \mathcal{V}^c} q_{p,\alpha} = q_{sum} - q_c, d(\mathbf{q_p}, \tilde{\mathbf{q}}_p) = E_p,\ q_{p,\alpha} \not> q_{p,\beta}$ for $c_\beta < c_\alpha$ s.t. $\alpha, \beta \in \mathcal{V}^c\}$.

  $\mathcal{L}_v^{'} = \{\mathbf{q}_v \in \mathbb{Z}_+^{L_v} \mid \sum_{\alpha \in \mathcal{V}} q_{v,\alpha} = q_c, q_{v,\alpha} \geq t \text{ for } \alpha \in \mathcal{V}\}$

  for $j_1 = 1 : |\mathcal{L}_p^{'}|$ **do**

    for $j_2 = 1 : |\mathcal{L}_v^{'}|$ **do**

      Insert $[\mathcal{L}_p^{'}(j_1) || \mathcal{L}_v^{'}(j_2)]$ into $\mathcal{L}_{final}$

    end for

  end for

end for

Meanwhile, we generate another list, denoted by $\mathcal{L}_v^{'}$, comprising the ARQ distributions on the virtual nodes satisfying the sum constraint, dictated by the running index q_c (see line 13). This way, a list of ARQ distributions for the virtual $N_{\gamma c}$-hop network, denoted by $\mathcal{L}_{final}$, is generated by considering all possible ARQ distributions on the virtual nodes (see line 14 to line 18). In line 16, the symbol $||$ denotes the concatenation operation. Finally, the PDP of the virtual $N_{\gamma c}$-hop network is computed for each of these distributions in $\mathcal{L}_{final}$, before picking the one which results in minimum PDP.

Since the ARQs on the physical nodes is solved using linear system of equations, aided by simple heuristics to satisfy the integer and sum constraints, we point out that the complexity of our approach is dominated by the total number of ARQ distributions that can be allotted to the virtual nodes. A detailed discussion on the complexity of our algorithms is presented in the next section.

4.4 Complexity Analysis and Simulation Results

In this section, we present simulation results to showcase the performance of the pair-wise cooperative scheme, cluster-wise cooperative scheme, and the fully-cumulative models. We compare these schemes in terms of the end-to-end PDP, and also in terms of the computational complexity of the underlying algorithms used to compute the near-optimal ARQ distribution.

4.4.1 Pair-wise Cooperative ARQ Scheme

We present the performance of the pair-wise cooperative ARQ scheme by incorporating both one-pair and two-pair cooperation. For the one-pair cooperation approach, i.e., $|\mathcal{P}| = 1$, we choose a pair of consecutive links wherein the number of unused ARQs of given link is used by the next link in the pair. In Fig. 4.2, we plot the PDP for a 4-hop network with the LOS vector

$\mathbf{c} = [0.6, 0.8, 0.2, 0.5]$ by pairing the last two links. Note that the optimal ARQ distribution for this strategy is of the form $[q_1, q_2, q_3 + q_4, 0]$. To showcase the advantages with respect to the non-cooperative scheme, we plot the PDP for different values of q_{sum} and SNR. For each combination of q_{sum} and SNR, the optimal ARQ distribution is obtained using Algorithm 2 as discussed in Section 4.3.3, wherein the input to the algorithm is a virtual 3-hop network containing one virtual node ($L_v = 1$) and two physical nodes ($L_p = 2$). The plots presented in Fig. 4.2 confirm our theoretical results that the PDP of the network significantly reduces in comparison with the non-cooperative scheme.

For the above mentioned simulation parameters, we plot the PDP in Fig. 4.3 after employing the ARQ distribution obtained by Algorithm 2 as well as the exhaustive search method. Note that the exhaustive search method in this context is the one applied on the virtual 3-hop network along with the constraint that the virtual node is given at least two ARQs. The plots show that our list generation algorithm provides near-optimal solution for all the considered values of q_{sum}. Finally, the complexity of the list generation algorithm for the one-pair cooperative ARQ scheme is also presented in Fig. 4.4 to showcase the difference in the list size with respect to the exhaustive search method.

Similar to the one-pair scheme, we present the performance of a two-pair cooperative ARQ scheme, wherein two pairs of non-contiguous links are formed thereby resulting in two virtual links. To showcase the results, we use $N = 5$ with the LOS vector $\mathbf{c} = [0.4, 0.6, 0.8, 0.1, 0.3]$, and then pair the first two and the last two links to form a virtual 3-hop network. In other words, the transmitter of the second link has the knowledge of ARQs given to the source node, and the transmitter of the fifth link has the knowledge of the ARQs allotted to that of the fourth link. For this configuration, we present the PDP of the network in Fig. 4.5 against different values of q_{sum} and SNR. To compute the PDP for each combination of q_{sum} and SNR, we use the proposed list-based algorithm to generate the ARQ distribution. The plots confirm significant improvement in the PDP as we transition from the non-cooperative scheme to the two-pair

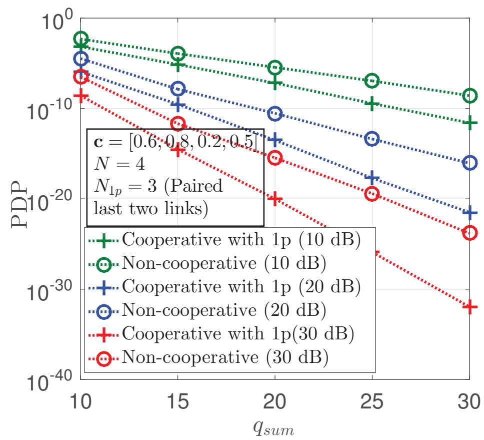

Figure 4.2: Plots depicting the improvement in the PDP when using a one-pair (1p) cooperative scheme.

cooperative scheme on this network. We also observe that the two-pair scheme outperforms the one-pair scheme, wherein only the last two links are paired to cooperative use the residual ARQs. This behavior is attributed to the fact that packet drop probability at the second link reduces owing to the use of residual number of unused ARQs at the source node. We highlight that the optimal ARQ distribution for the one-pair and the two-pair fully-cumulative scheme are of the form $[q_1 + q_2, 0, q_3, q_4 + q_5, 0]$, and $[q_1, q_2, q_3, q_4 + q_5, 0]$, respectively.

In the rest of this section, we quantify the computational complexity of the proposed list-based algorithm that is used to search the ARQ distribution for the pair-wise cooperative ARQ scheme. In the context of the non-cooperative ARQ scheme, it is well known that computational complexity of finding the optimal ARQ distribution through exhaustive search is $\binom{q_{sum}-1}{N-1}$. However, in the case of the pair-wise cooperative scheme, the computational complexity of the exhaustive search method is upper bounded by $\binom{q_{sum}-1}{N_{\gamma p}-1}$, where $N_{\gamma p}$ is the number of nodes in the virtual network such that $|\mathcal{P}| = \gamma$. To present the complexity numbers of the one-pair cooperative ARQ scheme, we plot the list size generated by our algorithm in Fig. 4.4 as a

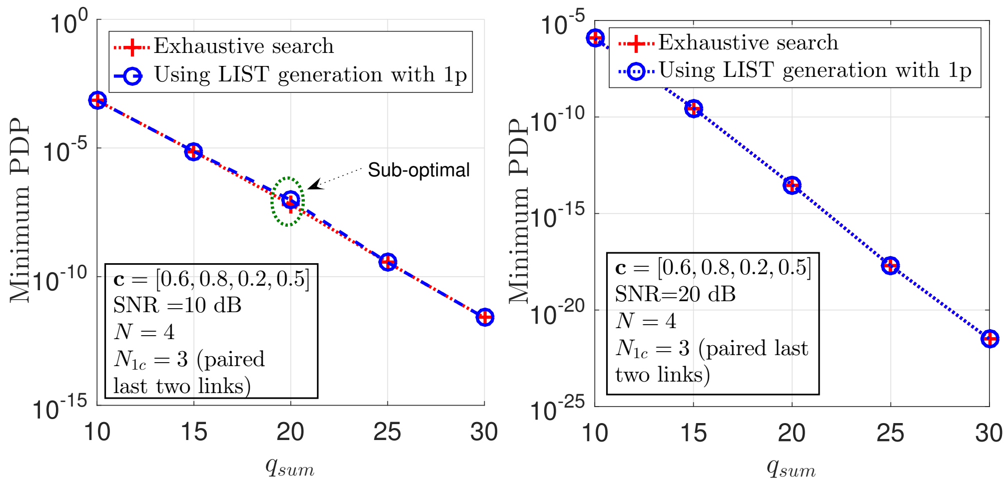

Figure 4.3: Plots depicting the PDP of the one-pair (1p) scheme when the ARQ distribution is computed using exhaustive search and the proposed algorithm.

function of q_{sum} for $N = 4$. The plots show that the list size is significantly shorter than the exhaustive search method. Note that in this context the exhaustive search method is on the virtual network containing three nodes with the constraint that each virtual node must be given at least two ARQs. Similar results on the complexity can also be presented for the case of two pairs with $N = 5$. In such a case, the list size is equal to the complexity of the exhaustive search on a three-hop network with the constraint that each node is given a minimum of two ARQs.

4.4.2 Cluster based Cooperative ARQ Scheme

We present the performance of the cluster-based cooperative scheme, and compare its PDP with that of the non-cooperative strategy. In this context, cluster refers to a set of more than two consecutive links that share the residual number of ARQs with the next node in the cluster. To showcase the results on PDP, we consider a 6-hop network with the LOS vector $\mathbf{c} = [0.8, 0.4, 0.7, 0.4, 0.2, 0.5]$. First, we form a single cluster containing three nodes by combining

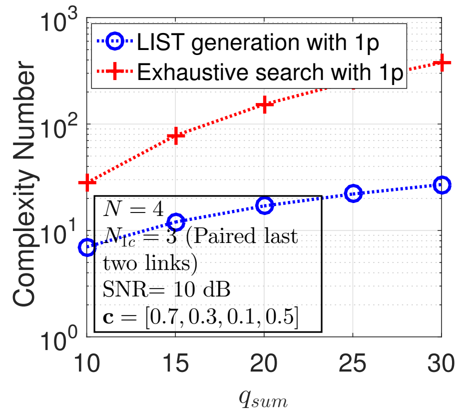

Figure 4.4: Complexity of the one-pair (1p) cooperative scheme.

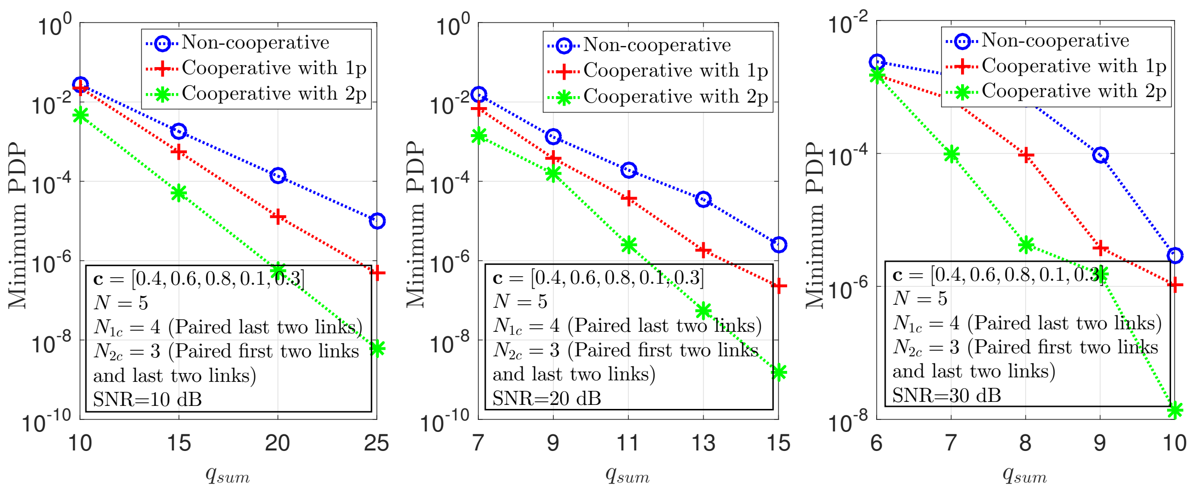

Figure 4.5: Plots depicting the improvement in the PDP when using a two-pair (2p) cooperative scheme.

the last three links of the network. As a result, the optimal ARQ distribution for this strategy is of the form $[q_1, q_2, q_3, q_4 + q_5 + q_6, 0, 0]$. Similarly, we form two disjoint clusters containing three nodes within each cluster, wherein the first cluster is formed by combining the first three

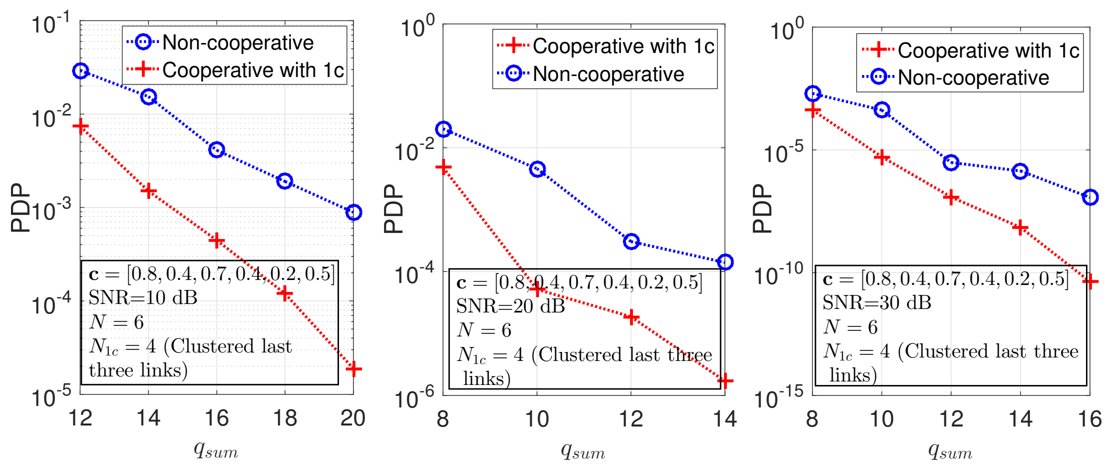

Figure 4.6: Plots depicting the improvement in the PDP when using a cooperative strategy with one-cluster (1c) scheme by combining 3 links.

links of the network, whereas the second cluster is formed by combining the last three links of the network. As a result, the optimal ARQ distribution for this strategy is of the form $[q_1 + q_2 + q_3, 0, 0, q_4 + q_5 + q_6, 0, 0]$. In Fig. 4.6, we compare the PDP of the one-cluster cooperative scheme against the non-cooperative strategy for various values of q_{sum} and SNR. The plots confirm that the one-cluster strategy provides significant improvement in the PDP. Furthermore, the PDP of the two-cluster cooperative scheme is also presented in Fig. 4.8, which displays further improvement over the non-cooperative strategy.

Similar to the pair-wise cooperative strategy, we quantify the computational complexity of our list-based algorithm in order to find the near-optimal solution for the cluster-wise cooperative ARQ scheme. Based on the nature of the proposed list-based algorithm, it is clear that the complexity is dominated by the number of virtual nodes that arises due to the clustering process. In particular, the computational complexity of the cluster-wise scheme is upper bounded by $\binom{L_p}{\lfloor \frac{L_p}{2} \rfloor} \sum_{i=tL_v}^{q_{ssum}-L_p} \binom{i-1}{L_v-1}$, where L_v is number of virtual nodes, L_p is number of physical nodes

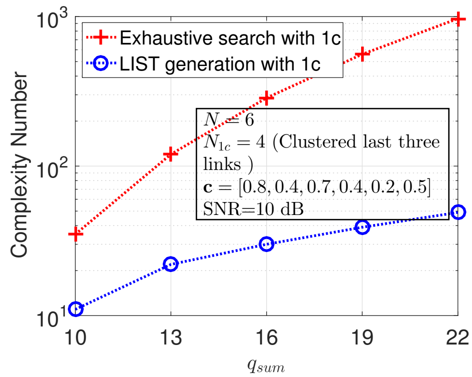

Figure 4.7: Complexity of the cooperative strategy with one-cluster (1c) scheme by combining 3 links.

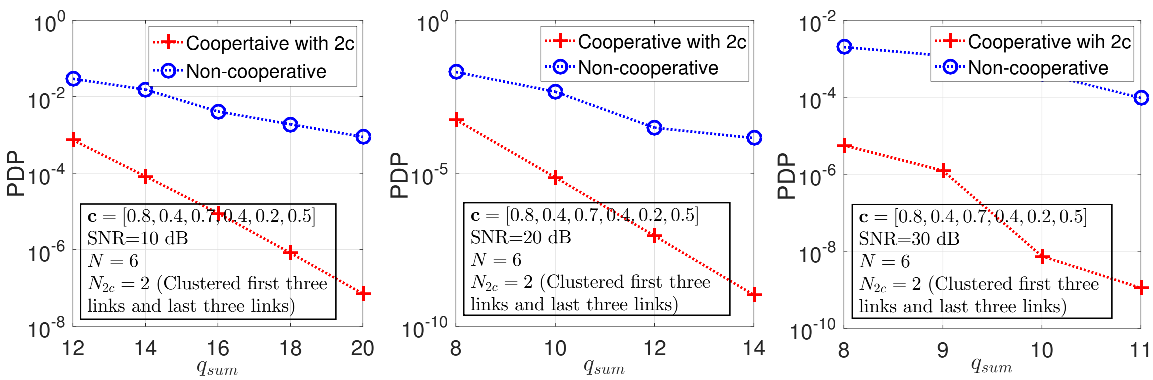

Figure 4.8: Plots depicting the improvement in the PDP when using a cooperative strategy with two-cluster (2c) scheme wherein each cluster is formed by combining 3 links.

that do not cooperate, t is the number of nodes in each cluster. To present the complexity numbers, we plot the list size generated by our algorithm in Fig. 4.7 as a function of q_{sum}. The plots show that the list size is significantly shorter than the exhaustive search method. Note that the exhaustive search method is on the virtual network containing four nodes with the constraint that each virtual node must be given at least $t = 3$ ARQs. In general, the list size of the exhaustive search method on the virtual network is upper bounded by $\binom{q_{sum}-1}{N_{\gamma c}-1}$, where $N_{\gamma c}$ is

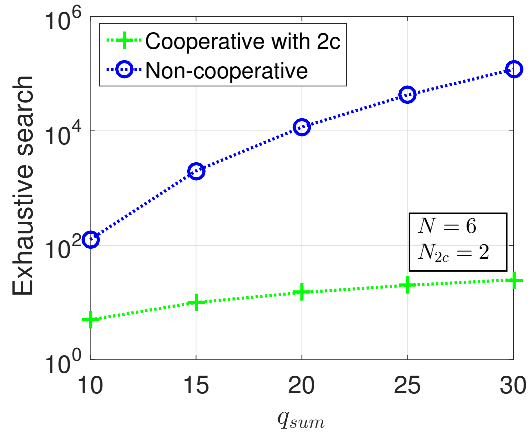

Figure 4.9: Complexity of the cooperative strategy with with two-cluster (2c) scheme wherein each cluster is formed by combining 3 links.

the number of nodes in the virtual network such that $|\mathcal{C}_t| = \gamma$. Similar results on the complexity are also presented in Fig. 4.9 for the case of two clusters. In this case, the list size is equal to the complexity of the exhaustive search on a two-hop network with the constraint that each node is given a minimum of $t = 3$ ARQs.

4.4.3 Fully-Cumulative ARQ Scheme

We present the PDP of the fully-cumulative scheme for a 4-hop network with LOS vector $\mathbf{c} = [0.7, 0.3, 0.1, 0.5]$. Owing to the underlying protocol, each node in the network has the knowledge of the number of ARQs of the preceding node. In addition, the packet structure contains a dedicated portion to carry the residual number of ARQs unused by all the preceding nodes. The plots on the PDP of the fully-cumulative scheme are shown in Fig. 4.10 for various values of q_{sum} and SNR. Furthermore, the PDP of the non-cooperative scheme, one-pair cooperative scheme (by pairing the last two links), and the one-cluster cooperative scheme (by combining the last three links to form a cluster) are also presented in the same figure. Note that the

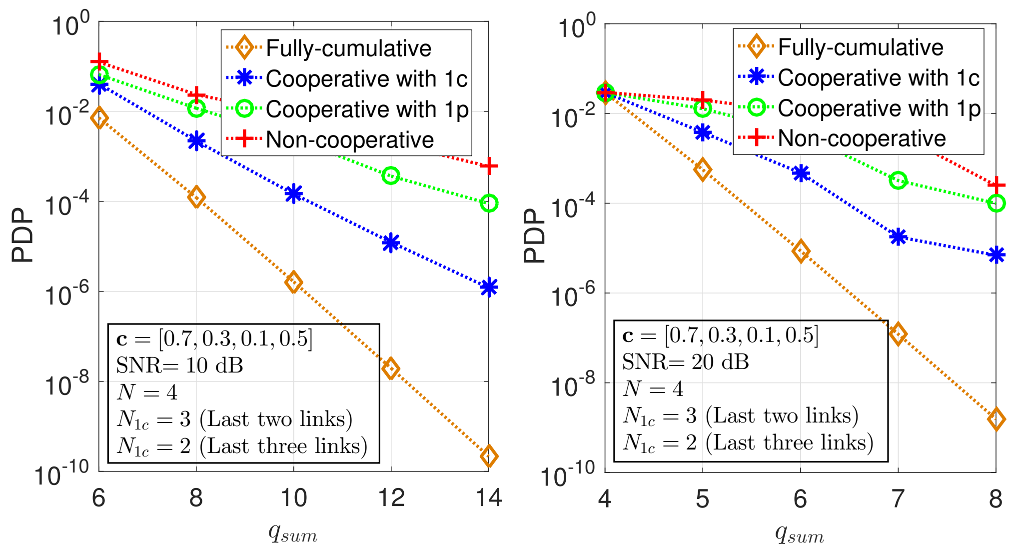

Figure 4.10: Plots depicting the improvement in the PDP when using a fully-cumulative strategy wherein one cluster is formed by combining all the links in the network.

optimal distribution for the fully-cumulative scheme, the pair-wise scheme and the cluster-wise scheme are of the form $[q_{sum}, 0, 0, 0]$, $[q_1, q_2, q_3 + q_4, 0]$, and $[q_1, q_2 + q_3 + q_4, 0, 0]$, respectively. The plots confirm that the fully-cumulative ARQ scheme is best in terms of minimizing the PDP among the schemes under comparison. However, this scheme is accompanied by a marginal increase in the communication-overhead owing to the use of a counter to carry the residual number of ARQs.

4.4.4 Delay Analysis of Cooperative ARQ Strategies

We recall that the cluster-wise cooperative scheme and the fully-cumulative scheme require a counter in the packet in order to convey residual number of ARQs of the preceding nodes in the chain. In terms of communication-overhead, the additional space for conveying residual ARQs is negligible as the counter takes at most $\log_2(q_{sum})$ bits. In terms of delay-overhead, the cluster-wise cooperative scheme and the fully-cumulative scheme are such that each node in the cluster (or the chain) needs to update the counter only once; this is when the packet is successfully decoded. Therefore, there is no additional delay at each node. To capture the delay profiles of all the four cooperative ARQ strategies in Fig. 4.10, we use simulation results to compute the PMF on the delay of the received packets when the schemes are implemented on a 4-hop network with LOS vector $\mathbf{c} = [0.7,\ 0.3,\ 0.1,\ 0.5]$. To generate the results, we use the optimal ARQ distribution with $q_{sum} = 8$, $R = 0.5,$ and 1, and SNR = 10 and 20 dB. With $T = 1$ microseconds denoting the time taken for each packet transmission (including the time taken for ACK/NACK), we fix the deadline for the packet to reach the destination as $q_{sum}T = 8$ microseconds. The PMFs that are generated using simulations are presented at the top of Fig. 4.11, which shows that the average delay offered by all the four schemes are less than the deadline. We also note that the fully-cumulative scheme incurs marginal increase in the average delay (at most 2%) as it consumes more number of re-transmissions to minimize the PDP.

For the rest of this section, we study the delay-overhead introduced by the cooperative ARQ strategies in Fig. 4.10 especially capturing the overhead in updating the counter at every node. Suppose that the counter portion of the packet is encrypted by every node to maintain confidentiality from eavesdropping. Then when the packet is successfully decoded at a relay node, it has to apply a suitable crypto-primitive to decrypt the counter portion of the packet. As a result, an additional delay is introduced on the packet. Given that this delay depends on the crypto-primitive architecture, we introduce a computation delay of T_c microseconds. Since the delay introduced on the packet for each transmission is $T = 1$ microsecond (including ACK/NACK), we study the effect of crypto-primitives by choosing $T_c = \alpha T$, where $\alpha = 0, 0.1, 0.5,$ and 1. If the effect of α is not considered when designing q_{sum}, then there is a non-zero probability that some packets may reach the destination beyond the deadline. Therefore, instead of solely focusing on PDP, we introduce a new metric referred to as probability of deadline violation (PDV), which is defined as the probability that the packets either get dropped in the network or do not reach the destination before the deadline. At the bottom of Fig. 4.11, we plot the PDV of the four schemes as a function of α for a 4-hop network with different parameters as chosen for the left side of Fig. 4.11. Since $q_{sum} = 8$, the deadline for packets to reach the destination is 8 microseconds. The plots confirm that: (i) The PDV of the non-cooperative strategy and the pair-wise cooperative strategy do not change with α since counter is not used in the packet. (ii) The PDV of the cluster-wise cooperative scheme and the fully-cumulative ARQ scheme increases with increasing values of α; this is because some of the nodes make use of the counter in the packet. (iii) The worst hit is the fully-cumulative scheme since every node has to open the counter, thereby adding significant delay to the packet. Overall, the simulation results at the bottom of Fig. 4.11 show that cooperative strategies outperform the non-cooperative strategy when the overhead in updating the counter is small. However, as the computational overhead in updating the counter increases, the performance of the cluster-wise and fully-cumulative strategies degrade.

4.5 Summary

In this chapter, we have studied several cooperative ARQ protocols to facilitate reliable and low-latency communication of messages over a line-of-sight dominated multi-hop network. We have characterized the PDP of these cooperative protocols, and have addressed the problem of minimizing their PDP under the sum constraint on the number of ARQs allotted to the nodes in the network. In this context, the sum constraint captures the bound on the time taken for the total number re-transmissions in the multi-hop network. We highlight that the fully-cumulative ARQ scheme offers the lowest PDP among the other schemes. However, this scheme requires every node to have the knowledge of the ARQs allotted to its preceding node, and also requires a counter in the packet, which is used to embed the number of unused ARQs by all preceding nodes.

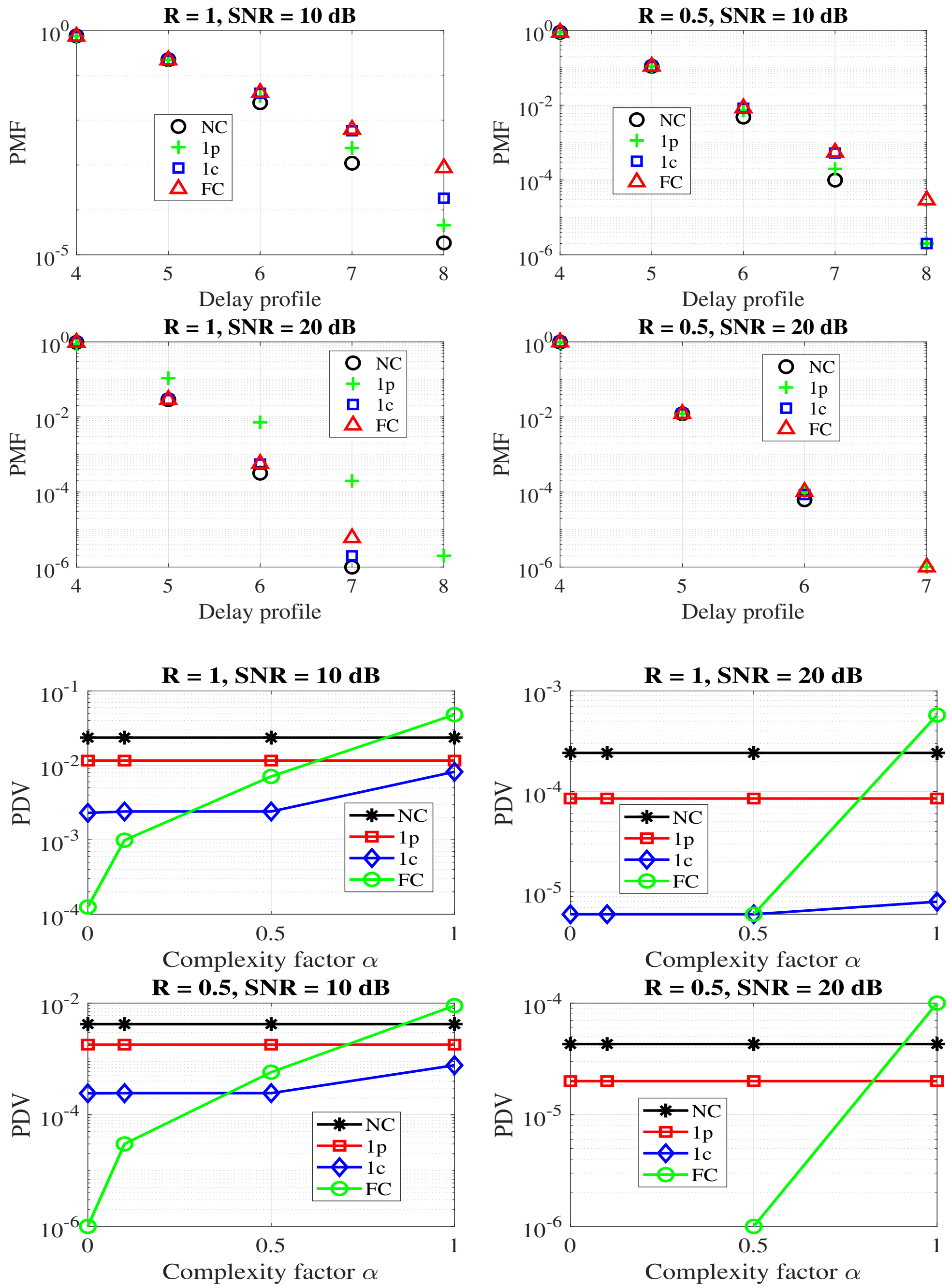

Figure 4.11: Simulation results using a 4-hop network with $\mathbf{c} = [0.7,\ 0.3,\ 0.1,\ 0.5]$, $q_{sum} = 8$ at various values of R and SNR. At the top: Delay profiles of the proposed schemes when $T = 1$ microsecond is the time taken for packet transmission each time in the network. At the bottom: plots on PDV as a function of α, where α captures the complexity parameter to update the counter. The legends “NC", “1p", “1c" and “FC" represent the non-cooperative, one-pair cooperative, one-cluster cooperative, and fully-cumulative strategies, respectively.

In the fully-cumulative scheme, if the counter is not used, then this will give rise to a new form of cooperative ARQ scheme, wherein every node can use the residual ARQs only from the preceding node, but not the residual ARQs from all the previous nodes. This new form of the cooperative strategy will be discussed in the next chapter.

Part III

One-Hop Listening ARQ Schemes for Achieving URLLC over Multi-Hop Networks

Chapter 5

Semi-Cumulative ARQ Sharing Strategies in Multi-Hop Networks

5.1 Introduction

Although the framework of non-cooperative strategy attempts to reduce the PDP by imposing latency-constraints in the form of an upper bound on the total number of ARQs, we observe that the non-cooperative strategy has a fundamental limit with which the PDP can be minimized. Also, if the counter is not employed in the fully-cumulative scheme, a new cooperative ARQ scheme will emerge in which each node can only use the residual ARQs from the preceding node and not the residual ARQs from all previous nodes. Motivated by these observations, in this chapter, we explore whether an ARQ based DF strategy can be proposed with cooperation among the relay nodes to further increase the reliability with no relaxations on the latency constraints on the packets. Towards that direction, we make the following contributions in this chapter [1]:

1) Under the class of ARQ based DF strategies for multi-hop networks, we propose a cooperative

[1] Part of the results presented in this chapter are available in publications [34, 35]

ARQ model, referred to as the semi-cumulative ARQ based DF strategy, wherein each transmitter in the network has the knowledge of the number of ARQs allotted to its preceding node in addition to the number of ARQs allotted to itself. We show that this cooperative framework assists a relay node in borrowing unused ARQs from the preceding node thereby increasing the reliability of the packets with no compromise in the latency constraints. We highlight that the benefits offered by the proposed cooperative strategy does not accompany additional overheads since every node only needs to count the number of failed attempts when decoding the packet received from the preceding node (see Section 5.2).

2) For the proposed semi-cumulative ARQ based DF strategy, we address the problem of computing the optimal ARQ distribution that minimizes the PDP subject to a sum constraint on the total number of ARQs. Towards that direction, first, we use the Fibonacci series to derive closed-form expressions on the PDP of the semi-cumulative strategy for arbitrary N and q_{sum}, and then formally prove that the proposed strategy outperforms the non-cooperative strategy (as discussed in Chapter 3). We highlight that the task of deriving the PDP expression is a non-trivial contribution owing to the memory property introduced by the idea of borrowing unused ARQs of the preceding nodes (see Section 5.3).

3) To solve the PDP minimization problem, first, we prove that the problem of computing the optimal ARQ distribution for an N-hop network can be reduced to the problem of computing the optimal ARQ distribution for an $(N-2)$-hop network, thereby showcasing a substantial reduction in the complexity (see Section 5.4). Subsequently, generalizing the reduction approach, we propose two classes of low-complexity algorithms that can be used to compute near-optimal ARQ distributions for any N-hop network (see Section 5.5). We also present extensive simulation results to showcase the efficacy of the proposed algorithms in terms of PDP reduction as well as computational complexity (see Section 5.6).

5.2 Semi-Cumulative Multi-Hop Network

Consider an N-hop network, as shown in Fig. 5.1, wherein a source node intends to communicate its messages to a destination through a set of $N-1$ relay nodes that operate using an ARQ based DF strategy. In this model, the multi-hop network is characterized by the LOS vector $\mathbf{c} = \{c_1, c_2, \ldots, c_N\}$ and the ARQ distribution $\mathbf{q} = \{q_1, q_2, \ldots, q_N\}$, such that $c_i \in [0, 1]$ represents the LOS component of the fading channel of the i-th hop and q_i represents the number of re-transmissions allotted to the transmitter of the i-th hop, for $1 \leq i \leq N$. Formally, let

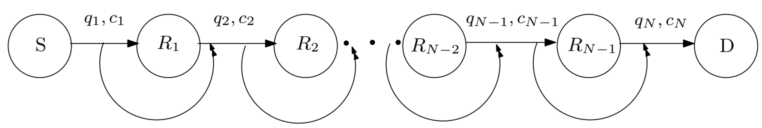

Figure 5.1: Illustration of an N-hop semi-cumulative scheme with channels dominated by LOS components.

$\mathcal{S} \subset \mathbb{C}^K$ denote the channel code employed at the source node of rate R bits per channel use, i.e., $R = \frac{1}{K}\log_2(|\mathcal{S}|)$. Let $\mathbf{x} \in \mathcal{S}$ denote the packet (traditionally referred to as a codeword) transmitted over the multi-hop network such that $\frac{1}{K}\mathbb{E}[|\mathbf{x}|^2] = 1$. When $\mathbf{x}$ is transmitted over the i-th link, for $1 \leq i \leq N$, the corresponding received signal after K channel uses is given by $\mathbf{y}_i = h_i\mathbf{x} + \mathbf{n}_i \in \mathbb{C}^K$, where h_i is a quasi-static Ricean fading channel given by $h_i = \sqrt{\frac{c_i}{2}}(1+\iota) + \sqrt{\frac{(1-c_i)}{2}}g_i$, such that $\iota = \sqrt{-1}$, g_i is distributed as $\mathcal{CN}(0,1)$, $\mathbf{n}_i$ is the additive white Gaussian noise (AWGN) vector at the receiver of the i-th link, distributed $\mathcal{CN}(0, \sigma^2\mathbf{I}_K)$. We assume that the receiver of each link has perfect knowledge of its channel, however, there is no knowledge about the channel at the transmitter side. Furthermore, it is possible that the instantaneous mutual information of the channel may not support the transmission rate R as the channel realization h_i is random and does not change over K channel uses. Therefore, if the mutual information of the channel is less than the transmission rate, then the receiver will not able to decode the packet correctly, and this event is referred to as the outage event whose probability is given by [2]

$$P_i = \text{Prob}\Big(R > \log_2(1 + |h_i|^2\alpha)\Big) = \text{F}_i\left(\frac{2^R - 1}{\alpha}\right), \tag{5.2}$$

where $\alpha = \frac{1}{\sigma^2}$ is the average signal-to-noise-ratio (SNR) of the i-th link, $\text{F}_i(x)$ is the cumulative distribution function of $|h_i|^2$, defined as $\text{F}_i\left(\frac{2^R-1}{\alpha}\right) = 1 - Q_1\left(\sqrt{\frac{2c_i}{(1-c_i)}}, \sqrt{\frac{2(2^R-1)}{\alpha(1-c_i)}}\right)$, such that $Q_1(\cdot,\cdot)$ is the first-order Marcum-Q function. Owing to the above mentioned outage events, a transmitter of the i-th hop is allotted q_i number of re-transmissions in order to successfully forward the packet to the next node in the network. Despite using this ARQ based DF strategy, if a transmitter is unable to transmit the packet within q_i number of ARQs, then the packet is said to be dropped in the network. Since the packet can be dropped at any hop of the network, we use PDP as the reliability metric of interest, which is defined as the fraction of packets that do not reach the destination.

To achieve higher reliability than the ARQ based DF protocol in [29], we propose the semi-cumulative model, as shown in Fig. 5.1, wherein every intermediate relay node can use

[2]In the regime of asymptotic block-lengths, i.e., $K \to \infty$, this event completely characterizes the decoding error probability of the i-th hop, denoted by P_i. In particular, with asymptotic block-lengths, P_i would be $\text{Prob}\Big(R > \log_2(1 + |h_i|^2\alpha)\Big)$, where $\alpha = \frac{1}{\sigma^2}$ is the average signal-to-noise-ratio (SNR) of the i-th link. However, in the regime of non-asymptotic block-lengths, i.e., when $K < \infty$, the corresponding non-asymptotic decoding error probability, as derived in [28], can be computed as

$$P_i = \int_{\mathbb{R}+} Q\left(\sqrt{\frac{K}{V(\Gamma_i)}}(C(\Gamma_i) - R)\right) f_{\Gamma_i}(\Gamma_i)d\Gamma_i, \tag{5.1}$$

where $\Gamma_i = |h_i|^2\alpha$ is the instantaneous SNR of the wireless channel in the i-th hop, $f_{\Gamma_i}(.)$ is the probability density function of the instantaneous SNR Γ_i, $C(\Gamma_i) = \log_2(1+\Gamma_i)$ is the Shannon channel capacity, $V(\Gamma_i) = (\Gamma_i/2)(\Gamma_i + 2)/((\Gamma_i + 1)^2)\log_2^2 e$ is the back-off factor for finite block-length, and finally, $Q(x) = (1/\sqrt{2\pi})\int_x^\infty e^{-\frac{u^2}{2}}du$. Note that the expression in (5.1), which is derived using the achievable rates in [32], is applicable for any K, and it collapses to the asymptotic outage probability expression as a special case when $K \to \infty$.

residual ARQs unused by its previous node in the chain. This simple idea stems from the fact that although q_i re-transmissions are allotted to a transmitter, the actual number of re-transmissions can be less than q_i owing to the stochastic nature of the wireless channel. To facilitate this, we assume that every node has the knowledge of the number of ARQs given to its preceding node in addition to the ARQs allotted to itself. Since a given node can use unused ARQs from its previous node, the total number of ARQs used by it can be more than the number of ARQs allotted to it. As a result, the next node in the chain, despite knowing the number of ARQs allotted to its preceding node, does not know how long to wait for the successful transmission of the packet. To fix this, each intermediate node will have to wait for a fixed amount of time to receive the packet from its previous node beyond which the packet is said to be dropped in the network. Unlike the non-cooperative strategy of [29], in this method, an intermediate node can get more re-transmissions than the number of allotted to it just by listening to the number of failed attempts of the preceding node. Although each relay node is allowed multiple transmissions (including the number of unused ARQs of its preceding node) to communicate the packet to the next node, there is a non-zero probability with the packet is dropped in the network since the sum of the ARQs allotted to all the nodes in the network is bounded, i.e., $\sum_{i=1}^{N} q_i = q_{sum}$. Henceforth, we denote the PDP of the semi-cumulative ARQ based DF strategy by $pdp_{sc,N}$, where sc in the subscript highlights the semi-cumulative scheme, and N denotes the number of hops in the network. Thus, in order to provide reliability along with low-latency constraint on the packets, in this chapter, we propose to solve Problem 5.1, as shown below.

Problem 5.1. *For an N-hop network with a given LOS vector* $\mathbf{c}$*, a given SNR $\alpha = \frac{1}{\sigma^2}$, and a given q_{sum}, solve*

$$q_1^*, q_2^*, \dots q_N^* = \arg \min_{q_1, q_2, \dots q_N} pdp_{sc,N}$$

subject to $q_i \in \mathbb{Z}_+ \; \forall i, q_i \geq 0$ *for* $i \geq 2, q_{i+1} \neq 0$ *if* $q_i = 0$ *or* 1, *and* $\sum_{i=1}^{N} q_i = q_{sum}$, *where* $i \in \{1, 2, \ldots N\}$.

Towards solving Problem 5.1, in the next section, we derive an expression for the PDP of the semi-cumulative scheme, and then formally prove that the semi-cumulative scheme outperforms the non-cooperative ARQ strategy in [29].

5.3 PDP Expression of Semi-Cumulative Scheme

Theorem 5.1. *The PDP expression for an* N*-hop semi-cumulative scheme is given by*

$$pdp_{sc,N} = P_1^{q_1} + (1 - P_1)P_2^{q_2}F_2 + \ldots + \prod_{i=1}^{N-1}(1 - P_i)P_N^{q_N}F_N, \tag{5.3}$$

where F_j*, for* $2 \leq j \leq N$*, is a function of* $P_1, P_2, \ldots, P_{j-1}$ *(as given in (5.2)) and* $q_1, q_2, \ldots, q_{j-1}$ *that can be computed using Fibonacci series.*

Proof. For a 2-hop network, the PDP expression, denoted by $pdp_{sc,2}$, can be written as $pdp_{sc,2} = pdp_{sc,1h} + pdp_{sc,2h}$, where $pdp_{sc,1h} = P_1^{q_1}$ and $pdp_{sc,2h} = (1 - P_1)P_2^{q_2}\left(\sum_{i=1}^{q_1} P_1^{q_1-i}P_2^{i-1}\right)$ represent the probability that the packet is dropped at the j-th hop for $j \in \{1, 2\}$. From the definition of the semi-cumulative scheme, the first node does not have any preceding node to borrow ARQs whereas the second node can borrow unused ARQs from the first node. In the expression for $pdp_{sc,2h}$, the term $\sum_{i=1}^{q_1} P_1^{q_1-i}P_2^{i-1}$, henceforth referred to as $\beta_E^{(1,2)}$, captures the probability that the packet is dropped in the second link despite using the residual ARQs from the first link. These types of terms are known as external borrowing ARQ terms. In short, we can rewrite $pdp_{sc,2}$ as $pdp_{sc,2} = P_1^{q_1} + (1 - P_1)P_2^{q_2}F_2$, where $F_2 = \beta_E^{(1,2)}$. Similarly, the PDP expression for a 3-hop network can be written as $pdp_{sc,3} = pdp_{sc,1h} + pdp_{sc,2h} + pdp_{sc,3h}$, where

$$pdp_{sc,3h} = (1 - P_1)(1 - P_2)P_3^{q_3}\left(\left(\sum_{i=1}^{q_1} P_1^{q_1-i}\right)\sum_{i=1}^{q_2} P_2^{q_2-i}P_3^{i-1} + \sum_{i=1}^{q_1} P_1^{q_1-i}\sum_{k=0}^{i-2} P_2^{q_2+k}\right),$$

captures the probability that the packet is dropped at the third link. In the above expression, the term $\left(\sum_{i=1}^{q_1} P_1^{q_1-i}\right)\left(\sum_{i=1}^{q_2} P_2^{q_2-i} P_3^{i-1}\right)$ represents the probability that the second node passes the packet without using the residual ARQs from the first node, whereas the third node makes use of all the residual ARQs of the second node. Here, $\beta_N^1 \triangleq \left(\sum_{i=1}^{q_1} P_1^{q_1-i}\right)$ is referred to as the no borrowing term from the first node, whereas $\beta_E^{2,3} \triangleq \sum_{i=1}^{q_2} P_2^{q_2-i} P_3^{i-1}$ is the external borrowing term as defined in the two-hop case. Along the same lines, the term $\beta_I^{1,2} \triangleq \sum_{i=1}^{q_1} P_1^{q_1-i} \sum_{k=0}^{i-2} P_2^{q_2+k}$ represents the probability that the second node passes the packet to the third node after using all the residual ARQs of the first node. We refer to this term as the internally borrowing term. In short, the PDP expression for the three-hop can be written as

$$pdp_{sc,3} = P_1^{q_1} + (1 - P_1)P_2^{q_2} F_2 + (1 - P_1)(1 - P_2)P_3^{q_3} F_3,$$

where $F_2 = \beta_E^{(1,2)}$ and $F_3 = \beta_N^{(1)} \beta_E^{(2,3)} + \beta_I^{(1,2)}$. In general, the PDP expression for an N-hop network can be written as $pdp_{sc,N} = pdp_{sc,1h} + pdp_{sc,2h} + \ldots + pdp_{sc,Nh}$, where $pdp_{sc,Nh}$ captures the probability that the packet is dropped in the last link. This implies that the packet has survived through the first set of $N-1$ nodes. Among the preceding $N-1$ nodes, a node passes the packet to its next node either by using a number of attempts within its allotted ARQs without having to use the residual ARQs of the preceding node, or by using a number of attempts exceeding the ARQs allotted to it, however, by using the residual ARQs of the preceding node. We shall denote these two possible ways as state 0 and state 1, respectively. This implies that the packet can reach the penultimate node wherein all possible states taken by the first $N-1$ nodes comes from the space $\{0, 1\}^{N-1}$. Although the number of such sequences is at most 2^{N-1}, not all those sequence states are valid in our case. This is because the first node in the network cannot take state 1 because it has no preceding node to borrow the ARQs. Similarly, given that the underlying protocol is of semi-cumulative nature, node-j, for $j > 2$, cannot forward the packet in state 1 if node-$(j-1)$ has already forwarded the packet in state 1. This is because node-$(j-1)$ has used more ARQs allotted to it, and therefore, node-j does not have any residual ARQs to its advantage. This implies that among all possible sequences $\{0,1\}^{N-1}$, we cannot have consecutive ones. This further implies that the total number of ways in which the packet arrives at the penultimate node is equal to the number of binary sequences of length $N-2$ that have no consecutive ones. Henceforth, let us refer to this number as $FB(N-2)$. Let us call the set of such binary sequences as $\mathcal{FB}_{N-2}$. In order to use it to obtain the ways in which packets survive till the penultimate node, we pick $\mathbf{x} \in \mathcal{FB}_{N-2}$, and then obtain a new sequence of length N as $\mathbf{x}' = [0 \; \mathbf{x} \; 0]$ if the last digit of $\mathbf{x}$ is 1. Similarly, we have $\mathbf{x}' = [0 \; \mathbf{x} \; 1]$ if the last digit of $\mathbf{x}$ is 0. The above changes are applicable because the first bit of $\mathbf{x}'$ has to be zero because of the first node, and moreover, if the $(N-1)$-th position is zero, that means the N-th node can make use of its residual ARQs. However, on the other hand, if the $(N-1)$-th position is one, then the only way the N-th node can drop the packet is by consuming all its ARQs. When the sequence $\mathbf{x}'$ is of the form $\mathbf{b} = [b_1 \; b_2 \; b_3 \ldots b_N]$, we can write the corresponding probability of survival as follows. Because of no consecutive ones, let us look for sub-sequences of '01' in the sequence $\mathbf{b}$. If the pattern '01' is found in the j-th and $(j+1)$-th positions for $j < N-2$, then use the expression $\beta_I^{j,j+1}$. If the pattern '01' is found in the $(N-1)$-th and N-th terms, then we use the expression $\beta_E^{N-1,N}$. Once the above expressions are placed, the rest of the zeros in the sequence are replaced by the term β_N^j if the zero is found in the j-th position. Finally, we multiply these terms to get one expression. Once a sequence is replaced by the expression, we add up all the terms to obtain F_N, which corresponds to a total of $FB(N-2)$ terms. Thus, we have $pdp_{sc,Nh} = \prod_{i=1}^{N-1}(1-P_i)P_N^{q_N}F_N$. This completes the proof. $\square$

Corollary 5.1. *Each term in F_N is a product of terms of the form $\beta_I^{j,j+1}$, $\beta_E^{N-1,N}$ and β_N^j, where $\beta_E^{N-1,N}$ can occur at most once, β_N^j can occur at most $N-2$ times, and $\beta_I^{j,j+1}$ can occur at most $\lceil \frac{N}{2} \rceil$ times.*

Lemma 5.1. *With $P = \max_{1 \leq i \leq N} P_i$ and when $P < \frac{1}{2}$, we have*

$$\beta_E^{N-1,N} < 2P \left(\sum_{j=1}^{q_{N-1}} P_{N-1}^{q_{N-1}-j} \right), \tag{5.4}$$

when $q_{N-1} > 1$. Similarly, we have

$$\beta_I^{j,j+1} < P^2 \left(\sum_{\alpha=1}^{q_j} P_j^{q_j-\alpha} \right) \left(\sum_{\gamma=1}^{q_{j+1}} P_{j+1}^{q_{j+1}-\gamma} \right), \tag{5.5}$$

when $q_{j+1} > 1$.

Proof. From the definition, we have $\beta_E^{N-1,N} = \sum_{i=1}^{q_{N-1}} P_{N-1}^{q_{N-1}-i} P_N^{i-1}$. Since $P = \max_i P_i$, we can upper bound it as $\beta_E^{N-1,N} < \sum_{i=1}^{q_{N-1}} P^{q_{N-1}-1} = q_{N-1} P^{q_{N-1}-1}$. Finally, since $q_{N-1} > 1$, we have $\beta_E^{N-1,N} < 2P$, and therefore (5.4) also holds good. Similarly, from the definition, we have $\beta_I^{j,j+1} = \sum_{\alpha=1}^{q_j} P_j^{q_j-\alpha} \sum_{\gamma=0}^{\alpha-2} P_{j+1}^{q_{j+1}+\gamma}$. Using $P = \max_i P_i$, we have $\sum_{\gamma=0}^{\alpha-2} P_{j+1}^{q_{j+1}+\gamma} \leq \sum_{\gamma=0}^{\alpha-2} P^{q_{j+1}+\gamma}$. Furthermore, we have $\sum_{\gamma=0}^{\alpha-2} P_{j+1}^{\gamma} \leq \sum_{\gamma=0}^{\infty} P^{\gamma} < \frac{1}{1-P} < 2$, wherein the last inequality holds since $P < \frac{1}{2}$. These inequalities imply that $\beta_I^{j,j+1} < \left(\sum_{\alpha=1}^{q_j} P_j^{q_j-\alpha} \right) 2P^{q_{j+1}}$. Therefore, when $q_{j+1} > 1$, we have $\beta_I^{j,j+1} < \left(\sum_{\alpha=1}^{q_j} P_j^{q_j-\alpha} \right) 2P^2$, and thus (5.5) also holds good. This completes the proof. □

Theorem 5.2. *For a given $\mathbf{q} = [q_1, q_2, \ldots, q_N]$, at high SNR values, the PDP of the semi-cumulative scheme is upper bounded by the PDP of non-cooperative scheme.*

Proof. We will prove this theorem by using the method of induction. For $N = 2$, the PDP expression for the non-cooperative scheme is $pdp_{nc,2} = P_1^{q_1} + (1 - P_1^{q_1}) P_2^{q_2}$. Similarly, the PDP expression for the semi-cumulative scheme is $pdp_{sc,2} = P_1^{q_1} + (1 - P_1) \left(\sum_{i=1}^{q_1} P_1^{q_1-i} P_2^{q_2+i-1} \right)$. When $q_1 > 1$, note that $P_2^{i-1} < 1$ for $i = 2, 3, \ldots, q_1$, and therefore, we have $pdp_{sc,2} < pdp_{nc,2}$. This completes the proof for $N = 2$. Assuming that the statement of the theorem is true for any k-hop network, we will prove the result for a $(k+1)$-network. The PDP expression of the semi-cumulative scheme for $(k+1)$-hop network is $pdp_{sc,k+1} = pdp_{sc,k} + F_{k+1} P_{k+1}^{q_{k+1}} \prod_{i=1}^{k} (1 - P_i)$, where $pdp_{sc,k}$ is the PDP of the semi-cumulative scheme for the k-hop network. From induction, we have $pdp_{sc,k} < pdp_{nc,k}$, and therefore, we only need to prove that $F_{k+1}\prod_{i=1}^{k}(1-P_i) < \prod_{i=1}^{k}(1-P_i^{q_i})$. In other words, we need to prove that $F_{k+1} < \prod_{i=1}^{k}\left(\sum_{j=1}^{q_i} P_i^{q_i-j}\right)$. From Theorem 5.1, we know that F_{k+1} can be written using binary sequences of length $k+1$ such that each sequence does not contain consecutive ones. Furthermore, we have shown that each term of F_{k+1} is a product of several terms of the form $\beta_I^{j,j+1}$, β_N^{j}, for $j < k-1$, and $\beta_E^{k,k+1}$. From Corollary 5.1, it is clear that there is only one term in F_{k+1} which has $\beta_E^{k,k+1}$ appearing once in conjunction with $\prod_{\alpha=1}^{k-1}\beta_N^{\alpha}$, and all other terms either have both $\beta_E^{k,k+1}$ and $\beta_I^{j,j+1}$, or only $\beta_I^{j,j+1}$. In addition, from Lemma 5.1, this implies that F_{k+1} can be upper bounded as $F_{k+1} < \prod_{i=1}^{k}(1-P_i^{q_i})\eta(P)$ where $\eta(P)$ is a polynomial in P of the form $2P + 2P^2(k+1-2) + \rho(P)$ such that $\rho(P)$ is a polynomial in P of degree at least three. At high SNR values, it is clear that $P << 1$, and therefore, we can show that $\eta(P) < 1$. This, in turn, implies that $F_{k+1} < \prod_{i=1}^{k}(1-P_i^{q_i})$. This completes the proof. □

Given that the expression for $pdp_{sc,N}$ is obtained, in the subsequent sections, we propose low-complexity algorithms to solve Problem 5.1 since implementing exhaustive search to find the optimal distribution of ARQs it is not practically feasible.

5.4 Optimal ARQ Distribution of the Semi-Cumulative Scheme

For an N-hop network with $\mathbf{q} = [q_1, q_2, \ldots, q_{N-1}, q_N]$, suppose that the ARQs for the first $N-2$ hops are fixed, and we are interested in computing the optimal values of q_{N-1} and q_N that minimizes the PDP. If we start with $\tilde{\mathbf{q}} = [q_1, q_2, \ldots, 0, q_{N-1}+q_N]$, it may give us a sub-optimal PDP. Therefore, using $\tilde{\mathbf{q}}$, as we keep transferring one ARQ from the last node to the penultimate node, we can expect the PDP to decrease, and then start to increase beyond a certain number of transfers. Towards understanding this transition of PDP, we are interested in understanding the structure of the ARQ distribution when the PDPs of network with $\mathbf{q} = [q_1, q_2, \ldots, q_{N-1}, q_N]$ and $\mathbf{q}^{'} = [q_1, q_2, \ldots, q_{N-1} + 1, q_N - 1]$ are equal. Once we obtain this relation, we can analytically compute the values of q_{N-1} and q_N for a given $q_1, q_2, \ldots, q_{N-2}$, which in turn reduces the search space for computing the optimal ARQ distribution. This result is formally captured in the following theorem.

Theorem 5.3. *To find the optimal distribution of ARQs for a given N-hop network, brute force search for an N-hop network can be reduced into brute force search for $(N-2)$-hop network by fixing ARQs $q_1, q_2, \ldots, q_{N-2}$.*

Proof. Consider an N-hop semi-cumulative scheme with $\mathbf{q} = [q_1, q_2, \ldots, q_N]$ where $\sum_{i=1}^{N} q_i = q_{sum}$. Let $pdp_{sc,N}$ and $pdp^{'}_{sc,N}$ represent the PDP of the N-hop network with $\mathbf{q} = [q_1, q_2, \ldots, q_{N-1}, q_N]$ and $\mathbf{q}^{'} = [q_1, q_2, \ldots, q_{N-1} + 1, q_N - 1]$ respectively. The PDP expressions with $\mathbf{q}$ and $\mathbf{q}^{'}$ can be respectively written as

$$\begin{aligned} pdp_{sc,N} &= pdp_{sc,1h} + \ldots + pdp_{sc,(N-1)h} + pdp_{sc,Nh}. \\ &= pdp_{sc,N-2} + pdp_{sc,(N-1)h} + pdp_{sc,Nh}, \\ pdp^{'}_{sc,N} &= pdp^{'}_{sc,1h} + \cdots + pdp^{'}_{sc,(N-1)h} + pdp^{'}_{sc,Nh}. \\ &= pdp^{'}_{sc,N-2} + pdp^{'}_{sc,(N-1)h} + pdp^{'}_{sc,Nh}, \end{aligned}$$

where the individual expressions are the probabilities that the packet is dropped at the intermediate links. It is straightforward to note that $pdp_{sc,jh} = pdp^{'}_{sc,jh}$ for $1 \leq j \leq N-2$ since the first $N-2$ terms are the same in $\mathbf{q}$ and $\mathbf{q}^{'}$. Therefore, on equating $pdp_{sc,N} = pdp^{'}_{sc,N}$, we get

$$\begin{gathered} pdp_{sc,(N-1)h} + pdp_{sc,Nh} = pdp^{'}_{sc,(N-1)h} + pdp^{'}_{sc,Nh}, \\ pdp_{sc,(N-1)h} - pdp^{'}_{sc,(N-1)h} = -(pdp_{sc,Nh} - pdp^{'}_{sc,Nh}), \end{gathered}$$

where we can write $pdp^{'}_{sc,(N-1)h} = P_{N-1}\left(pdp_{sc,(N-1)h}\right)$ because at the $(N-1)$-th hop, every term of $F^{'}_{N-1}$ gets multiplied by P_{N-1} since one ARQ has been transferred from the N-th hop.

Hence, we can write

$$pdp_{sc,(N-1)h}(1-P_{N-1}) \;\; = \;\; -(pdp_{sc,Nh}-pdp^{'}_{sc,Nh}).$$

On expanding the above equation and including $(1-P_{N-1})$ in the product loop, we write

$$\left(\prod_{i=1}^{N-1}(1-P_i)P_i^{q_i}\right)\left(\frac{F_{N-1}}{\prod_{i=1}^{N-2}P_i^{q_i}}\right) = \left(\prod_{i=1}^{N-1}(1-P_i)P_i^{q_i}\right)P_N^{q_N}\frac{(P_N^{-1}F_N^{'}-F_N)}{\prod_{i=1}^{N-1}P_i^{q_i}},$$

where $F_N^{'}$ is the term obtained using the Fibonacci series corresponding to $pdp^{'}_{sc,Nh}$. The above equality can be further simplified as

$$\left(\frac{F_{N-1}}{\prod_{i=1}^{N-2}P_i^{q_i}}\right) = P_N^{q_N}\frac{(P_N^{-1}F_N^{'}-F_N)}{\prod_{i=1}^{N-1}P_i^{q_i}}. \tag{5.6}$$

In the rest of the proof, we will show that $\frac{P_N^{-1}F_N^{'}-F_N}{P_{N-1}^{q_{N-1}}}$ does not contain q_{N-1} in it. Towards that direction, note that both $F_N^{'}$ and F_N contain the same number of terms in their expansion using Fibonacci series, however, with the difference that the terms q_N and q_{N-1} in F_N appear as q_N-1 and $q_{N-1}+1$ in $F_N^{'}$, respectively. When constructing $F_N^{'}$ and F_N using binary sequences of length N, we partition the terms of $F_N^{'}$ and F_N into two categories, namely: the sequences that end with '01' and sequences that end with '10'. This is because the states of the nodes before the last two digits are the same for both $F_N^{'}$ and F_N. As a result, for the sequences that end with '01', we can take the term $\beta_E^{N-1,N}$ common, and only focus on its effect in $\frac{F_N^{'}-P_NF_N}{P_{N-1}^{q_{N-1}}}$. Similarly, for the sequences that end with '10', we can take the term $\beta_I^{N-2,N-1}$ common, and only focus on its effect in $\frac{P_N^{-1}F_N^{'}-F_N}{P_{N-1}^{q_{N-1}}}$. To handle the former case, the term $\beta_E^{N-1,N}$ from $F_N^{'}$ is of the form $\sum_{i=1}^{q_{N-1}+1}P_{N-1}^{q_{N-1}+1-i}P_N^{i-1} = P_{N-1}^{q_{N-1}}\left(\sum_{i=1}^{q_{N-1}+1}P_{N-1}^{1-i}P_N^{i-1}\right)$, whereas the term $\beta_E^{N-1,N}$ from F_N is of the form $\sum_{i=1}^{q_{N-1}}P_{N-1}^{q_{N-1}-i}P_N^{i-1} = P_{N-1}^{q_{N-1}}\left(\sum_{i=1}^{q_{N-1}}P_{N-1}^{-i}P_N^{i-1}\right)$. Therefore, the difference of the two corresponding terms in $\frac{P_N^{-1}F_N^{'}-F_N}{P_{N-1}^{q_{N-1}}}$ is $\frac{1}{P_N}$, and this is because of the equality

$$\sum_{i=1}^{q_{N-1}}P_{N-1}^{-i}P_N^{i-1} - \sum_{i=1}^{q_{N-1}+1}P_{N-1}^{1-i}P_N^{i-2} = -\frac{1}{P_N}. \tag{5.7}$$

This completes the proof that $\frac{P_N^{-1}F_N^{'}-F_N}{P_{N-1}^{q_{N-1}}}$ does not contain q_{N-1} in it from sequences ending with '01'. To handle the sequences that end with '10', the term $\beta_I^{N-2,N-1}$ contributing to $F_N^{'}$ is of the form $\sum_{i=1}^{q_{N-2}} P_{N-2}^{q_{N-2}-i} \sum_{k=0}^{i-2} P_{N-1}^{q_{N-1}+1+k}$. Similarly, the term $\beta_I^{N-2,N-1}$ contributing to F_N is of the form $\sum_{i=1}^{q_{N-2}} P_{N-2}^{q_{N-2}-i} \sum_{k=0}^{i-2} P_{N-1}^{q_{N-1}+k}$. Therefore, when evaluated as $P_N^{-1}F_N^{'} - F_N$, the term $P_{N-1}^{q_{N-1}}$ can be taken common from both the terms, and therefore, the term $\frac{F_N^{'}-P_N F_N}{P_{N-1}^{q_{N-1}}}$ does not contain q_{N-1} in it from sequences ending with '10'.

Henceforth, (5.6) is written as $R_{1,N} = P_N^{q_N} R_{2,N}$, wherein $R_{1,N} \triangleq \left(\frac{F_{N-1}}{\prod_{i=1}^{N-2} P_i^{q_i}}\right)$ and $R_{2,N} \triangleq \frac{(P_N^{-1}F_N^{'}-F_N)}{\prod_{i=1}^{N-1} P_i^{q_i}}$ do not contain the terms $P_N^{q_N}$ and q_{N-1}. Hence, $R_{1,N}$ and $R_{2,N}$ are constants since $\{P_i \mid i = 1, 2, \ldots, N\}$ and $\{q_i \mid i = 1, 2, \ldots, N-2\}$ are fixed. Now, we can rewrite the equality condition as $P_N^{q_N} = \frac{R_{1,N}}{R_{2,N}}$, or as $q_N = \frac{\left(\log \frac{R_{1,N}}{R_{2,N}}\right)}{\log P_N}$. Note that in our work, we have a condition that $q_i \in \mathbb{Z}_+$, however, the solution of $q_N = \frac{\left(\log \frac{R_{1,N}}{R_{2,N}}\right)}{\log P_N}$ may belong to $\mathbb{R}$. It implies that to find the optimal solution which lies in $\mathbb{Z}_+$, we need to obtain either $\lceil q_N \rceil$ or $\lfloor q_N \rfloor$ from the equality condition. It can be observed that $\lceil q_N \rceil$ will decrease $P_N^{\lceil q_N \rceil}$, and this implies that $pdp_{sc,N} > pdp_{sc,N}^{'}$, and this is a sub-optimal solution because when we give one more ARQ from the last hop to the second last hop, PDP decreases. On the other hand, if we use $\lfloor q_N \rfloor$, then $P_N^{\lfloor q_N \rfloor}$ increases, which implies $pdp_{sc,N} < pdp_{sc,N}^{'}$. Therefore, on giving one more ARQ from the last hop to second last hop, PDP increases, and this implies that using $q_N = \lfloor \frac{\left(\log \frac{R_{1,N}}{R_{2,N}}\right)}{\log P_N} \rfloor$ in $\mathbf{q}$ captures the optimal solution conditioned on the first $N-2$ ARQ numbers. Thus, on fixing $q_1, q_2, \ldots, q_{N-2}$, we can analytically compute q_N, and also compute q_{N-1} using the relation $q_{N-1} = q_{sum} - \sum_{t=1, t\neq N-1}^{N} q_t$. $\square$

5.5 Low-Complexity Algorithms

From Theorem 5.3, we have proved that the search space for the N-hop network can be reduced to the search space of an $(N-2)$-hop network. Henceforth, we refer to this reduction as a one-fold technique. For a large value of N, we observe that the one-fold technique may not be feasible to implement in practice. Therefore, we propose low-complexity algorithms to further reduce the search space for the optimization problem.

5.5.1 List Generation using Multi-Folding

In the proposed multi-folding algorithm, as presented in Algorithm 3, instead of folding the network once from N-hop to $(N-2)$-hop, we fold it multiple times to $(N-4)$-hop, $(N-6)$-hop and so on up to a 2-hop network or a 1-hop network depending on whether N is even or odd, respectively. When the network is reduced (or folded) to a j-hop network, we need to provide a sum of $\tilde{q}_{sum,j}$ ARQs to it, and it is clear that $\tilde{q}_{sum,j}$ can take all possible values in a range $[j, q_{sum} - (N-j) + 1]$. When folding the network up to j-hops, for $j \geq 4$ and $j \geq 3$ when N is even and odd, respectively, we fix the ARQs for the first $(j-2)$-hops and then compute q_{j-1} and q_j using Theorem 5.3 for each value of $\tilde{q}_{sum,j}$. Subsequently, we create a list of ARQ distributions $[q_1, \ldots, q_j]$, denoted by $\mathcal{L}_j$, by varying the values of $\tilde{q}_{sum,j}$. Following a similar procedure, the candidates of $\mathcal{L}_j$ are used to generate $\mathcal{L}_{j+2}$ for the $(j+2)$-hop network by using Theorem 5.3 for each value of $\tilde{q}_{sum,j+2}$. This way, a list of ARQ distributions are obtained through $\mathcal{L}_N$ for the original N-hop network. It is clear that the size of the search space $\mathcal{L}_N$ reduces with increase in the number of folds.

5.5.2 Multi-Folding based Greedy Algorithm

To further reduce the size of the search space from that of Algorithm 3, we propose to retain the ARQ distribution that gives us minimum PDP for a given $\tilde{q}_{sum,j}$ from the list $\mathcal{L}_j$. This way, only one ARQ distribution survives for a given $\tilde{q}_{sum,j}$, thereby significantly reducing the list size when the algorithm traverses to $\tilde{q}_{sum,N}$. In the process of obtaining q_j for each $\tilde{q}_{sum,j}$, we note that only the floor of the ratio $\frac{\log R_j}{\log P_j}$ is chosen to obtain q_j. However, by observing that the optimal distribution of the folded network may not contribute to the optimal distribution of the original N-hop network, we also propose to select the ARQ distribution by ceiling the ratio $\frac{\log R_j}{\log P_j}$, where $R_j = \frac{R_{1,j}}{R_{2,j}}$. In other words, for each $\tilde{q}_{sum,j}$ in $\mathcal{L}_j$ we choose the ARQ distribution that minimizes the PDP for the j-hop network, and for that selected ARQ distribution, we also pick the ARQ distribution obtained by giving one ARQ from the last node to the penultimate node. It is straightforward to observe that this technique gives us a significantly shorter list compared to the multi-folding approach.

5.6 Simulation Results and Complexity Analysis

In the first part, we show that the packets of the ARQ based SC strategy that reach the destination arrive within the given deadline constraint with a high probability, provided the delay overheads from ACK/NACK are sufficiently small. To generate the results, q_{sum} is obtained as $\lfloor \frac{\tau_{total}}{\tau_p + \tau_d} \rfloor$ without considering the resources for ACK/NACK in the reverse channel, where τ_{total}, τ_d and τ_p are as defined in Section 5.1. Subsequently, we introduce different resolution of delays from NACK, say τ_{NACK} time units, and then study its impact on the end-to-end delay on the packets. Assuming $\tau_p + \tau_d = 1$ microsecond, we set the deadline for end-to-end packet delay as q_{sum} microseconds. Then, by sending an ensemble of 10^6 packets to the destination through the SC strategy, we compute the following metrics when $\tau_{NACK} \in \{0.2, 0.4, 0.6, 1\}$ microseconds: (i) the fraction of packets that were dropped in the network (denoted by W_{Drop}) due to insufficient ARQs at the intermediate nodes, (ii) the fraction of packets that reach the destination after the deadline (denoted by $W_{Deadline}$), and finally, (iii) the average end-to-end delay on the packets. These metrics are plotted in Fig. 5.2 for various values of SNR at a specific value of N and the LOS vector $\mathbf{c}$. The plots suggest that the average delay is significantly lower than that of the deadline especially when τ_{NACK} is small, owing to the opportunistic nature of ARQ strategies. However, as τ_{NACK} increases, the average delay is pushed closer to the deadline. Furthermore,

Algorithm 3 Multi-folding list algorithm

Require: $N, q_{sum}, \mathbf{P} = [P_1, P_2, \ldots, P_N]$.

Ensure: $\mathcal{L}_{final}$ - List of ARQ distributions in search space.

1: $\mathcal{L}_k = \{\phi\}$ (A null set) $\forall\, k = 1, 2, \ldots, N$.
2: **if** $N = odd$ **then**
3:     Start with fixing q_1.
4:     $\mathcal{L}_1 = \{[1, q_{sum} - (N-1) + 1]\}$.
5:     Assign $p = 3$.
6:     **for** $j = p : 2 : N$ **do**
7:         Assign $C = 1$.
8:         **for** $i_1 = 1 : |\mathcal{L}_{j-2}|$ **do**
9:             $[q_1, \ldots, q_{j-2}] = \mathcal{L}_{j-2}(i_1)$
10:             Compute $q_j = \lfloor \frac{\log R_j}{\log P_j} \rfloor$ where $R_j = \frac{R_{1,j}}{R_{2,j}}$.
11:             **for** $\tilde{q}_{sum,j} = j : (q_{sum} - (N-j) + 1)$ **do**.
12:                 Compute $q_{j-1} = \tilde{q}_{sum,j} - \sum_{t=1, t\neq j-1}^{j} q_t$.
13:                 **if** $q_{j-1} \geq 0$ **then**
14:                     Insert $[\mathcal{L}_{j-2}(i_1)||q_{j-1}||q_j]$ into $\mathcal{L}_j(C)$.
15:                     Assign $C = C + 1$.
16:                 **end if**
17:             **end for**
18:         **end for**
19:     **end for**
20:     $\mathcal{L}_{final} = \{\mathcal{L}_N | q_{j+1} \neq 0$ for $q_j = 0$ or $1\}$.
21: **else if** $N = even$ **then**
22:     Start with fixing q_1 and q_2.
23:     $\mathcal{L}_2 = \{\{q_1, q_2\} \in \mathbb{Z}_+^2 | q_1 + q_2 \in [2, q_{sum} - (N-2) + 1]\}$
24:     Assign $p = 4$.
25:     Repeat steps from line number 6 to 20.
26: **end if**

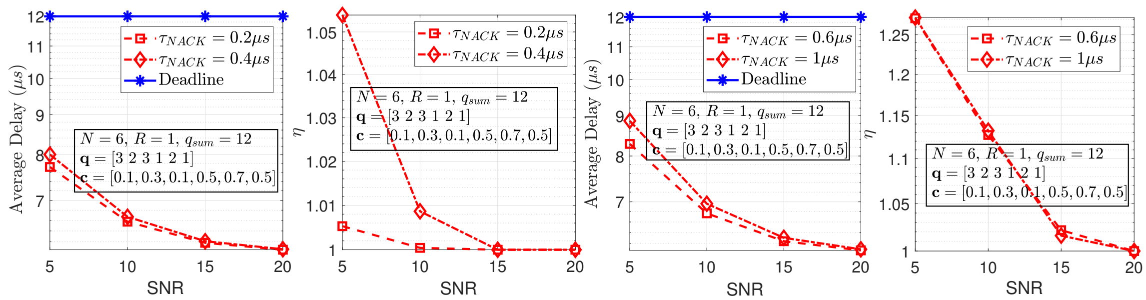

Figure 5.2: Variation of average delay on the packets and the deadline violation parameter (η) for various τ_{NACK} when using the SC strategy.

to capture the behaviour of deadline violations due to higher τ_{NACK}, in Fig. 5.2, we also plot $\eta = \frac{W_{Drop}+W_{Deadline}}{W_{Drop}}$. The plots confirm that when τ_{NACK} is sufficiently small compared to $\tau_p + \tau_d$ (see $\tau_{NACK} = 0.2\mu s$ at SNR = 15, 20 dB), the packets that reach the destination arrive within the deadline with an overwhelming probability as $\eta = 1$ at those values. In the rest of the section, we present simulation results to analyse the PDP of the SC strategy for various values of N, q_{sum}, and LOS vectors. Henceforth, to generate the simulation results for a given LOS vector $\mathbf{c}$ and SNR, we use the saddle-point approximation in [28, Theorem 2] on (5.2) to compute $\{P_i, 1 \leq i \leq N\}$. As emphasized in [28, Section V], these approximations are tight for Ricean channels when $R > 0.5$ and when the block-length K is in few hundreds. Therefore, for the proposed approximation to be valid, we use the block-length $K = 500$ in this simulation setup. In general, when the saddle-point approximation in [28, Theorem 2] is not tight, $\{P_i, 1 \leq i \leq N\}$ in (5.2) must be computed using numerical methods. First, in Fig. 5.3, we present simulation results to compare the PDP of the SC strategy with that of the non-cooperative strategy [30]. Although we have proved the dominance of our strategy theoretically, the plots confirm that the PDP of the SC strategy outperforms the PDP of the non-cooperative strategy with no increase in the communication-overhead on the packet. Furthermore, to showcase the benefits of using the multi-fold algorithm and the greedy algorithm, we plot the minimum PDP offered by these algorithms for $N = 5$ and $N = 6$ in Fig. 5.3. The plots confirm that while the multi-fold algorithm provides near-optimal ARQ distribution, the greedy algorithm is successful in offering the optimal ARQ distributions

In terms of complexity, for an N-hop network, the size of the search space for the SC strategy is upper bounded by $\binom{q_{sum}+N-1}{N-1}$. However, with the multi-fold algorithm, we have shown that the search space can be reduced. To showcase the reduction, we plot the size of the search space ($\mathcal{L}_N$) of the multi-fold algorithm for $N = 5$ and $N = 6$. For these cases, since we can fold the network at most twice, we have shown the results for both one-fold and two-fold cases. The simulation results, as shown in Fig. 5.4, display significant reduction in the list size as we move to one-fold and two-fold. Finally, the plots also show that the list size of the greedy algorithm is shorter than the multi-fold case, and it is, therefore, amenable to implementation in practice.

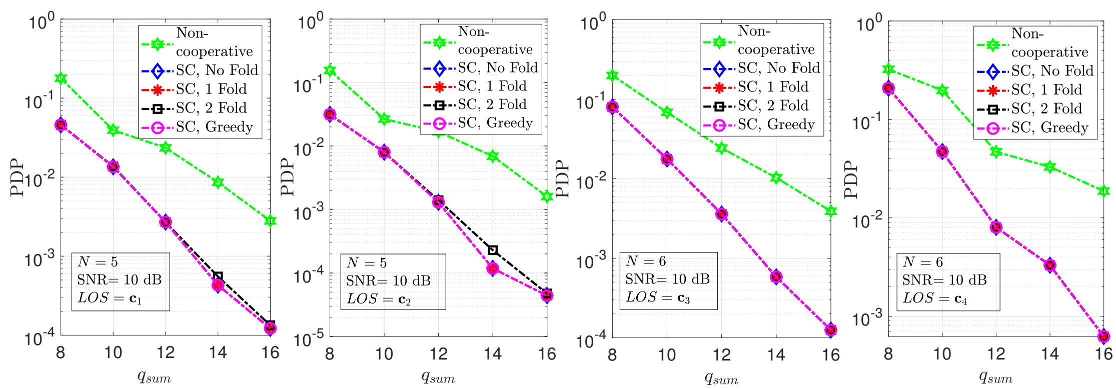

Figure 5.3: PDP plots when using an SC strategy with exhaustive search (no fold), 1-fold, 2-fold and greedy strategies with $\mathbf{c}_1 = [0.1, 0.3, 0.1, 0.5, 0.2]$, $\mathbf{c}_2 = [0.5, 0.5, 0.5, 0.5, 0.5]$, $\mathbf{c}_3 = [0.9, 0.2, 0.4, 0.7, 0.1, 0.5]$ and $\mathbf{c}_4 = [0.3, 0.3, 0.3, 0.3, 0.3, 0.3]$ at SNR = 10 dB and rate $R = 1$.

Although the above presented results showcase the reduction in the overall list size for computing the minimum PDP, they do not capture the number of computations at the destination in order to arrive at the final lists. If we include the computations required to apply the results of Theorem

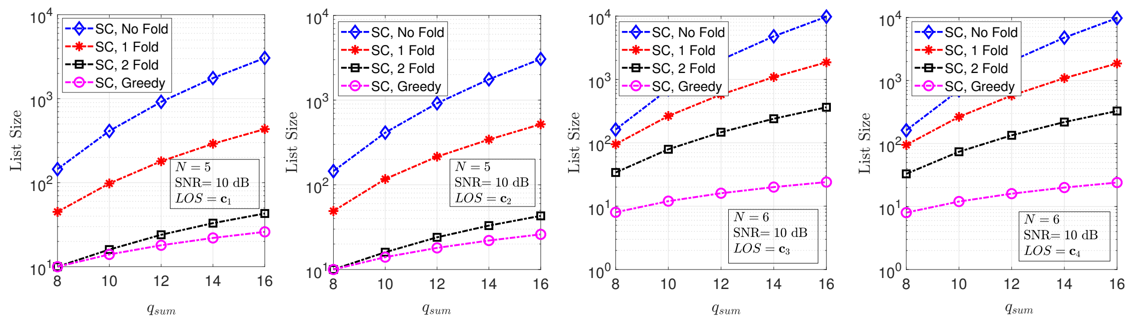

Figure 5.4: List sizes for an SC strategy with exhaustive search, 1-fold, 2-fold and greedy strategies with $\mathbf{c}_1 = [0.1, 0.3, 0.1, 0.5, 0.2]$, $\mathbf{c}_2 = [0.5, 0.5, 0.5, 0.5, 0.5]$, $\mathbf{c}_3 = [0.9, 0.2, 0.4, 0.7, 0.1, 0.5]$ and $\mathbf{c}_4 = [0.3, 0.3, 0.3, 0.3, 0.3, 0.3]$ at SNR$=10$ dB and rate $R = 1$.

5.3 at each level of multi-folding, it is clear that the greedy algorithm offers minimum complexity owing to fewer surviving distributions at each level. From the above results, we conclude that the proposed SC strategy can be preferred to the non-cooperative strategy. In the next section, we explore methods to further improve the reliability of the SC strategy by including the counters in the packet.

5.7 Summary

In this chapter, we have proposed a semi-cumulative ARQ scheme wherein intermediate relays of the network can use the residual ARQs from their previous node thereby reducing the PDP of the network with no compromise in the latency constraints. We have derived the Fibonacci series based closed-form expressions on the PDP of the semi-cumulative scheme for any given value of N and q_{sum}, and have subsequently addressed solving the optimization problem of minimizing the PDP under a sum constraint on the total number of ARQs.

Chapter 6

Cluster Based Semi-Cumulative ARQs Sharing Strategy in Multi-Hop Networks

6.1 Introduction

In the Chapter 4, the cooperative strategies use a counter in the packet so that the unused ARQs by a node can be used by the succeeding nodes in order to further reduce the PDP. Although the cooperative strategies of Chapter 4 are appealing, the use of counters in the packet contributes to additional communication-overhead in the packet. Therefore, we ask: (i) *Are there cooperative strategies for ARQ based DF protocols that DO NOT use a counter in the packet, and yet utilize the unused ARQs without violating the latency constraint?*, and (ii) *If the use of counter is allowed, are there cooperative strategies that outperform the strategies given in Chapter 4?* Also, in Chapter 5, despite using the SC strategy, there may be residual ARQs at some nodes which go unused. This is because a node cannot listen to the number of incorrect decoding events of all the preceding nodes in the chain. Towards circumvent these problems, in this chapter [1]:

[1]Part of the results presented in this chapter are available in publications [35]

1) We propose a Cluster-based Semi-Cumulative (CSC) strategy in which a group of consecutive nodes form a cluster, such that every node in the cluster forwards the information on the residual ARQs of its preceding node to the next node in the cluster through a counter in the packet. Therefore, a node in the cluster can make use of the residual ARQs of all the preceding nodes in the cluster. However, the nodes outside the cluster continue to use only the residual ARQs of their immediately preceding node without the need for a counter. This way, we further reduce the PDP of the network from that of the SC strategy without violating the sum constraint. Given the memory property introduced by the residual ARQs, we first provide a method to write the PDP of the CSC strategy, and then formulate an optimization problem to minimize the PDP subject to the sum constraint on the total ARQs. Furthermore, theoretical results on the ARQ distribution within the cluster and outside the cluster are also provided before proposing several low-complexity algorithms to solve the optimization.

2) Through extensive simulation results, we show that the proposed low-complexity algorithms for the CSC strategy provide near-optimal ARQ distributions in minimizing the PDP. Furthermore, we show that SC strategy and its cluster variant respectively outperform the non-cooperative strategy and its cluster variant (as discussed in Chapters 3 and 4) for a given number of total ARQs. In addition, unlike the cluster based cooperative strategy, we show that the performance of the CSC strategy depends on the position of the cluster in the network. This is because the unused ARQs of the last node of the cluster will have to be used by the next node in the chain. We also present simulation results on packet delay profiles to highlight the impact of cooperation on latency performance. We show that with no overhead to access residual ARQs from the packet, the CSC strategies incur a marginal increase in average delay compared to the SC strategy. This is due to more re-transmissions to provide lower PDP, and the need for updating the counter once at each intermediate relay. However, it allows a majority of packets to reach the destination within the deadline.

6.2 Cluster Based Semi-Cumulative Strategy

In the SC strategy, the benefit of cooperation is limited if an intermediate node uses more ARQs than the number allotted to it. For instance, in a 3-hop network with ARQ distribution $[q_1, q_2, q_3] = [4, 3, 5]$, suppose that the first hop consumes 2 attempts and the second hop consumes 4 attempts by utilizing one residual ARQ from the first hop. Although one residual ARQ is still unused from the first two hops, the third hop cannot utilize this because the second hop has used more ARQs than its allotted quota. On the other hand, if the transmitter of the third hop had the knowledge of residual ARQs entering the transmitter of the second hop, then it would have used that one unused ARQs. Thus, to take advantage of the residual ARQs of the preceding nodes, we require a counter in the packet that would be updated with the residual ARQs at each hop. Formally, the set of consecutive nodes in the network that use a counter to share the residual ARQs in the packet is referred as a cluster (similar to Chapter 4). To explain the cluster-based idea, with q_1 denoting the number of ARQs allotted to the first hop, let the first node in the cluster make $q_1 - r_1$ number of attempts to successfully transmit the packet to the second node in the chain, for some $0 \leq r_1 \leq q_1 - 1$. After that, when the second node receives the packet successfully, it updates the counter with a number equal to the sum of ARQs allotted to itself and the residual ARQs coming to the previous hop, i.e., $q_2 + r_1$ ARQs, and then transmits the packet to the next node. If the second node of the cluster uses $q_2 + r_1 - r_2$ attempts, then the third node updates the counter with $q_3 + r_2$ ARQs before transmitting the packet. This way, each receiving node in the cluster updates the counter only once and recovers the total number of unused ARQs by the previous nodes.

Using the above idea, we propose a Cluster based Semi-Cumulative (CSC) strategy on an N-hop network wherein we make a group of nodes that acts as a cluster, as exemplified in Fig. 6.1. As the grouping of nodes can be done anywhere in the network, we propose three cases,

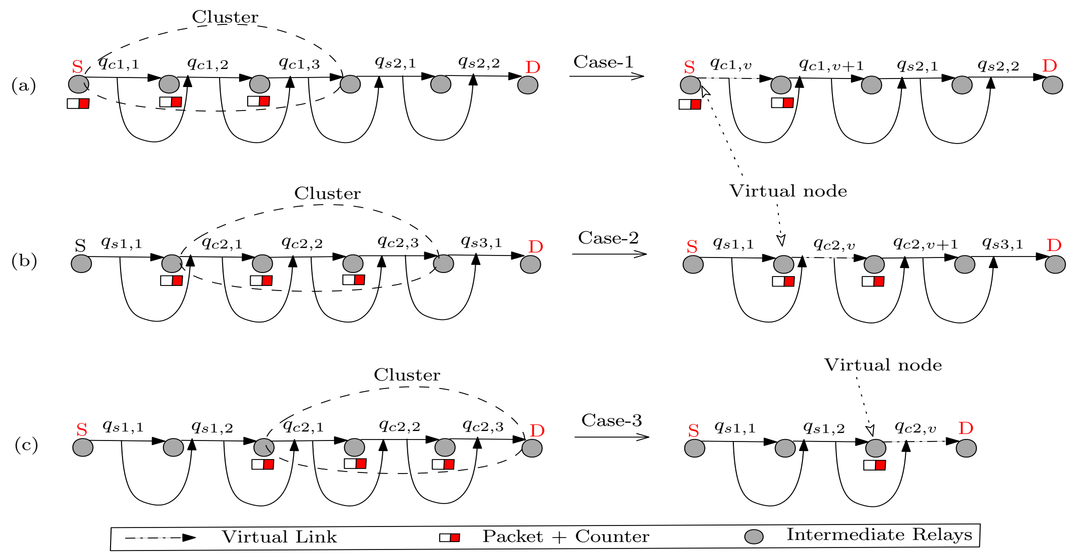

Figure 6.1: An illustrative example of a 5-hop network where a cluster is formed by grouping 3 consecutive nodes: (i) the cluster at the beginning (Case-1), (ii) the cluster at an intermediate position (Case-2), and (iii) the cluster at the end (Case-3).

namely, Case-1: cluster placed at the beginning by grouping a set of first few nodes. The network is made up of two portions, a cluster portion followed by a semi-cumulative portion (see (a) in Fig. 6.1). Case-2: cluster placed at an intermediate position. The N-hop network is made up of three portions, a semi-cumulative portion followed by a cluster, which in-turn is followed by a semi-cumulative portion (see (b) in Fig. 6.1). Case-3: cluster placed at the end by grouping a set of last few nodes in the network. The network is made up of two portions, a semi-cumulative portion followed by a cluster (see (c) in Fig. 6.1). Overall, the nodes inside the cluster can utilize the unused ARQs of all the preceding nodes in the cluster, whereas the nodes in the semi-cumulative portion(s) utilize the unused ARQs of its preceding node only. Henceforth, a multi-hop network employing the CSC strategy is referred to as the CSC network.

Similar to the SC strategy (as discussed in Chapter 5), we are interested in computing the optimal ARQ distribution on the N-hop CSC network such that its PDP is minimized for a given q_{sum}. Let N_{su} and N_{sw} represent the hop sizes of the semi-cumulative portions, and N_{cy} represent the hop size of the cluster. Here the first subscript indicates the type of the sub-network, and the subscripts $u \in \{1\}$ and $y \in \{1,2\}$ and $w \in \{2,3\}$ are jointly used to represent their placement in the N-hop network. Consequently, we have $N_{su} + N_{cy} + N_{sw} = N$. In particular, the valid combinations of u, y, w that capture the three cases of the CSC network are Case-1: $y = 1$ along with $w = 2$ implying that a cluster of size N_{c1} is followed by a semi-cumulative network of size N_{s2}. Case-2: $y = 2$ along with $u = 1$ and $w = 3$ implying that a cluster of size N_{c2} is between two semi-cumulative networks of size N_{s1} and N_{s3}. Finally, Case-3: $u = 1$ and $y = 2$ implying that a cluster of size N_{c2} follows a semi-cumulative network of size N_{s1}. Furthermore, let the ARQ distribution on the nodes in the cluster be denoted by $\mathbf{q}_{cy} = [q_{cy,1}, q_{cy,2}, \ldots, q_{cy,N_{cy}}]$, and the ARQ distribution on the nodes of the semi-cumulative portions be $\mathbf{q}_{su} = [q_{su,1}, q_{su,2}, \ldots, q_{su,N_{su}}]$ and $\mathbf{q}_{sw} = [q_{sw,1}, q_{sw,2}, \ldots, q_{sw,N_{sw}}]$. Given that a sum constraint is imposed on the ARQs, we have $\sum_{k=1}^{N_{su}} q_{su,k} + \sum_{k=1}^{N_{cy}} q_{cy,k} + \sum_{k=1}^{N_{sw}} q_{sw,k} = q_{sum}$ as long as the combinations of u, y, w are valid. Thus, in contrast to the usual notation on the ARQ vector $\mathbf{q} = [q_1, q_2, \ldots, q_N]$, we use $\mathbf{q} = [\underbrace{q_{su,1}, q_{su,2}, \ldots, q_{su,N_{su}}}_{\mathbf{q}_{su}}, \underbrace{q_{cy,1}, q_{cy,2}, \cdots q_{cy,N_{cy}}}_{\mathbf{q}_{cy}}, \underbrace{q_{sw,1}, q_{sw,2}, \ldots, q_{sw,N_{sw}}}_{\mathbf{q}_{sw}}]$, for valid combinations of u, y, w. Similarly, we use $\mathbf{P} = [P_{su,1}, P_{su,2}, \ldots, P_{su,N_{su}} P_{cy,1}, P_{cy,2}, \ldots P_{cy,N_{cy}}, P_{sw,1}, P_{sw,2}, \ldots, P_{sw,N_{sw}}]$ instead of $\mathbf{P} = [P_1, P_2, \ldots, P_N]$, to highlight the association of the outage probabilities to either the semi-cumulative portion or the cluster. With the above notations on a CSC network, the optimization problem for ARQ distribution is given in Problem 6.1 by encompassing all the three cases. As presented in Problem 6.1, $pdp_{cs,N}$ represents the PDP at the destination wherein the subscript cs highlights the CSC network.

Problem 6.1. *For an N-hop CSC network with a given LOS vector* $\mathbf{c}$, N_{cy}, N_{su}, N_{sw}, *SNR* $\alpha = \frac{1}{\sigma^2}$, *and* q_{sum}, *solve* $\arg\min_{\mathbf{q}} pdp_{cs,N}$, *subject to* $\mathbf{q} \in \{0 \cup \mathbb{Z}_+\}^N$ *such that* $\sum_{k=1}^{N_{su}} q_{su,k} +$

$\sum_{k=1}^{N_{cy}} q_{cy,k} + \sum_{k=1}^{N_{sw}} q_{sw,k} = q_{sum}$ *for all valid combinations of* u, y, w.

Towards solving the above problem, the following questions must be answered, (i) How to write the closed-form expression for $pdp_{cs,N}$ in Case-1, Case-2 and Case-3, (ii) Once the expression for $pdp_{cs,N}$ is written, how to allocate the ARQs to the nodes that are part of the cluster, and those that are outside the cluster. A particularly interesting question under the question in (ii) is "Does the optimal ARQ distribution on the nodes in the cluster depend on the position of the cluster? To justify the relevance of this question, note that in both Case-1 and Case-2, the node that follows the cluster can use the residual ARQs of the last node of the cluster by listening to its number of failed transmissions. In contrast, in Case-3, there is no node after the cluster that requires unused ARQs. Thus, whether or not non-zero ARQs must be allotted to the last node of the cluster depends on the position of the cluster. Henceforth, in the rest of this section, we follow a 3-step approach to solve Problem 6.1: **Step 1**: For a given q_{sum}, $\mathbf{q}_{su}$ $\mathbf{q}_{sw}$, characterize the structure of $\mathbf{q}_{cy}$ that minimizes the PDP. **Step 2**: Using the results of **Step 1**, form a virtual semi-cumulative network, and minimize its PDP. **Step 3**: Apply the structure of $\mathbf{q}_{cy}$ obtained in **Step 1** on the solution of **Step 2**.

As part of **Step 1**, we propose to solve Problem 6.2 that addresses the optimal ARQ distribution within the cluster conditioned on the ARQs allotted to the semi-cumulative network(s). In contrast to Problem 6.1, Problem 6.2 addresses to maximize the Packet Survival Probability (PSP) at the last node of the cluster for any given residual ARQs. Formally, the PSP at the N_{cy}-th node of the cluster is denoted by $psp_{cy,N_{cy}}(r_{cy,N_{cy}-1})$, where $r_{cy,N_{cy}-1}$ denotes the number of residual ARQs coming from the $(N_{cy}-1)$-th node of the cluster. We remark that it is imperative to take the PSP approach due to the presence of a semi-cumulative network after the cluster.

Problem 6.2. *For an* N*-hop CSC network with a given LOS vector* $\mathbf{c}$*, SNR* $\alpha = \frac{1}{\sigma^2}$*,* $\mathbf{q}_{su}$*,* $\mathbf{q}_{sw}$ *and* q_{sum}*, solve* $\arg \underset{q_{cy,1}, q_{cy,2}, \cdots q_{cy,N_{cy}-1}}{\max} psp_{cy,N_{cy}}(r_{cy,N_{cy}-1}), \forall r_{cy,N_{cy}-1}$*, where* $[q_{cy,1}, q_{cy,2}, \ldots$

$q_{cy,N_{cy}-1}] \in \{0 \cup \mathbb{Z}_+\}^{N_{cy}-1}$, *such that* $\sum_{k=1}^{N_{su}} q_{su,k} + \sum_{k=1}^{N_{cy}} q_{cy,k} + \sum_{k=1}^{N_{sw}} q_{sw,k} = q_{sum}$ *for all valid* u, y, w.

6.2.1 Theoretical Results for CSC Strategy

First, we consider a CSC network with either Case-1 or Case-2. For a given $\mathbf{q}_{su}$ and $\mathbf{q}_{sw}$, the following theorem proves that except the last node of the cluster, all the ARQs on the nodes inside the cluster must be transferred to the first node of the cluster in order to maximize the PSP at the last node of the cluster.

Theorem 6.1. *In Case-1 and Case-2, giving all the ARQs of the first* $N_{cy} - 1$ *nodes of the cluster to the first node of the cluster maximizes the PSP at the* N_{cy}*-th hop of the cluster for any given residual ARQs* $r_{cy,(N_{cy}-1)}$ *at the* N_{cy}*-th hop.*

Proof. We divide the proof into two parts depending on the placement of the cluster in the network.

Case I : With the cluster placed at the beginning, we have $N_{c1} + N_{s2} = N$. Let $\mathbf{q}_{c1} = \{q_{c1,1}, q_{c1,2}, \ldots, q_{c1,N_{c1}}\}$ and $\mathbf{q}_{s2} = \{q_{s2,1}, q_{s2,2}, \ldots, q_{s2,N_{s2}}\}$ represent the ARQ distribution of the networks $c1$ and $s2$, respectively. Therefore, the overall ARQ distribution of the N-hop network satisfies the constraint $\sum_{k_1=1}^{N_{c1}} q_{c1,k_1} + \sum_{k_2=1}^{N_{s2}} q_{s2,k_2} = q_{sum}$. We prove the theorem using the induction method. For the initialization step, let $N_{c1} = 3$ and $\mathbf{q}_{c1} = \{q_{c1,1}, q_{c1,2}, q_{c1,3}\}$. Since PSP is the metric of interest, let the residual ARQ arriving at the 3^{rd} hop be $r_{c1,2}$, where the range of $r_{c1,2}$ is $[0, q_{c1,1} + q_{c1,2} - 2]$. We prove that $\{q_{c1,1} + q_{c1,2}, 0\}$ maximizes the PSP at the 3^{rd} hop of the cluster for any residual ARQ. For the ARQ distribution $\{q_{c1,1}, q_{c1,2}\}$, let the consumption profile of ARQs at the first two hops of the cluster be $\tilde{\mathbf{q}}_{c1} = \{\tilde{q}_{c1,1}, \tilde{q}_{c1,2}\}$ such that $\tilde{q}_{c1,1} \leq q_{c1,1}$ and $q_{c1,1} + q_{c1,2} = \tilde{q}_{c1,1} + \tilde{q}_{c1,2} + r_{c1,2}$. In order to result in $r_{c1,2}$ residual ARQs, the PSP at the 3^{rd} hop of the cluster is $psp_{c1,3}(r_{c1,2}) = (1 - P_{c1,1})(1 - P_{c1,2})(\sum_{i=1}^{q_{c1,1}} P_{c1,1}^{i-1} P_{c1,2}^{q_{c1,1}+q_{c1,2}-r_{c1,2}-i-1})$. Note that when the exponent term of $P_{c1,2}$ goes negative, we discard the corresponding terms from PSP expression as those terms are invalid. Similarly, with ARQ distribution $\{q_{c1,1} + q_{c1,2}, 0\}$, the PSP at the 3^{rd} hop with $r_{c1,2}$ residual ARQs is $psp^{'}_{c1,3}(r_{c1,2}) = (1 - P_{c1,1})(1 - P_{c1,2})(\sum_{i=1}^{q_{c1,1}+q_{c1,2}} P_{c1,1}^{i-1} P_{c1,2}^{q_{c1,1}+q_{c1,2}-r_{c1,2}-i-1})$, and similar to $psp_{c1,3}$, we discard the terms with negative exponent on $P_{c1,2}$. After solving the difference of $psp^{'}_{c1,3}(r_{c1,2})$ and $psp_{c1,3}(r_{c1,2})$, we get $psp^{'}_{c1,3}(r_{c1,2}) - psp_{c1,3}(r_{c1,2}) = (1 - P_{c1,1})(1 - P_{c1,2})(\sum_{i=1}^{q_{c1,2}} P_{c1,1}^{i-1} P_{c1,2}^{q_{c1,1}+q_{c1,2}-r_{c1,2}-i-1})$. It can be observed that $psp^{'}_{c1,3}(r_{c1,2}) \geq psp_{c1,3}(r_{c1,2})$ (because $q_{c1,2} \geq 0$) where equality holds when $q_{c1,2} = 0$ or 1. Therefore, the initialization step is proved. Now, we assume that the result is also true for $N_{c1} = t + 1$ for any $t \geq 3$. As a consequence, the optimal ARQ distribution which maximizes the PSP at the $(t + 1)$-th hop for any given $r_{c1,t}$ is $\{q_{c1,1} + q_{c1,2} + \ldots + q_{c1,t}, 0, \ldots, 0\}$. Now, we have to prove that the same result is true for $N_{c1} = t + 2$.

Let $r_{c1,(t+1)}$ be the number of residual ARQs at $(t + 2)$-th node, where the range of $r_{c1,(t+1)}$ is $[0, \sum_{i=1}^{t+1} q_{c1,i} - (t + 1)]$. Let $psp_{c1,(t+2)}(r_{c1,t+1})$ and $psp_{c1,t+1}(r_{c1,t})$ be the PSP at the $(t + 2)$ and $(t + 1)$ nodes, respectively. Conditioned on a given $q_{c1,t+1}$, we want to write $psp_{c1,t+2}(r_{c1,t+1})$ in terms of $psp_{c1,t+1}(r_{c1,t})$. This means for a given $r_{c1,t}$, the $(t + 1)$-th node must make $(q_{c1,t+1} + r_{c1,t} - r_{c1,t+1})$ attempts out of which one is successful and the others are unsuccessful. The PSP at $(t + 2)$-th node is

$$psp_{c1,t+2}(r_{c1,t+1}) = (1 - P_{c1,t+1}) \sum_{r_{c1,t}} psp_{c1,t+1}(r_{c1,t}) P_{c1,t+1}^{q_{c1,t+1}+r_{c1,t}-r_{c1,t+1}-1}, \tag{6.1}$$

where $(1 - P_{c1,t+1}) P_{c1,t+1}^{q_{c1,t+1}+r_{c1,t}-r_{c1,t+1}-1}$ represents the probability that $q_{c1,t+1} + r_{c1,t} - r_{c1,t+1}$ ARQs are consumed at $(t + 1)$-th node. Note that once $r_{c1,t}$ is fixed, the second term in the summation of (6.1) is fixed. This implies that to maximize $psp_{c1,t+2}(r_{c1,t+1})$ for a given $q_{c1,t+1}$, we need to maximize $psp_{c1,t+1}(r_{c1,t})$ for every $r_{c1,t}$ in the valid range. From the induction step, we have already assumed that the ARQ distribution $\{q_{c1,1} + q_{c1,2} + \ldots + q_{c1,t}, 0, \ldots, 0\}$ maximizes the PSP for any residue at the $(t + 1)$-th hop. Therefore, by invoking the result from $(t + 1)$-hop network, the optimal ARQ distribution for the $(t + 2)$-hop network conditioned on $q_{c1,t+1}$ is of the form $\mathbf{q}_{c1,t+1} = \{q_{c1,1} + q_{c1,2} + \ldots + q_{c1,t}, 0, \ldots, 0, q_{c1,t+1}\}$. As the last step of this proof, we need to show that $q_{c1,t+1}$ must be transferred to the first node of the cluster in order to maximise the PSP for any residue at the $(t+2)$-th hop. Using proof by contradiction, let us assume that $q_{c1,t+1} > 1$ maximizes the PSP for any residue at the $(t+2)$-th hop. In that case, let us focus on the ARQ consumption profile $\tilde{\mathbf{q}}_{c1,t+1} = \{\tilde{q}_{c1,1}, \tilde{q}_{c1,2}, \ldots, \tilde{q}_{c1,t+1}\}$ of the first $t+1$ nodes of the cluster that results in $r_{c1,t+1} = 0$ at node $t+2$. We immediately note that when $q_{t+1} > 1$ in $\mathbf{q}_{c1,t+1}$, it is not possible to have ARQ consumption of the form $\{q_{c1,1}+q_{c1,2}+\ldots+q_{c1,t}-(t-1)+1, 1, \ldots, 1, q_{c1,t+1}-1\}$ because a node cannot borrow from its succeeding node. Therefore, the mass point value on the above consumption profile is 0. On the other hand, if $q_{c1,t+1} = 0$ or 1 in $\mathbf{q}_{c1,t+1}$, we can obtain every possible ARQ consumption profile, and hence in this case, we have a non-zero mass point value against each ARQ consumption profile. This is a contradiction as it results in higher value of $psp_{c1,t+2}(r_{c1,t+1} = 0)$ compared to that when $q_{t+1} > 1$ in $\mathbf{q}_{c1,t+1}$. Thus, the optimal ARQ distribution that maximizes PSP for every residue at the $(t+2)$-th hop is $\{q_{c1,1} + q_{c1,2} + \ldots + q_{c1,t+1}, 0, \ldots, 0\}$. Although the ARQ on the $(t+1)$-th node can be either 0 or 1, we have used 0 in this proof.

Case II : In this case, the cluster is placed in between two semi-cumulative networks such that $N = N_{s1} + N_{c2} + N_{s3}$. As a consequence, the first node of the cluster can make use of the residual ARQs from its previous node in addition to the ARQs allotted to it. Let $r_{s1,N_{s1}} \in [0, q_{s1,N_{s1}} - 1]$ be the residual ARQs from the last node of the $s1$ network. Therefore, the first node of the cluster can use ARQs in the range $[q_{c2,1}, q_{c2,1} + r_{s1,N_{s1}}]$. Since a semi-cumulative network precedes the cluster, the PSP at the first and the last nodes of the cluster are given in (6.2) and (6.3), respectively.

$$psp_{c2,1}(r_{s1,N_{s1}}) = (1 - P_{s1,N_{s1}}^{q_{s1,N_{s1}}}) \sum_{r_{s1,N_{s1}-1}} psp_{s1,N_{s1}}(r_{s1,N_{s1}-1}) P_{s1,N_{s1}}^{q_{s1,N_{s1}}-1-r_{s1,N_{s1}}}, \tag{6.2}$$

$$psp_{c2,N_{c2}}(r_{c2,N_{c2}-1}) = psp_{s1,N_{s1}}(r_{s1,N_{s1}-1}) psp_{c2,N_{c2}}(r_{c2,N_{c2}-1} | r_{s1,N_{s1}-1}, c2, 1, \ldots, q_{c2,N_{c2}-1}). \tag{6.3}$$

The LHS of (6.2) is fixed since the ARQ distribution on the semi-cumulative part $s1$ is fixed. However, the LHS of (6.3) depends on the ARQs allotted to the first $N_{c2}-1$ nodes of the cluster as well as how the residual ARQ entering the first node of the cluster is used. It can be observed that in order to maximize the LHS of (6.3) we need to maximize the second term of (6.3) because $psp_{s1,N_{s1}}(r_{s1,N_{s1}-1})$ is constant. By using Case I, we can maximize the second term by transferring all the ARQs of the first $N_{c2}-1$ nodes along with the residual ARQs from $s1$ to the first node of the cluster. This completes the proof. □

The following theorem shows that the above discussed ARQ distribution within the cluster minimizes the average PDP at the destination for a given ARQ allocation on the semi-cumulative networks.

Theorem 6.2. *In Case-1 and Case-2, for a given $\mathbf{q}_{su}$ and $\mathbf{q}_{sw}$, by maximizing the PSP at the N_{cy}-th hop of the cluster for any residual ARQs, we minimize the average PDP at the destination.*

Proof. First, we consider Case-1 where the cluster of size N_{c1} is followed by a semi-cumulative network of size N_{s2}. For this case, we have already proved that the ARQ distribution $\{q_{c1,1}+q_{c1,2}+\ldots+q_{c1,N_{c1}-1}, 0, \ldots, 0\}$ maximizes the PSP at N_{c1}-th hop of the cluster for every $r_{c1,N_{c1}-1}$. Furthermore, let $q_{c1,N_{c1}}$ be the number of ARQs given to the last node of the cluster, and therefore, the N_{c1}-th node can use upto $q_{c1,N_{c1}}+r_{c1,N_{c1}-1}$ ARQs. If N_{c1}-th node uses more than $q_{c1,N_{c1}}$ attempts (because of residual ARQs as indicated in the packet), then the first node of the $s2$ network cannot get ARQ benefits. However, if the N_{c1}-node uses fewer than $q_{c1,N_{c1}}$ attempts, then the first node of the $s2$ network can borrow the residual ARQs unused by the N_{c1}-th node. Formally, we can write the PSP at the first node of $s2$ network with respect to the PSP at N_{c1}-th hop as $psp_{s2,1}(r_{c1,N_{c1}}) = (1-P_{c1,N_{c1}})\sum_{r_{c1,N_{c1}-1}} psp_{c1,N_{c1}}(r_{c1,N_{c1}-1})P_{c1,N_{c1}}^{q_{c1,N_{c1}}-1+r_{c1,N_{c1}-1}-r_{c1,N_{c1}}}$, where $r_{c1,N_{c1}-1}$ and $r_{c1,N_{c1}}$ represent the number of residual ARQs arriving at N_{c1}-th node and first node of $s2$ network, respectively. Similarly, the PSP at the second node of the $s2$ network and at the destination are respectively given by $psp_{s2,2}(r_{s2,1}) = (1-P_{s2,1})\sum_{r_{c1,N_{c1}}} psp_{s2,1}(r_{c1,N_{c1}})P_{s2,1}^{q_{s2,1}-1+r_{c1,N_{c1}}-r_{s2,1}}$,

$psp_{s2,D}(r_{s2,N_{s2}}) = (1-P_{s2,N_{s2}}) \sum_{r_{s2,N_{s2}-1}} psp_{s2,N_{s2}}(r_{s2,N_{s2}-1}) P_{s2,N_{s2}}^{q_{s2,N_{s2}}-1+r_{s2,N_{s2}-1}-r_{s2,N_{s2}}}$, where D represents the destination of the N-hop network (i.e. the N_{s2}-th hop). It must be observed that since $psp_{c1,N_{c1}}(r_{c1,N_{c1}-1})$ is maximized for each $r_{c1,N_{c1}-1}$, the term $psp_{s2,1}(r_{c1,N_{c1}})$ is maximized for each $r_{c1,N_{c1}}$. Furthermore, by repeating the steps in the similar way, we maximize the PSP at the destination for every $r_{s2,N_{s2}}$. As a result, the average PSP is maximized at the destination, which in turn minimizes the average PDP at the destination. This completes the proof when the cluster is placed at the beginning. We can use a similar approach to prove the statement of the theorem for Case-2. □

Theorem 6.3. *In Case-3, for a given $\mathbf{q}_{su}$ and q_{sum}, giving all the ARQs of the N_{cy} nodes of the cluster to the first node of the cluster minimizes the PDP at the destination*

Proof. The proof is along the similar lines of Theorem 6.1 and Theorem 6.2 with the exception that the ARQs allotted to the last node of the cluster must also be transferred to the first node of the cluster; this is because there is no semi-cumulative network following the cluster in Case-3. □

For a given CSC network, once the ARQs on the N nodes is known, we have so far completed **Step 1** in our approach. In order to complete **Step 2**, it is important to write the expression for the PDP of the network. Given that it is challenging to write the PDP expression owing to the memory property in the strategy, we provide a set of rules to write the PDP expression for any N. Similar to the PDP expression for the SC strategy, we continue to make use of binary sequences, however, without using the structure of Fibonacci series owing to the presence of the cluster in the network. For the ease of explaining our procedure, if the cluster is placed other than the last position in the N-hop network, we split the cluster into two parts. The first part of the cluster is the group of first $N_{cy}-1$ nodes, henceforth referred to as the virtual node called node v. Similarly, the last node of the cluster is referred to as node $v+1$. On the other hand, when the cluster is placed at the last position of the network, all the N_{cy} nodes of the cluster are treated as node v.

Therefore, we replace the cluster by either two virtual hops or one virtual hop depending on its location in the network. Consequently, we will replace the physical N-hop network by a virtual $\tilde{N}$-hop network, where $\tilde{N} = N-(N_{c1}-1)+1$, $\tilde{N} = N-(N_{c2}-1)+1$, and $\tilde{N} = N-(N_{c2})+1$, for Case-1, Case-2, and Case-3, respectively. On this $\tilde{N}$-hop network, the effective ARQ vector is given by $\mathbf{q} = [q_{su,1}, q_{su,2}, \ldots, q_{su,N_{su}}, q_v, q_{v+1}, q_{sw,1}, q_{sw,2}, \ldots, q_{sw,N_{sw}}] \in \{0 \cup \mathbb{Z}_+\}^{\tilde{N}}$, where $q_v = \sum_{i=1}^{N_{cy}-1} q_{cy,i}$, and $q_{v+1} = q_{cy,N_{cy}}$ for Case-1 and Case-2, whereas $q_v = \sum_{i=1}^{N_{cy}} q_{cy,i}$ for Case-3. The procedure for writing the PDP expression for this virtual $\tilde{N}$-hop network is explained in the next theorem.

Theorem 6.4. *After replacing the cluster by node v and node $v+1$ in an N-hop CSC network, the PDP expression for the $\tilde{N}$-hop network can be written in closed-form.*

Proof. For the virtual $\tilde{N}$-hop network, we can write the PDP expression at the destination as $pdp_{cs,\tilde{N}} = pdp_{cs,1h} + pdp_{cs,2h} + \ldots + pdp_{cs,\tilde{N}h}$, where $pdp_{cs,kh}$, for $1 \leq k \leq \tilde{N}$, denotes the probability that the packet is dropped at the k-th hop. To write the expression of $pdp_{cs,kh}$, for each $k \in \{1, 2, \ldots, \tilde{N}\}$, we recommend using a set of k-length binary sequences to characterize the packet surviving event till the k-th hop. While writing the expression for $pdp_{cs,kh}$, a bit '$0'$ in the m-th position of the sequence, for $1 \leq m < k$, indicates that the m-th node forwards the packet to its next node without borrowing the residual ARQs from its preceding node. Similarly, bit '$1'$ indicates that the m-th node forwards the packet after borrowing the residual ARQs from its preceding node. For a given k, we discard the invalid sequences that do not follow the rules of CSC network These rules are: (a) sequences starting with 1 at the MSB must be discarded since the first node is the source, (b) sequence containing all 0 is invalid, (c) in a sequence, two consecutive $'1'$ can only occur at the positions of v and $v+1$ nodes of the cluster, because node $v+1$ of the cluster can borrow from node v irrespective of whether node v node has borrowed from its preceding node or not. This event occurs when the cluster is placed other than last position in the N-hop network. Therefore, a k-length sequence is invalid if it contains consecutive ones at any positions except for the above mentioned case, and (d) the last bit of the sequence can be either '$0'$ or '$1'$ as it represents whether the packet is dropped at the k-th node with no borrowing or borrowing of residual ARQs from the $(k-1)$-th node, respectively. First, we collect the set of valid k-length sequences, and then based on whether k is odd or even, we propose a procedure to use the k-length sequences to construct $pdp_{cs,kh}$. We divide the k-length sequence into the chunks of bits from the left to the right such that first chunk contains only the MSB bit and the subsequent bits are divided into the chunks of two bits. If k is odd, we end up with a chunk of two bits at the end, otherwise, we end up with a chunk of one bit. To map the bits to corresponding PDP terms, we parse the chunks of the sequence from left to right, and then replace the chunks with the terms as listed in Table 6.1. Among the terms in the table, the notation $psp_v(\cdot)$ represents the probability that packet reaches node $v+1$ after consuming as many ARQs in the argument by the virtual node. Similarly, $pdp_v(\cdot)$ represents the probability that the packet is dropped by the virtual node despite consuming as many ARQs in the argument. In this process, it is important to identify the locations of node v and node $v+1$ for every k. This way, each valid binary sequence is written as the product of terms of the form $\beta_{\gamma_1,\gamma_2 I}$, $\beta_{\gamma_1,\gamma_2 I}$, $\beta_{\gamma_1,\gamma_2 E}$, and $\beta_{\gamma_1,\gamma_2 E}$ as given in Table 6.1, for valid combinations of $\gamma_1 \in \{0, 1, 01, 10, 11\}$ and $\gamma_2 \in \{s, v, v+1, ss, sv, v(v+1), (v+1)s\}$. Once a k-length sequence is replaced by the corresponding expression, we add the expressions of all the valid k-length sequences to obtain B_k. Finally, using B_k, we obtain (i) $pdp_{cs,kh} = \prod_{i=1}^{k-1}(1-P_{s1,i})(P_{s1,k})^{q_{s1,k}} B_k$, when $k \leq N_{s1}$, (ii) $pdp_{cs,kh} = \prod_{i=1}^{k-1}(1-P_{s1,i})B_k$, when $k = N_{s1}+1$, (iii) $pdp_{cs,kh} = \prod_{i=1}^{k-2}(1-P_{s1,i})(P_{v+1})^{q_{v+1}} B_k$, when $k = N_{s1}+2$ (referred to as a special case in Table 6.1) and (iv) $pdp_{cs,kh} = (1-P_{v+1})\prod_{i=1}^{N_{s1}}(1-P_{s1,i})\prod_{j=1}^{t-1}(1-P_{s2,j})(P_{s2,t})^{q_{s2,t}} B_k$, when $k \geq N_{s1}+2+t$, where $1 \leq t \leq N_{s2}$ for all above cases. Overall, we get $pdp_{cs,\tilde{N}} = \sum_{i=1}^{\tilde{N}} pdp_{cs,ih}$. $\square$

From the above results, we have shown that the PDP for the CSC strategy can be written using the ARQ vector is $\mathbf{q} = [q_{su,1}, \ldots, q_{su,N_{su}}, q_v, q_{v+1}, q_{sw,1}, \ldots, q_{sw,N_{sw}}] \in \{0 \cup \mathbb{Z}_+\}^{\tilde{N}}$, where

Table 6.1: Binary sequence based PDP expression for $\tilde{N}$-hop CSC network

Nodes	Chunk size	Chunks	βs terms	Expression
First position				
s (MSB)	1	0	$\beta_{0,sI}$	$\sum_{i=1}^{q_{s1,1}} P_{s1,1}^{q_{s1,1}-i}$,
v (MSB)	1	0	$\beta_{0,vI}$	$\sum_{i=1}^{\bar{q}_v} psp_v(q_v-i+1)$, where $\bar{q}_v = q_v - (N_{c1}-2)$.
Middle positions $m < \tilde{N}$				For (s,s) combinations, the indices $m, m-1, m+1$ refer to positions of the nodes inside the semi-cumulative portions
s, s	2	00	$\beta_{00,ssI}$	$\big(\sum_{i=1}^{q_{sx,(m-1)}} P_{sx,m-1}^{q_{sx,m-1}-i}\big)\big(\sum_{j=1}^{q_{sx,m}} P_{sx,m}^{q_{sx,m}-j}\big)$, for $x \in \{u,w\}$.
s, s	2	01	$\beta_{01,ssI}$	$\big(\sum_{i=1}^{q_{sx,m-1}} P_{sx,m-1}^{q_{sx,m-1}-i}\big)\big(\sum_{j=0}^{i-2} P_{sx,m}^{q_{sx,m}+j}\big)$.
s, s	2	10	$\beta_{10,ssI}$	$\big(\sum_{j=0}^{i-2} P_{sx,m}^{q_{sx,m}+j}\big)\big(\sum_{k=1}^{q_{sx,(m+1)}} P_{sx,(m+1)}^{q_{sx,(m+1)}-k}\big)$, where i is residual ARQs from previous node.
s, v	2	00	$\beta_{00,svI}$	$\big(\sum_{i=1}^{q_{s1,N_{s1}}} P_{s1,N_{s1}}^{q_{s1,N_{s1}}-i}\big)\big(\sum_{j=1}^{\bar{q}_v} psp_v(q_v-j+1)\big)$.
s, v	2	01	$\beta_{01,svI}$	$\big(\sum_{i=1}^{q_{s1,N_{s1}}} P_{s1,N_{s1}}^{q_{s1,N_{s1}}-i}\big)\big(\sum_{j=1}^{i-1} psp_v(q_v+i-1-j+1)\big)$.
s, v	2	10	$\beta_{10,svI}$	$\big(\sum_{j=0}^{i-2} P_{s1,N_{s1}}^{q_{s1,N_{s1}}+j}\big)\big(\sum_{k=1}^{\bar{q}_v} psp_v(q_v-k+1)\big)$, where i is residual ARQs from previous node.
$v+1$, s	2	00	$\beta_{00,(v+1)sI}$	$\big(\sum_{i=1}^{q_{v+1}} P_{v+1}^{q_{v+1}-i}\big)\big(\sum_{i=1}^{q_{sw,1}} P_{sw,1}^{q_{sw,1}-i}\big)$.
$v+1$, s	2	01	$\beta_{01,(v+1)sI}$	$\big(\sum_{i=1}^{q_{v+1}} P_{v+1}^{q_{cy,v+1}-i}\big)\big(\sum_{j=0}^{i-2} P_{sw,1}^{q_{sw,1}+j}\big)$.
$v+1$, s	2	10	$\beta_{10,(v+1)sI}$	$\big(\sum_{j=0}^{i-1} P_{v+1}^{q_{v+1}+j}\big)\big(\sum_{k=1}^{q_{sw,1}} P_{sw,1}^{q_{sw,1}-k}\big)$.
$v, v+1$	2	00	$\beta_{00,v(v+1)I}$	$\big(\sum_{i=1}^{\bar{q}_v} psp_v(q_v-i+1)\big)\big(\sum_{j=1}^{q_{v+1}} P_{v+1}^{q_{v+1}-j}\big)$.
$v, v+1$	2	01	$\beta_{01,v(v+1)I}$	$\big(\sum_{i=1}^{\bar{q}_v} psp_v(q_v-i+1)\big)\big(\sum_{j=0}^{i-1} P_{v+1}^{q_{v+1}+j}\big)$.
$v, v+1$	2	10	$\beta_{10,v(v+1)I}$	$\big(\sum_{j=1}^{i-1} psp_v(q_v+i-1-j+1)\big)\big(\sum_{k=1}^{q_{v+1}} P_{v+1}^{q_{v+1}-k}\big)$, where i is the residual ARQs from previous node.
$v, v+1$	2	11	$\beta_{11,v(v+1)I}$	$\big(\sum_{j=1}^{i-1} psp_v(q_v+i-1-j+1)\big)\big(\sum_{k=0}^{j-1} P_{v+1}^{q_{v+1}+k}\big)$, where i is the residual ARQs from previous node.
Last positions				
s, s	2	01	$\beta_{01,ssE}$	$\sum_{i=1}^{q_{sw,N_{sw}}-1} P_{sw,N_{sw}-1}^{q_{sw,N_{sw}-1}-i} P_{sw,N_{sw}}^{i-1}$,
s, s	2	10	$\beta_{10,ssE}$	$\sum_{j=0}^{i-2} P_{s1,N_{s1}-1}^{q_{s1,N_{sw}-1}+j}$
s, v	2	01	$\beta_{01,svE}$	$\sum_{i=1}^{q_{s1,N_{s1}}} P_{s1,N_{s1}}^{q_{s1,N_{s1}}-i} PDP_v(q_v+i-1)$,
s, v	2	10	$\beta_{10,svE}$	$\sum_{j=0}^{i-2} P_{s1,N_{s1}}^{q_{s1,N_{s1}}+j} PDP_v(q_v)$
$v+1$, s	2	01	$\beta_{01,(v+1)sE}$	$\sum_{i=1}^{q_{v+1}} P_{v+1}^{q_{v+1}-i} P_{sw,1}^{i-1}$,
$v+1$, s	2	10	$\beta_{10,(v+1)sE}$	$\sum_{j=0}^{i-2} P_{v+1}^{q_{v+1}+j}$
$v, v+1^*$	2	01	$\beta_{01,v(v+1)E}$	$\big(\sum_{i=1}^{\bar{q}_v} psp_v(q_v-i+1)\big) P_{v+1}^{i-1}$.
$v, v+1^*$	2	11	$\beta_{11,v(v+1E}$	$\big(\sum_{j=1}^{i-1} psp_v(q_v+i-1-j+1)\big) P_{v+1}^{j-1}$, where i is the residual ARQs from previous node.
* Special case				
s (LSB)	1	0	$\beta_{0,sE}$	1,
s (LSB)	1	1	$\beta_{1,sE}$	$P_{sw,N_{sw}}^{i-1}$ where i is the residual ARQs from the previous node.
v (LSB)	1	1	$\beta_{1,vE}$	$PDP_v(q_v+i-1)$ where i is the residual ARQs from the previous node.

$q_v = \sum_{i=1}^{N_{cy}-1} q_{cy,i}$, and $q_{v+1} = q_{cy,N_{cy}}$. Since $\mathbf{q}$ satisfies the sum constraint $\sum_{k=1}^{N_{su}} q_{su,k} + q_v + q_{v+1} + \sum_{k=1}^{N_{sw}} q_{sw,k} = q_{sum}$, it is straightforward to compute the optimal ARQ distribution $[q^*_{su,1}, \ldots, q^*_{su,N_{su}}, q^*_v, q^*_{v+1}, q^*_{sw,1}, \ldots, q^*_{sw,N_{sw}}]$ through exhaustive search. Subsequently, we can complete **Step 3** by obtaining the optimal ARQ distribution of the N-hop network as $q^*_{cy,1} = q^*_v$, $q^*_{cy,N_{cy}} = q^*_{v+1}$ and $q^*_{cy,j} = 0$ for $1 < j < N_{cy}$. However, since exhaustive search is not practical, we present theoretical results on complexity reduction to compute the optimal ARQ distribution for all the cases under **Step 2**.

6.2.2 Complexity Reduction for Case-1 and Case-2

In the following theorem, we show that the optimal ARQ distribution of the $\tilde{N}$-hop network can be computed by using the search space for the first $\tilde{N} - 2$ values of $\mathbf{q}$.

Theorem 6.5. *For Case-1 and Case-2, to find the optimal ARQ distribution, the brute force search of an $\tilde{N}$-hop network can be reduced to the brute force search for $(\tilde{N} - 2)$-hop network.*

Proof. Let the ARQ distribution of the $\tilde{N}$-hop network be $\tilde{\mathbf{q}}_{cs,\tilde{N}} = [\tilde{q}_{cs,1}, \tilde{q}_{cs,2}, \ldots, \tilde{q}_{cs,\tilde{N}-1}, \tilde{q}_{cs,\tilde{N}}]$, where either $\tilde{q}_{cs,\tilde{N}-1} = q_{sw,N_{sw}-1}, \tilde{q}_{cs,\tilde{N}} = q_{sw,N_{sw}}$, or $\tilde{q}_{cs,\tilde{N}-1} = q_{v+1}, \tilde{q}_{cs,\tilde{N}} = q_{sw,1}$. Similarly, let the outage probability vector be $[\tilde{P}_{cs,1}, \tilde{P}_{cs,2}, \ldots, \tilde{P}_{cs,\tilde{N}}]$ = $[P_{su,1}, \ldots, P_{su,N_{su}}, P_v, P_{v+1}, P_{sw,1}, \ldots, P_{sw,N_{sw}}]$, where $P_v = psp_v(\{q_{cy,1}, \ldots, q_{cy,N_{cy}-1}\})$ is as defined in Theorem 6.4, $P_{v+1} = P_{cy,N_{cy}}$. Our approach for the proof is to fix the ARQs for the first $\tilde{N} - 2$ nodes, and then transfer the ARQs from the last node to the penultimate node until the PDP is minimized. When we give one ARQ from last node to the penultimate node, we obtain $\tilde{\mathbf{q}}'_{cs,\tilde{N}} = [\tilde{q}_{cs,1}, \tilde{q}_{cs,2}, \ldots, \tilde{q}_{cs,\tilde{N}-1} + 1, \tilde{q}_{cs,\tilde{N}} - 1]$. Now, the PDP expressions with $\tilde{\mathbf{q}}_{cs,\tilde{N}}$ and $\tilde{\mathbf{q}}'_{cs,\tilde{N}}$, respectively, can be written as $pdp_{cs,\tilde{N}} = pdp_{cs,\tilde{N}-2} + pdp_{cs,(\tilde{N}-1)h} + pdp_{cs,\tilde{N}h}$, $pdp'_{cs,\tilde{N}} = pdp'_{cs,\tilde{N}-2} + pdp'_{cs,(\tilde{N}-1)h} + pdp'_{cs,\tilde{N}h}$, where the individual expressions are the probabilities that the packet is dropped at the intermediate links. Also, it can be noted that the last two nodes must be either only semi-cumulative nodes or a $v + 1$ node that is followed by a semi-cumulative node. It is straightforward to note that

$pdp_{cs,jh} = pdp^{'}_{cs,jh}$ for $1 \leq j \leq \tilde{N}-2$ since the first $\tilde{N}-2$ terms are the same in $\tilde{\mathbf{q}}_{cs,\tilde{N}}$ and $\tilde{\mathbf{q}}^{'}_{cs,\tilde{N}}$, when cluster is placed other than the last position. Therefore, on equating $pdp_{cs,\tilde{N}} = pdp^{'}_{cs,\tilde{N}}$, we get $pdp_{cs,(N-1)h} - pdp^{'}_{cs,(\tilde{N}-1)h} = -(pdp_{cs,\tilde{N}h} - pdp^{'}_{cs,\tilde{N}h})$, where we can write $pdp^{'}_{cs,(\tilde{N}-1)h} = \tilde{P}_{cs,\tilde{N}-1}(pdp_{cs,(\tilde{N}-1)h})$ because at the $(\tilde{N}-1)$-th hop, every term of $B^{'}_{\tilde{N}-1}$ gets multiplied by $\tilde{P}_{cs,\tilde{N}-1}$ (outage probability of the $(\tilde{N}-1)$-th hop) since one ARQ has been transferred from the $\tilde{N}$-th hop. Hence, we can write $pdp_{cs,(\tilde{N}-1)h}(1-\tilde{P}_{cs,\tilde{N}-1}) = -(pdp_{cs,\tilde{N}h} - pdp^{'}_{cs,\tilde{N}h})$. On expanding the above equation and including $(1-\tilde{P}_{cs,\tilde{N}-1})$ in the product loop, we can write

$$\frac{B_{\tilde{N}-1}}{\prod_{i=1,i\neq j,s.t.\tilde{P}_{cs,j}=P_v}^{\tilde{N}-2} \tilde{P}_{cs,i}^{q_{cs,i}}} = \tilde{P}_{cs,\tilde{N}}^{\tilde{q}_{cs,\tilde{N}}} \frac{(\tilde{P}^{-1}_{cs,\tilde{N}} B^{'}_{\tilde{N}} - B_{\tilde{N}})}{\prod_{i=1,i\neq j,s.t.\tilde{P}_{cs,j}=P_v}^{\tilde{N}-1} \tilde{P}_{csi}^{\tilde{q}_{cs,i}}}. \tag{6.4}$$

where $B_{\tilde{N}}$ and $B^{'}_{\tilde{N}}$ are the terms obtained using the binary sequence representation corresponding to $pdp_{cs,\tilde{N}h}$ and $pdp^{'}_{cs,\tilde{N}h}$, respectively. In the rest of the proof, we will show that $\big(\tilde{P}^{-1}_{cs,\tilde{N}} B^{'}_{\tilde{N}} - B_{\tilde{N}}\big)/\big(\tilde{P}^{\tilde{q}_{cs,\tilde{N}-1}}_{cs,\tilde{N}-1}\big)$ does not contain $\tilde{q}_{cs,\tilde{N}-1}$ in it. Towards that direction, note that both $B^{'}_{\tilde{N}}$ and $B_{\tilde{N}}$ contain the same number of terms in their expansion using binary sequences, however, with the difference that the terms $\tilde{q}_{cs,\tilde{N}}$ and $\tilde{q}_{cs,\tilde{N}-1}$ in $B_{\tilde{N}}$ appear as $\tilde{q}_{cs,\tilde{N}}-1$ and $\tilde{q}_{cs,\tilde{N}-1}+1$ in $B^{'}_{\tilde{N}}$, respectively. When constructing $B^{'}_{\tilde{N}}$ and $B_{\tilde{N}}$ using binary sequences of length $\tilde{N}$, we partition the terms of $B^{'}_{\tilde{N}}$ and $B_{\tilde{N}}$ into two categories, namely: the sequences that end with '01' and sequences that end with '10'. This is because the states of the nodes before the last two digits are the same for both $B^{'}_{\tilde{N}}$ and $B_{\tilde{N}}$. As a result, for the sequences that end with '01', we can take the term $\beta_{01,\gamma_2 E}$, for $\gamma_2 \in \{(v+1)s, ss\}$ at the locations $\tilde{N}-1$ and $\tilde{N}$, common, and only focus on its effect in $(B^{'}_{\tilde{N}} - \tilde{P}_{cs,\tilde{N}} B_{\tilde{N}}/\tilde{P}^{\tilde{q}_{cs,\tilde{N}-1}}_{cs,\tilde{N}-1})$. Similarly, for the sequences that end with '10', we can take the term $\beta_{10,\gamma_2 I}$, for $\gamma_2 \in \{v(v+1), (v+1)s, ss\}$ common at the locations $\tilde{N}-2$ and $\tilde{N}-1$ and only focus on its effect in $(\tilde{P}^{-1}_{cs,\tilde{N}} B^{'}_{\tilde{N}} - B_{\tilde{N}})/(\tilde{P}^{\tilde{q}_{cs,\tilde{N}-1}}_{cs,\tilde{N}-1})$. To handle the former case, the term $\beta_{01,\gamma_2 E}$ from $B^{'}_{\tilde{N}}$ is of the form $\sum_{i=1}^{\tilde{q}_{cs,\tilde{N}-1}+1} \tilde{P}^{\tilde{q}_{cs,\tilde{N}-1}+1-i}_{cs,\tilde{N}-1} \tilde{P}^{i-1}_{cs,\tilde{N}}$, whereas the term $\beta_{01,\gamma_2 E}$ from $B_{\tilde{N}}$ is of the form $\sum_{i=1}^{\tilde{q}_{cs,\tilde{N}-1}} \tilde{P}^{\tilde{q}_{cs,\tilde{N}-1}-i}_{cs,\tilde{N}-1} \tilde{P}^{i-1}_{cs,\tilde{N}}$. By taking $\tilde{P}^{\tilde{q}_{cs,\tilde{N}-1}}_{cs,\tilde{N}-1}$ common from both the above terms, the difference of the two corresponding terms in $\tilde{P}^{-1}_{cs,\tilde{N}} B^{'}_{\tilde{N}} - B_{\tilde{N}}/\tilde{P}^{\tilde{q}_{cs,\tilde{N}-1}}_{cs,\tilde{N}-1}$ is

$1/\tilde{P}_{cs,\tilde{N}}$, and this is because of the equality

$$\sum_{i=1}^{\tilde{q}_{cs,\tilde{N}-1}} \tilde{P}_{cs,\tilde{N}-1}^{-i} \tilde{P}_{cs,\tilde{N}}^{i-1} - \sum_{i=1}^{\tilde{q}_{cs,\tilde{N}-1}+1} \tilde{P}_{cs,\tilde{N}-1}^{1-i} \tilde{P}_{cs,\tilde{N}}^{i-2} = -\frac{1}{\tilde{P}_{cs,\tilde{N}}}. \tag{6.5}$$

This completes the proof that $(\tilde{P}_N^{-1} B'_{\tilde{N}} - B_{\tilde{N}})/\tilde{P}_{cs,\tilde{N}-1}^{\tilde{q}_{cs,\tilde{N}-1}}$ does not contain $\tilde{q}_{cs,\tilde{N}-1}$ in it from sequences ending with '01'. Note that this argument holds when $\gamma_2 \in \{(v+1)s, ss\}$. To handle the sequences that end with '10', we can have three types of terms based on the positions of node v and node $v+1$ and their status of borrowing the residual ARQs. One type of term is $\beta_{01,ssE}$ which contributes to $B'_{\tilde{N}}$ a term of the form $\sum_{i=1}^{\tilde{q}_{cs,\tilde{N}-2}} \tilde{P}_{cs,\tilde{N}-2}^{\tilde{q}_{cs,\tilde{N}-2}-i} \sum_{k=0}^{i-2} \tilde{P}_{cs,\tilde{N}-1}^{\tilde{q}_{cs,\tilde{N}-1}+1+k}$. Similarly, the term $\beta_{01,ssE}$ contributes to $B_{\tilde{N}}$ a term of the form $\sum_{i=1}^{q_{cs,\tilde{N}-2}} \tilde{P}_{cs,\tilde{N}-2}^{\tilde{q}_{cs,\tilde{N}-2}-i} \sum_{k=0}^{i-2} P_{cs,\tilde{N}-1}^{\tilde{q}_{\tilde{N}-1}+k}$. After taking out $\tilde{P}_{cs,\tilde{N}-1}^{\tilde{q}_{\tilde{N}-1}}$ common, we can evaluate that $(\tilde{P}_{cs,\tilde{N}}^{-1} B'_{\tilde{N}} - B_{\tilde{N}})/\tilde{P}_{cs,\tilde{N}-1}^{q_{\tilde{N}-1}}$ does not contain $\tilde{q}_{\tilde{N}-1}$. Furthermore, the other type of term is $\beta_{01,v(v+1)E}$ which contributing to $B'_{\tilde{N}}$ a term of the form $\sum_{i=1}^{\tilde{q}_{cs,\tilde{N}-2}} psp_{cs,\tilde{N}-2}(\tilde{q}_{cs,\tilde{N}-2} - i + 1) \sum_{j=1}^{i-1} \tilde{P}_{cs,\tilde{N}-1}^{\tilde{q}_{cs,\tilde{N}-1}+j}$. Similarly, the term $\beta_{01,v(v+1)E}$ contributing to $B_{\tilde{N}}$ is of the form $\sum_{i=1}^{\tilde{q}_{cs,\tilde{N}-2}} psp_{cs,\tilde{N}-2}(\tilde{q}_{cs,\tilde{N}-2} - i + 1) \sum_{j=1}^{i-1} \tilde{P}_{cs,\tilde{N}-1}^{\tilde{q}_{\tilde{N}-1}+j-1}$. Therefore, after taking out $\tilde{P}_{cs,\tilde{N}-1}^{\tilde{q}_{\tilde{N}-1}}$ common, we can show that $(\tilde{P}_{cs,\tilde{N}}^{-1} B'_{\tilde{N}} - B_{\tilde{N}})/\tilde{P}_{cs,\tilde{N}-1}^{\tilde{q}_{\tilde{N}-1}}$ does not contain $\tilde{q}_{\tilde{N}-1}$. Similar result can also be proved for the third type of term $\beta_{11,v(v+1)E}$. This completes the proof that $(B'_{\tilde{N}} - \tilde{P}_{cs,\tilde{N}} B_{\tilde{N}})/(\tilde{P}_{cs,\tilde{N}-1}^{\tilde{q}_{cs,\tilde{N}-1}})$ does not contain $\tilde{q}_{cs,\tilde{N}-1}$ in it from sequences ending with '10'.

Henceforth, equation (6.4) is written as $\tilde{R}_{1,\tilde{N}} = \tilde{P}_{cs,\tilde{N}}^{\tilde{q}_{cs,\tilde{N}}} \tilde{R}_{2,\tilde{N}}$, wherein $\tilde{R}_{1,\tilde{N}} \triangleq B_{\tilde{N}-1}/ (\prod_{i=1, i\neq j, s.t. \tilde{P}_{cs,j}=P_v}^{\tilde{N}-2} \tilde{P}_{cs,i}^{q_{cs,i}})$ and $\tilde{R}_{2,N} \triangleq (\tilde{P}_{cs,\tilde{N}}^{-1} B'_{\tilde{N}} - B_{\tilde{N}})/(\prod_{i=1, i\neq j, s.t. \tilde{P}_{cs,j}=P_v}^{\tilde{N}-1} \tilde{P}_{cs,i}^{\tilde{q}_{cs,i}})$ do not contain the terms $\tilde{P}_{cs,\tilde{N}}^{\tilde{q}_{cs,\tilde{N}}}$ and $\tilde{q}_{cs,\tilde{N}-1}$. Hence, $\tilde{R}_{1,\tilde{N}}$ and $\tilde{R}_{2,\tilde{N}}$ are constants since $\{\tilde{P}_{cs,i} \mid i = 1, 2, \ldots, \tilde{N}\}$ and $\{\tilde{q}_{cs,i} \mid i = 1, 2, \ldots, \tilde{N} - 2\}$ are fixed. Now, we can rewrite the equality condition as $\tilde{P}_{\tilde{N}}^{\tilde{q}_{\tilde{N}}} = \tilde{R}_{1,\tilde{N}}/\tilde{R}_{2,\tilde{N}}$, or as $\tilde{q}_{cs,\tilde{N}} = \log(\tilde{R}_{1,\tilde{N}}/\tilde{R}_{2,\tilde{N}})/\log \tilde{P}_{cs,\tilde{N}}$. Note that in our work, we have a condition that $q_{cs,i} \in \mathbb{Z}_+$, however, the solution of $\tilde{q}_{cs,\tilde{N}} = \log(\tilde{R}_{1,\tilde{N}}/\tilde{R}_{2,\tilde{N}})/\log \tilde{P}_{cs,\tilde{N}}$ may not belong to $\mathbb{Z}_+$. It implies that to find the optimal solution which lies in $\mathbb{Z}_+$, we need to obtain either $\lceil \tilde{q}_{cs,\tilde{N}} \rceil$ or $\lfloor \tilde{q}_{cs,\tilde{N}} \rfloor$ from the equality condition. It can be observed that $\lceil \tilde{q}_{cs,\tilde{N}} \rceil$

will decrease $\tilde{P}_{cs,\tilde{N}}^{\lceil q_{cs,\tilde{N}} \rceil}$, and this implies that $pdp_{cs,\tilde{N}} > pdp'_{cs,\tilde{N}}$, and this is a sub-optimal solution because when we give one more ARQ from the last hop to the second last hop, PDP decreases. On the other hand, if we use $\lfloor \tilde{q}_{cs,\tilde{N}} \rfloor$, then $\tilde{P}_{cs,\tilde{N}}^{\lfloor \tilde{q}_{cs,\tilde{N}} \rfloor}$ increases, which implies $pdp_{cs,\tilde{N}} < pdp'_{cs,\tilde{N}}$. Therefore, on giving one more ARQ from the last hop to second last hop, PDP increases, and this implies that using $\tilde{q}_{cs,\tilde{N}} = \lfloor \frac{\left(\log \frac{\tilde{R}_{1,\tilde{N}}}{\tilde{R}_{2,\tilde{N}}}\right)}{\log \tilde{P}_{cs,\tilde{N}}} \rfloor$ in $\tilde{\mathbf{q}}_{cs,\tilde{N}}$ captures the optimal solution conditioned on the first $\tilde{N}-2$ ARQ numbers. Thus, on fixing $\tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_{\tilde{N}-2}$, we can analytically compute $\tilde{q}_{\tilde{N}}$, and also compute $\tilde{q}_{\tilde{N}-1}$ using the relation $\tilde{q}_{cs,\tilde{N}-1} = q_{sum} - \sum_{t=1,t\neq\tilde{N}-1}^{\tilde{N}} \tilde{q}_{cs,t}$. □

Henceforth, the reduction technique in Theorem 6.5 is referred to as the one-fold technique since the search space for the $\tilde{N}$-hop network is reduced to that of an $(\tilde{N}-2)$-hop network. In the next section, we present the results on complexity reduction for Case-3.

6.2.3 Complexity Reduction for Case-3

When the cluster is placed at the last position in the network, it is not possible to obtain the results along the lines of Theorem 6.5. This is because the PDP at node v is not of the form of the PDP of a semi-cumulative node. To circumvent this problem, we present a new one-fold algorithm, which is as explained below. First, we split the N-hop network into two parts namely $\hat{N}_1$-hop and $\hat{N}_2$-hop sub-networks such that $\hat{N}_1 = N_{s1} + 1$ includes all the nodes of the semi-cumulative network along with the first node of the cluster, and $\hat{N}_2 = N_{c2} - 1$ includes all the nodes of the cluster except the first node. Let the outage probabilities and ARQ vector of $\hat{N}_1$-hop network be given by $\hat{\mathbf{P}}_1 = [\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_{\hat{N}_1}] = [P_{s1,1}, P_{s1,2}, \ldots, P_{s1,N_{s1}}, P_{c2,1}]$, where $\hat{P}_{\hat{N}_1} = P_{c2,1}$ (outage probability of the first node of the cluster) and $\hat{\mathbf{q}}_1 = [\hat{q}_1, \hat{q}_2, \ldots, \hat{q}_{\hat{N}_1}] = [q_{s1,1}, q_{s1,2}, \ldots, q_{s1,N_{s1}}, q_{c2,1}]$. Since the $\hat{N}_1$-hop network is also semi-cumulative, we apply Theorem 5.3 (from Chapter 5) on it by feeding a total of $q_{sum,\hat{N}_1} = q_{sum} - (N_{c2} - 1)$ ARQs. This way, we compute the list of ARQ distributions for the $\hat{N}_1$-hop network, which is as large as the search space for $(\hat{N}_1 - 2)$-hop network. Subsequently, for each ARQ distribution in the list, we transfer $(N_{c2} - 1)$ ARQs to $q_{\hat{N}_1}$ by assigning $q_{c2,j} = 0$ for $2 \leq j \leq N_{c2}$, and then compute the optimal ARQ distribution of the $\hat{N}$-hop network. Through this process, the size of the list is reduced to that of a $(\hat{N}_1 - 2)$-hop network thereby reducing the complexity. The pseudocode for the proposed method is provided in Algorithm 5.

6.3 Low-Complexity Algorithms for CSC Strategy

For large values of $\tilde{N}$, the one-fold technique might not be feasible to implement in practice. Therefore, to further reduce the complexity, we propose multi-folding and greedy algorithms for all the three cases.

6.3.1 List Generation using Multi-Folding for the CSC Strategy

In the proposed multi-folding algorithm, instead of folding the network once from $\tilde{N}$-hop to $(\tilde{N} - 2)$-hop, we fold it multiple times to $(\tilde{N} - 4)$-hop, $(\tilde{N} - 6)$-hop and so on, upto a 2-hop network or a 1-hop network depending on $\tilde{N}$, and the positions of node v and node $v + 1$ in the network. The pseudocodes of the multi-folding algorithm are presented in Algorithm 4 for Case-1, Algorithm 5 for Case-3, and Algorithm 6 for Case-2. In Case-1, we use Algorithm 4 with $\tilde{N} = N - (N_{c2} - 1) + 1$. For the first j-hop network, such that $j \in \{3, 4\}$, we fix a total number of ARQs, denoted by $q_{sum,j}$, in the range $[(j + N_{c1} - 2), q_{sum,\tilde{N}} - (\tilde{N} - j) + 1]$, and then compute q_{j-1} and q_j using the ratio $\tilde{R}_j$ from Theorem 6.5. Subsequently, using the list from the j-hop network, a list of ARQs is obtained for the $j + 2$-hop network, eventually generating a list for the $\tilde{N}$-hop network. Similarly, in Case-3, we use Algorithm 5 with $\tilde{N} = N - (N_{c2} - 1)$. The pseudocode presented in the algorithm captures the ideas presented in Section 6.2.3. In this case, the multi-folding algorithm as discussed in Section 5.5 (from Chapter 5) is applicable on

Algorithm 4 Multi-folding list algorithm for Case-1 of the CSC strategy

Require: N, N_{c1}, $\tilde{N}$, q_{sum}, Outage probabilities of the links

Ensure: $\mathcal{L}_{final}$ - List of ARQ distributions in search space

1: $\mathcal{L}_k = \{\phi\}$ for $k = 1, 2, \ldots, \tilde{N}$

2: **if** $\tilde{N}$ is odd **then** ▷ Start with fixing $\tilde{q}_{cs,1}$ and $\tilde{q}_{cs,2}$

3: $\quad$ $\mathcal{L}_1 = \{[N_{c1} - 1, q_{sum} - (\tilde{N} - 1) + 1]\}$

4: $\quad$ Assign $p = 3$

5: $\quad$ **for** $j = p : 2 : \tilde{N}$ **do**

6: $\quad\quad$ **for** $i_1 = 1 : |\mathcal{L}_{j-2}|$ **do**

7: $\quad\quad\quad$ $[\tilde{q}_{cs,1}, \ldots, \tilde{q}_{cs,j-2}] = \mathcal{L}_{j-2}(i_1)$

8: $\quad\quad\quad$ Compute $\tilde{q}_{cs,j}$ from $[\tilde{q}_{cs,1}, \ldots, \tilde{q}_{cs,j-2}]$ by using Theorem 6.5

9: $\quad\quad\quad$ **for** $\tilde{q}_{sum,j} = (j + (N_{c1} - 1) - 1) : (q_{sum} - (\tilde{N} - j) + 1)$ **do**.

10: $\quad\quad\quad\quad$ Compute $\tilde{q}_{cs,j-1} = \tilde{q}_{sum,j} - \sum_{t=1,t\neq j-1}^{j} \tilde{q}_{cs,t}$.

11: $\quad\quad\quad\quad$ Insert $[\mathcal{L}_{j-2}(i_1)||\tilde{q}_{cs,j-1}||\tilde{q}_{cs,j}]$ to $\mathcal{L}_j$ if $\tilde{q}_{cs,j-1} \geq 0$

12: $\quad\quad\quad$ **end for**

13: $\quad\quad$ **end for**

14: $\quad$ **end for**

15: $\quad$ $\mathcal{L}_{final} = \mathcal{L}_{\tilde{N}}$.

16: **else if** $\tilde{N}$ is even **then** ▷ Start with fixing $\tilde{q}_{cs,1}$ and $\tilde{q}_{cs,2}$

17: $\quad$ $\mathcal{L}_2 = \{\{\tilde{q}_{cs,1}, \tilde{q}_{cs,2}\} \in \mathbb{Z}_+^2 | \tilde{q}_{cs,1} + \tilde{q}_{cs,2} \in [N_{c1}, q_{sum} - (\tilde{N} - 2) + 1]\}$

18: $\quad$ Assign $p = 4$, and repeat steps from line number 5 to 15

19: **end if**

the $\hat{N}_1$-hop network which only comprises semi-cumulative nodes. Finally, in Case-2, we use Algorithm 6 with $\tilde{N} = N - (N_{c2} - 1) + 1$. Here, when the network is reduced (or folded) to a j-hop network, we need to identify the locations of node v and node $v + 1$ in the folded network. This is because the ARQs for node v and node $v + 1$ must not be computed in the same iteration as it does not result in complexity reduction. Therefore, we split the virtual $\tilde{N}$-hop network into three parts, namely: $\tilde{N}_1$-hop, $\tilde{N}_2$-hop and $\tilde{N}_3$-hop networks such that $\tilde{N} = \tilde{N}_1 + \tilde{N}_2 + \tilde{N}_3$, wherein $(\tilde{N}_1 + \tilde{N}_2)$-hop networks contain the semi-cumulative nodes preceding the cluster along with node v, and $\tilde{N}_3$-hop network contains node $v + 1$ along with the semi-cumulative nodes that follow the cluster. Note that this case invokes results from Algorithm 5 while folding within the $\tilde{N}_1$-hop network.

6.3.2 Multi-Folding Algorithms for the CSC Strategy

To further reduce the size of the search space from that in Algorithms 4, 5 and 6, we propose to retain the ARQ distribution that gives the minimum PDP for a given $\tilde{q}_{sum,j}$ from the list $\mathcal{L}_j$. This way, only one ARQ distribution survives for a given $\tilde{q}_{sum,j}$, thereby significantly reducing the list size when the algorithm traverses to $\tilde{q}_{sum,\hat{N}_1}$ (refer to Algorithms 5, 6) and $\tilde{q}_{sum,\tilde{N}_1}$ (refer to Algorithm 4). In addition, for the ARQ distribution that survives for a given $\tilde{q}_{sum,j}$, we generate one more ARQ distribution by giving one ARQ from the last node to the penultimate node for every j. This is along the similar lines of the greedy algorithm for the multi-folding algorithm in the SC strategy. If we capture the number of computations required to arrive at the final lists, it is clear that the greedy algorithm offers minimum complexity owing to fewer surviving distributions at each level.

Algorithm 5 Multi-fold algorithm for Case-3 of the CSC strategy

Require: $\hat{N}_1, \hat{N}_2, q_{sum}, q_{sum,\hat{N}_1} = q_{sum} - (N_{c2} - 1), \hat{\mathbf{P}}_1 = [\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_{\hat{N}_1}]$

Ensure: $\mathcal{L}_{final}$ - List of ARQ distributions in search space

1: **if** $\hat{N}_1$ is odd **then**
2: $\quad\mathcal{L}_1 = \{[1, q_{sum,\hat{N}_1} - (\hat{N}_1 - 1) + 1]\}$
3: $\quad$Assign $p = 3$
4: $\quad$**for** $j = p : 2 : \hat{N}_1$ **do**
5: $\quad\quad$**for** $i_1 = 1 : |\mathcal{L}_{j-2}|$ **do**
6: $\quad\quad\quad[\hat{q}_1, \ldots, \hat{q}_{j-2}] = \mathcal{L}_{j-2}(i_1)$
7: $\quad\quad\quad$Compute $\hat{q}_j$ from $[\hat{q}_1, \ldots, \hat{q}_{j-2}]$ using Theorem 5.3 (from Chapter 5)
8: $\quad\quad\quad$**for** $\tilde{q}_{sum,j} = j : (q_{sum,\hat{N}_1} - (\hat{N}_1 - j) + 1)$ **do**
9: $\quad\quad\quad\quad$Compute $\hat{q}_{j-1} = \tilde{q}_{sum,j} - \sum_{t=1,t\neq j-1}^{j} \hat{q}_t$
10: $\quad\quad\quad\quad$Insert $[\mathcal{L}_{j-2}(i_1)||\hat{q}_{j-1}||\hat{q}_j]$ in $\mathcal{L}_j$ if $\hat{q}_{j-1} \geq 0$ && $j < \hat{N}_1$
11: $\quad\quad\quad\quad$Insert $[\mathcal{L}_{j-2}(i_1)||\hat{q}_{j-1}||\hat{q}_j + \hat{N}_2]$ in $\mathcal{L}_j$ if $\hat{q}_{j-1} \geq 0$ && $j = \hat{N}_1$
12: $\quad\quad\quad$**end for**
13: $\quad\quad$**end for**
14: $\quad$**end for**
15: $\quad\mathcal{L}_{final} = \mathcal{L}_{\hat{N}_1}$.
16: **else if** $\hat{N}_1$ is even **then** $\triangleright$ Start with fixing $\hat{q}_1$ and $\hat{q}_2$
17: $\quad\mathcal{L}_2 = \{\{\hat{q}_1, \hat{q}_2\} \in \mathbb{Z}_+^2 | \hat{q}_1 + \hat{q}_2 \in [2, q_{sum,\hat{N}_1} - (\hat{N} - 2) + 1]\}$
18: $\quad$Assign $p = 4$, and repeat steps from line number 4 to 15
19: **end if**

Algorithm 6 Multi-folding list algorithm for Case-2 of the CSC strategy

Require: N, N_{s1}, N_{c2}, N_{s3}, $\tilde{N}$, q_{sum}, $\tilde{\mathbf{P}}_{cs} = [\tilde{P}_{cs,1}, \tilde{P}_{cs,2}, \ldots, \tilde{P}_{cs,\tilde{N}}]$

Ensure: $\mathcal{L}_{final}$ - List of ARQ distributions in search space

$\mathcal{L}_k = \{\phi\}$ $k = 1, 2, \ldots, \tilde{N}$

Split network $\tilde{N} = \tilde{N}_1 + \tilde{N}_2 + \tilde{N}_3$ such that $\tilde{N}_1 = N_{s1} + 1$, $\tilde{N}_2 = N_{c2} - 2$ and $\tilde{N}_3 = N_{s3} + 1$

Assign $\mathbf{q}_{sum,\tilde{N}_1} = [N_{s1} + 1, q_{sum} - (\tilde{N}_2 + \tilde{N}_3)]$

for $k_1 = 1 : |\mathbf{q}_{sum,\tilde{N}_1}|$ **do**

  Call Algorithm 5 with $q_{sum,\hat{N}_1} = \mathbf{q}_{sum,\tilde{N}_1}(k_1)$, $\hat{N}_1 = \tilde{N}_1$, $\hat{\mathbf{P}}_1 = \tilde{\mathbf{P}}_{\tilde{N}_1}$ and $\hat{N}_2 = \tilde{N}_2$

  $\mathcal{L}_{\tilde{N}_1,k_1} = \{\mathcal{L}_{\hat{N}_1} | \tilde{q}_{j+1} \neq 0$ for $\tilde{q}_j = 0$ or 1 where $j \in \{1, 2, \ldots, \tilde{N}_1 - 2\}$ and $\tilde{q}_{\tilde{N}_1} \geq N_{c2} - 1$ for $\tilde{q}_{\tilde{N}_1 - 1} \leq 1$ $\}$

  Insert $\mathcal{L}_{\tilde{N}_1,k_1}$ into $\mathcal{L}_{\tilde{N}_1}$

end for

Assign $p = \tilde{N}_1 + 2$

Assign $\bar{N} = \tilde{N} - 1$ if $\tilde{N}_3$ is odd, otherwise, assign $\bar{N} = \tilde{N}$

for $j = p : 2 : \bar{N}$ **do**

  for $i_1 = 1 : |\mathcal{L}_{j-2}|$ **do**

    $[\tilde{q}_1, \ldots, \tilde{q}_{j-2}] = \mathcal{L}_{j-2}(i_1)$

    Compute $\tilde{q}_j$ from $[\tilde{q}_1, \ldots, \tilde{q}_{j-2}]$ using Theorem 6.5

    for $\tilde{q}_{sum,j} = j + \tilde{N}_2 : (q_{sum} - (\tilde{N} - j) + 1)$ **do**.

      Compute $\tilde{q}_{j-1} = \tilde{q}_{sum,j} - \sum_{t=1,t\neq j-1}^{j} \tilde{q}_t$.

      Insert $[\mathcal{L}_{j-2}(i_1) || \tilde{q}_{j-1} || \tilde{q}_j]$ in $\mathcal{L}_j$ if $\tilde{q}_{j-1} \geq 0$

    end for

  end for

end for

if $\tilde{N}_3$ is odd **then**

  for $j_1 = 1 : |\mathcal{L}_{\tilde{N}-1}|$ **do**

    Compute $\tilde{q}_{\tilde{N}} = q_{sum} - sum(\mathcal{L}_{\tilde{N}-1}(j_1))$, where $sum(\cdot)$ is the sum of elements in $\mathcal{L}_{\tilde{N}-1}(j_1)$

    Insert $[\mathcal{L}_{\tilde{N}-1}(j_1) || \tilde{q}_{\tilde{N}}]$ in $\mathcal{L}_{\tilde{N}}$ if $\tilde{q}_{\tilde{N}} \geq 0$

  end for

end if

$\mathcal{L}_{final} = \mathcal{L}_{\tilde{N}}$

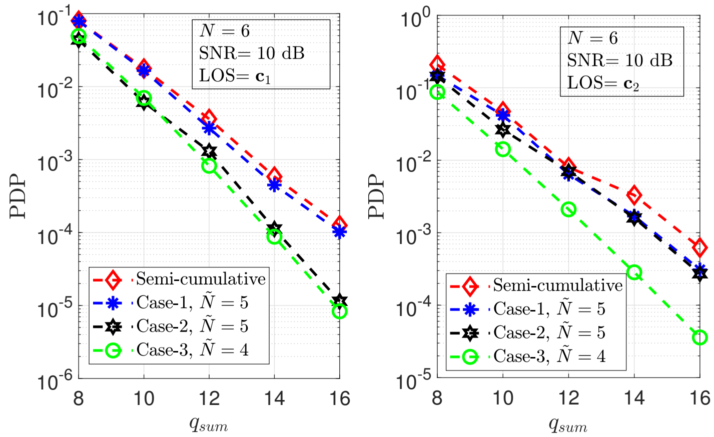

Figure 6.2: PDP comparison: Case-1 (first three hops constitute cluster), Case-2 (third hop to fifth hop forming a cluster) and Case-3 (fourth hop to sixth hop forming a cluster) with $\mathbf{c}_1 = [0.9, 0.2, 0.4, 0.7, 0.1, 0.5]$ and $\mathbf{c}_2 = [0.3, 0.3, 0.3, 0.3, 0.3, 0.3]$ at rate $R = 1$.

6.4 Simulation Results on Cluster based Semi-Cumulative Strategy

In this section, we present simulation results to showcase the benefits of using a cluster inside the N-hop semi-cumulative network for Case-1, Case-2 and Case-3. First, we present simulation results on the PDP and the complexity reduction of the proposed algorithms. For the experiment set up, we consider a 6-hop network such that Case-1 has $N_{c1} = 3$ and $N_{s2} = 3$. Similarly, in Case-2, we have $N_{s1} = 2$, $N_{c2} = 3$ and $N_{s3} = 1$, and in Case-3, we have $N_{s1} = 3$ and $N_{c2} = 3$. Similar to Section 5.6 (from Chapter 5), we use the saddle-point approximation in [28, Theorem 2] to compute $\{P_i, 1 \leq i \leq N\}$, for a given $\mathbf{c}$ and SNR, and also use $K = 500$.

In Fig. 6.2, we plot the minimum PDP offered by the optimal ARQ distribution for each case as a function of q_{sum}. From Fig. 6.2, it is clear that Case-3 shows a great improvement in PDP because the number of nodes contributing to node v is three, whereas in Case-1 and Case-2, we

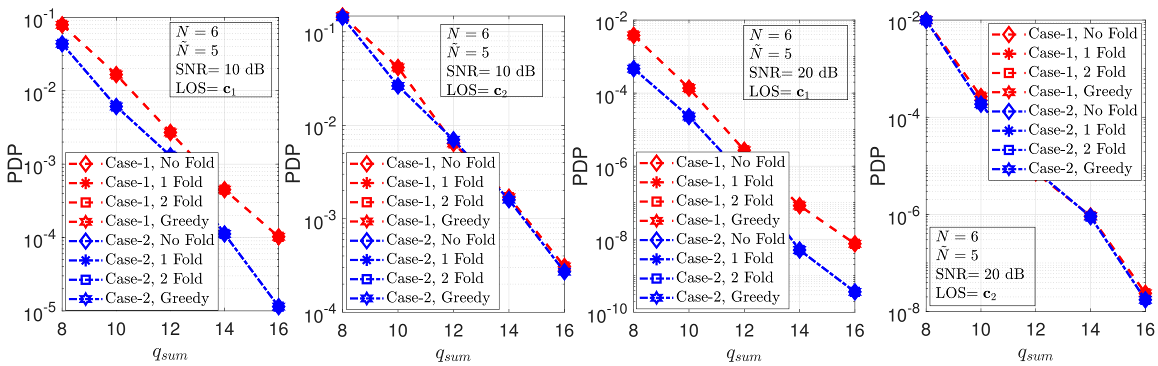

Figure 6.3: Plots depicting the PDP comparison for Case-1 (first three hops constitute cluster) and Case-2 (third hop to fifth hop forming a cluster) with no-fold, 1-fold, 2-fold and greedy strategies where $\mathbf{c}_1 = [0.9, 0.2, 0.4, 0.7, 0.1, 0.5]$ and $\mathbf{c}_2 = [0.3, 0.3, 0.3, 0.3, 0.3, 0.3]$.

have node $v + 1$, such that if it uses residual ARQs from its previous node, then, the next node in the chain cannot make use of residual ARQs from node $v + 1$. To showcase the significance of multi-folding and the greedy algorithms for Case-1 and Case-2, in Fig. 6.3, we plot the minimum PDP from the list of ARQ distributions by making use of exhaustive search, one-fold, two-fold and greedy algorithms. We use the same LOS vectors as in Fig. 6.2, i.e., $\mathbf{c}_1$ and $\mathbf{c}_2$ at SNR = 10 dB. Similarly for Case-3, the results on PDP analysis are shown in the left-side of Fig. 6.4. It can be observed that folding techniques provide near-optimal ARQ distributions.

To evaluate the complexity reduction of our algorithms, we plot the list size for all the three cases as a function of q_{sum} in the right side of Fig. 6.4 and in Fig. 6.5. From the plots, we observe a significant reduction in the list size by using the one-fold method, and further reduction in the list size when using the multi-fold and the greedy algorithms. In Case-3, as $\tilde{N} = 4$, we could not apply the multi-folding technique because the network-size does not allow to fold more than once. However, for large $\tilde{N}$, we can apply the multi-folding and the greedy algorithms to reduce the list size.

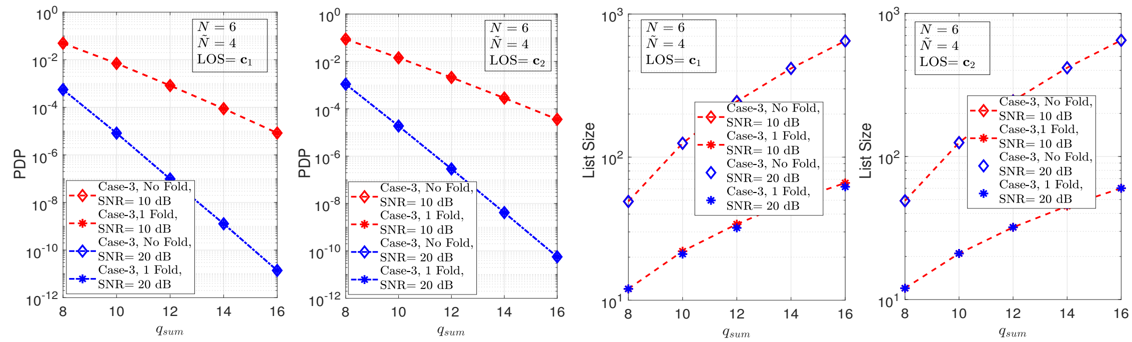

Figure 6.4: On the left: Plots illustrating the PDP comparison for Case-3 with no-fold, 1-fold algorithm where $\mathbf{c}_1 = [0.9, 0.2, 0.4, 0.7, 0.1, 0.5]$ and $\mathbf{c}_2 = [0.3, 0.3, 0.3, 0.3, 0.3, 0.3]$, SNR = 10 dB, rate $R = 1$. In this case, the last three hops constitute the cluster. On the right: Plots depicting the reduction in list size for Case-3 with exhaustive search, and 1-fold algorithms.

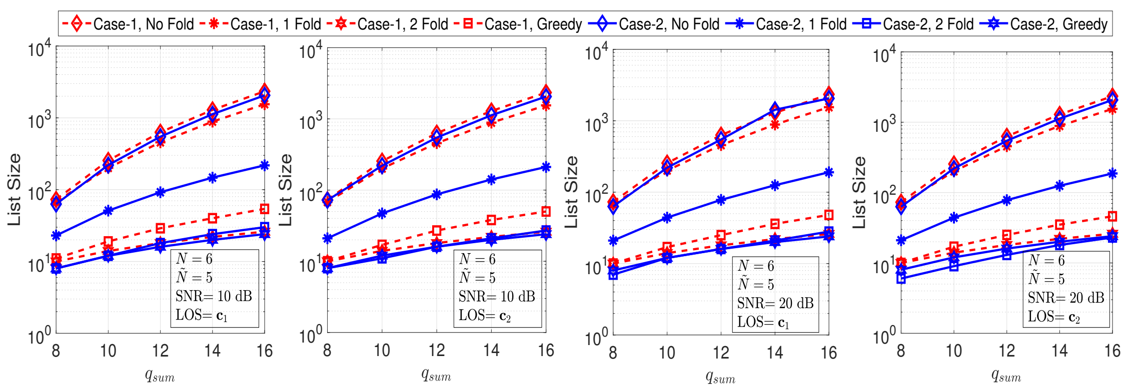

Figure 6.5: Plots depicting the reduction in list size for Case-1 (first three hops constitute cluster) and Case-2 (third hop to fifth hop forming a cluster) with no fold, 1 Fold, 2 Fold and Greedy strategies where $\mathbf{c}_1 = [0.9, 0.2, 0.4, 0.7, 0.1, 0.5]$ and $\mathbf{c}_2 = [0.3, 0.3, 0.3, 0.3, 0.3, 0.3]$.

In the rest of this section, we present delay analysis on packets to study the delay-overhead introduced by the use of the cluster in Case-1, Case-2 and Case-3 scenarios. Let us assume that we want to secure the packets from eavesdropping, and therefore, every node in the cluster encrypts the counter portion of the packet after updating it with the residual ARQs. As a result, when the packet is successfully decoded at the next node, it needs to decrypt it by using an appropriate crypto-primitive. Since this procedure results in an additional processing delay on the packet, we represent this delay by T_c microseconds. Assuming that the delay introduced on the packet per hop for each transmission is $T = 1$ microseconds (including both ACK/NACK), we analyse the effect of crypto-primitives on end-to-end delay by choosing $T_c = \alpha T$, where $\alpha = 0, 0.5,$ and 1. To quantify the delay profile of the packets in addition to PDP, we define a new metric referred to as probability of deadline violation (PDV), which can be defined as the probability that the packets either do not reach the destination before the deadline or get dropped in the network. In Fig. 6.6, we plot the PDV of the four strategies as a function of α for a 6-hop network with different parameters. Since $q_{sum} = 12$, the deadline for packets to reach the destination is 12 microseconds. The plots confirm that: (i) The PDV of the semi-cumulative network do not change with α. (ii) The PDV of the CSC strategy increases with increasing values of α; this is because some of the nodes make use of the counter in the packet. (iii) The worst-hit are Case-2 and Case-3 strategies with small q_{sum} as every node in the cluster has to modify the counter, thereby adding a significant delay of $N_{c1}T_c$ microseconds to the packet. However, in Case-1, the nodes except the first node of the cluster need to encrypt and decrypt, and hence, the additional delay from the cluster is $(N_{c1} - 1)T_c$ microseconds. Overall, the simulation results of Fig. 6.6 show that CSC strategy outperforms the SC strategy when α is small.

6.5 Summary

In this chapter, we have proposed a new family of cooperative ARQ strategies to assist low-latency communication in multi-hop networks. We have derived closed-form expressions on the PDP of the proposed strategies, and have solved the non-linear optimization problem of minimizing their PDP under a sum constraint on the total number of ARQs. We have shown that

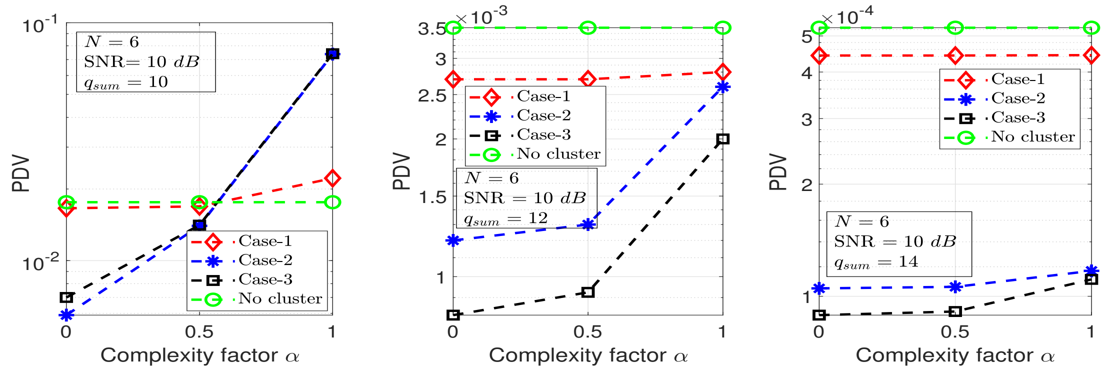

Figure 6.6: Simulation results on probability of deadline violation using a 6-hop network with $\mathbf{c} = [0.9, 0.2, 0.4, 0.7, 0.1, 0.5]$, at rate $R = 1$ and SNR = $10\ dB$ with 10^6 packets. For comparison, we have Case-1, Case-2, Case-3 and the SC strategy (with no cluster).

our strategies outperform the known strategies in this space.

Part IV

CC-HARQ Based Relaying Strategies for Achieving URLLC over Multi-Hop Networks

Chapter 7

CC-HARQ Strategies in Multi-Hop Networks under Slow-Fading Scenario

7.1 Introduction

In the previous chapters, the Type-1 ARQ protocol was employed, wherein the receiver node at each hop transmits an ACK or NACK depending on the success in decoding. Subsequently, on every retransmission, the receiver node discards the previous version of the packet and only uses the latest packet to decode the information. We note that when these solutions are used in the scenario where the devices are static, and their surrounding does not change over time (slow-fading setting), e.g., factory settings, Type-1 ARQs are not applicable for two apparent reasons: Firstly, in Type-1 ARQ, the previous packet is discarded; therefore, it implies the wastage of resources. Secondly, as the channel may not vary with time, there is no benefit in discarding the previous packets and then decoding the new packet (that is re-transmitted) under the same channel conditions. To address the mentioned issues, Hybrid Automatic Repeat Request (HARQ) is a promising technique wherein, on every retransmission, the receiver node combines the latest packet along with its previous copies to decode the information. Among many variants of HARQ schemes in practical systems, a popular scheme is Chase-Combining HARQ (CC-HARQ), wherein each retransmission block is identical to the original code block, and all the received blocks are combined using the maximum-ratio combining technique at the receiver [17].

Motivated by the drawbacks of using Type-1 ARQs in wireless networks with slowly varying channels, in this chapter, we propose a CC-HARQ based DF strategy for achieving high reliability under delay-bounded scenarios. In particular, we consider a multi-hop network dominated by slowly varying wireless channels with arbitrary line-of-sight (LOS) components. Along the similar lines of previous chapters, we impose an upper bound on the total number of ARQs required in the network following the CC-HARQ strategy based on the end-to-end delay requirements of the application. Subsequently, we formulate an optimization problem of allocating the optimal ARQ distribution to each intermediate link such that the packets reach the destination with high reliability.[1] Due to the intractability of the objective function in the optimization problem, first, we propose an approximation for the objective function at a high signal-to-noise-ratio (SNR). After that, we obtain the necessary and sufficient conditions on the near-optimal ARQ distribution followed by a low-complexity algorithm to solve the optimization problem. We show that our CC-HARQ based framework outperforms the Type-1 ARQ based strategy discussed in Chapter 3 and Chapter 4 when channels are slowly varying.

In terms of novelty, this is the first CC-HARQ based strategy that optimizes the reliability aspects of multi-hop networks with bounded-delay constraints. Moreover, the analytical results of this chapter cannot be viewed as a straightforward extension of previous chapters since the objective function used for optimization is unique to the CC-HARQ strategy, which was not

[1] Although it appears that allocating q_{sum} ARQs to the first node of the network seems optimal, such an approach would require each node to explicitly communicate the residual ARQs in the packet. This, in turn, leads to additional communication-overhead in the packet, and such a provision may not be allowed in certain applications.

dealt with bounded-delay applications hitherto [2].

The main contributions of this chapter are summarized as follows:

1. First, we deal with CC-HARQ based strategies for multi-hop networks with slow-fading channels (denoted as CC-HARQ-SF), wherein the channels are assumed to be static over the allotted attempts at each link. Under this scenario, we propose two types of strategies, namely: non-cumulative strategy, wherein the intermediate nodes only have the knowledge of ARQs allotted to themselves but not others, and the fully-cumulative strategy, wherein the intermediate nodes have the knowledge on ARQs allotted to the other nodes in addition to themselves. For reliability analysis, we use the packet-drop-probability (PDP) metric, which is defined as the fraction of packets that do not reach the destination. For the non-cumulative strategy, we derive a closed-form expression on PDP and formulate an optimization problem of minimizing the PDP for a given q_{sum}. We show that the optimization problem is non-tractable as it contains a first-order Marcum-Q function. Towards obtaining near-optimal ARQ distributions, we propose tight approximation on the first-order Marcum-Q functions, and then present non-trivial theoretical results for synthesizing a low-complexity algorithm. Through extensive simulations, we show that our analysis on the near-optimal ARQ distribution gives us the desired results with affordable complexity (see Section 7.5). For the fully-cumulative strategy, we provide theoretical results on the optimal ARQ distribution in closed-form, and show that it provides lower PDP than that of the non-cumulative strategy (see Section 7.6). However, we also point out that the PDP benefits offered by the fully-cumulative strategy come at the cost of marginal increase in communication-overhead, as they make use of a counter in the packet to forward the residual ARQs to the rest of the nodes in the up-stream.

2. After that, we present a detailed analysis on end-to-end delay by considering the following

[2]Parts of the result presented in this chapter are available in publications [36, 37]

metrics: (i) average end-to-end delay, (ii) packet deadline violation (PDV), which is defined by the number of packets reaching the destination after the given deadline, and (iii) delay profile, which represents the percentage of packets reaching the destination at a certain time for a given deadline. By using the aforementioned delay-metrics, we provide valuable insights on the merits and demerits of our strategies in achieving high-reliability with bounded constraints on end-to-end delay (see Section 7.7). In the next section, we introduce the system model for the CC-HARQ based strategy.

7.2 CC-HARQ Based Multi-Hop Network Model

Consider a network with N hops, as shown in Fig. 7.1, consisting of a source node (S), a set of $N-1$ relays $R_1, R_2, \ldots, R_{N-1}$ and a destination node (D). By aggregating the information bits in the form of packets, we communicate these packets from S to D by using the $N-1$ intermediate relays. We assume that the channel between any two successive nodes is characterized by Rician fading that varies slowly over time. We model the complex baseband channel of the k-th link, for $1 \leq k \leq N$, as

$$h_k = \sqrt{\frac{c_k}{2}}(1+\iota) + \sqrt{\frac{(1-c_k)}{2}}g_k,$$

where $\iota = \sqrt{-1}$, $0 \leq c_k \leq 1$ is the LOS component, $1 - c_k$ is the Non-LOS component, and g_k is a Gaussian random variable with distribution $\mathcal{CN}(0,1)$. In this channel model, c_k is a deterministic quantity, which characterizes different degrees of Rician fading channels, and also makes sure that $\mathbb{E}[|h_k|^2] = 1$ holds for any c_k. At the extreme ends, it is well known that $c_k = 0$ gives us the Rayleigh fading channels and $c_k = 1$ provides the Gaussian channels.

By assuming that the intermediate relays are sufficiently far apart from each other, throughout this chapter, we use the vector $\mathbf{c} = [c_1, c_2, \ldots, c_N]$ to denote the LOS components of the N-hop network. Let $\mathcal{C} \subset \mathbb{C}^L$ denote the channel code used at the source of rate R bits per channel use,

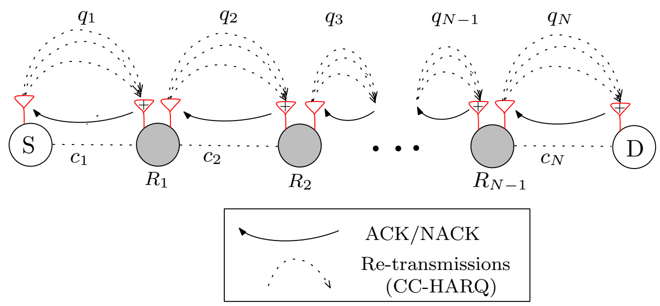

Figure 7.1: Illustration of an N-hop network with a source node (S), the relay nodes $R_1, \ldots, R_{N-1}$ and the destination node (D) follow the CC-HARQ protocol at each intermediate link. Also, each intermediate link can be characterized by an LOS component c_k $\forall k \in [N]$.

i.e.,

$$R = \frac{1}{L}\log(|\mathcal{C}|).$$

Let $\mathbf{x} \in \mathcal{C}$ denote a packet (which is a codeword in the code) transmitted by the source node such that its average energy per channel use is unity. When $\mathbf{x}$ is sent over the k-th link, for $1 \leq k \leq N$, the corresponding baseband symbols gathered at the receiver over L channel uses is $\mathbf{y}_k = h_k\mathbf{x} + \mathbf{w}_k \in \mathbb{C}^L$, where $\mathbf{w}_k$ is the additive white Gaussian noise (AWGN) at the receiver of the k-th link, distributed as $\mathcal{CN}(0, \sigma^2\mathbf{I}_L)$. We assume that h_k is known at the receiver of the k-th link owing to channel estimation; however, the transmitter of the k-th link does not know h_k. Since h_k is sampled from an underlying distribution and the realization remains constant for L channel uses, the transmission rate R may not be less than the instantaneous mutual information of the k-th link. Therefore, in such cases, the corresponding relay node will fail to correctly decode the packet. The probability of such an event is well captured by [3]

$$P_k = \text{Pr}\Big(R > \log_2(1 + |h_k|^2\gamma)\Big) = \text{F}_k\left(\frac{2^R - 1}{\gamma}\right), \tag{7.1}$$

where $\gamma = \frac{1}{\sigma^2}$ is the average signal-to-noise-ratio (SNR) of the k-th link, $\text{F}_k(x)$ is the cumulative distribution function of $|h_k|^2$, defined as

$$\text{F}_k\left(\frac{2^R - 1}{\gamma}\right) = 1 - Q_1\left(\sqrt{\frac{2c_k}{(1-c_k)}}, \sqrt{\frac{2(2^R-1)}{\gamma(1-c_k)}}\right), \tag{7.2}$$

such that $Q_1(\cdot,\cdot)$ is the first-order Marcum-Q function [27]. To support the transmission rate, we follow the CC-HARQ strategy wherein a receiver node asks the transmitter node for retransmission of the packet and combines the received packet with the previous failed attempts to recover the information. To explain further, for a given N-hop network model, a transmitter node gets an ACK or NACK from the next node in the chain, indicating the success or failure of the transmission, respectively. Upon receiving a NACK, the transmitter retransmits the packet, and the receiver combines the current packet with previously received copies of the packet. Let q_k be the maximum number of attempts given to the transmitter of the k-th link to retransmit the packet on demand. Consolidating the number of attempts given to each link, the ARQ distribution of the multi-hop network is represented by the vector $\mathbf{q} = [q_1, q_2, \ldots, q_N]$. Since we are addressing bounded delay applications, we impose the sum constraint $\sum_{i=1}^{N} q_i = q_{sum}$, for some $q_{sum} \in \mathbb{Z}_+$, which captures an upper bound on the end-to-end delay on the packets.

Note that the packet does not reach the destination if an intermediate node fails to deliver the packet to its next node despite using the allotted number of attempts. Since the packet can be dropped in any of the links, we use packet-drop-probability (PDP) as our reliability metric of

[3] It is well known that the expression for P_k given in (7.1) captures the error expression in asymptotic block-length regimes. However, it has been shown in [28, Theorem 2] that (7.1) serves as a saddle-point approximation on the error-probability expressions in non-asymptotic block lengths, especially when the block-length K is in the order of a few hundreds.

interest, given by

$$p_d = P_{1q_1} + \sum_{k=2}^{N} P_{kq_k}\left(\prod_{j=1}^{k-1}(1 - P_{jq_j})\right), \tag{7.3}$$

where P_{kq_k} represents the outage event at k-th link after using the q_k attempts with the CC-HARQ protocol. In particular, we have $P_{kq_k} = \Pr\left(R > \log_2(1 + (\sum_{j=1}^{q_k}|h_{kj}|^2)\gamma)\right)$, where h_{kj} is the channel realization at the k-th link for the j-th attempt. The expression on P_{kq_k} is obtained due to the maximum ratio combining technique. Under the scenario, when the channel remains fixed over multiple attempts, we can rewrite the above equation as $\Pr\left(R > \log_2(1 + |h_{k1}|^2 q_k\gamma)\right)$ which is equal to $\mathrm{F}_k\left(\frac{2^R-1}{q_k\gamma}\right)$. It can be observed that on every retransmission, the packet gets added to its previous copies due to CC-HARQ protocol, which results in an increased effective SNR of the given link. We call this strategy as CC-HARQ-SF, where 'SF' stands for slow-fading scenario.

For the proposed CC-HARQ-SF based multi-hop network, we are interested in minimizing the PDP given in expression (7.3) for a given sum constraint on the total number of ARQs given by $q_{sum} \in \mathbb{Z}_+$ such that $\sum_{i=1}^{N} q_i = q_{sum}$. Henceforth, we formulate our optimization problem in Problem 7.1 as shown below. Throughout this chapter, we refer to the solution of Problem 7.1, as the optimal ARQ distribution.

Problem 7.1. *For a given* $\mathbf{c}$ *and* $\gamma = \frac{1}{\sigma^2}$, *solve*

$$q_1^*, q_2^*, \ldots q_N^* = \arg \min_{q_1, q_2, \ldots q_N} p_d$$

$$\textit{subject to } q_k \geq 1, q_k \in \mathbb{Z}_+ \ \forall k \in [N], \sum_{i=1}^{N} q_i = q_{sum}.$$

7.3 Analysis on the Optimal ARQ Distribution using CC-HARQ-SF

The PDP expression in (7.3) is dependent on the Marcum-Q function, and tackling the Marcum-Q function analytically is challenging because it contains a modified Bessel function of first kind [27]. Therefore, in this section, we first present an approximation of the Marcum-Q function at a high SNR and then use it to provide necessary and sufficient conditions on the optimal ARQ distribution when using the approximations of the Marcum-Q function.

7.3.1 Approximation on the Marcum-Q function

Theorem 7.1. *For a given N-hop network and at high SNR regime, we can approximate the Marcum-Q function as $Q_1(a_k, b_k) \approx \tilde{Q}_1(a_k, b_k) \triangleq 1 - (b_k^2/2)e^{-a_k^2/2}$ for all $k \in [N]$.*

Proof. In the context of the N-hop network discussed in the previous section, the Marcum-Q function of first-order associated with the k-th hop is given by

$$e^{-\frac{a_k^2}{2}} e^{-\frac{b_k^2}{2}} \sum_{i=0}^{\infty} \left(\frac{a_k^2}{2}\right)^i \left(\sum_{m=0}^{\infty} \left(\frac{a_k b_k}{2}\right)^{2m} \frac{1}{m!(m+i)!}\right),$$

where $a_k = \sqrt{(2c_k)/(1-c_k)}$, $b_k = \sqrt{2(2^R-1)/\gamma(1-c_k)}$ such that $\gamma = 1/\sigma^2$. We highlight that a_k depends on the LOS component and b_k is dependent on SNR. It can be observed that at high SNR, b_k is very small, and henceforth, we use the small values of b_k to derive the approximation. On expanding $e^{-b_k^2/2}$ using Taylor series, and neglecting the higher power terms, we can approximate $e^{-b_k^2/2}$ by $(1-b_k^2/2)$. Similarly, by expanding the internal summation starting with index m, we can neglect the higher power terms of b_k and approximate the summation terms as

$$\sum_{m=0}^{\infty} \left(\frac{a_k b_k}{2}\right)^{2m} \frac{1}{m!\Gamma(m+i+1)} \approx \frac{1}{i!} + \frac{1}{(1+i)!}\left(\frac{a_k^2}{2}\right)b_k^2.$$

Therefore, we can rewrite $Q_1(a_k, b_k)$ as

$$Q_1(a_k, b_k) \approx e^{-\frac{a_k^2}{2}}\left(1-\frac{b_k^2}{2}\right)\sum_{i=0}^{\infty}\left(\frac{a_k^2}{2}\right)^i\left(\frac{1}{i!}+\frac{1}{(1+i)!}\left(\frac{a_k^2}{4}\right)b_k^2\right).$$

Now, by using $\sum_{n=0}^{\infty} x^n/n! = e^x$ and re-arranging the terms in above equation, we can write

$$Q_1(a_k, b_k) \approx e^{-\frac{a_k^2}{2}}\left(1-\frac{b_k^2}{2}\right)\left[e^{\frac{a_k^2}{2}}+\frac{b_k^2}{2}\left(e^{\frac{a_k^2}{2}}-1\right)\right].$$

On further solving the above equation and by neglecting the higher power terms of b_k, we obtain

$$Q_1(a_k, b_k) \approx \tilde{Q}_1(a_k, b_k) \triangleq 1-\frac{b_k^2}{2}e^{\frac{-a_k^2}{2}}. \tag{7.4}$$

This completes the proof. □

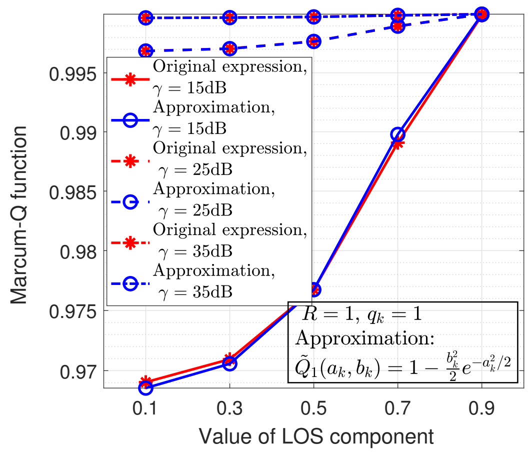

Figure 7.2: Comparison of original expression of Marcum Q-function with its proposed approximation at different values of SNR.

To validate the approximation given in (7.4) for different values of SNR and the LOS components, we present the simulation results in Fig. 7.2 where the original expression of Marcum-Q function is compared with its approximated expression given in (7.4). It can be observed from Fig. 7.2 that the approximation on the Marcum-Q function almost coincides with the original expression.

By using the approximation in (7.4), we can approximate (7.1) as

$$\tilde{P}_k = 1 - \tilde{Q}_1(a_k, b_k) \triangleq \frac{b_k^2}{2} e^{\frac{-a_k^2}{2}} = \frac{\phi}{(1-c_k)} e^{\frac{-a_k^2}{2}},$$

where $\phi = \frac{2^R - 1}{\gamma}$, and $\tilde{P}_k$ denotes the approximation on P_k. Similarly, for a given q_k retransmissions, we can approximate P_{kq_k} as $\tilde{P}_{kq_k}$, given by

$$\tilde{P}_{kq_k} \triangleq \frac{\phi}{q_k(1-c_k)} e^{\frac{-a_k^2}{2}}. \tag{7.5}$$

Finally, by using (7.5), we can write the approximate version of (7.3) as

$$\tilde{p}_d = \tilde{P}_{1q_1} + \sum_{k=2}^{N} \tilde{P}_{kq_k} \left(\prod_{j=1}^{k-1} (1 - \tilde{P}_{jq_j}) \right), \tag{7.6}$$

where $\tilde{p}_d$ denotes the approximation on p_d. Henceforth, using the approximated expression on p_d as above, we formulate an optimization problem in Problem 7.2 as shown below. Throughout this chapter, we refer to the solution of Problem 7.2, as the near-optimal ARQ distribution since the objective function in Problem 7.2 is an approximation on the objective function in Problem 7.1.

Problem 7.2. *For a given* $\mathbf{c}$ *and a high SNR* $\gamma = \frac{1}{\sigma^2}$, *solve*

$$q_1^*, q_2^*, \ldots q_N^* = \arg \mathop{\mathit{min}}_{q_1, q_2, \ldots q_N} \tilde{p}_d$$

$$\textit{subject to } q_k \geq 1, q_k \in \mathbb{Z}_+ \; \forall k, \sum_{i=1}^{N} q_i = \; q_{sum}.$$

7.3.2 Sufficient and Necessary Conditions on the Near-Optimal ARQ Distribution in CC-HARQ-SF Strategy

Before we obtain the necessary and sufficient conditions on the optimal ARQ distribution, we show that a link with a higher LOS component must not be given more ARQs than the link with a lower LOS component.

Theorem 7.2. *For a given LOS vector* $\mathbf{c}$ *and a high SNR regime in a CC-HARQ-SF based multi-hop network, the solution to Problem 7.2 satisfies the property that whenever* $c_i \geq c_j$*, we have* $q_i \leq q_j \; \forall \; i, j \; \in [N]$.

Proof. To highlight the location of c_i and c_j, we write $\mathbf{c}$ as $[c_1, c_2, \ldots, c_i, \ldots, c_j, \ldots, c_{N-1}, c_N]$ such that $j > i$. Assume that $c_j > c_i$, and q_i and q_j denote the number of ARQs allotted to the i-th and j-th link, respectively. Furthermore, let us suppose that $q_i = q_j = q$ and we have one ARQ with us. Now, the problem is deciding whether to allocate that additional ARQ to the i-th link or the j-link that results in lower PDP. Towards solving this problem, we consider an equivalent multi-hop network with LOS vector $\mathbf{c}' = [c_1, c_2, \ldots, c_{N-1}, \ldots, c_N, \ldots, c_i, c_j]$, wherein $\mathbf{c}'$ is obtained from $\mathbf{c}$ by swapping c_i with c_{N-1} and c_j with c_N. By using a similar approach given in Theorem 1 (from Chapter 3), we can show that the PDP of the multi-hop networks with the LOS vectors $\mathbf{c}$ and $\mathbf{c}'$ are identical. Furthermore, the PDP of the N-hop network with LOS vector $\mathbf{c}'$, is written as

$$\tilde{p}_d = \tilde{P}_{1q_1} + \tilde{P}_{2q_2}(1 - \tilde{P}_{1q_1}) + \ldots + \left(\tilde{P}_{iq_i} + \tilde{P}_{jq_j}(1 - \tilde{P}_{iq_i})\right) \prod_{k \in [N] \setminus \{i,j\}} (1 - \tilde{P}_{kq_k}).$$

Note that $\tilde{P}_{iq_i}$ and $\tilde{P}_{jq_j}$ appear only in the last term of the above expression. Since the question of allocating the additional ARQ is dependent only on the expression $\tilde{P}_{iq_i} + \tilde{P}_{jq_j}(1 - \tilde{P}_{iq_i})$, we henceforth do not use the entire expression for PDP. Additionally, since $q_i = q_j = q$, we obtain one of the following expressions when allocating the additional ARQ,

$$\begin{aligned} A &= \tilde{P}_{i(q+1)} + \tilde{P}_{jq}(1 - \tilde{P}_{i(q+1)}), \\ B &= \tilde{P}_{iq} + \tilde{P}_{j(q+1)}(1 - \tilde{P}_{iq}). \end{aligned}$$

Since $c_i < c_j$, we know that $\tilde{P}_{iq} > \tilde{P}_{jq}$ for $q = 1$. To prove the statement of the theorem, we use the approximation given in (7.5) to show that $A < B$. Therefore, using (7.5), we write

$$A = \frac{\phi e^{\frac{-a_i^2}{2}}}{(q+1)(1-c_i)} + \frac{\phi e^{\frac{-a_j^2}{2}}}{q(1-c_j)}\left(1 - \frac{\phi e^{\frac{-a_i^2}{2}}}{(q+1)(1-c_i)}\right),$$

$$B = \frac{\phi e^{\frac{-a_i^2}{2}}}{q(1-c_i)} + \frac{\phi e^{\frac{-a_j^2}{2}}}{(q+1)(1-c_j)}\left(1 - \frac{\phi e^{\frac{-a_i^2}{2}}}{q(1-c_i)}\right).$$

On solving the difference $A - B$, we get

$$\phi\left(\frac{e^{-a_j^2/2} - e^{-a_i^2/2} + c_j e^{-a_i^2/2} - c_i e^{-a_j^2/2}}{(1-c_i)(1-c_j)}\right)\underbrace{\left(\frac{1}{q} - \frac{1}{q+1}\right)}.$$

In the above equation, note that the second product term in the bracket is positive. Therefore, to prove that $A - B < 0$, first, we show that $e^{-a_j^2/2} - e^{-a_i^2/2} + c_j e^{-a_i^2/2} - c_i e^{-a_j^2/2} \le 0$ for any $i, j \in [N]$. To proceed further, we start by assuming that the above inequality is true, and by rewriting the above equation, we get $e^{-a_j^2/2}(1-c_i) \le e^{-a_i^2/2}(1-c_j)$ which in turns equal to $\dfrac{(1-c_i)}{(1-c_j)} \le \dfrac{e^{-a_i^2/2}}{e^{-a_j^2/2}}$. Now, by taking the logarithm on both sides and on expanding both a_i and a_j, we obtain the inequality $\log\left(\dfrac{1-c_i}{1-c_j}\right) \le \dfrac{c_j - c_i}{(1-c_i)(1-c_j)}$. Let $x = \dfrac{1-c_i}{1-c_j}$, and therefore, $x - 1 = \dfrac{1-c_i}{1-c_j} - 1 = \dfrac{c_j - c_i}{(1-c_j)}$. Furthermore, by using x, we can rewrite the above inequality as $\log x \le \dfrac{x-1}{(1-c_i)}$, where $\dfrac{1}{1-c_i} \ge 1$ because $c_i \le 1$. Moreover, by using the standard inequality i.e. $\log x \le (x-1)$, we can prove that the inequality $\log x \le \dfrac{x-1}{(1-c_i)}$ is true for any $i, j \in [N]$. Hence, this completes the proof that $A - B < 0$. □

Henceforth, we use the set $\mathbb{S} = \{\mathbf{q} \in \mathbb{Z}_{+}^{N} \mid \sum_{j=1}^{N} q_j = q_{sum} \;\&\; q_j \geq 1 \;\forall\, j\}$ to define the search space for the optimal ARQ distribution. Furthermore, for a given $\mathbf{q} \in \mathbb{S}$, its neighbors are defined as below.

Definition 7.1. *The set of neighbors for a given* $\mathbf{q} \in \mathbb{S}$ *is defined as* $\mathcal{H}(\mathbf{q}) = \{\bar{\mathbf{q}} \in \mathbb{S} \mid d(\mathbf{q}, \bar{\mathbf{q}}) = 2\}$, *where* $d(\mathbf{q}, \bar{\mathbf{q}})$ *denotes the Hamming distance between* $\mathbf{q}$ *and* $\bar{\mathbf{q}}$.

In the following definition, we present a local minima of the space $\mathbb{S}$ by evaluating the PDP of the CC-HARQ-SF based multi-hop network over the vectors in $\mathbb{S}$.

Definition 7.2. *To be a local minima of* $\mathbb{S}$*, an ARQ distribution* $\mathbf{q}^* \in \mathbb{S}$ *must satisfy the condition* $p_d(\mathbf{q}^*) \leq p_d(\mathbf{q})$, *for every* $\mathbf{q} \in \mathcal{H}(\mathbf{q}^*)$*, such that* $p_d(\mathbf{q}^*)$ *and* $p_d(\mathbf{q})$ *represent the PDP evaluated at the distributions* $\mathbf{q}^*$ *and* $\mathbf{q}$*, respectively.*

Using the above definition, we derive a set of necessary and sufficient conditions on the local minima in the following theorem.

Theorem 7.3. *For a given* N*-hop network with LOS vector* $\mathbf{c}$*, the ARQ distribution* $\mathbf{q}^* = [q_1^*, q_2^*, \ldots, q_N^*]$ *is said to be a local minima if and only if* q_i^* *and* q_j^* *for* $i \neq j$ *satisfy the following bounds*

$$q_j^{*2} K_i - q_j^* (K_i + K_j K_i) - C_1 \;\; \leq \;\; 0, \tag{7.7}$$

$$q_j^{*2} K_i + q_j^* (K_i - K_j K_i) - C_2 \;\; \geq \;\; 0, \tag{7.8}$$

where $C_1 = -K_j K_i + q_i^{*2} K_j + q_i^* (K_j - K_j K_i)$, $C_2 = K_j K_i + q_i^{*2} K_j - q_i^* (K_j + K_j K_i)$ *with* K_t *for* $t \in \{i, j\}$ *given by*

$$K_t = \frac{\phi}{1 - c_t} e^{\left(\dfrac{-c_t}{1 - c_t}\right)}. \tag{7.9}$$

Proof. According to Definition 7.1, it can be observed that a neighbor of $\mathbf{q}^*$ in the search space $\mathbb{S}$ differs in two positions with respect to $\mathbf{q}^*$. Let these neighbors are of the form $\hat{\mathbf{q}}_{+} =$

$[q_1^*, q_2^*, \ldots, q_i^* + 1, \ldots, q_j^* - 1, \ldots, q_N^*]$ and $\hat{\mathbf{q}}_- = [q_1^*, q_2^*, \ldots, q_i^* - 1, \ldots, q_j^* + 1, \ldots, q_N^*]$ that differs in two positions at i and j provided $q_i^* - 1 \geq 1$ and $q_j^* - 1 \geq 1$. Because of the type of search space and the expression of PDP, we invoke the results from Theorem 1 (Chapter 3) that the PDP remains identical after swapping intermediate links. Therefore, instead of considering the multi-hop network with LOS vector $\mathbf{c} = [c_1, c_2, \ldots, c_i, \ldots, c_j, \ldots, c_{N-1}, c_N]$, we consider its permuted version with the LOS vector $\mathbf{c} = [c_1, c_2, \ldots, c_{N-1}, \ldots, c_N, \ldots, c_i, c_j]$, wherein the i-th link is swapped with $(N-1)$-th link, and the j-th link is swapped with N-th link. Consequently, the local minima and its two neighbors are respectively of the form $\mathbf{q}^* = [q_1^*, q_2^*, \ldots, q_{N-1}^*, \ldots, q_N^*, \ldots, q_i^*, q_j^*]$, $\hat{\mathbf{q}}_+ = [q_1^*, q_2^*, \ldots, q_{N-1}^*, \ldots, q_N^*, \ldots, q_i^* + 1, q_j^* - 1]$ and $\hat{\mathbf{q}}_- = [q_1^*, q_2^*, \ldots, q_{N-1}^*, \ldots, q_N^*, \ldots, q_i^* - 1, q_j^* + 1]$. By using the definition of local minima, we have the inequalities

$$\tilde{p}_d(\mathbf{q}^*) \leq \tilde{p}_d(\hat{\mathbf{q}}_+), \text{ and } \tilde{p}_d(\mathbf{q}^*) \leq \tilde{p}_d(\hat{\mathbf{q}}_-), \tag{7.10}$$

where $\tilde{p}_d(\mathbf{q}^*)$, $\tilde{p}_d(\hat{\mathbf{q}}_+)$ and $\tilde{p}_d(\hat{\mathbf{q}}_-)$ represent the PDP evaluated at the distributions $\mathbf{q}^*$, $\hat{\mathbf{q}}_+$, and $\hat{\mathbf{q}}_-$, respectively. Because of the fact that $\hat{\mathbf{q}}_+$ and $\hat{\mathbf{q}}_-$ differ only in the last two positions and the structure of the PDP, it is possible to demonstrate that the inequalities in (7.10) are equivalent to

$$\begin{aligned} \tilde{P}_{iq_i^*} + \tilde{P}_{jq_j^*}\left(1 - \tilde{P}_{iq_i^*}\right) &\leq \tilde{P}_{i(q_i^*+1)} + \tilde{P}_{j(q_j^*-1)}(1 - \tilde{P}_{i(q_i^*+1)}), \\ \tilde{P}_{iq_i^*} + \tilde{P}_{jq_j^*}\left(1 - \tilde{P}_{iq_j^*}\right) &\leq \tilde{P}_{i(q_i^*-1)} + \tilde{P}_{j(q_j^*+1)}(1 - \tilde{P}_{i(q_i^*-1)}), \end{aligned} \tag{7.11}$$

respectively. First, let us proceed with (7.11) to derive a necessary and sufficient conditions on q_i^* and q_j^*. On expanding $\tilde{P}_{iq}$ and $\tilde{P}_{jq}$, and by using the approximation on outage probability given in (7.5), the inequality in (7.11) can be rewritten as

$$\frac{K_i}{q_i^*} + \frac{K_j}{q_j^*}\left(1 - \frac{K_i}{q_i^*}\right) \leq \frac{K_i}{q_i^* + 1} + \frac{K_j}{q_j^* - 1}\left(1 - \frac{K_i}{q_i^* + 1}\right),$$

where K_i and K_j can be obtained from (7.9). On solving the above equation, we obtain

$$\frac{K_i}{q_i^*} - \frac{K_i}{q_i^* + 1} + \frac{K_j}{q_j^*} - \frac{K_j}{q_j^* - 1} \leq \frac{K_iK_j}{q_i^*q_j^*} - \frac{K_iK_j}{(q_i^* + 1)(q_j^* - 1)}.$$

After further modifications, we can rewrite the above equation as

$$q_j^{*2}K_i - q_j^*(K_i + K_iK_j) \leq -K_iK_j + q_i^{*2}K_j + q_i^*(K_j - K_iK_j). \tag{7.12}$$

In the above equation, we can replace $(-K_iK_j + q_i^{*2}K_j + q_i^*(K_j - K_iK_j))$ by C_1, and by rearranging the terms, we get (7.7). This completes the proof for the first necessary condition. The second necessary condition for (7.11) can be proved along the similar lines of the above proof to obtain (7.8). It can be observed that the two conditions of this theorem are also sufficient because the bounds are obtained by rearranging the terms in the condition on local minima. □

7.4 Low-Complexity List-Decoding Algorithm

Using Theorem 7.3, we are ready to synthesize a low-complexity algorithm to solve Problem 7.2.

proposition 7.1. *If the ARQ distribution* $\mathbf{q}$ *is chosen such that* $\frac{q_j^*}{q_i^*} = \sqrt{\frac{K_j}{K_i}}$*, for* $i \neq j$*, then* $\mathbf{q}$ *is a local minima of the search space at a high SNR.*

Proof. At a high SNR, ϕ is very small, and therefore, the product K_iK_j is negligible. Therefore, we can rewrite (7.7) and (7.8) as

$$\begin{aligned} q_j^{*2}K_i - q_j^*K_i &\leq q_i^{*2}K_j + q_i^*K_j, \\ q_j^{*2}K_i + q_j^*K_i &\geq q_i^{*2}K_j - q_i^*K_j, \end{aligned}$$

respectively. On rearranging the above equations, we get

$$\begin{aligned} q_j^{*2}K_i - q_i^{*2}K_j &\leq q_j^*K_i + q_i^*K_j, \\ q_j^{*2}K_i - q_i^{*2}K_j &\geq -q_j^*K_i - q_i^*K_j, \end{aligned}$$

respectively. Now, by equating $q_j^*K_i + q_i^*K_j = \epsilon_{i,j}$, the above equations become

$$q_j^{*2}K_i - q_i^{*2}K_j \leq \epsilon_{i,j}, \tag{7.13}$$

$$q_j^{*2}K_i - q_i^{*2}K_j \geq -\epsilon_{i,j}, \tag{7.14}$$

where $\epsilon_{i,j}$ is a positive number dependent on K_i, K_j, q_i^*, q_j^*. If we choose q_i^* and q_j^* such that $q_j^{*2}K_i - q_i^{*2}K_j = 0$, for $i \neq j$, then it ensures that the inequalities in (7.13) and (7.14) are trivially satisfied. Thus, choosing

$$\frac{q_j^*}{q_i^*} = \sqrt{\frac{K_j}{K_i}}, \tag{7.15}$$

for every pair $i, j \in [N]$ such that $i \neq j$, satisfies the sufficient conditions given in (7.13) and (7.14) at high SNR values. This completes the proof. □

Based on the results in Proposition 7.1, we formulate Problem 7.3, as given below, as a means of solving Problem 7.2 at a high SNR.

Problem 7.3. *For a given* $\{K_1, K_2, \ldots, K_N\}$ *and* q_{sum}*, find* $q_1^*, q_2^*, \ldots q_N^*$ *such that*

$$\frac{q_j^*}{q_i^*} = \sqrt{\frac{K_j}{K_i}}, \ \forall\, i, j \in [N] \text{ where } i \neq j,$$

$q_i^* \geq 1, q_i^* \in \mathbb{Z}_+, \ \forall i, \ \sum_{i=1}^{N} q_i^* = \ q_{sum}$.

However, from Problem 7.3, it is observed that a solution is not guaranteed because the ratio $\sqrt{\frac{K_j}{K_i}}$, which is computed based on the LOS components and the SNR, need not be in $\mathbb{Q}$. Therefore, we first propose a method to solve Problem 7.3 without the integer constraints, i.e., to find an ARQ distribution $\mathbf{q} \in \mathbb{R}^N$. This type of problem can be solved by using the system of linear equations of the form $\mathbf{Rq} = \mathbf{s}$ to obtain $\mathbf{q}_{real} = \mathbf{R}^{-1}\mathbf{s}$, where $\mathbf{q} = [q_1, q_2, \ldots, q_N]^T$, $\mathbf{s} = [0, 0, \ldots, 0, q_{sum}]^T$ and $\mathbf{R} \in \mathbb{R}^{N \times N}$ such that $\mathbf{R}(j, j) = 1$ for $1 \leq j \leq N$, $\mathbf{R}(N, j) = 1$, for $1 \leq j \leq N$, $\mathbf{R}(j, j+1) = -r_{j,j+1}$, for $r_{j,j+1} \triangleq \sqrt{\frac{K_{j+1}}{K_j}}$ and $1 \leq j \leq N-1$, and the rest of the entries of $\mathbf{R}$ are zeros. Subsequently, using the solution $\mathbf{q}_{real}$ in $\mathbb{R}^N$, we propose to obtain a solution in the true search space $\mathbb{S}$ by generating a list as given in Algorithm 7 (also discussed in Chapter 3). Finally, once the list is generated, the ARQ distribution that minimizes the PDP would be the solution of the proposed algorithm. In the next section, we present the extensive simulation results to show the performances of our theoretical results.

Algorithm 7 List Creation Based Algorithm

Require: $\mathbf{R}$, $\mathbf{s}$, q_{sum}, $\mathbf{c} = [c_1, c_2, \ldots, c_N]$

Ensure: $\mathcal{L} \subset \mathbb{S}$ - List of ARQ distributions in search space $\mathbb{S}$.

1: Compute $\mathbf{q}_{real} = \mathbf{R}^{-1}\mathbf{s}$, $\tilde{\mathbf{q}} = \lceil \mathbf{q}_{real} \rceil$.
2: **for** $i = 1 : N$ **do**
3:     **if** $\tilde{q}_i = 0$ **then**
4:         $\tilde{q}_i = \tilde{q}_i + 1$
5:     **end if**
6: **end for**
7: Compute $E = \left(\sum_{i=1}^{N} \tilde{q}_i\right) - q_{sum}$
8: $\mathcal{L} = \{\mathbf{q} \in \mathbb{S} \mid d(\mathbf{q}, \tilde{\mathbf{q}}) = E, q_j \ngtr q_i \text{ for } c_i < c_j\}$.

7.5 Simulation Results

In the first part of this section, we present simulation results on the delay profiles of the packets under the CC-HARQ-SF protocol. We show that when the ACK/NACK delay overheads are sufficiently small as compared to the payload, there is a high probability with which the packets arrive at the destination within the given deadline. To obtain the results, first, we take $q_{sum} = \lfloor \frac{\tau_{total}}{\tau_p + \tau_d} \rfloor$, by assuming $\tau_{NACK} = 0$ and $\tau_p + \tau_d = 1$ microsecond where $\tau_{total}, \tau_{NACK}, \tau_p$ and τ_d (with μs unit) are defined in Chapter 1. Then, we consider non-zero values of $\tau_{NACK} = \{0.05, 0.2, 0.8\}\mu s$ during the packet flow, and present the results for the following metrics: (i) the number of packets that did not reach the destination due to lack of ARQs present at the intermediate nodes (defined by P_{drop}), (ii) the number of packets that reach the destination

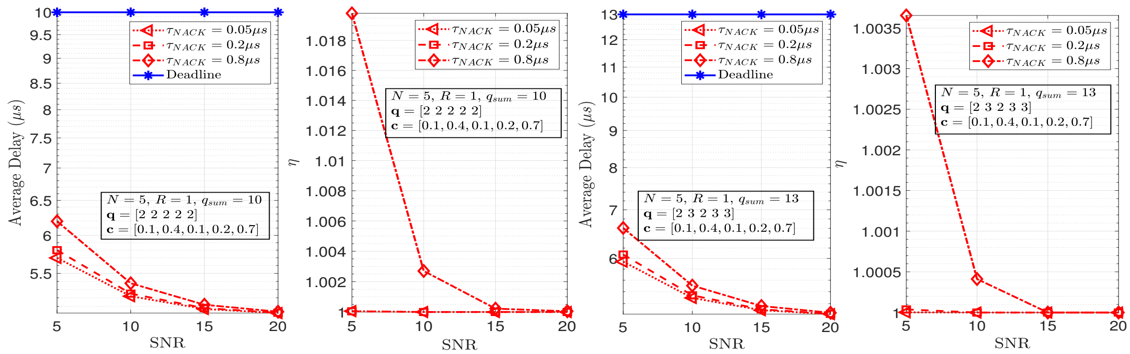

Figure 7.3: Illustration on the average delay on packets and deadline violation parameter (η) for several values of $\tau_{NACK} = \{0.05, 0.2, 0.8\}$ microseconds while implementing CC-HARQ-SF strategy.

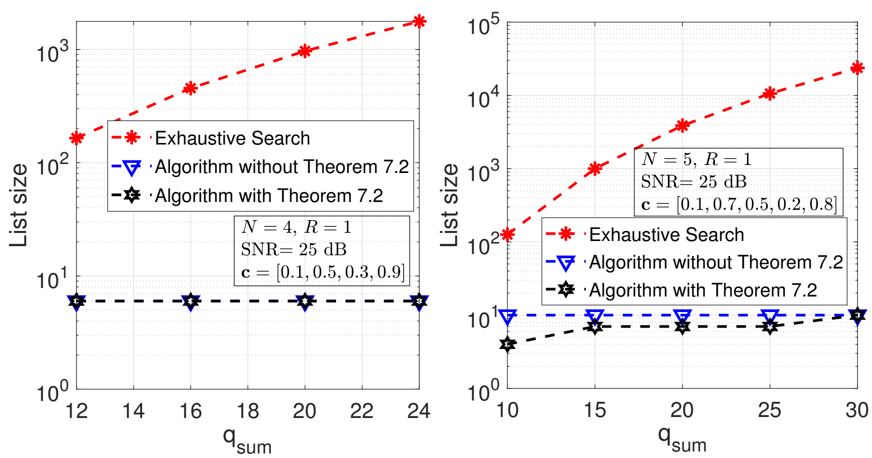

Figure 7.4: Comparison of list size between exhaustive search and the proposed low-complexity algorithm with and without Theorem 7.2 .

after the given deadline (defined by $P_{deadline}$), and (iii) the average delay on the packets. The plots are as shown in Fig. 7.3 for a given N and a LOS vector $\mathbf{c}$ with different q_{sum} and various SNR values. It can be observed that when τ_{NACK} is very small, specifically at values $\tau_{NACK} = \{0.05, 0.2\}$ microseconds, the average delay is sufficiently lower than the q_{sum} (the

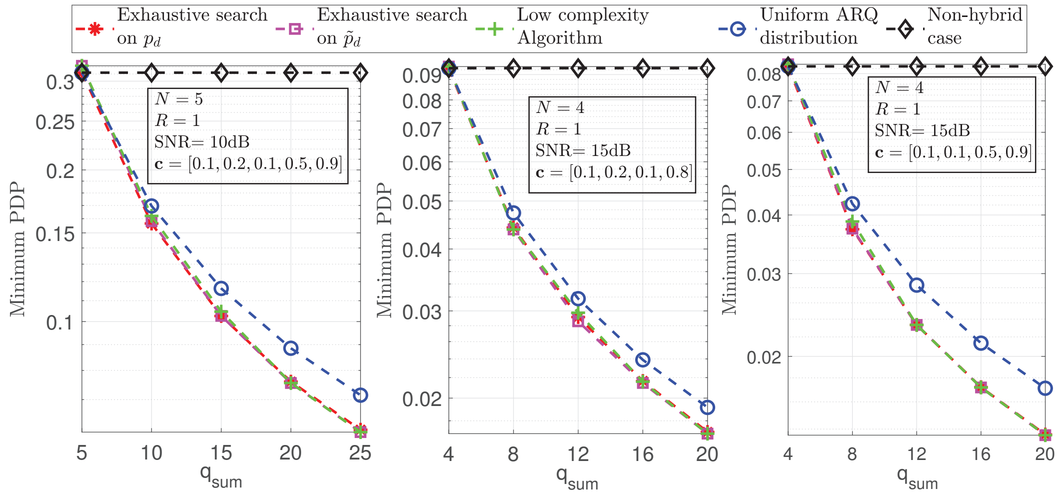

Figure 7.5: PDP comparison in CC-HARQ-SF based non-cumulative network between (i) exhaustive search using original PDP expression, p_d, (ii) exhaustive search using approximated PDP expression, $\tilde{p}_d$, (iii) proposed low-complexity algorithm (given in Algorithm 7), and (iv) uniform ARQ distribution along with ARQ distribution in a non-HARQ strategy (as discussed in Chapters 3, 4). With uniform distribution, each link is first distributed with $\lfloor q_{sum}/N \rfloor$ ARQs, and the remaining ARQs are equally shared by the first q_{sum} mod N links.

deadline). Also, the deadline violation parameter $\eta = \frac{P_{drop}+P_{deadline}}{P_{drop}}$ is nearly or equal (in the case of $\tau_{NACK} = 0.05\mu s$) to one. However, when $\tau_{NACK} = 0.8\mu s$, the average delay slightly moves towards the deadline and also, η exceeds one, showing the deadline violation of the packets.

In the rest of this section, we present the simulation results to validate our theoretical analysis and to showcase the benefits of using the CC-HARQ-SF protocol over a non-HARQ based multi-hop model. On the one hand, the computational complexity for solving Problem 7.1 and Problem 7.2 using an exhaustive search is $\binom{q_{sum}-1}{N-1}$. On the other hand, the computational complexity of our method is determined by the complexity of computing the inverse of a matrix and that of

Algorithm 7. Furthermore, while the complexity for computing the inverse is $O(N^3)$, the number of computations for generating the list is bounded by $\binom{N}{E}$, where E is the leftover number of ARQs after the ceiling operation. To demonstrate the benefits of using our low-complexity method, we plot the curves for the list size of an exhaustive search and the proposed algorithm in Fig. 7.4 for several values of q_{sum} with $N = 4$ and $N = 5$. Specifically, we plot the list size both with and without incorporating the results of Theorem 7.2. When using Theorem 7.2, by subtracting one ARQ from all possible $\binom{N}{E}$ positions from $\tilde{\mathbf{q}}$, we discard those ARQ distributions which follow the rule $q_i > q_j$ whenever $c_i > c_j$. As a result, we see that the list size shortens after incorporating the rule of Theorem 7.2 for $N = 5$. Furthermore, based on the simulation results, we observe that the ARQ distribution, which minimizes the PDP from the list $\mathcal{L}$ matches the result of exhaustive search, thereby confirming that our list contains the optimal ARQ distribution. Although, we used high SNR results of Proposition 7.1 to synthesize the list-decoding method, we observe that the size of the list reduces significantly at low and medium-range of SNR values.

Finally, to showcase the benefits of using CC-HARQ-SF in a multi-hop network, we plot the minimum PDP against several values of q_{sum} in Fig. 7.5 by using (i) an exhaustive search on p_d (as given in (7.3)), (ii) an exhaustive search on $\tilde{p}_d$ (as given in (7.6)), (iii) the proposed low-complexity algorithm, (iv) uniform ARQ distribution in CC-HARQ-SF based strategy, and (v) non-HARQ strategy. There are mainly two observations from the simulation results in Fig. 7.5: (a) the PDP improves in the case of optimal ARQ distribution (with the exhaustive search on both p_d and $\tilde{p}_d$) over uniform ARQ distribution and non-hybrid based ARQ distribution, and (b) our proposed approximation $\tilde{p}_d$ produces reasonably accurate results.

In the next section, we discuss the fully-cumulative strategy in CC-HARQ-SF based multi-hop model with slow-fading intermediate channels.

7.6 CC-HARQ-SF based Fully-Cumulative Network

In the CC-HARQ-SF based non-cumulative network, the residual ARQs at any node cannot be transferred to its succeeding nodes. This is because the nodes in the network do not have the knowledge on the ARQs given to the other nodes, thereby leading to wastage of ARQ resources. Due to this limitation of the CC-HARQ-SF based non-cumulative network, in this section, we consider a setup where every node is aware of the ARQs assigned to the node before it, unlike the non-cumulative model. Therefore, the following node can use the unused ARQs from the preceding node. In addition, we also assume that the packet structure contains a particular portion, referred to as the *counter*, in order to carry the number of ARQs not utilized by the preceding nodes in the network. We emphasize that the need to include the counter in the packet structure has a vital role since a given relay node can only hear the residual ARQs from the node immediately preceding one [30, Section V]. When using a counter, every node can use the residual ARQs from all its preceding nodes in addition to its ARQs, thus reducing the end-to-end PDP. Henceforth, throughout this chapter, we refer to this multi-hop network as the CC-HARQ-SF based fully-cumulative network.

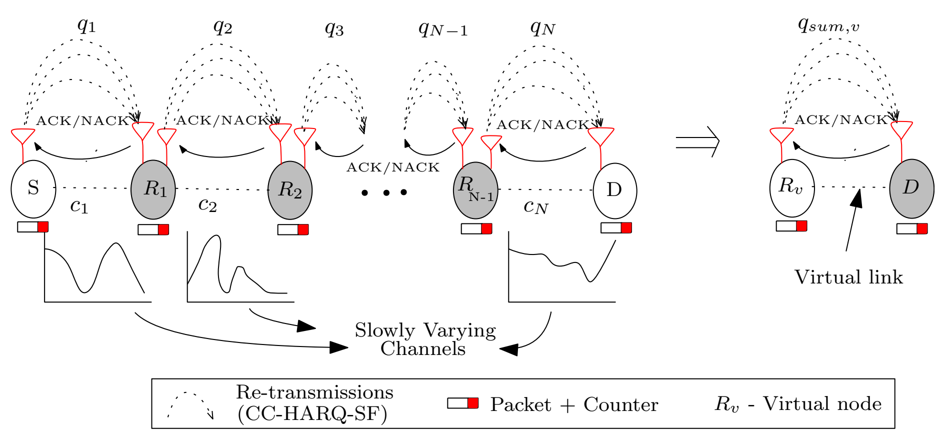

Figure 7.6: An illustration for a CC-HARQ-SF based fully-cumulative network.

To explain the model further, assume that on the first link, $q_1^{'}$ out of q_1 ARQs are used to successfully transmit the packet to the next node by implementing CC-HARQ-SF, and as a consequence, $q_1^{''} = q_1 - q_1^{'}$ ARQs are not utilised. Since the following node in the chain is aware of both q_1 ARQs and the number of attempts made by the preceding node for the successful transmission of the packet, it can use $q_2 + q_1^{''}$ ARQs to transmit the packet to its successor node. Furthermore, to assist the successor node in the chain to make aware of any residual number of ARQs, the second node encodes the number $q_2 + q_1^{''}$ in the counter and then starts transmitting the packet to the successor node. If the second node uses $q_2^{'}$ attempts, then the third node (which is the successor node in the chain) can use $q_3 + q_2 + q_1^{''} - q_2^{'}$ number of attempts. Similarly, the third node encodes the number $q_3 + q_2 + q_1^{''} - q_2^{'}$ in the counter and then starts transmitting the packet to its successor node. This way, each intermediate relay node gains additional ARQ for packet transmission by using (i) the knowledge of the number of ARQs allotted to itself, (ii) the number encoded in the counter portion of the packet, and (iii) the number of unsuccessful attempts made by the preceding node. We emphasize that every relay node updates the counter in the packet only once, and as a result, the additional delay incurred by this update process is negligible. An illustration of the fully-cumulative ARQ scheme is given in Fig. 7.6. In this scheme, note that as the preceding node may have more ARQs than allotted to it (because of additional ARQ accumulated from its preceding node), the next node in the chain does not know the maximum number of attempts that would be made by its preceding node. Therefore, a given node, upon sending a NACK to its preceding node, will wait for the packet only for a specified amount of time, say τ seconds. If it does not receive the packet within τ units of time, then it implies that the preceding node has exhausted the allotted number of ARQs, and therefore, the packet is said to be dropped in the network. It is intuitive that this process of cumulatively adding the unused ARQs at each hop will reduce the PDP in comparison with the non-cumulative strategy without changing the sum constraint $q_{\text{sum}} = \sum_{i=1}^{N} q_i$. To showcase the benefits, we have shown the simulation results for a 3-hop network in Fig. 7.7 where it can be observed that the CC-HARQ-SF based fully-cumulative network outperforms the CC-HARQ-SF based non-cumulative network.

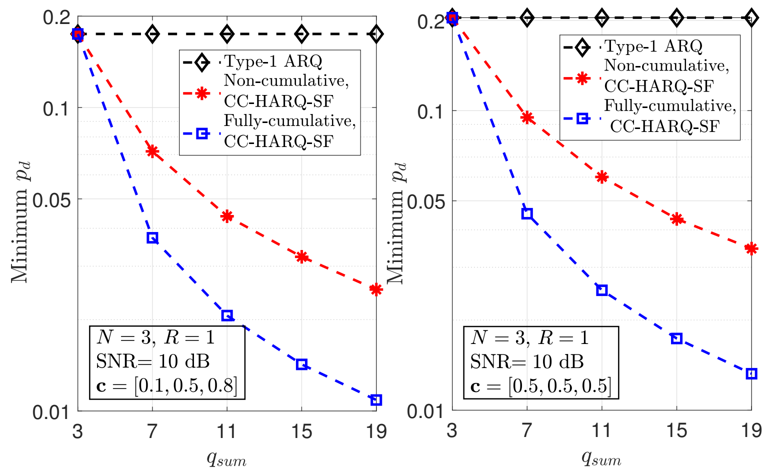

Figure 7.7: Comparison of the minimum PDP, denoted by p_d, for : (i) CC-HARQ-SF based non-cumulative network, (ii) CC-HARQ-SF based fully-cumulative network, and (iii) non-hybrid ARQ strategy .

The PDP expression for this scheme can be obtained similar to that of (7.3), however, the residual ARQs from the preceding nodes are also taken into account when computing the PDP at each node. For example, for a 3-hop network with the ARQ distribution $\mathbf{q} = [q_1, q_2, q_3]$, the PDP expression can be given as

$$\begin{aligned} p_d \;=\; & P_{1q_1} + \sum_{i=0}^{q_1-1}(P_{1i} - P_{1(i+1)})P_{2(q_2+q_1-(i+1))} + \sum_{i=0}^{q_1-1}(P_{1i} - P_{1(i+1)}) \sum_{j=0}^{q_2+q_1-(i+2)} \\ & (P_{2j} - P_{2(j+1)})P_{3(q_3+q_2+q_1-(i+j+2))}. \end{aligned} \tag{7.16}$$

In the above equation, $P_{ki} - P_{k(i+1)}$ represents that the packet at k-th hop (where $k \in [1, 3]$) gets dropped till its i-th attempt and is transmitted successfully to its next node at $(i + 1)$-th attempt. In a similar manner, we can write the PDP expression for any N-hop network. Now, we present the result on the optimal ARQ distribution for the CC-HARQ-SF based fully-cumulative network.

Theorem 7.4. *For a given ARQ distribution* $\mathbf{q} = [q_1, q_2, \ldots, q_N]$ *and* q_{sum} *in a CC-HARQ-SF based fully-cumulative network, the optimal ARQ distribution can be given by* $[q_{sum}, 0, \ldots, 0]$.

Proof. First, we want to prove the result for $N = 2$. With ARQ distribution $[q_1, q_2]$, the PDP can be written as

$$pdp_2 = P_{1q_1} + \sum_{i=0}^{q_1-1} (P_{1i} - P_{1(i+1)}) P_{2(q_2+q_1-(i+1))}, \tag{7.17}$$

where $P_{10} = 1$. Furthermore, let us transfer one ARQ from the second link to the first link by considering that $q_2 > 1$. Therefore, the PDP expression in this case can be written as

$$\begin{aligned} pdp_2^{s'} &= P_{1(q_1+1)} + \sum_{i=0}^{q_1} (P_{1i} - P_{1(i+1)}) P_{2(q_2-1+q_1+1-(i+1))}, \\ &= P_{1(q_1+1)} + \sum_{i=0}^{q_1} (P_{1i} - P_{1(i+1)}) P_{2(q_2+q_1-(i+1))}. \end{aligned}$$

On calculating $pdp_2 - pdp_2^{s'}$, we get

$$\begin{aligned} pdp_2 - pdp_2^{s'} &= P_{1q_1} - P_{1(q_1+1)} + \sum_{i=0}^{q_1-1} (P_{1i} - P_{1(i+1)}) P_{2(q_2+q_1-(i+1))} \\ &\quad - \sum_{i=0}^{q_1} (P_{1i} - P_{1(i+1)}) P_{2(q_2+q_1-(i+1))}, \\ &= (P_{1q_1} - P_{1(q_1+1)})(1 - P_{2(q_2-1)}). \end{aligned}$$

It can be observed that $P_{1q_1} > P_{1(q_1+1)}$ and $(1 - P_{2(q_2-1)}) \geq 0$, because $P_{2(q_2-1)} < 1$. Therefore, we conclude that the above equation results in a positive number and hence, $pdp_2 > pdp_2^{s'}$. Furthermore, if we keep transferring one ARQ from the second link to the first link and calculating the difference between the old PDP and the new PDP (after transferring one ARQ), the difference is always positive. In this way, the optimal ARQ distribution can be given by $[q_1 + q_2, 0]$. This completes the proof for $N = 2$.

Now, by using the hypothesis step of induction, let us assume that the result is true for $N = k$. To prove the result for $N = (k+1)$, let $\mathbf{q} = [q_1, q_2, \ldots, q_{k+1}]$ be the ARQ distribution across the $k+1$ nodes. Since, the nodes are applying the fully-cumulative scheme, we assume that the set of nodes from the 2^{nd} to the $(k+1)$-th node as a single virtual node. Henceforth, we refer to this virtual node as R_v. After forming a virtual node, now, we have a network comprising source node, R_v and a destination node. Also, the R_v node has given total $q_{sum,v} = \sum_{i=2}^{k+1} q_i$ number of transmissions and $q_{sum,v}$ is internally distributed among the k nodes constituting the virtual node, R_v. Let $PDP_v(q_{sum,v})$ represent the PDP at the virtual node. Now, the PDP expression for this above mentioned two-hop network can be written as

$$PDP_{2v} = P_{1q_1} + \sum_{i=0}^{q_1-1} (P_{1i} - P_{1(i+1)}) PDP_v(q_{sum,v} + q_1 - (i+1)), \tag{7.18}$$

where $P_{10} = 1$. Now, in the above equation, it can be observed that for the fixed q_1, P_{1i} and $P_{1(i+1)}$ for each iteration 'i', the PDP expression can be minimized by minimizing the $PDP_v(q_{sum,v} + q_1 - (i+1))$. Also, by using the assumption on $N = k$ from the induction step, we know that $[q_{sum}, 0, \ldots, 0]$ will minimize PDP_v. Furthermore, for (7.18), we have the ARQ distribution $[q_1, q_{sum,v}, 0, \ldots, 0]$. In order to find the optimal ARQ distribution of (7.18), we use similar approach as for $N = 2$. That is, we start transferring one ARQ from the virtual node to the first link and write the new PDP expression as

$$\begin{aligned} PDP_{2v}^{s'} &= P_{1(q_1+1)} + \sum_{i=0}^{q_1} (P_{1i} - P_{1(i+1)}) PDP_v(q_{sum,v} - 1 + q_1 + 1 - (i+1)) \\ &= P_{1(q_1+1)} + \sum_{i=0}^{q_1} (P_{1i} - P_{1(i+1)}) PDP_v(q_{sum,v} + q_1 - (i+1)) \end{aligned}$$

On calculating $PDP_{2v} - PDP_{2v}^{s'}$, we get

$$PDP_{2v} - PDP_{2v}^{s'} = (P_{1q_1} - P_{1(q_1+1)})(1 - PDP_v(q_{sum,v} - 1))$$

It can be observed that $0 \geq P_{1q_1} - P_{1(q_1+1)} \leq 1$ and $(1 - PDP_v(q_{sum,v} - 1)) \geq 0$, therefore, it implies $PDP_{2v} \geq PDP_{2v}^{s'}$. Hence, on transferring one ARQ from the virtual node to the first link, the PDP will decrease. Therefore, we can conclude that the optimal ARQ distribution is $[q_{sum}, 0, \ldots, 0]$ for a fully-cumulative network. $\square$

7.7 Delay Analysis for CC-HARQ-SF based Non-Cumulative Network

In this section, we present a detailed analysis on end-to-end delay for CC-HARQ-SF strategies. First, we show that packets in a non-cumulative network have a higher probability of reaching the destination before the deadline, assuming that the delay overheads from ACK/NACK are sufficiently small. To demonstrate the results, we obtain q_{sum} as $\lfloor \frac{\tau_{total}}{\tau_p + \tau_d} \rfloor$ without considering the resources for ACK/NACK in the reverse channel, where τ_{total}, τ_d and τ_p are as defined in Chapter 1. Subsequently, we introduce different resolution of delays from NACK, say τ_{NACK} time units, and then observe the impact on the end-to-end delay on the packets. Assuming $\tau_p + \tau_d = 1$ microsecond, we set the deadline for end-to-end packet delay as q_{sum} microseconds. Then, by sending an ensemble of 10^6 packets to the destination through the CC-HARQ-SF strategy, we compute the following metrics when $\tau_{NACK} \in \{0.05, 0.2, 0.8\}$ microseconds: (i) the fraction of packets that were dropped in the network (denoted by P_{Drop}) due to insufficient ARQs at the intermediate nodes, (ii) the fraction of packets that reach the destination after the deadline (denoted by $P_{Deadline}$), and finally, (iii) the average end-to-end delay on the packets. These metrics are plotted in Fig. 7.8 for various values of SNR for a specific value of N and the LOS vector $\mathbf{c}$. The plots suggest that the average delay is significantly lower than that of the deadline especially when τ_{NACK} is small, owing to the opportunistic nature of CC-HARQ-SF strategies. However, as τ_{NACK} increases, the average delay is pushed slightly closer to the deadline. Furthermore, to capture the behaviour of deadline violations due to higher τ_{NACK}, in Fig. 7.8, we also plot $\eta = \frac{P_{Drop} + P_{Deadline}}{P_{Drop}}$. The plots confirm that when τ_{NACK} is sufficiently

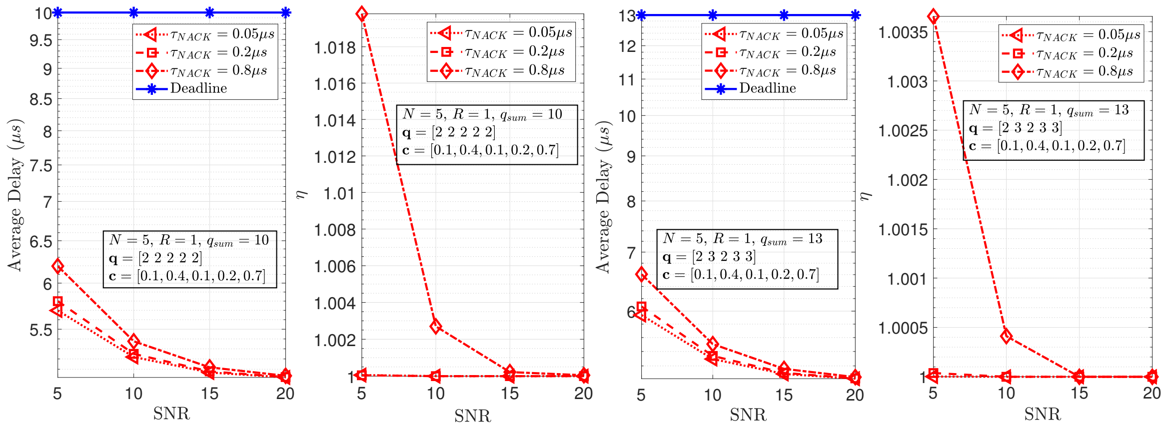

Figure 7.8: Variation of average delay on the packets and the deadline violation parameter (η) for various τ_{NACK} in CC-HARQ-SF based strategies at SNR= 0 dB.

small compared to $\tau_p + \tau_d$ (see $\tau_{NACK} = 0.05\mu s$ at SNR = 10, 15, 20 dB), the packets that reach the destination arrive within the deadline with an overwhelming probability as $\eta = 1$ at those values. Owing to the use of ACK/NACK, [30] has already shown that the average delay offered by the ARQ based strategy is much smaller compared to other strategies for packet retransmissions. We highlight that our results improve upon that of [30] because of the combining nature of the ARQs at the receiver.

In the rest of this section, we present the other delay metrics for our non-cumulative and fully-cumulative strategies. Let us assume that we want to secure the packets from eavesdropping, and therefore, every node in the cluster encrypts the counter portion of the packet after updating it with the residual ARQs. As a result, when the packet is successfully decoded at the next node, it needs to decrypt it by using an appropriate crypto-primitive. Since this procedure results in an additional processing delay on the packet, we represent this delay by T_c microseconds. Assuming that the delay introduced on the packet per hop for each transmission is $T = \tau_p + \tau_d = 1$ microsecond. Now, we analyse the effect of crypto-primitives on end-to-end delay by choosing $T_c = \alpha T$, where $\alpha = 0, 0.5,$ and 1. In Fig. 7.9, we have shown the effect of α on the end-to-end delay by plotting the delay profiles (in percentage) of both non-cumulative and fully-cumulative strategies. The delay profile refers to the percentage of packets reaching the destination in a certain amount of time. For generating the plots, we have used an ensemble of 10^6 packets and we considered $N = 5$, $R = 1$, SNR= 5 dB, $q_{sum} = 12$, and $\mathbf{c} = \{0.1, 0.5, 0.1, 0.3, 0.7\}$. It can be noted that the percentage of packets that reaches the destination after the given deadline can be treated under the category of deadline violation. Evidently, delay profiles are same for the non-cumulative strategy irrespective of the value of α, because there is no counter present in it. However, it can be observed in a fully-cumulative network that as α increases, the percentage of packets violating the deadline (where deadline is $12\mu s$) increases. It can also be visualized by the width of the rectangle that as α increases, the width of the rectangle increases (wherein the rectangle shows the percentage of packets violating the deadline). Furthermore, if the effect of

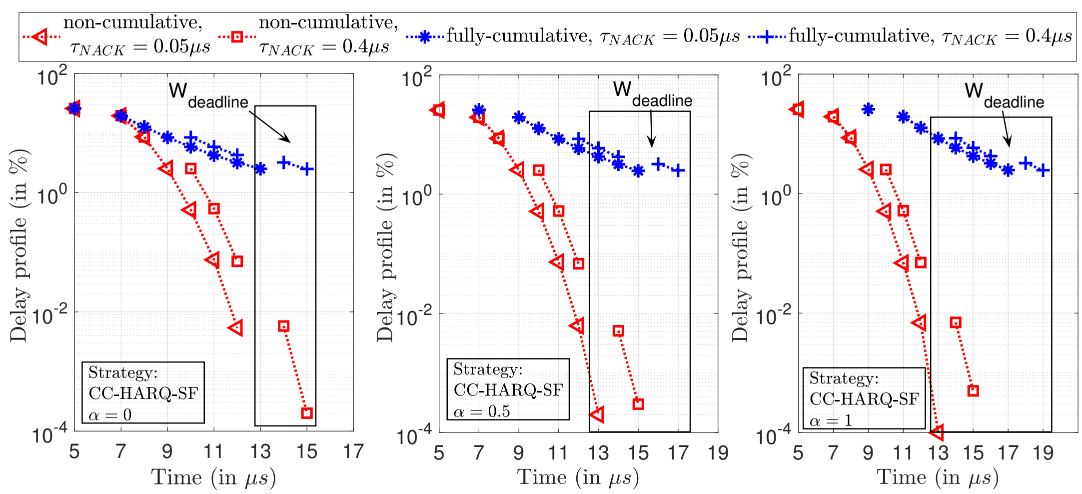

Figure 7.9: Simulation results on delay profiles for CC-HARQ-SF based strategies using a 5-hop network with $\mathbf{c} = [0.1, 0.5, 0.1, 0.3, 0.7]$ and $q_{sum} = 12$ at rate $R = 1$ and SNR= 5dB with 10^6 packets wherein some packets are dropped either due to outage and some are dropped due to deadline violation (marked in the rectangle).

α is not considered when designing q_{sum}, when $\alpha = 0$, then there is a non-zero probability that

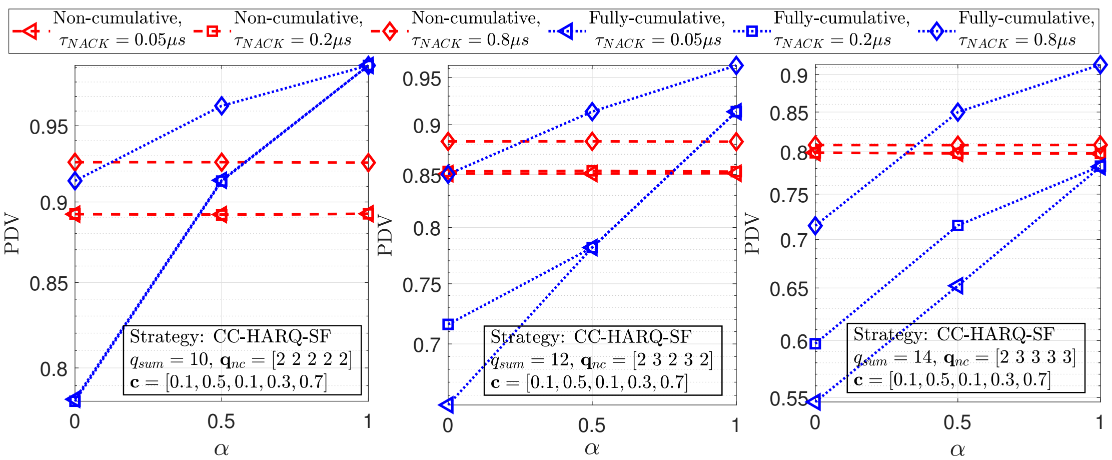

Figure 7.10: Illustration on PDV for non-cumulative and fully-cumulative network while implementing CC-HARQ-SF strategies for different τ_{NACK} and α values.

some packets may reach the destination beyond the deadline, however it is very small (see the first plot in Fig.7.9).

Now, in addition to PDP, we define a new metric referred to as probability of deadline violation (PDV), which can be defined as the PDV $= \frac{P_{drop}+P_{deadline}}{\text{Total packets}}$. In Fig. 7.10, we plot the PDV for our CC-HARQ-SF based non-cumulative and fully-cumulative networks as a function of α for a 5-hop network with different q_{sum}. In the plots, $\mathbf{q_{nc}}$ represents the ARQ distribution taken for generating the plots for non-cumulative strategy and $\{q_{sum}, 0, \ldots, 0\}$ is the ARQ distribution for the fully-cumulative network. Also, for a given q_{sum}, the deadline for packets to reach the destination is q_{sum} microseconds. The plots confirm that: (i) PDV of the non-cumulative network do not change with α, (ii) PDV of the fully-cumulative network increases with increasing values of α; this is because $N-1$ nodes made use of the counter in the packet, thereby adding a significant delay of $(N-1)T_c$ microseconds to the packet (because in fully-cumulative network, the nodes except the first node use the counter).

7.8 Summary

In this chapter, we have presented a novel CC-HARQ-SF framework to achieve high reliability with bounded-delay constraints in a slow-fading multi-hop network. In particular, by using the CC-HARQ-SF protocol at the intermediate relay, we have posed the problem of distributing a given number of ARQs such that the PDP is minimized. We have shown that the problem is non-tractable because the underlying PDP expression contains the Marcum-Q function. Towards solving the problem, (i) we have provided a novel approximation on the Marcum-Q function in the high SNR regime and have subsequently used the approximation by formulating a similar optimization problem, (ii) we have derived a set of necessary and sufficient conditions on the near-optimal ARQ distribution by using the approximated version of the optimization problem, and finally, (iii) we have proposed a low-complexity list-based algorithm to solve the optimization problem that yields near-optimal ARQ distribution. We have validated our approximation and the efficacy of our algorithm through extensive simulation results.

Chapter 8

CC-HARQ Strategies in Multi-Hop Networks under Fast-Fading Scenario

8.1 Introduction

In the previous chapter, we discussed the CC-HARQ strategy over a multi-hop model in a slow-fading scenario. However, when the channels are not static and change over multiple ARQ attempts, the solutions and the analysis on PDP from Chapter 7 are no longer applicable. Therefore, in this chapter, we address the issue of obtaining optimal ARQ distribution for the CC-HARQ strategy under a fast-fading scenario [1]. In particular, the main contributions of this chapter are summarized as follows:

1. We deal with CC-HARQ based strategies in fast-fading scenarios (denoted as CC-HARQ-FF), wherein the channels are statistically independent across allotted attempts at each link. Similar to CC-HARQ-SF, we propose non-cumulative and fully-cumulative strategies under this case. For the non-cumulative strategy, we provide a closed-form expression on

[1]Parts of the result presented in this chapter are available in publications [37]

PDP, and show that the optimization problem is non-tractable and more challenging (as compared to CC-HARQ-SF) as it contains higher-order Marcum-Q functions. Therefore, to obtain near-optimal ARQ distributions, we first propose approximations on the Marcum-Q functions of higher-orders, and then use them to formulate a new optimization problem. Furthermore, we derive necessary and sufficient conditions on the solutions for the new optimization problem, and then propose a low-complexity algorithm to solve the same. Using extensive simulations, we showcase the accuracy of our theoretical results and the proposed algorithms (see Section 8.3). For the non-cumulative strategy, we provide theoretical results on the optimal ARQ distribution for a fully-cumulative network in closed-form, and show that the fully-cumulative strategy provides lower PDP than the non-cumulative counterpart at the cost of marginal increase in communication-overhead (see Section 8.4).

2. For each of our strategies, we present a detailed analysis on end-to-end delay by considering the following metrics: (i) average end-to-end delay, (ii) packet deadline violation (PDV), which is defined by the number of packets reaching the destination after the given deadline, and (iii) delay profile, which represents the percentage of packets reaching the destination at a certain time for a given deadline. By using the aforementioned delay-metrics, we provide valuable insights on the merits and demerits of our strategies in achieving high-reliability with bounded constraints on end-to-end delay (see Section 8.5).

3. Finally, we also address the open question of whether CC-HARQ based strategy outperforms performs Type-1 ARQs in a multi-hop network. Surprisingly, in this fast-fading scenario, we note that the CC-HARQ strategy may or may not outperform Type-1 ARQ strategy depending on the LOS components at each link and the underlying SNR. To exemplify this behaviour, in Fig. 8.1, we present the minimum PDP of CC-HARQ based strategies for a two-hop network under two different combinations of the LOS components.

Irrespective of these observations, our contributions on the optimal ARQ distribution for CC-HARQ based strategies are significant since the proposed low-complexity algorithms are applicable for a wide range of LOS parameters.

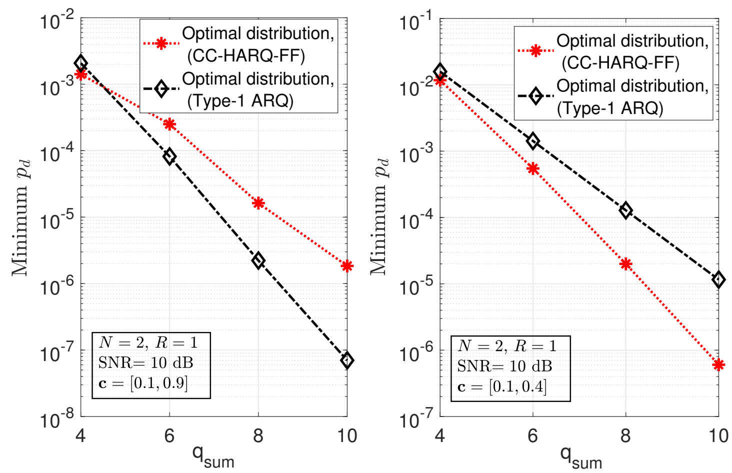

Figure 8.1: Comparison of minimum PDP (denoted by p_d) against different values of q_{sum} for a 2-hop network implementing CC-HARQ-FF and Type-1 ARQs strategies [29].

8.2 CC-HARQ based Multi-Hop Network Model

Consider a network with N-hops, as shown in Fig. 8.2, consisting of a source node (S), a set of $N-1$ relays $R_1, R_2, \ldots, R_{N-1}$ and a destination node (D). Along the similar lines of previous chapter (in particular, Section 7.2), using the CC-HARQ protocol, the error event at the k-th link after using the q_k attempts can be written as $P_{kq_k} = \Pr\left(R > \log_2(1 + \sum_{j=1}^{q_k} |h_{kj}|^2\gamma)\right)$, due to the maximum-ratio combining technique. For CC-HARQ-FF, wherein the channel realizations across the q_k attempts are statistically independent, we denote P_{kq_k} by $P^f_{kq_k}$, which can be written as $P^f_{kq_k} = \Pr\left(R > \log_2(1 + \sum_{j=1}^{q_k} |h_{kj}|^2\gamma)\right)$. In this case, we highlight that only the NLOS component across the attempts take independent realizations, whereas the LOS components

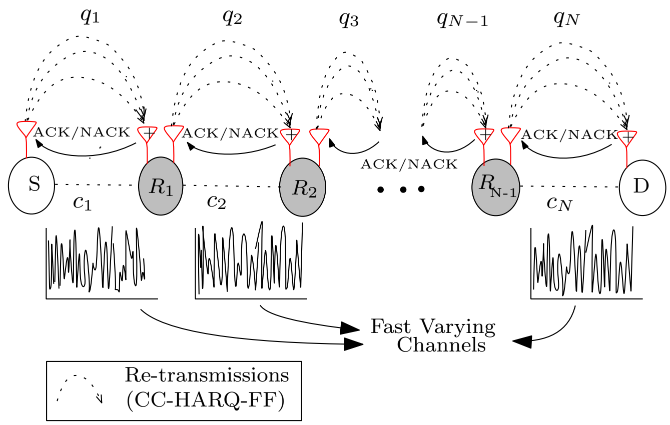

Figure 8.2: Illustration of an N-hop network with a source node (S), the relay nodes $R_1, \ldots, R_{N-1}$ and the destination node (D) follow the CC-HARQ protocol at each intermediate link. Also, each intermediate link can be characterized by an LOS component c_k $\forall k \in [N]$ and the non-LOS component captures fast varying behavior of the channel.

remain c_k. Here, the superscript 'f' is used to represent the fast-fading scenario across the attempts. Also, it is important to note that $P^f_{kq_k}$ is a function of q_k-th order Marcum-Q function because $\sum_{j=1}^{q_k} |h_{kj}|^2$ can be characterized by the non-central chi-square distribution with degrees of freedom $2q_k$.

In the following section, we introduce an end-to-end reliability metric for the CC-HARQ protocol under fast-fading scenarios and then present an analysis on "how to distribute the ARQs so as to optimize the reliability metric".

8.3 Optimal ARQ Distribution in CC-HARQ-FF based Non-Cumulative Network

In this section, we analyse the optimal ARQ distribution for the CC-HARQ-FF network wherein the channels across the attempts are assumed to be statistically independent, as given in Fig. 8.2. Along the similar lines of previous chapter, for a CC-HARQ-FF based non-cumulative network, the PDP expression can be written as

$$p_d^f = P_{1q_1}^f + \sum_{k=2}^{N} P_{kq_k}^f \left(\prod_{j=1}^{k-1} (1 - P_{jq_j}^f) \right), \tag{8.1}$$

where $P_{kq_k}^f = \Pr\left(R > \log_2(1 + (\sum_{j=1}^{q_k} |h_{kj}|^2)\gamma)\right)$. Now, we are interested in allocating the ARQs to the nodes to minimize the PDP in (8.1) for a given sum constraint $q_{sum} \in \mathbb{Z}_+$ such that $\sum_{i=1}^{N} q_i = q_{sum}$. Therefore, we formulate our optimization problem in Problem 8.1 as shown below. Throughout this chapter, we refer to the solution of Problem 8.1 as the optimal ARQ distribution for CC-HARQ-FF based non-cumulative network.

Problem 8.1. *For a given* $\mathbf{c}$, $\gamma = \frac{1}{\sigma^2}$ *and* q_{sum}, *solve*

$$q_1^*, q_2^*, \dots q_N^* = \arg \min_{q_1, q_2, \dots q_N} p_d^f$$

$$\textit{subject to } q_k \geq 1, q_k \in \mathbb{Z}_+, \sum_{i=1}^{N} q_i = q_{sum}.$$

It can be observed that $P_{kq_k}^f$ is dependent on q_k-th order Marcum-Q function, and tackling a higher-order Marcum-Q function analytically is challenging because it contains a lower incomplete gamma function. This way, the PDP expression is different from the CC-HARQ-SF strategy wherein the PDP expression is dependent on the first-order Marcum-Q function. Therefore, in this section, we first present an approximation on the higher-order Marcum-Q function at a higher SNR and then use the approximation to provide the necessary and sufficient conditions on the near-optimal ARQ distribution.

8.3.1 Approximation on the q_k-th Order Marcum-Q Function

Theorem 8.1. *For a given N-hop network and a higher SNR regime, we can approximate the generalized Marcum-Q function of order q_k, denoted by $Q_{q_k}(a_k, b_k)$, as $Q_{q_k}(a_k, b_k) \approx \tilde{Q}_{q_k}(a_k, b_k) \triangleq 1 - \frac{1}{q_k!}\left(\frac{b_k^2}{2}\right)^{q_k} e^{\frac{-a_k^2}{2}}$ for all $k \in [N]$.*

Proof. In the context of the N-hop network discussed in the previous section, the generalized Marcum-Q function of order q_k associated with the k-th hop is given by

$$Q_{q_k}(a_k, b_k) = 1 - e^{-\frac{a_k^2}{2}} \sum_{i=0}^{\infty} \frac{1}{i!} \left(\frac{\Lambda(q_k + i, \frac{b_k^2}{2})}{\Gamma(q_k + i)} \right) \left(\frac{a_k^2}{2} \right)^i, \tag{8.2}$$

where $\Lambda(\cdot, \cdot)$ is a lower incomplete gamma function, $\Gamma(\cdot)$ is a gamma function defined as $\Gamma(n) = (n-1)!$, $a_k = \sqrt{\frac{2c_k}{(1-c_k)}}$ and $b_k = \sqrt{\frac{2(2^R-1)}{\gamma(1-c_k)}}$ such that $\gamma = \frac{1}{\sigma^2}$. Furthermore, to simplify $\Lambda(\cdot, \cdot)$, we use the Holomorphic extension of incomplete gamma function given by

$$\Lambda\left(q_k + i, \frac{b_k^2}{2}\right) = \Gamma(q_k + i) e^{-\frac{b_k^2}{2}} \left(\frac{b_k^2}{2}\right)^{q_k} \sum_{j=0}^{\infty} \frac{\left(\frac{b_k^2}{2}\right)^{i+j}}{\Gamma(q_k + i + j + 1)}. \tag{8.3}$$

By using (8.2) and (8.3), we can write

$$Q_{q_k}(a_k, b_k) = 1 - e^{-\frac{a_k^2}{2}} e^{-\frac{b_k^2}{2}} \left(\frac{b_k^2}{2}\right)^{q_k} \sum_{i=0}^{\infty} \frac{1}{i!} \left(\frac{a_k^2}{2}\right)^i \sum_{j=0}^{\infty} \frac{\left(\frac{b_k^2}{2}\right)^{i+j}}{\Gamma(q_k + i + j + 1)}.$$

Similar to Theorem 7.1 (from Chapter 7), at a higher SNR, we use small values of b_k to derive the approximation. On expanding $e^{-\frac{b_k^2}{2}}$ using Taylor series, and neglecting the higher power terms, we can approximate $e^{\frac{-b_k^2}{2}}$ by $(1-\frac{b_k^2}{2})$. Similarly, by expanding the summations and neglecting the higher power terms of b_k (powers more than two) we approximate $Q_{q_k}(a_k, b_k)$ as

$$\begin{aligned} Q_{q_k}(a_k,b_k) \quad &\approx \quad 1-e^{-\frac{a_k^2}{2}}\left(1-\frac{b_k^2}{2}\right)\left(\frac{b_k^2}{2}\right)^{q_k}\left[\frac{1}{q_k!}+\frac{(b_k^2)/2}{(q_k+1)!}+\left(\frac{b_k^2}{2}\right)\sum_{j=1}^{\infty}\frac{1}{j!}\left(\frac{a_k^2}{2}\right)^{j}\right], \\ &\quad\quad 1-e^{-\frac{a_k^2}{2}}\left(1-\frac{b_k^2}{2}\right)\left(\frac{b_k^2}{2}\right)^{q_k}\left[\frac{1}{q_k!}+\frac{(b_k^2)/2}{(q_k+1)!}+\frac{b_k^2}{2}(e^{a_k^2/2}-1)\right]. \end{aligned}$$

On further neglecting the higher power terms, we get

$$Q_{q_k}(a_k,b_k) \approx \tilde{Q}_{q_k}(a_k,b_k) \triangleq 1-\frac{1}{q_k!}\left(\frac{b_k^2}{2}\right)^{q_k} e^{-\frac{a_k^2}{2}}. \tag{8.4}$$

This completes the proof. □

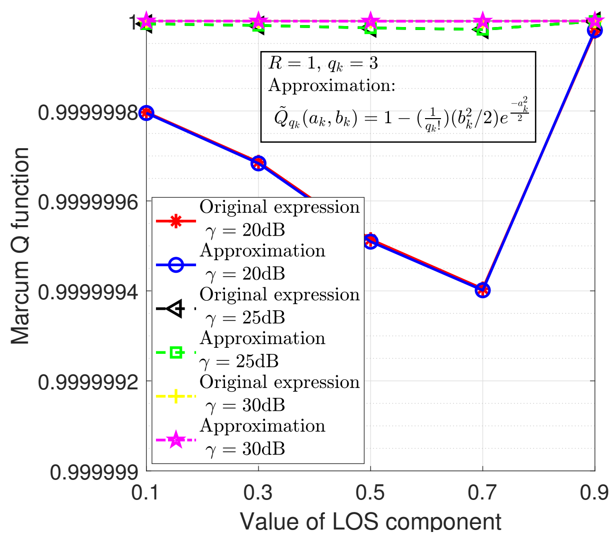

Figure 8.3: Comparison of an original expression of q_k-th order Marcum Q-function with its proposed approximation (in (8.4)) at different SNR.

To validate the approximation in (8.4) for different values of SNR and the LOS components, we present simulation results in Fig. 8.3 where the original expression of Marcum-Q function of higher-order is compared with its approximated expression in (8.4). It can be observed from

Fig. 8.3 that the approximation on the Marcum-Q function almost coincides with the original expression and hence, it verifies the accuracy of our approximation.

Now, by using the approximation in (8.4), we can approximate $P^f_{kq_k}$ in (8.1) as

$$\tilde{P}^f_{kq_k} = 1 - \tilde{Q}_{q_k}(a_k, b_k) \triangleq \frac{1}{q_k!}\left(\frac{b_k^2}{2}\right)^{q_k} e^{-\frac{a_k^2}{2}} = \frac{1}{q_k!}\left(\frac{\phi}{1-c_k}\right)^{q_k} e^{\frac{-a_k^2}{2}}, \tag{8.5}$$

where $\phi = \frac{2^R-1}{\gamma}$, and $\tilde{P}^f_{kq_k}$ denotes the approximation on $P^f_{kq_k}$. Finally, by using (8.5), we can write the approximate version of (8.1) as

$$\tilde{p}^f_d = \tilde{P}^f_{1q_1} + \sum_{k=2}^{N} \tilde{P}^f_{kq_k}\left(\prod_{j=1}^{k-1}(1 - \tilde{P}^f_{jq_j})\right), \tag{8.6}$$

where $\tilde{p}^f_d$ denotes the approximate version of p^f_d. Henceforth, using the approximated expression on p^f_d, we formulate an optimization problem in Problem 8.2 as shown below. Throughout this chapter, we refer to the solution of Problem 8.2 as the near-optimal ARQ distribution for CC-HARQ-FF based non-cumulative network since the objective function in Problem 8.2 is an approximate version of the objective function in Problem 8.1.

Problem 8.2. *For a given* $\mathbf{c}$, *a higher SNR* $\gamma = \frac{1}{\sigma^2}$ *and* q_{sum}, *solve*

$$q_1^*, q_2^*, \dots q_N^* = \arg \min_{q_1, q_2, \dots q_N} \tilde{p}^f_d$$
$$\textit{subject to } q_k \geq 1, q_k \in \mathbb{Z}_+, \sum_{i=1}^{N} q_i = q_{sum}.$$

It can be noted that the expressions in (7.3) (from Chapter 7) and (8.1) look similar. However, due the presence of the distinct order Marcum-Q functions, the strategies for solving the optimization problems are different. Now, in the next section, we provide novel results on the necessary and sufficient conditions on the near-optimal ARQ distribution for CC-HARQ-FF based non-cumulative networks.

8.3.2 Analysis on the Near-Optimal ARQ Distribution for CC-HARQ-FF based Non-Cumulative Network

Before we obtain the necessary and sufficient conditions on the near-optimal ARQ distribution, we show that based on a given LOS component, how we can decide on which link should get more ARQs than others.

Theorem 8.2. *For a given LOS vector* $\mathbf{c}$ *and high SNR regime in a CC-HARQ-FF based multi-hop network, the solution to Problem 8.2 satisfies the property that whenever* $c_i \geq c_j$*,* $q > 1$ *and* $q \geq \frac{1}{1-c_i}$*, we have* $q_i \geq q_j \ \forall i, j \ \in [N]$ *and whenever* $q > 1$ *and* $q \leq \frac{1}{1-c_j}$*, we have* $q_i \leq q_j \ \forall i, j \ \in [N]$.

Proof. Along the similar lines of Theorem 7.2 (from Chapter 7), we only consider the i-th and j-th links and assume that they have 'q' ARQs to start with. Let $A = \tilde{P}^f_{iq} + \tilde{P}^f_{j(q+1)}(1 - \tilde{P}^f_{iq})$ and $B = \tilde{P}^f_{i(q+1)} + \tilde{P}^f_{jq}(1 - \tilde{P}^f_{i(q+1)})$ with $q_i = q_j = q$. By rearranging the terms in A and B, we get $A = \tilde{P}^f_{iq} + \tilde{P}^f_{j(q+1)} - \tilde{P}^f_{iq}\tilde{P}^f_{j(q+1)}$ and $B = \tilde{P}^f_{i(q+1)} + \tilde{P}^f_{jq} - \tilde{P}^f_{i(q+1)}\tilde{P}^f_{jq}$. On taking the difference between A and B, we get $A - B = \tilde{P}^f_{iq} - \tilde{P}^f_{i(q+1)} + \tilde{P}^f_{j(q+1)} - \tilde{P}^f_{jq} - \tilde{P}^f_{iq}\tilde{P}^f_{j(q+1)} + \tilde{P}^f_{i(q+1)}\tilde{P}^f_{jq}$. Using the approximation on $\tilde{P}^f_k$ in (8.5), we can rewrite the difference as

$$\begin{aligned} A - B \;&=\; \frac{1}{q!}\left(\frac{b_i^2}{2}\right)^q e^{\frac{-a_i^2}{2}}\left(1 - \frac{1}{(q+1)}\left(\frac{b_i^2}{2}\right)\right) + \frac{1}{q!}\left(\frac{b_j^2}{2}\right)^q e^{\frac{-a_j^2}{2}}\left(\frac{1}{(q+1)}\left(\frac{b_j^2}{2}\right) - 1\right) \\ &\quad + \frac{1}{(q!)^2}\left(\frac{b_i^2 b_j^2}{2}\right)^q \frac{e^{(-a_i^2 - a_j^2)/2}}{q+1}\left(-\frac{b_j^2}{2} + \frac{b_i^2}{2}\right). \end{aligned}$$

From the above equation, it can be observed that first term in the above equation is positive and the second term is negative. Also, at high SNR, we can ignore high power terms of b_i or b_j. Therefore, by neglecting the third term from the above equation, we get

$$A - B \;=\; \frac{1}{q!}\left(\frac{b_i^2}{2}\right)^q e^{\frac{-a_i^2}{2}}\left(1 - \frac{1}{(q+1)}\left(\frac{b_i^2}{2}\right)\right) + \frac{1}{q!}\left(\frac{b_j^2}{2}\right)^q e^{\frac{-a_j^2}{2}}\left(\frac{1}{(q+1)}\left(\frac{b_j^2}{2}\right) - 1\right).$$

Now, we want to estimate that whether the first term is greater than the second term or not. Let us start by assuming that the second term is greater than the first term. Therefore, we can write

$$\frac{1}{q!}\left(\frac{b_j^2}{2}\right)^q e^{\frac{-a_j^2}{2}}\left(\frac{1}{(q+1)}\left(\frac{b_j^2}{2}\right)-1\right) \geq \frac{1}{q!}\left(\frac{b_i^2}{2}\right)^q e^{\frac{-a_i^2}{2}}\left(1-\frac{1}{(q+1)}\left(\frac{b_i^2}{2}\right)\right).$$

After rearranging and solving the above equation, we get

$$\frac{e^{-a_j^2/2}}{e^{-a_i^2/2}}\left(\frac{1-\frac{b_j^2}{2(q+1)}}{1-\frac{b_i^2}{2(q+1)}}\right) \geq \left(\frac{b_i^2}{b_j^2}\right)^q.$$

Furthermore, by expanding the $b_k^2 = \frac{2\phi}{(1-c_k)} \forall k \in \{i,j\}$ on RHS of the equation and making $\frac{b_j^2}{2(q+1)} \approx 0$ and $\frac{b_i^2}{2(q+1)} \approx 0$, we obtain

$$\frac{e^{-a_j^2/2}}{e^{-a_i^2/2}} \geq \left(\frac{1-c_i}{1-c_j}\right)^q.$$

Now, by taking logarithm on both the side and after rearranging, we get

$$\frac{c_i - c_j}{(1-c_i)(1-c_j)} \geq q \log\left(\frac{1-c_i}{1-c_j}\right). \tag{8.7}$$

By using the standard inequality $\log x \leq x-1$, it can be observed that (8.7) is true for $q=1$. However, (8.7) may not be true for $q>1$, because of the inequality $\frac{x-1}{x} \leq \log x$. Hence, we conclude that whenever $q>1$ and also, $q \geq \frac{1}{(1-c_i)}$, (8.7) is false. It implies that $A-B \geq 0$ and hence, in this situation, higher LOS must be given more ARQs. Similarly, when $q>1$ and $q \leq \frac{1}{(1-c_j)}$, (8.7) is true. It implies that $A-B \leq 0$ and higher LOS must be given lower ARQs.

Similar to Theorem 7.3, by using Definition 7.2, we provide the necessary and sufficient conditions on near-optimal ARQ distribution for CC-HARQ-FF. □

Theorem 8.3. *For a given N-hop network with LOS vector* $\mathbf{c}$*, the ARQ distribution* $\mathbf{q}^* = [q_1^*, q_2^*, \ldots, q_N^*]$ *is said to be a local minima for CC-HARQ-FF strategy if and only if* q_i^* *and* q_j^*

for $i \neq j$ satisfy the following bounds

$$\frac{E_i B_i^{q_i^*}}{q_i^*!}\left(1-\frac{B_i}{q_i^*+1}\right)+\frac{E_j B_j^{(q_j^*-1)}}{(q_j^*-1)!}\left(\frac{B_j}{q_j^*}-1\right)\leq\frac{E_i E_j B_i^{q_i^*} B_j^{(q_j^*-1)}}{q_i^*!(q_j^*-1)!}\left(\frac{B_j}{q_j^*}-\frac{B_i}{q_i^*+1}\right), \quad (8.8)$$

$$\frac{E_i B_i^{(q_i^*-1)}}{(q_i^*-1)!}\left(1-\frac{B_i}{q_i^*}\right)+\frac{E_j B_j^{q_j^*}}{q_j^*!}\left(\frac{B_j}{q_j^*+1}-1\right)\geq\frac{E_i E_j B_i^{(q_i^*-1)} B_j^{q_j^*}}{(q_i^*-1)!q_j^*!}\left(\frac{B_j}{q_j^*+1}-\frac{B_i}{q_i^*}\right), \quad (8.9)$$

where $E_i = e^{-a_i^2/2}$, $E_j = e^{-a_j^2/2}$, $B_i = \phi/(1-c_i)$ and $B_j = \phi/(1-c_j)$ with $\phi = (2^R-1)/\gamma$.

Proof. Along the similar lines of Theorem 7.3 (from Chapter 7), by using the definition of local minima, we have the inequalities

$$\tilde{p}_d^f(\mathbf{q}^*) \leq \tilde{p}_d^f(\hat{\mathbf{q}}_+), \text{ and } \tilde{p}_d^f(\mathbf{q}^*) \leq \tilde{p}_d^f(\hat{\mathbf{q}}_-), \quad (8.10)$$

where $\tilde{p}_d^f(\mathbf{q}^*)$, $\tilde{p}_d^f(\hat{\mathbf{q}}_+)$ and $\tilde{p}_d^f(\hat{\mathbf{q}}_-)$ represent the PDP evaluated at the distributions $\mathbf{q}^*$, $\hat{\mathbf{q}}_+$, and $\hat{\mathbf{q}}_-$, respectively. Because of the fact that $\hat{\mathbf{q}}_+$ and $\hat{\mathbf{q}}_-$ differ only in the last two positions and the structure of the PDP, it is possible to demonstrate that the inequalities in (8.10) are equivalent to

$$\tilde{P}^f_{iq_i^*} + \tilde{P}^f_{jq_j^*}(1-\tilde{P}^f_{iq_i^*}) \leq \tilde{P}^f_{i(q_i^*+1)} + \tilde{P}^f_{j(q_j^*-1)}(1-\tilde{P}^f_{i(q_i^*+1)}),$$
$$\tilde{P}^f_{iq_i^*} + \tilde{P}^f_{jq_j^*}(1-\tilde{P}^f_{iq_i^*}) \leq \tilde{P}^f_{i(q_i^*-1)} + \tilde{P}^f_{j(q_j^*+1)}(1-\tilde{P}^f_{i(q_i^*-1)}).$$

Now, by using the approximation on $\tilde{P}^f_{iq}$ from (8.5), we can rewrite the above two equations as

$$\frac{e^{\frac{-a_i^2}{2}}}{q_i^*!}\left(\frac{\phi}{1-c_i}\right)^{q_i^*}+\frac{e^{\frac{-a_j^2}{2}}}{q_j^*!}\left(\frac{\phi}{1-c_j}\right)^{q_j^*}\left(1-\frac{e^{\frac{-a_i^2}{2}}}{q_i^*!}\left(\frac{\phi}{1-c_i}\right)^{q_i^*}\right)\leq\frac{e^{\frac{-a_i^2}{2}}}{(q_i^*+1)!}$$
$$\left(\frac{\phi}{1-c_i}\right)^{(q_i^*+1)}+\frac{e^{\frac{-a_j^2}{2}}}{(q_j^*-1)!}\left(\frac{\phi}{1-c_j}\right)^{(q_j^*-1)}\left(1-\frac{e^{\frac{-a_i^2}{2}}}{(q_i^*+1)!}\left(\frac{\phi}{1-c_i}\right)^{(q_i^*+1)}\right),$$
$$\frac{e^{\frac{-a_i^2}{2}}}{q_i^*!}\left(\frac{\phi}{1-c_i}\right)^{q_i^*}+\frac{e^{\frac{-a_j^2}{2}}}{q_j^*!}\left(\frac{\phi}{1-c_j}\right)^{q_j^*}\left(1-\frac{e^{\frac{-a_i^2}{2}}}{q_i^*!}\left(\frac{\phi}{1-c_i}\right)^{q_i^*}\right)\leq\frac{e^{\frac{-a_i^2}{2}}}{(q_i^*-1)!}$$
$$\left(\frac{\phi}{1-c_i}\right)^{(q_i^*-1)}+\frac{e^{\frac{-a_j^2}{2}}}{(q_j^*+1)!}\left(\frac{\phi}{1-c_j}\right)^{(q_j^*+1)}\left(1-\frac{e^{\frac{-a_i^2}{2}}}{(q_i^*-1)!}\left(\frac{\phi}{1-c_i}\right)^{(q_i^*-1)}\right).$$

Assuming $E_i = e^{-a_i^2/2}$ and $B_i = \phi/(1-c_i)$ for $i \in [1, N]$, we can simultaneously rewrite the above two inequalities as

$$\frac{E_i}{q_i^*!}B_i^{q_i^*} + \frac{E_j}{q_j^*!}B_j^{q_j^*}\left(1 - \frac{E_i}{q_i^*!}B_i^{q_i^*}\right) \leq \frac{E_i}{(q_i^*+1)!}B_i^{(q_i^*+1)} + \frac{E_j}{(q_j^*-1)!}B_j^{(q_j^*-1)}\left(1 - \frac{E_i}{(q_i^*+1)!}B_i^{(q_i^*+1)}\right),$$
$$\frac{E_i}{q_i^*!}B_i^{q_i^*} + \frac{E_j}{q_j^*!}B_j^{q_j^*}\left(1 - \frac{E_i}{q_i^*!}B_i^{q_i^*}\right) \leq \frac{E_i}{(q_i^*-1)!}B_i^{(q_i^*-1)} + \frac{E_j}{(q_j^*+1)!}B_j^{(q_j^*+1)}\left(1 - \frac{E_i}{(q_i^*-1)!}B_i^{(q_i^*-1)}\right).$$

By rearranging the above two inequalities, we can obtain the two inequalities as

$$\frac{E_i B_i^{q_i^*}}{q_i^*!}\left(1 - \frac{B_i}{q_i^*+1}\right) + \frac{E_j B_j^{(q_j^*-1)}}{(q_j^*-1)!}\left(\frac{B_j}{q_j^*} - 1\right) \leq \frac{E_i E_j B_i^{q_i^*} B_j^{(q_j^*-1)}}{q_i^*!(q_j^*-1)!}\left(\frac{B_j}{q_j^*} - \frac{B_i}{q_i^*+1}\right), \tag{8.11}$$
$$\frac{E_i B_i^{(q_i^*-1)}}{(q_i^*-1)!}\left(1 - \frac{B_i}{q_i^*}\right) + \frac{E_j B_j^{q_j^*}}{q_j^*!}\left(\frac{B_j}{q_j^*+1} - 1\right) \geq \frac{E_i E_j B_i^{(q_i^*-1)} B_j^{q_j^*}}{(q_i^*-1)!q_j^*!}\left(\frac{B_j}{q_j^*+1} - \frac{B_i}{q_i^*}\right), \tag{8.12}$$

respectively. This completes the proof. □

It can be observed that the above two conditions are both necessary and sufficient because the bounds are obtained by rearranging the terms in the condition on local minima.

8.3.3 Low-Complexity Algorithm for CC-HARQ-FF based Non-Cumulative Network

It can be observed from (8.11) that the conditions are non-linear with integer constraints. Due to higher-order non-linearity, it is not possible to solve the conditions analytically on the near-optimal ARQ distribution. Hence, towards finding the near-optimal ARQ distribution, in this section, first, we approximate the necessary and sufficient conditions at high SNR. After that, we propose a Fold-To-Make-List (FTML) low-complexity algorithm based on numerical-methods, as given in Algorithm 8.

Towards approximating the necessary and sufficient conditions at high SNR, let us rearrange (8.9) as

$$\frac{E_i B_i^{(q_i^*-1)}}{(q_i^*-1)!}\left(\frac{B_i}{q_i^*} - 1\right) + \frac{E_j B_j^{q_j^*}}{q_j^*!}\left(1 - \frac{B_j}{q_j^*+1}\right) \leq \frac{E_i E_j B_i^{(q_i^*-1)} B_j^{q_j^*}}{(q_i^*-1)!q_j^*!}\left(\frac{B_i}{q_i^*} - \frac{B_j}{q_j^*+1}\right),$$

and add the above inequality with (8.8), to obtain

$$\frac{E_i B_i^{(q_i^*-1)}}{(q_i^*-1)!}\left(\frac{2B_i}{q_i^*}-1-\frac{B_i^2}{q_i^*(q_i^*+1)}\right)+\frac{E_j B_j^{(q_j^*-1)}}{(q_j^*-1)!}\left(\frac{2B_j}{q_j^*}-1-\frac{B_j^2}{q_j^*(q_j^*+1)}\right)\leq$$
$$\frac{E_i E_j B_i^{(q_i^*-1)} B_j^{(q_j^*-1)}}{(q_i^*-1)!(q_j^*-1)!}\left(\frac{2B_iB_j}{q_i^*q_j^*}-\frac{B_i^2}{q_i^*(q_i^*+1)}-\frac{B_j^2}{q_j^*(q_j^*+1)}\right).$$

At high SNR, first we focus on the LHS of the expression, wherein we can ignore terms which contain B_i^2 and B_j^2. This is because they are negligible compared to unity. Likewise, on the RHS of the expression, we can ignore the terms which contain B_i^2, B_j^2 and B_iB_j due to the same reason as specified above. Therefore, by ignoring the above-mentioned terms, we can obtain the inequalities as shown below

$$\begin{aligned}\frac{E_i B_i^{(q_i^*-1)}}{(q_i^*-1)!}\left(\frac{2B_i}{q_i^*}-1\right)+\frac{E_j B_j^{(q_j^*-1)}}{(q_j^*-1)!}\left(\frac{2B_j}{q_j^*}-1\right) &\leq 0,\\ \frac{E_i B_i^{q_i^*}}{q_{i!}^*}\left(2-\frac{q_i^*}{B_i}\right)+\frac{E_j B_j^{q_j^*}}{q_j^*!}\left(2-\frac{q_j^*}{B_j}\right) &\leq 0,\end{aligned}$$

where the second inequality can be obtained from the first inequality using algebraic manipulations. Furthermore, there are multiple ways in which q_i^* and q_j^* can follow the above inequality. One such way is to assume that $\frac{E_i B_i^{q_i^*}}{q_i^*}=\frac{E_j B_j^{q_j^*}}{q_j^*!}$. This is because, by writing $\frac{E_j B_j^{q_j^*}}{q_j^*!}=\frac{E_i B_i^{q_i^*}}{q_i^*}$, we can obtain

$$\frac{E_i B_i^{q_i^*}}{q_i^*}\left(4-\frac{q_i^*}{B_i}-\frac{q_j^*}{B_j}\right)\leq 0,$$

which is also true. Therefore, we impose the condition

$$\frac{E_i B_i^{q_i^*}}{q_i^*!}=\frac{E_j B_j^{q_j^*}}{q_j^*!}.$$

for every $i,j\in[N]$ using the approximation on the sufficient conditions. Furthermore, by using the Stirling's formulae i.e. $n!\approx\sqrt{2\pi n}(n/e)^n$, we can rewrite the above equation as

$$\frac{E_i}{\sqrt{2\pi q_i^*}}\left(\frac{eB_i}{q_i^*}\right)^{q_i^*}=\frac{E_j}{\sqrt{2\pi q_j^*}}\left(\frac{eB_j}{q_j^*}\right)^{q_j^*}.$$

Now, on further solving the above equation, we finally get

$$\sqrt{q_j^*}\left(\frac{q_j^*}{eB_j}\right)^{q_j^*} = E_j\frac{\sqrt{q_i^*}}{E_i}\left(\frac{q_i^*}{eB_i}\right)^{q_i^*}. \tag{8.13}$$

Now, to obtain the near-optimal ARQ distribution by using the above expression, we formulate an equivalent optimization problem of Problem 8.2 as given below:

Problem 8.3. *For a given* N, $\mathbf{E} = \{E_1, E_2, \ldots, E_N\}$, $\mathbf{B} = \{B_1, B_2, \ldots, B_N\}$, *and* q_{sum}, *find* q_i *and* q_j *such that*

$$\sqrt{q_j^*}\left(\frac{q_j^*}{eB_j}\right)^{q_j^*} = E_j\frac{\sqrt{q_i^*}}{E_i}\left(\frac{q_i^*}{eB_i}\right)^{q_i^*}, \ \forall i,j \in [1,N] \text{ where } i \neq j,$$

$q_i \geq 1, q_i \in \mathbb{Z}_+, B_i = \frac{2^R-1}{\gamma(1-c_i)}, E_i = e^{\frac{c_i}{(1-c_i)}}$ $\forall i \in [1,N]$, $\sum_{i=1}^{N} q_i = \ q_{sum}$.

In the rest of this section, our objective is to find the near-optimal ARQ distribution for Problem 8.3. Note that the equation in Problem 8.3 is non-tractable because we cannot solve it analytically due to the non-linearity and integer constraints. Therefore, we obtain the near-optimal ARQ distribution with the help of an FTML low-complexity algorithm based on numerical-methods as proposed in Algorithm 8. In Algorithm 8, we define $\mathfrak{D}_{\mathrm{j}}(q_j^*) \triangleq \sqrt{q_j^*}(q_j^*/eB_j)^{q_j^*}$ and $\mathfrak{C}_{\mathrm{i}}(q_i^*) \triangleq \frac{\sqrt{q_i^*}}{E_i}(q_i^*/eB_i)^{q_i^*}$ $\forall i,j \in [1,N]$ & $i \neq j$, where $\mathfrak{C}_{\mathrm{i}}(q_i^*)$ is a constant term for a given q_i^*, and thus we can rewrite (8.13) as $\mathfrak{D}_{\mathrm{j}}(q_j^*) = E_j\mathfrak{C}_{\mathrm{i}}(q_i^*)$.

To explain Algorithm 8 in detail, it can be observed that for a fixed q_i^*, the RHS of $\mathfrak{D}_{\mathrm{j}}(q_j^*) = E_j\mathfrak{C}_{\mathrm{i}}(q_i^*)$ is a constant term. Therefore, by fixing q_1^* (wherein $q_1 \in [q_{sum} - (N-1)]$), we can find the ARQ distribution q_j^*, $\forall j \in [2,N]$, by forming pairs such as $(q_1,q_2), (q_1,q_3), \ldots,$ and so on. Thereby, for each possible value of q_1^*, the other ARQ distribution can be calculated. Thus, a list of suitable ARQ distribution can be created, which is denoted by $\mathcal{L}^f$ (in Algorithm 8) and from the list, the near-optimal ARQ distribution can be obtained. Furthermore, as the

Algorithm 8 Fold-To-Make-List (FTML)- A Numerical-Methods Based Algorithm For CC-HARQ-FF Based Non-Cumulative Network

Require: $\mathbf{E}$, $\mathbf{B}$, q_{sum}, $\mathbf{c} = [c_1, c_2, \ldots, c_N]$.

Ensure: $\mathcal{L}^f \subset \mathbb{S}$ - List of ARQ distributions in search space $\mathbb{S}$.

$\mathcal{L}^I = \{\phi\}$ - Internal array of list of ARQ distributions in $\mathbb{Z}_+$.
for $q_1^* = 1 : q_{sum} - (N-1)$ **do**
    $\mathcal{L}^I_{q_1^*,1} = \{q_1^*\}$.
    for $j = 2 : N$ **do**
        Assign $m_1 = 1$, $m_2 = q_{sum} - (N-1)$.
        while $m_2 - m_1 \geq 2$ **do**
            for $k = 1 : 2$ **do**
                Calculate $\mathfrak{D}_{\mathrm{j}}(m_k) = \sqrt{m_k}(m_k/eB_j)^{m_k}$.
                Compute $d_k = |\mathfrak{D}_{\mathrm{j}}(m_k) - E_j \mathfrak{C}_1(q_1^*)|$.
            end for
            if $d_1 \geq d_2$ **then**
                Assign $m_1 = \lfloor \frac{m_1+m_2}{2} \rfloor$.
            else
                Assign $m_2 = \lfloor \frac{m_1+m_2}{2} \rfloor$.
            end if
        end while
        for $l = 1 : 2$ **do**
            Compute $\mathfrak{D}_{\mathrm{j}}(m_l) = \sqrt{m_l}(m_l/eB_j)^{m_l}$.
            Compute $e_l = |\mathfrak{D}_{\mathrm{j}}(m_l) - E_j \mathfrak{C}_1(q_1^*)|$.
        end for
        if $e_1 \leq e_2$ **then**
            Assign $q_j^* = m_1$.
        else
            Assign $q_j^* = m_2$.
        end if
        Insert q_j^* in $\mathcal{L}^I_{q_1^*,j}$.
    end for
end for
for $t = 1 : \mathrm{rows}(\mathcal{L}^I)$ **do**
    Compute $I_t = \left(\sum_{i=1}^{N} \mathcal{L}^I_{t,i}\right) - q_{sum}$.
    $\mathcal{L}^f_t = \{\mathbf{q} \in \mathbb{S} \mid d(\mathbf{q}, \tilde{\mathbf{q}}) = I_t, q_i, q_j \geq 1\}$.
end for

LHS of $\mathfrak{D}_{\mathrm{j}}(q_j^*) = E_j \mathfrak{C}_{\mathrm{i}}(q_i^*)$ is a non-linear term, we can apply a folding technique based on numerical-methods to solve it. In particular, we use iterations, wherein the search space for q_j^* is reduced by half in each iteration such that it converges into an appropriate value of q_j^*.

To exemplify the steps, let us take the first pair (q_1^*, q_2^*). After fixing q_1^*, the RHS of $\mathfrak{D}_2(q_2^*) = E_2 \mathfrak{C}_1(q_1^*)$ becomes a constant term and our objective is to find the suitable value of q_2^* such that RHS becomes approximately equal to LHS. Therefore, to find the q_2^*, first, we define $m_1 = min(q_2^*)$ and $m_2 = max(q_2^*)$, wherein 'min' is used for the minimum value and 'max' is used for the maximum value of ARQs that can be given to q_2^*, respectively. In the first iteration, $m_1 = 1$ and $m_2 = q_{sum} - (N-1)$ because the minimum value of $q_2^* = 1$ and maximum value of $q_2^* = q_{sum} - (N-1)$. And then, in each iteration, the value of either m_1 or m_2 gets changed based on the algorithm. In particular, we are changing and assigning the new value to either m_1 or m_2 as $\lfloor (m_1 + m_2)/2 \rfloor$, where the deviation of RHS to LHS is higher in the expression $\mathfrak{D}_2(m_k) = E_2 \mathfrak{C}_1(q_1^*)\ \forall k \in [1, 2]$. This is because, our objective is to make RHS closer to the LHS by eliminating ARQ distribution that results in higher deviation of RHS from LHS. *We emphasize that the folding algorithm helps because of the fact that the RHS is an increasing function of* $q_j^*\ \forall j$. Now, after few iterations, when the difference $m_2 - m_1 < 2$, the iteration is stopped (refer to the lines from 6 to 16 in Algorithm 8). After the iteration stops, we obtain the final values of m_1 and m_2, on which we can again find the deviation of RHS to LHS in $\mathfrak{D}_2(m_k) = E_2 \mathfrak{C}_1(q_1^*)\ \forall k \in [1, 2]$. This time, we choose and assign q_2^* to either m_1 or m_2, wherein the deviation is lower (refer to the lines from 17 to 25 in Algorithm 8). This is because, we want to ensure that the difference between RHS and LHS is minimum to obtain q_2^*. Now, we repeat all the above steps to find other ARQ distribution for a fixed q_1^* and therefore, we obtain a list of ARQ distribution, denoted by $\mathcal{L}^I$ (as given in Algorithm 8). Note that, for every $j \in [2, N]$, the starting value of $m_1 = 1$ and $m_2 = q_{sum} - (N-1)$. Now, the list may contain infeasible ARQ distribution because the sum of ARQ distribution along the rows of $\mathcal{L}^I$ can exceed q_{sum}. Therefore, for each ARQ distribution in $\mathcal{L}^I$, we remove the extra ARQs, calculated by I_t (refer to line 31 in Algorithm 8), such that the ARQ distributions are in the desired search space (refer to lines from 29 to 32 in Algorithm 8). We emphasize that using Algorithm 8, we reduce the list size by a significant amount. Also, in order to reduce the list size further, we can invoke the results from Theorem 8.2.

We validate the accuracy of our Algorithm 8 using simulations in the next section. Formally, we present simulation results on PDP and complexity analysis for CC-HARQ-FF based non-cumulative network in the next section.

8.3.4 Simulation Results for CC-HARQ-FF based Non-Cumulative Network

To validate all the results on the optimal ARQ distribution, we present the plots in Fig. 8.4 for $N = 4$ and $N = 5$ at different q_{sum}. The following conclusions can be drawn through the plots: (i) the minimum PDP obtained by using an approximated expression in (8.6) almost coincides with the minimum PDP obtained from the original PDP expression in (8.1). Hence, it shows the accuracy of our approximation on the Marcum-Q function of a higher-order, (ii) the proposed low-complexity algorithm gives us the near-optimal ARQ distribution, hence validating the accuracy of our proposed low-complexity algorithm, and (iii) near-optimal ARQ distribution can be obtained by using the uniform ARQ distribution (unlike the case of CC-HARQ-SF) because the dominating factor is in terms of the number of ARQs allotted to a link (as in the power form of our objective function). This way, while solving Problem 8.3 (which is equivalent solving the Problem 8.2) using numerical-methods, the near-optimal ARQ distribution is closer to the uniform distribution.

Furthermore, similar to Section 7.5 (from Chapter 7), the list size of the exhaustive search in case of original expression (in (8.1)) and the approximated expression (in (8.6)) is $\binom{q_{sum}-1}{N-1}$. However, by using the proposed low-complexity method given in Algorithm 8, we can reduce

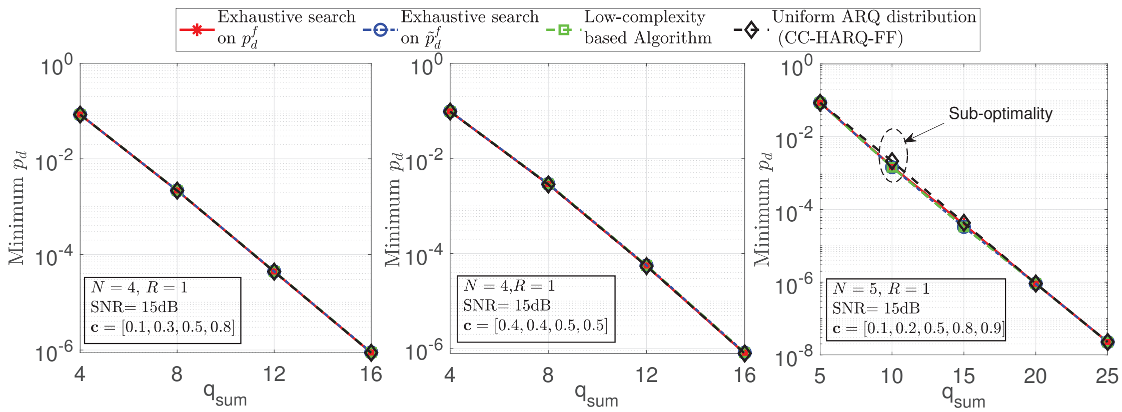

Figure 8.4: PDP comparison in CC-HARQ-FF based non-cumulative network between (i) exhaustive search using original PDP expression, p_d^f, (ii) exhaustive search using approximated PDP expression, $\tilde{p}_d^f$, (iii) proposed FTML low-complexity algorithm (given in Algorithm 8), and (iv) uniform ARQ distribution. With uniform distribution, each link is first distributed with $\lfloor q_{sum}/N \rfloor$ ARQs, and the remaining ARQs are equally shared by the first q_{sum} mod N links.

the list size significantly as shown in Fig. 8.5. Also, we have shown that the uniform distribution is giving near-optimal ARQ distribution. Therefore, by using the uniform distribution along with Theorem 8.2 (when q_{sum} is not divisible by N) provides the near-optimal ARQ distribution.

In the next section, we provide results for CC-HARQ-FF based fully-cumulative network.

8.4 CC-HARQ-FF based Fully-Cumulative Network

The idea of fully-cumulative network is similar to that in Section 7.7 with the only exception that the PDP expression, in this case, will consider fast fading intermediate channels. For instance, we can write the PDP expression for a 3-hop CC-HARQ-FF based fully-cumulative network as

$$\begin{aligned} p_d^f \;&=\; P_{1q_1}^f + \sum_{i=0}^{q_1-1}(P_{1i}^f - P_{1(i+1)}^f)P_{2(q_2+q_1-(i+1))}^f + \sum_{i=0}^{q_1-1}(P_{1i}^f - P_{1(i+1)}^f)\sum_{j=0}^{q_2+q_1-(i+2)} \\ &\quad (P_{2j}^f - P_{2(j+1)}^f)P_{3(q_3+q_2+q_1-(i+j+2))}^f \end{aligned} \tag{8.14}$$

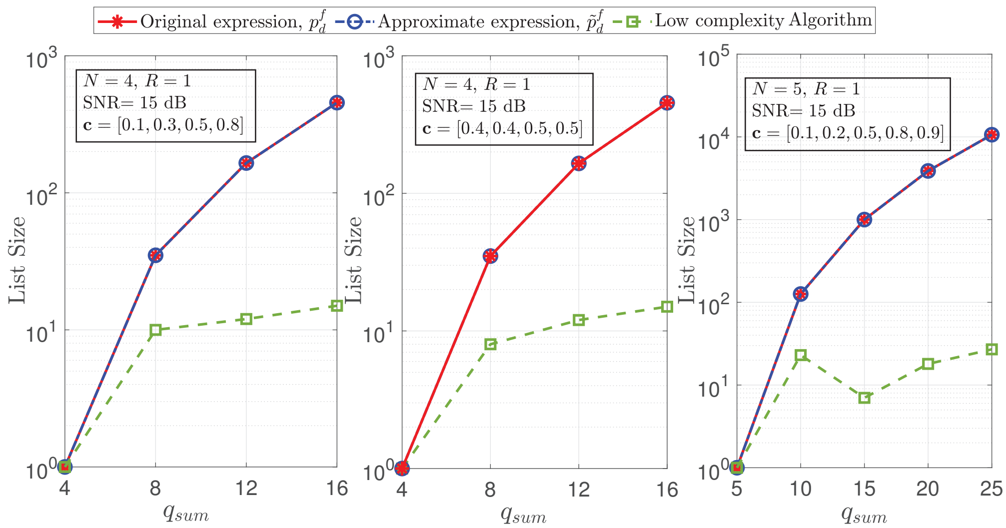

Figure 8.5: Comparison of list size between (i) exhaustive search in original PDP expression, p_d^f (in (8.1)), (ii) exhaustive search in approximated PDP expression, $\tilde{p}_d^f$ (in (8.6)), and (iii) the proposed FTML low-complexity algorithm (given in Algorithm 8) for CC-HARQ-FF based non-cumulative network.

where $P_{10}^f = 1$ and $(P_{ki}^f - P_{k(i+1)}^f)$ represents that the packet at k-th hop (where $k \in [1, 3]$) gets dropped till its i-th attempt and transmitted successfully to its next node at $(i+1)$-th attempt. It can be noted that the difference $(P_{ki}^f - P_{k(i+1)}^f)$ is not similar to $(P_{ki} - P_{k(i+1)})$ (from Chapter 7) because of the Marcum-Q functions of distinct orders. Also, by using the similar strategy, it is trivial to write the PDP expression for any given hop size. Now, we want to minimize the PDP by finding the optimal ARQ distribution for a fully-cumulative network.

Theorem 8.4. *For a given ARQ distribution* $\mathbf{q} = [q_1, q_2, \ldots, q_N]$ *and* q_{sum} *in a CC-HARQ-FF based fully-cumulative network, the optimal ARQ distribution can be given by* $[q_{sum} - (N - 1), 1, \ldots, 1]$.

Proof. First, we want to prove the result for $N = 2$. We start with ARQ distribution $[q_1, q_2]$ and

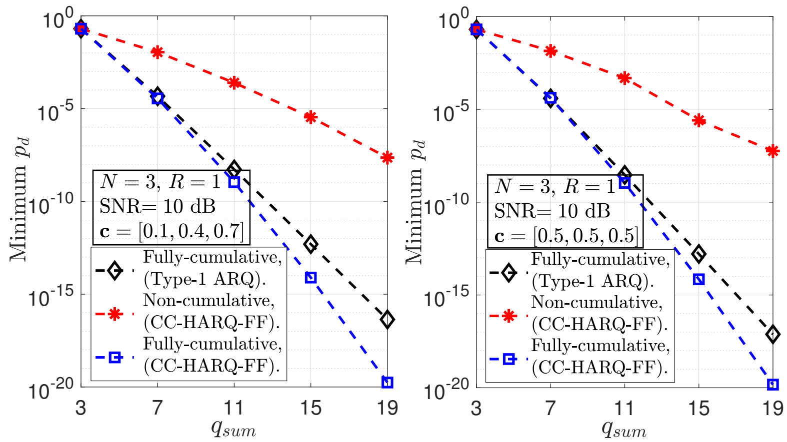

Figure 8.6: Comparison of PDP for the CC-HARQ-FF based non-cumulative network and fully-cumulative network at various values of q_{sum}.

the PDP can be written as

$$pdp_2^f = P_{1q_1}^f + \sum_{i=0}^{q_1-1}(P_{1i}^f - P_{1(i+1)}^f)P_{2(q_2+q_1-(i+1))}^f, \tag{8.15}$$

where $P_{10}^f = 1$ and the running index 'i' captures the unused residual ARQs from its previous link. Furthermore, let us transfer one ARQ from the second link to the first link by considering that $q_2 > 1$. Therefore, the PDP expression in this case can be written as

$$\begin{aligned} pdp_2^{f'} &= P_{1(q_1+1)}^f + \sum_{i=0}^{q_1}(P_{1i}^f - P_{1(i+1)}^f)P_{2(q_2-1+q_1+1-(i+1))}^f, \\ &= P_{1(q_1+1)}^f + \sum_{i=0}^{q_1}(P_{1i}^f - P_{1(i+1)}^f)P_{2(q_2+q_1-(i+1))}^f \end{aligned}$$

Similar to the results in Theorem 7.4 (from Chapter 7), we conclude that $pdp_2^f > pdp_2^{f'}$ and the optimal ARQ distribution can be given by $[q_1+q_2-1, 1]$. Unlike the CC-HARQ-SF based fully-cumulative network, we cannot have $[q_1+q_2, 0]$ as the optimal ARQ distribution. This is because, in CC-HARQ-FF, the order of the Marcum-Q function is equivalent to the ARQs given to the link and zero-order in not valid. This completes the proof for $N=2$. Along the similar lines of Theorem 7.4 (from Chapter 7), we can write

$$PDP^{f}_{2v} = P^{f}_{1q_1} + \sum_{i=0}^{q_1-1}(P^{f}_{1i} - P^{f}_{1(i+1)})PDP^{f}_{v}(q_{sum,v} + q_1 - (i+1)). \tag{8.16}$$

In the above equation, for the fixed q_1, P^f_{1i} and $P^f_{1(i+1)}$ for i-th iteration, the PDP expression can be minimized by minimizing the $PDP^f_v(q_{sum,v} + q_1 - (i+1))$. Also, by using the assumption on $N = k$ from the induction step, we know that $[q_{sum} - (N-1), 1, \ldots, 1]$ will minimize the PDP^f_v. Furthermore, for (8.16), we have the ARQ distribution $[q_1, q_{sum,v} - (N-2), 1, \ldots, 1]$. In order to find the optimal ARQ distribution of (8.16), we use similar approach as for $N = 2$. That is, we start transferring one ARQ from the virtual node to the first link and the new PDP expression can be written as

$$\begin{aligned} PDP^{f'}_{2v} &= P^{f}_{1(q_1+1)} + \sum_{i=0}^{q_1}(P^{f}_{1i} - P^{f}_{1(i+1)})PDP^{f}_{v}(q_{sum,v} - 1 + q_1 + 1 - (i+1)), \\ &= P^{f}_{1(q_1+1)} + \sum_{i=0}^{q_1}(P^{f}_{1i} - P^{f}_{1(i+1)})PDP^{f}_{v}(q_{sum,v} + q_1 - (i+1)). \end{aligned}$$

Similar to Theorem 7.4 (from Chapter 7), $PDP^f_{2v} \geq PDP^{f'}_{2v}$ and we can conclude that the optimal ARQ distribution is $[q_{sum} - (N-1), 1, \ldots, 1]$ for a fully-cumulative network. Hence, it completes the proof. □

To showcase the benefit of using the fully-cumulative network, we plot the results in Fig. 8.6. It can be observed that there is a significant reduction in the PDP when we use a fully-cumulative scheme. However, as mentioned in Section 7.6 (from Chapter 7), the fully-cumulative network uses the counter that needs to be updated at each hop to convey the residual ARQs to the next node in the chain. Therefore, it may increase the delay overhead compared to the non-cumulative network. This is the cost associated with the fully-cumulative network in order to reduce the PDP of a non-cumulative network.

In the next section, we provide a detailed analysis on the latency for CC-HARQ-FF based non-cumulative and fully-cumulative networks.

8.5 Simulation Results on Delay Analysis for CC-HARQ-FF Strategy

Similar to Section 7.7, in this section, we present a thorough analysis on the end-to-end delay for CC-HARQ-FF strategies by using the same set of delay and deadline-violation metrics. First, we show that the packets in a non-cumulative network reach the destination before the deadline with a higher probability, provided that the delay overheads from ACK/NACK are sufficiently small. To demonstrate the results, we obtain q_{sum} as $\lfloor \frac{\tau_{total}}{\tau_p + \tau_d} \rfloor$ without considering the resources for ACK/NACK in the reverse channel, where τ_{total}, τ_d and τ_p are as defined in Chapter 1. Subsequently, we introduce different resolution of delays from NACK, say τ_{NACK} time units, and then observe the impact on the end-to-end delay on the packets. Assuming $\tau_p + \tau_d = 1$ microsecond, we set the deadline for end-to-end packet delay as q_{sum} microseconds. Then, by sending an ensemble of 10^6 packets to the destination through the CC-HARQ-FF strategy, we compute the following metrics when $\tau_{NACK} \in \{0.05, 0.2, 0.8\}$ microseconds: (i) P_{Drop}, (ii) $P_{Deadline}$, and finally, (iii) the average end-to-end delay on the packets. These metrics are plotted in Fig. 8.7 for various values of SNR at a specific value of N and the LOS vector $\mathbf{c}$. The plots

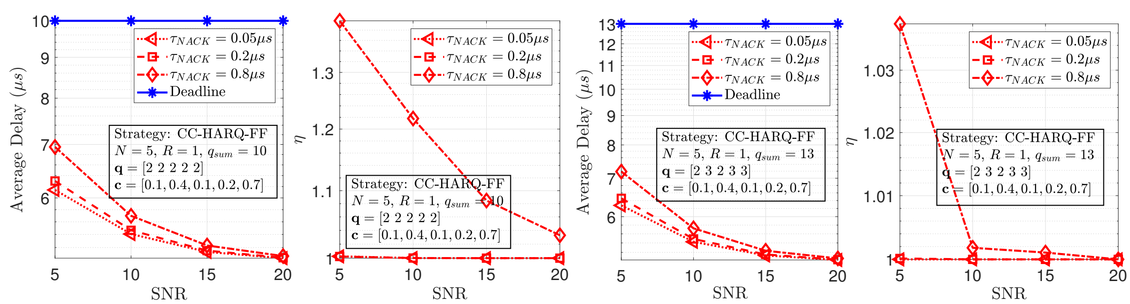

Figure 8.7: Variation of average delay on the packets and the deadline violation parameter (η) for various τ_{NACK} in CC-HARQ-FF based strategies at SNR$= 0$ dB.

suggest that the average delay is significantly lower than that of the deadline especially when

τ_{NACK} is small, owing to the opportunistic nature of CC-HARQ-FF protocol. However, as τ_{NACK} increases, the average delay is pushed slightly closer to the deadline. Furthermore, to capture the behaviour of deadline violations due to higher τ_{NACK}, in Fig. 8.7, we also plot $\eta = \frac{P_{Drop}+P_{Deadline}}{P_{Drop}}$. The plots confirm that when τ_{NACK} is sufficiently small compared to $\tau_p + \tau_d$ (see $\tau_{NACK} = 0.05\mu s$ at SNR = 10, 15, 20 dB), the packets that reach the destination arrive within the deadline with an overwhelming probability as $\eta = 1$ at those values.

In the rest of this section, we present other delay metrics for non-cumulative and fully-cumulative strategies. Similar to Section 7.7, we assume that that the delay introduced on the packet per hop for each transmission is $T = \tau_p + \tau_d = 1$ microsecond, $\tau_{NACK} \in \{0.05, 0.2, 0.8\}$ microseconds and $T_c = \alpha T$, where $\alpha = 0, 0.5,$ and 1. By using these parameters, in Fig. 8.8, we have shown the delay profiles (in percentage) of both non-cumulative and fully-cumulative strategies. For generating the plots, we considered $N = 5$, $q_{sum} = 12$, $R = 1$, SNR= 5 dB, and $\mathbf{c} = \{0.1, 0.5, 0.1, 0.3, 0.7\}$ and an ensemble of 10^6 packets. It can be observed that the delay profiles remain unchanged for the non-cumulative strategy irrespective of the value of α because there is no counter present in it. However, it can be observed in a fully-cumulative network that as α increases, the percentage of packets violating the deadline is more. This can be visualized by the width of the rectangle in Fig. 8.8. Furthermore, when $\alpha = 0$ while designing q_{sum}, there is a non-zero probability that some packets may reach the destination beyond the deadline; however, it is minimal (see the first plot in Fig. 8.8).

Now, in Fig. 8.9, we plot PDV for our CC-HARQ-FF based non-cumulative and fully-cumulative networks as a function of α for a 5-hop network with different q_{sum}. In the plots, $\mathbf{q_{nc}}$ represents the ARQ distribution taken for generating the plots for non-cumulative strategy and $\{q_{sum} - (N-1), 1, \ldots, 1\}$ for the fully-cumulative network. For a given q_{sum}, the deadline for packets to reach the destination is q_{sum} microseconds. The following conclusions can be drawn from the plots: (i) the PDV of the non-cumulative network does not change with α, (ii) the PDV of the fully-cumulative network increases with increasing values of α; this is because $N-1$

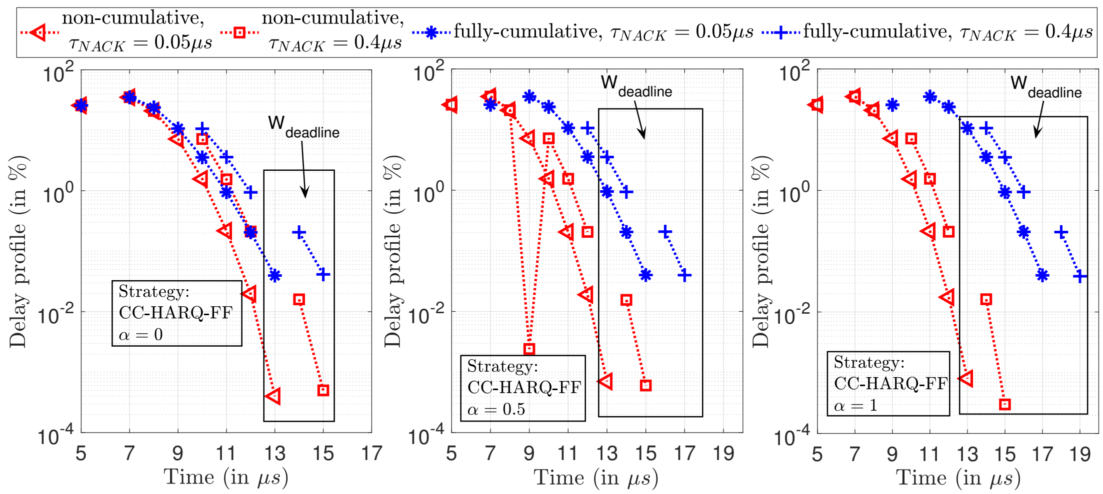

Figure 8.8: Simulation results on delay profiles (in %) for CC-HARQ-FF based strategies using a 5-hop network with $\mathbf{c} = [0.1, 0.5, 0.1, 0.3, 0.7]$ and $q_{sum} = 12$ at rate $R = 1$ and SNR= 5 dB with 10^6 packets wherein some packets are dropped either due to outage and some are dropped due to deadline violation (marked in the rectangle).

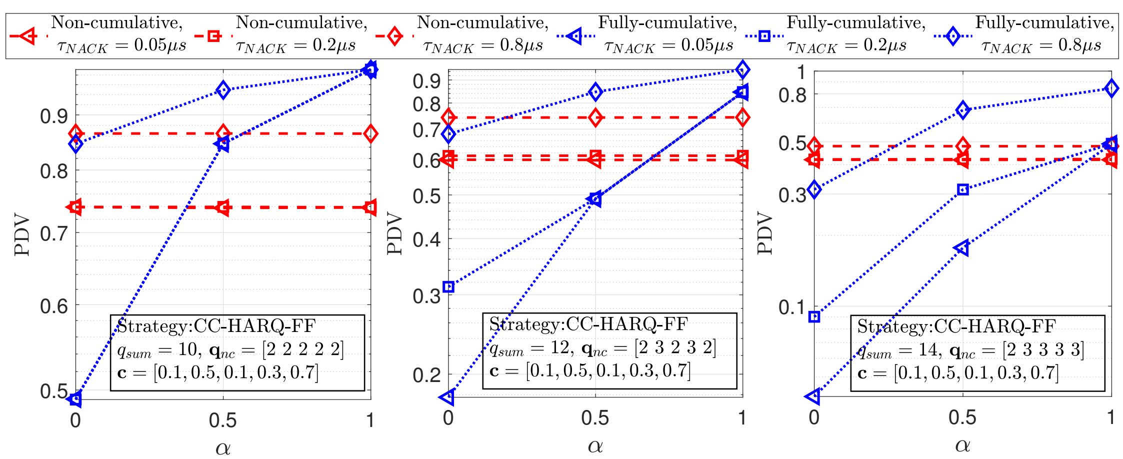

Figure 8.9: Illustration on PDV for non-cumulative and fully-cumulative network implementing the CC-HARQ-FF strategy for different τ_{NACK} and α values.

nodes make use of the counter in the packet, thereby adding a significant delay of $(N-1)T_c$ microseconds to the packet.

8.6 Summary

In this chapter, we have proposed novel CC-HARQ based ARQ strategies for delay-bounded applications in multi-hop networks. In particular, we have addressed the problem of optimally distributing the ARQs to minimize the PDP for a given end-to-end deadline. Specifically, we have considered the fast varying intermediate channels where we have considered non-cumulative and fully-cumulative strategies for multi-hop networks. Towards formulating the optimization problems, we have derived closed-form expressions on the PDP and have solved the non-tractable optimization problems for our strategies. We have shown that the methods of solving the optimization problems are distinct for each strategy because they contain the Marcum-Q function of different orders. In addition, all the theoretical results and algorithms have been validated through simulations along with a detailed analysis on end-to-end delay for each of our proposed strategies.

Part V

Final Remarks

Chapter 9

Conclusions and Future Directions

9.1 Conclusions

This thesis addressed the issue of optimally distributing the Automatic Repeat Requests (ARQs) in the multi-hop networks. The contributions of this thesis are as follows:

1. We proposed a non-cooperative ARQ based DF strategy, wherein each node only knows the number of ARQs allotted to itself; it does not know the ARQs assigned to the other nodes in the network. With such constraints, we formulated an optimization problem of distributing an appropriate number of ARQs at each link such that the PDP is minimized for a given q_{sum}. We showed that the optimization problem is non-linear with non-negative integer constraints on the solution and therefore, it is extremely challenging to solve it. Towards finding the solution of the optimization problem, we derived sufficient and necessary conditions on the optimal ARQ distribution, and then proposed a low-complexity algorithm. Through simulation results, we showed that the proposed algorithm is amenable to implementation in practice, and also provides the optimal solution of the problem. We also presented the simulation results on the delay profiles of the packets by comparing the

non-cooperative ARQ strategy with a standard baseline that does not require the use of ACK/NACK. The simulation results showed that the ARQ strategy assists in reducing the average delay on the packets since the idea of asking for re-transmissions outweigh the delay-overhead introduced by NACK.

2. After allocating an appropriate number of ARQs at each node, we identified that there may be unused ARQs at some relays owing to the stochastic nature of the wireless channels. Therefore, without violating the sum constraint, we proposed a variety of cooperative ARQ strategies wherein the unused ARQs of one node can be used by other nodes with negligible increase in the communication-overhead. We referred to such strategies as the fully-cumulative ARQ schemes, cluster-based ARQ schemes, semi-cumulative ARQ schemes, and cluster-based semi-cumulative ARQ schemes. Subsequently, we characterized the PDP expressions of each of the above schemes, and then solved the problem of allocating an appropriate number of ARQs to the relay nodes so as to minimize the PDP under the sum constraint on the total number of ARQs. Through extensive analysis and simulation results, we showed that the non-cooperative ARQ strategy and the fully-cumulative ARQ strategy respectively offered the worst and the best PDP performance among the schemes, whereas the class of semi-cumulative ARQ schemes trade-offed PDP with the communication-overhead between these two extreme classes of ARQ based DF strategies. Overall, in contrast to the idea of optimizing the reliability of each link in a hop-by-hop manner, our approach jointly optimized the ARQ allocation across the links because of the constraints on end-to-end delay.

3. We proposed CC-HARQ based strategies for multi-hop networks with slow-fading channels (denoted as CC-HARQ-SF), wherein the channels are assumed to be static over the allotted attempts at each link. Under this scenarios, we proposed two types of strategies, namely: non-cumulative strategy and the fully-cumulative strategy. For the non-cumulative

strategy, we derived a closed-form expression on PDP and formulated an optimization problem of minimizing the PDP for a given q_{sum}. We showed that the optimization problem is non-tractable as it contains Marcum-Q function of first-order. Towards obtaining near-optimal ARQ distributions, we proposed a tight approximation on the first-order Marcum-Q functions, and then presented non-trivial theoretical results for synthesizing a low-complexity algorithm. Through extensive simulations, we showed that our analysis on the near-optimal ARQ distribution gives us the desired results with affordable complexity. For the fully-cumulative strategy, we provided theoretical results on the optimal ARQ distribution in the closed-form.

4. We proposed CC-HARQ based strategies for fast-fading scenarios (denoted as CC-HARQ-FF), wherein the channels are statistically independent across allotted attempts at each link. Similar to CC-HARQ-SF strategies, under this scenarios, we proposed two types of strategies, namely: non-cumulative strategy and the fully-cumulative strategy. For the non-cumulative strategy, we derived a closed-form expression on PDP and formulated an optimization problem of minimizing the PDP for a given q_{sum}. We showed that the optimization problem is non-tractable as it contains Marcum-Q function of higher-order. Towards obtaining near-optimal ARQ distributions, we proposed a tight approximation on the higher-order Marcum-Q functions, and then presented non-trivial theoretical results for synthesizing a low-complexity algorithm. Similar to CC-HARQ-SF strategies, using extensive simulations, we showed that our analysis on the near-optimal ARQ distribution gives us the desired results with manageable complexity. For the fully-cumulative strategy, we presented theoretical results on the optimal ARQ distribution in closed-form.

5. For each of our strategies, we have presented a detailed analysis on end-to-end delay by considering the following metrics: (i) average end-to-end delay, (ii) packet deadline violation (PDV), which is defined by the number of packets reaching the destination after

Table 9.1: Summary of various protocols presented in this thesis

Protocol	**ARQ knowledge at a node**	**Overheads**	**PDP**	**Complexity to compute optimal ARQ distribution**
Non-cooperative	ARQs allotted to itself	no counter needed	higher than cooperative	low
Pair-wise semi-cumulative	ARQs allotted to itself and the unused ARQs of the preceding node in the pair	no counter	lower than non-cooperative	higher than non-cooperative
Cluster-wise semi-cumulative	ARQs allotted to itself and unused ARQs of the preceding nodes in the cluster	counter needed	lower than pair-wise	same as pair-wise scheme
Fully-cumulative	ARQs allotted to itself and unused ARQs of all preceding nodes	counter needed	lowest among the schemes	optimal ARQ distribution known in closed form

the given deadline, and (iii) delay profile, which represents the percentage of packets reaching the destination at a certain time for a given deadline. By using the aforementioned delay-metrics, we have provided valuable insights on the merits and demerits of our strategies in achieving high-reliability with bounded constraints on end-to-end delay [30], [35], and [37].

The summary of various protocols presented in this thesis are given in Table 9.1.

9.2 Future Directions

The ideas and the techniques proposed in this thesis can be extended in several ways. Some possible directions for future work are as follows:

1. In this thesis, we have assumed a multi-hop network with a single path with equal process-

ing time at each hop. One of the possible directions for future work is to extend the above technique with multiple paths and have unequal processing time for the packet at each hop.

2. In Chapter 7 and Chapter 8, we considered a non-cooperative and fully-cumulative model for a CC-HARQ based multi-hop network under both slow-fading and fast-fading scenarios. Similar to Chapter 5 and Chapter 6, it would be interesting to analyse the behaviour of PDP for a semi-cumulative based CC-HARQ based model.

Appendix-I

Biodata

Jaya (aka Jaya Goel) is currently working towards her Ph.D. in the Bharti School of Telecommunication Technology and Management at IIT Delhi, India. She joined the Ph.D. program in January 2018. She received her M.Tech. degree in Electronics & Communication Engineering from Guru Gobind Singh Indraprastha University (GGSIPU), Delhi in 2016. She worked at the Institute of Informatics and Communication (IIC), Delhi University for one year before joining Ph.D. Her areas of interest include wireless communication, information theory, optimization problem, wireless security, wireless sensor networks, IoT etc.

List of Publications Based on this Thesis

In Peer Reviewed International Journal

1. Jaya Goel and J. Harshan, “Listen to Others? Failures: Cooperative ARQ Schemes for Low-Latency Communication over Multi-Hop Networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 09, pp. 6049–6063, Sept. 2021.

2. Jaya Goel and J. Harshan, “One-Hop Listening Based ARQs for Low-Latency Communication in Multi-Hop Networks," accepted for publication in *IEEE Transactions on Wireless*

Communication, August 2022. (Available in Early Access on IEEE Xplore).

3. Jaya Goel and J. Harshan, “Hybrid-ARQ Based Relaying Strategies for Enhancing Reliability in Delay-Bounded Networks," submitted in *IEEE Transactions on Communication*, (under review).

In Peer Reviewed International Conferences

1. Jaya Goel and J. Harshan, “On the Optimal ARQ Distribution for Low-Latency Communication over Line-of-Sight Dominated Multi-Hop Networks," in the Proc *18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt 2020)*.

2. Jaya Goel and J. Harshan, “Minimal Overhead ARQ Sharing Strategies for URLLC in Multi-Hop Networks," in the Proc *IEEE Vehicular Technology Conference (VTC2021-Spring)*.

3. Jaya Goel and J. Harshan, “Optimization of ARQ Distribution for HARQ Strategies in Delay-Bounded Networks," in the Proc *20th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt 2022)*.

Bibliography

[1] Y. Zhou, N. Cheng, N. Lu, and X. S. Shen, “Multi-UAV-aided networks: Aerial-ground cooperative vehicular networking architecture," *IEEE Vehicular Technology Magazine*, vol. 10, no. 4, pp. 36–44, Dec. 2015.

[2] F. S. Shaikh and R. Wismüller, “Routing in multi-hop cellular Device-to-Device (D2D) networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 2622–2657, June 2018.

[3] O. S. Badarneh and F. S. Almehmadi, “Performance of multihop wireless networks in α-μ fading channels perturbed by an additive generalized Gaussian noise," *IEEE Communications Letters*, vol. 20, no. 5, pp. 986–989, May 2016.

[4] G. Pocovi, H. Shariatmadari, G. Berardinelli, K. Pedersen, J. Steiner, and Z. Li, “Achieving ultra-reliable low-latency communications: Challenges and envisioned system enhancements," *IEEE Network*, vol. 32, no. 2, pp. 8–15, March 2018.

[5] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, “Ultra-reliable and low-latency communications in 5G downlink: Physical layer aspects," *IEEE Transactions on Wireless Communications*, vol. 25, no. 3, pp. 124–130, June 2018.

[6] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. A. Ashraf, B. Almeroth, J. Voigt, I. Riedel, A. Puschmann, A. Mitschele-Thiel, M. Muller, T. Elste, and M. Windisch, “Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 70–78, Feb. 2017.

[7] D. Chase, “Code Combining- A Maximum-Likelihood decoding approach for combining an arbitrary number of noisy packets," *IEEE Transactions on Communications*, vol. 33, no. 5, pp. 385–393, May 1985.

[8] Y. Chen, N. Zhao, Z. Ding, and M. Alouini, “Multiple UAVs as relays: Multi-hop single link versus multiple dual-hop links," *IEEE Transactions on Wireless Communications*, vol. 17, no. 9, pp. 6348–6359, Sept. 2018.

[9] H. Ren, C. Pan, K. Wang, Y. Deng, M. Elkashlan, and A. Nallanathan, “Achievable data rate for URLLC-enabled UAV systems with 3-D channel model," *IEEE Wireless Communications Letters*, vol. 8, no. 6, pp. 1587–1590, Dec. 2019.

[10] C. She, C. Liu, T. Q. S. Quek, C. Yang, and Y. Li, “Ultra-reliable and low-latency communications in unmanned aerial vehicle communication systems," *IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3768–3781, May 2019.

[11] M. Mozaffari, W. Saad, M. Bennis and M. Debbah,“Mobile Unmanned Aerial Vehicles (UAVs) for Energy-Efficient Internet of Things Communications," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7574–7589, Nov. 2017.

[12] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, “Modeling air-to-ground path loss for low altitude platforms in urban environments," in the Proc. of *2014 IEEE Global Communications Conference*, Austin, TX, pp. 2898–2904, Dec. 2014.

[13] A. Al-Hourani, S. Kandeepan, and S. Lardner, “Optimal lap altitude for maximum coverage," *IEEE Wireless Commun. Letter*, vol. 3, no. 6, pp. 569–572, Dec. 2014.

[14] X. Lin et al.,“The sky is not the limit: LTE for unmanned aerial vehicles?, *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 204–210, Apr. 2018.

[15] J. N. Laneman, D. N. C. Tse and G. W. Wornell, “Cooperative diversity in wireless networks: Efficient protocols and outage behavior," in *IEEE Trans. on Information Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[16] H. Wiemann, M. Meyer, R. Ludwig, and Chang Pae O, “A novel multi-hop ARQ concept," in the Proc. of *IEEE Vehicular Technology Conference*, pp. 3097–3101, May 2005.

[17] T. V. Chaitanya, E. G. Larsson, “Optimal power allocation for hybrid ARQ with chase combining in iid Rayleigh fading channels," *IEEE Transactions on Communications*, vol. 61, no. 05, pp. 1835–1846, May 2013.

[18] C. Madapatha et al. “On integrated access and backhaul networks: current status and potentials," *IEEE Open Journal of the Communications Society*, pp. 1374–1389, 2020.

[19] C. Shen, and M. P. Fitz, “Hybrid ARQ in multiple-antenna slow fading channels: Performance limits and optimal linear dispersion code design," *IEEE Transactions on Information Theory*, vol. 57, no. 9, pp. 5863–83, Sept. 2011.

[20] Faisal Nadeem et al., “Analysis and Optimization of HARQ for URLLC," *IEEE Globecomm workshop GC Wkshps*, Dec 2021.

[21] G. Caire and D. Tuninetti, “The throughput of hybrid-ARQ protocols for the Gaussian collision channel," *IEEE Transactions on Information Theory*, vol. 47, no. 05, pp. 1971–1988, July 2001.

[22] D. Tuninetti, “On the benefits of partial channel state information for repetition protocols in block fading channels," *IEEE Transactions on Information Theory*, vol 57, no. 8, pp. 5036–5053, August 2011.

[23] M. K. Sharma and C. R. Murthy, “Packet drop probability analysis of dual energy harvesting links with retransmission," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3646–3660, Dec. 2016.

[24] M. Tacca and P. Monti and A. Fumagalli, “Cooperative and reliable ARQ protocols for energy harvesting wireless sensor nodes," *IEEE Transactions on Wireless Communications*, vol. 6, no. 7, pp. 2519–2529, July 2007.

[25] L. Le and E. Hossain, “An analytical model for ARQ cooperative diversity in multi-hop wireless networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 5, pp. 1786–1791, May 2008.

[26] M. Serror, C. Dombrowski, K. Wehrle, and J. Gross, “Channel coding versus cooperative ARQ: Reducing outage probability in ultra-low latency wireless communications," in the Proc. of *IEEE Globecom Workshops*, Dec. 2015.

[27] G. E. Corazza and G. Ferrari, “New bounds for the Marcum Q-function," *IEEE Transactions on Information Theory*, vol. 48, no. 11, pp. 3003–3008, Nov. 2002.

[28] P. Mary et al. “Finite Blocklength Information Theory: What is the Practical Impact on Wireless Communications?," IEEE Globecom Workshops *(GC Wkshps)*, Washington DC, USA, pp. 1–6, 2016.

[29] Jaya Goel and J. Harshan, “On the Optimal ARQ Distribution for Low-Latency Communication over Line-of-Sight Dominated Multi-Hop Networks," in the Proc. of *WiOpt 2020*, Greece, June 2020.

[30] Jaya Goel and J. Harshan, “Listen to Others? Failures: Cooperative ARQ Schemes for Low-Latency Communication over Multi-Hop Networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 09, pp. 6049–6063, Sept. 2021.

[31] J. Mei, K. Zheng, L. Zhao, Y. Teng, and X. Wang, “A latency and reliability guaranteed resource allocation scheme for LTE V2V communication systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 3850–3860, June 2018.

[32] Y. Polyanskiy, H. V. Poor and S. Verdu, “Channel Coding Rate in the Finite Blocklength Regime," in *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[33] 3GPP TS36.211, “LTE Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation," v13.0.0, 2016-01.

[34] Jaya Goel and J. Harshan, “Minimal Overhead ARQ Sharing Strategies for URLLC in Multi-Hop Networks," 2021 IEEE 93rd Vehicular Technology Conference, *VTC 2021-Spring*, pp. 1–7, June 2021.

[35] Jaya Goel and J. Harshan, “One-Hop Listening Based ARQs for Low-Latency Communication in Multi-Hop Networks," accepted in *IEEE Transactions on Wireless Communication*, August 2022.

[36] Jaya Goel and J. Harshan, “Optimization of ARQ Distribution for HARQ Strategies in Delay-Bounded Networks," in the proceeding of 20th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks *(WiOpt)*, Sept. 2022. Also available on ArXiv.

[37] Jaya Goel and J. Harshan, “Hybrid-ARQ Based Relaying Strategies for Enhancing Reliability in Delay-Bounded Networks," submitted in *IEEE Transactions on Communication*, (under review). Also available on ArXiv.