Tiny Codes for Guaranteeable Delay

Derya Malak, Muriel Médard, and Edmund M. Yeh

Abstract

Future 5G systems will need to support ultra-reliable low-latency communications scenarios. From a latency-reliability viewpoint, it is inefficient to rely on average utility-based system design. Therefore, we introduce the notion of guaranteeable delay which is the average delay plus three standard deviations of the mean. We investigate the trade-off between guaranteeable delay and throughput for point-to-point wireless erasure links with unreliable and delayed feedback, by bringing together signal flow techniques to the area of coding. We use tiny codes, i.e. sliding window by coding with just 2 packets, and design three variations of selective-repeat ARQ protocols, by building on the baseline scheme, i.e. uncoded ARQ, developed by Ausavapattanakun and Nosratinia: (i) Hybrid ARQ with soft combining at the receiver; (ii) cumulative feedback-based ARQ without rate adaptation; and (iii) Coded ARQ with rate adaptation based on the cumulative feedback. Contrasting the performance of these protocols with uncoded ARQ, we demonstrate that HARQ performs only slightly better, cumulative feedback-based ARQ does not provide significant throughput while it has better average delay, and Coded ARQ can provide gains up to about 40% in terms of throughput. Coded ARQ also provides delay guarantees, and is robust to various challenges such as imperfect and delayed feedback, burst erasures, and round-trip time fluctuations. This feature may be preferable for meeting the strict end-to-end latency and reliability requirements of future use cases of ultra-reliable low-latency communications in 5G, such as mission-critical communications and industrial control for critical control messaging.

Index Terms

Coding, feedback, Gilbert-Elliott, ARQ, HARQ, signal-flow graph, erasure, burst, guaranteeable delay.

D. Malak and M. Médard are with the Research Laboratory of Electronics (RLE), at The Massachusetts Institute of Technology, Cambridge, MA 02139 USA (email: {deryam, medard}@mit.edu). E. Yeh is with Electrical and Computer Engineering Department, at Northeastern University, Boston, MA 02115, USA (email: eyeh@ece.neu.edu). Article last revised: February 5, 2019.

I. Introduction

Ultra reliability and low latency in 5G are key factors for many applications ranging from industrial control (automation) [1], tactile Internet [2], interactive gaming, remote healthcare, financial services, smart cities, public safety, defense [3], to mission-critical communications such as autonomous driving, drones, virtual and enhanced reality (wearable computing devices) [4]–[8]. Ultra-reliable low-latency communications (URLLC) for a large number of Internet of Things (IoT) devices will be an important use case for future 5G communication networks [9]. Such scenarios have strict requirements in terms of capacity, end-to-end latency (on the order of about a few milliseconds) and reliability (higher than 99.9999%). 5G will need to support a round-trip time (RTT) of about 1 millisecond, an order of magnitude faster than 4G, along with necessary overhead for resource allocation and access in 5G networks. Such severe latency constraints together with the associated control information introduce a plethora of challenges in terms of the protocol stack design, control/user plane, and the core network [10].

Inevitable feedback channel impairments may cause unreliability in packet delivery. A negative acknowledgement (NACK), falsely received as positive acknowledgement (ACK) results in undesirable packet outage. Repetition of a packet and forward error-correction (FEC) help repair the loss of the packets over non-deterministic channel conditions. The role of feedback is to increase the reliability in packet delivery and the channel efficiency by limiting the repetitions. Automatic Repeat reQuest (ARQ) and hybrid ARQ (HARQ), which combines FEC and ARQ error control, have been used in 5G mobile networks [11], to boost the performance of wireless technologies such as HSPA, WiMAX and LTE [12]. A network coding based HARQ algorithm, which combines FEC- and network-coding-based ARQ to maximize the throughput and video quality for wireless video broadcast, has been proposed in [13]. ARQ and HARQ perform together to provide robustness in 4G LTE networks, and a system with reliable packet delivery. Failure in HARQ is compensated for by ARQ at the expense of extra latency for the packet [14]. While enhanced mobile broadband (eMBB) aims at high spectral efficiency, it can also rely on HARQ retransmissions to achieve high reliability. However, this might not be the case for URLLC due to the hard latency constraints [4]. In this case, it is required to depart from average delay-based models, and design wireless systems that provide delay guarantees.

A. Related Work

Different classes of codes have been proposed to correct errors over packet erasure channels. Block codes require a packet stream to be partitioned into blocks, each block being treated independently from the rest. Block codes for error correction have been considered in [15]. Streaming codes, e.g. convolutional codes, have the flexibility of grouping the blocks of information in an appropriate way, and decoding the part of the sequence with fewer erasures. They can correct more errors than classical block codes when considering the erasure channel [15], [16]. Fountain codes have efficient encoding and decoding algorithms, and are capacity-achieving. However, they are not suitable for streaming because the decoding delay is proportional to the size of the data [17].

Network RTT can be estimated using delay measurements reported from the receiver's acknowledgments [18]. RTT can also be estimated using traffic already flowing between two points without requiring precise time synchronization between each point [19]. However, measuring the time-varying RTT actually experienced by an application is a substantial challenge, particularly when the RTT fluctuates more frequently than one can sample the path. The unpredictability of RTT might have massive effects on upper layer protocols. Therefore, it is required to design robust protocols which are more predictable across statistics, hence are more stable when RTT is unreliable.

Feedback and coding over a broadcast erasure channel have been combined in [20] to optimize decoding delay when perfect feedback is available from the receivers. The achievable rate has been optimized using feedback and coding, under the condition that each received packet is either useless or can be immediately decoded by the destination [21]. An extension of ARQ for coded networks has been proposed in [22] to minimize the queue size at the transmitter. This approach combines the benefits of network coding and ARQ by acknowledging degrees of freedom (DoF) instead of original packets. It enables the feedback-based control of the tradeoff between throughput and decoding delay [23]. The proposed scheme in [22] is robust to delayed or imperfect feedback. However, none of these examples jointly investigates the delay and throughput when the feedback is imperfect.

For schemes requiring feedback, it is generally assumed that feedback is lossless and delay-free [22], [23]. Imperfect and delayed feedback may cause unreliability in packet delivery. Inevitable feedback channel impairments and burst errors may impede the protocol stability. The situation becomes worse under RTT fluctuations along with the delayed feedback. A method of acknowledging packet delivery for retransmission protocols with unreliable feedback has been proposed in [12]. Based on backwards composite acknowledgment from multiple packets, the scheduler can exploit the channel quality to increase reliability at the cost of a small increase in average delay. Attempts to increase feedback reliability via repetition coding might be costly to the receiver node while erroneous feedback detection may increase packet delivery latency and diminish throughput and reliability. In LTE, blind HARQ retransmissions of a packet are proposed to avoid feedback complexity and increase reliability [24]. However, this approach can severely decrease resource utilization efficiency.

Using FEC, in-order delivery delay over packet erasure channels can be reduced [15], and the performance of SR ARQ protocols can be boosted. Delay bounds for convolutional codes have been provided in [25]. Packet dropping to reduce playback delay of streaming over an erasure channel has been investigated in [26]. Delay-optimal codes without feedback for burst erasure channels, and the decoding delay of codes for more general erasure models have been analyzed in [27]. Despite all these prior attempts, feedback and coding have been difficult to blend, and to the best of our knowledge, unreliable feedback has not been analyzed in the area of coding before.

Throughput-delay tradeoffs of low-latency communications have been studied at the physical layer. Delay-limited link capacity has been investigated in [28], which focused on minimizing the average delay instead of the worst-case delay. Complexity of various channel coding schemes for URLLC in 5G has been investigated in [29]. At the network layer, recent work includes end-to-end delay bounds in wireless networks using large deviations theory [30], edge caching [31], the use of short transmission time interval [32], HARQ retransmissions to meet target reliability rate in the uplink [33] or in the downlink [34], non-orthogonal multiple access [35], and delay-limited throughput [36]. From a coding perspective, average delay of network coding in downlink has been studied [37]. A coded ARQ scheme for delay-free feedback has been proposed in [38]. To the best of our knowledge, coding has only been studied from an average delay perspective, but minimizing the worst-case delay, i.e. providing delay guarantees, is crucial in 5G system design that also supports URLLC.

B. Contributions

We investigate the trade-off between throughput and guaranteeable delay over packet erasure channels with unreliable and delayed feedback. By building on the uncoded baseline scheme in [39], we propose three protocols: (i) Hybrid ARQ (HARQ) with soft combining at the receiver, where the feedback is not cumulative; (ii) Cumulative feedback-based ARQ (CF ARQ) without rate adaptation, where feedback includes the extra information regarding the previous packets; and (iii) Coded ARQ with rate adaptation based on the cumulative feedback.

We use tiny codes, i.e. sliding window with just 2 packets. For the Gilbert-Elliott channels, we analyze the distributions of transmission time and delay by exploiting signal-flow techniques. We provide exact closed-form expressions of throughput and delay for memoryless channels, as functions of the system parameters such as the timeout, the RTT and the packet erasure rate. Contrasting the new protocols with uncoded ARQ, we demonstrate that HARQ performs only slightly better. While CF ARQ has lower average delay and has benefits under burst erasures or high erasure rates, it does not provide a significant throughput gain (up to about 18%). Coded ARQ can provide throughput gains up to about 40% when the average erasure burst is higher than 3. It also provides delay guarantees, and is robust to various challenges such as imperfect and delayed feedback, high erasure rates, burst erasures, and RTT fluctuations. Coded ARQ is more predictable across statistics, hence is more stable.

To the best of our knowledge, we bring signal-flow techniques for the first time, to the area of coding, melding two areas that have hitherto been quite separate. Our results permit analysis of the perennially vexing problem of accurately accounting for delay when coding.

II. System Model

We consider a point-to-point channel model consisting of a sender and a receiver. As illustrated in Fig. 1, on the forward link, the sender attempts to transmit a packet to the receiver, and upon the successful reception of the packet, on the reverse link, the receiver acknowledges the sender by transmitting a feedback. Erasure errors can occur in both the forward and reverse channels. However, an ACK cannot be decoded as a NACK, and vice versa. For the convenience of the reader, we follow the notation of [39]. The status of a transmission at time t is a Bernoulli random variable taking values in $\mathcal{X} = \{0, 1\}$, where 0 denotes an error-free packet, and 1 means the packet is erased. The erasure rate ϵ is a function of channel condition. Both for the forward and reverse links we use a Gilbert-Elliott (GE) channel model [40], which is a binary-state Markov process S_t with states G (good) and B (bad), i.e. $\mathcal{S} = \{G, B\}$, and probability transition matrix $\mathbf{P}$. The packet erasure rates in states G and B are

ϵ_G and ϵ_B, respectively. We let $\boldsymbol{\epsilon} = [\epsilon_G, \epsilon_B]$. Since the channel state is not the same as the channel observation, the process X_t is a hidden Markov model (HMM)[1], which is driven by the process S_t.

The channel state information is not available at the transmitter and the receiver. Hence, the transmitter does not know the state of the forward link at time t, but it observes the status of the feedback at time $t-1$, which is a Bernoulli random variable. Similarly, the receiver does not know the status of the reverse link, but it observes the status of a transmission at time t. The joint probabilities of channel state and observation at time t can be computed using the state-transition matrix of the GE channel: $\mathbf{P} = [p_{ij}] \in \mathbb{R}^{2\times 2}$, $i, j \in \mathcal{S}$, where $p_{GB} = 1 - p_{GG} = q$, $p_{BG} = 1 - p_{BB} = r$, i.e., the first and second rows correspond to the transition probabilities of states G and B, given the channel state at time $t-1$. Solving $\pi\mathbf{P} = \pi$ and $\pi\mathbf{1} = 1$, where $\mathbf{1}$ is a column vector of ones, the stationary vector of $\mathbf{P}$ is $\pi = [\frac{r}{r+q}, \ \frac{q}{r+q}]$. The erasure rate is $\epsilon = \pi\boldsymbol{\epsilon}^\intercal$. Given r, ϵ_G, ϵ_B, and ϵ, we have $q = r\big(\frac{\epsilon_B - \epsilon_G}{\epsilon_B - \epsilon} - 1\big)$. Note that $1/r$ represents the average erasure burst, and burst errors occur when r is low. The joint probabilities of channel state and observation at time t, given the channel state at time $t-1$, are

$$\mathbb{P}(S_t = j, X_t = 1 | S_{t-1} = i) = \mathbb{P}(S_t = j | S_{t-1} = i)\mathbb{P}(X_t = 1 | S_t = j) = p_{ij}\epsilon_j, \quad i, j \in \mathcal{S}.$$

Let $\mathbf{P}_1 = \mathbf{P}\cdot\mathrm{diag}\{\boldsymbol{\epsilon}\}$ be the error matrix on the forward (or reverse) link. Similarly, $\mathbf{P}_0 = \mathbf{P}\cdot\mathrm{diag}\{\mathbf{1}-\boldsymbol{\epsilon}\}$ is the success matrix in either link. The entries of $\mathbf{P}_0$ and $\mathbf{P}_1$ are the joint state-transition probabilities given the channel observations [39]. Hence, the HMM can be characterized by $\{\mathcal{S}, \mathcal{X}, \mathbf{P}_0, \mathbf{P}_1\}$.

Consider the forward link $\{\mathcal{S}^{(f)}, \mathcal{X}^{(f)}, \mathbf{P}_0^{(f)}, \mathbf{P}_1^{(f)}\}$ and the reverse link $\{\mathcal{S}^{(r)}, \mathcal{X}^{(r)}, \mathbf{P}_0^{(r)}, \mathbf{P}_1^{(r)}\}$ that are mutually independent. The composite channel is characterized by $\{\mathcal{S}^{(c)}, \mathcal{X}^{(c)}, \mathbf{P}_{00}^{(c)}, \mathbf{P}_{01}^{(c)}, \mathbf{P}_{10}^{(c)}, \mathbf{P}_{11}^{(c)}\}$, where $\mathcal{S}^{(c)} = \mathcal{S}^{(f)} \times \mathcal{S}^{(r)}$ are the composite channel states, i.e. the Cartesian product of forward and reverse states, and $\mathcal{X}^{(c)} = \mathcal{X}^{(f)} \times \mathcal{X}^{(r)} = \{00, 01, 10, 11\}$ is the combined observation set. Note that $X_t^{(c)} = 00$ means both the forward and reverse channels are good, while $X_t^{(c)} = 10$ means the forward channel is erroneous and the reverse channel is good. For $X_t^{(c)} = 11$, the joint probability of the combined observation and the composite state at time t, given the composite state at time $t-1$, is

$$\mathbb{P}(S_t^{(c)} = (j, m), X_t^{(c)} = 11 | S_{t-1}^{(c)} = (i, k)) = (p_{ij}^{(f)}\epsilon_j^{(f)}) \cdot (p_{km}^{(r)}\epsilon_m^{(r)}). \tag{1}$$

Using the Kronecker product notation $\otimes$, we have $\mathbf{P}_{ij}^{(c)} = \mathbf{P}_i^{(f)} \otimes \mathbf{P}_j^{(r)}$ for the combined observation

[1] HMM is a statistical Markov process with unobserved states [41]. Although the state is not directly observed, the output dependent on the state can be observed.

at time t, i.e., $X_t^{(c)} = ij$, $i, j \in \mathcal{X}$. We assume that both the forward and the reverse channels have the same parameters[2] r, ϵ_G, ϵ_B, and ϵ. Hence, the state-transition matrix for both the forward and reverse channels is given by $\mathbf{P}$. In the rest of the paper, we will drop the superscript $^{(c)}$ and denote the observation probability matrices by $\mathbf{P}_{00}$, $\mathbf{P}_{01}$, $\mathbf{P}_{10}$ and $\mathbf{P}_{11}$. Similar to [39], let $\mathbf{P}_{0x} = \mathbf{P}_{00} + \mathbf{P}_{01}$ and $\mathbf{P}_{1x} = \mathbf{P}_{10} + \mathbf{P}_{11}$ be the success and error probability matrices on the forward channel, respectively, and let $\mathbf{P}_{x0} = \mathbf{P}_{00} + \mathbf{P}_{10}$ and $\mathbf{P}_{x1} = \mathbf{P}_{01} + \mathbf{P}_{11}$ be the success and error probability matrices on the reverse channel, respectively. The matrices $\mathbf{P}$, $\mathbf{P}_0$, and $\mathbf{P}_1$ will denote the composite channel matrices, i.e. the Kronecker product of the forward and reverse channel matrices. The matrices for the GE channel are provided in Appendix A.

III. Analysis of ARQ

In this section, we describe the protocol for the proposed channel model, signal-flow graphs, as well as a primer on the MSFGs for throughput and delay of ARQ protocols. We analyze the throughput and guaranteeable delay of uncoded ARQ, and provide exact expressions for memoryless channels.

A. Protocol

We use a slotted Selective Repeat (SR) ARQ protocol for data transmission. With SR ARQ, the sender sends a number of packets specified by a window size without the need to wait for individual ACK from the receiver. SR ARQ allows the receiver to accept packets out of order, which can be stored in a buffer and sorted at the receiver to ensure in-order delivery[3]. The receiver may selectively reject the packets, and the sender individually retransmits packets that have timed out. All data packets are available at the transmitter prior to any transmission, and the receiver does not have buffer overflows. There is a handshake mechanism between the sender and receiver that initiates a synchronous transmission. After the start of transmission, the RTT is k slots, i.e. it takes $k-1$ time slots between the transmission of a packet and receipt of its feedback.

[2] The forward and reverse channels do not necessarily have the same parameters. In practice, data packets and feedback packets typically have different lengths and different coding levels. One can use the same method to obtain the results for different channel parameters [42]. Furthermore, data packets generally travel downstream from the sender towards receivers, and feedback packets travel upstream from receivers to the sender [43]. Hence, to compensate the channel asymmetry, more bandwidth can be allocated on downlink.

[3] If there is full feedback, ARQ achieves 100% throughput and the lowest possible packet delay over an erasure channel, and it is composable across links [22]. However, when the network is lossy, link-by-link ARQ cannot achieve the capacity of a general network.

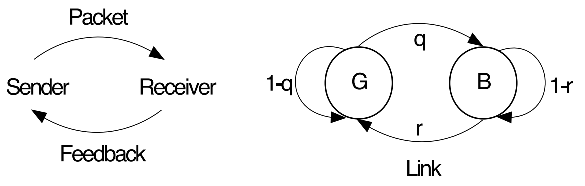

Fig. 1: Point-to-point channel model.

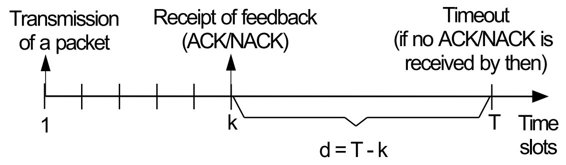

Fig. 2: SR ARQ protocol description.

At the sender, when a packet is (re)transmitted, the timeout associated with this packet is set to T, which is greater than or equal to the RTT k. Upon the reception of the first feedback, the waiting will be aborted after the timer expires, i.e., after $d = T - k$ slots. The feedback – ACK/NACK sent by the receiver indicating if it has correctly received a data packet – includes the information about all correctly received packets. The ACK/NACK is sent in each slot. Thus, the packet whose ACK is lost will be acknowledged by the subsequent ACKs/NACKs. If a succeeding ACK/NACK is successfully received before the timeout, the packet will not be retransmitted. Otherwise, the sender retransmits the packet until it receives an ACK. Hence, we do not have an upper bound on the maximum number of retransmissions of the same packet to guarantee its reliable delivery. If a packet is lost, the packet will be retransmitted immediately (if its NACK is received), or after the timer expires (if the NACK is also lost). The protocol is shown in Fig. 2.

B. A Primer on Signal-Flow Graphs

A signal-flow graph is a diagram that consists of a set of nodes that denote the different states of the system, and a set of directed branches that represent the functional relationships among the states. The analysis of finite-state HMMs can be streamlined by using signal-flow graphs, and labeling the branches of flow graphs with observation probabilities [44], [45]. We next detail how to build the flow graphs for the analysis of SR ARQ.

In the current paper, the nodes of the flow graphs correspond to the states of the transmitter. Upon the initial state that a new packet is transmitted (input node I), the transmitter goes from one state to the other. The output node (O) represents correct reception of ACK by the sender, and other nodes are hidden states. A certain value for the random variable X, that for example models the transmission or delay time for ARQ protocols as in [39], [46]–[48], corresponds to a state transition. The value of

X along with its probability p appear in the branch gain as pz^X. Hence, the input-output gain of the graph is a polynomial in z, whose coefficients are the probabilities of corresponding values of X. This polynomial denotes the probability-generating function (PGF) for X, i.e., $\mathbb{E}[z^X]$. Flow graphs with vector node values and branches labeled with observation probability matrices are called matrix signal-flow graphs[4] (MSFGs) [39]. The graph can be simplified using the basic equivalence operations, i.e. parallel, series, and self-loop, and the matrix gain can be computed. Then, the input-output relationship is given by the matrix-generating function (MGF) $\mathbf{\Phi}(z)$.

C. Probability Distributions of the Transmission Time and the Delay

We derive the MGFs for the transmission time and the delay of different SR ARQ protocols. The transmission time τ is defined as the number of packets transmitted per successful packet, while the delay D is the time from when a packet is first transmitted to when its ACK is successfully received at the sender. Both τ and D are random variables with positive integer outcomes. The PGFs $\Phi_\tau(z)$ and $\Phi_D(z)$ of τ and D are derived using their MGFs by pre- and postmultiplications of row and column vectors, respectively. We will discuss how to obtain the MSFGs and the MGFs for the transmission time and delay in Sect. III-D. We now discuss in detail how to obtain $\Phi(z)$'s from $\mathbf{\Phi}(z)$'s.

For the GE channel model, the probability of transmitting a new packet depends on the channel state. From Fig. 1, given that $\epsilon_G = 0$, the probability of transmitting a new packet in state G is $\pi_G(1-q)+\pi_B r$. Similarly, the probability of transmitting a new packet in state B is $(\pi_G q+\pi_B(1-r))(1-\epsilon_B)$. Therefore, the probability vector of transmitting a new packet is

$$\pi_I = \pi\mathbf{P}_0 = [\pi_G(1-q)+\pi_B r,\ (\pi_G q+\pi_B(1-r))(1-\epsilon_B)].$$

In Appendix A, we detail how to compute the probability vectors $\mathbf{P}_0$ and $\mathbf{P}_1$ for the GE channel model as well as the probabilities for the memoryless channel model.

Proposition 1. Distribution of transmission time τ [39]. *The PGF of the transmission time τ is*

$$\phi_\tau(z) = \frac{\pi_I\mathbf{\Phi}_\tau(z)\mathbf{1}}{\pi_I\mathbf{1}} = \frac{1}{1-\epsilon}\pi\mathbf{P}_0\mathbf{\Phi}_\tau(z)\mathbf{1}, \qquad (2)$$

[4] MSFGs have been extensively used in the state-space formulation of feedback theory [49]. They can also be used to model channel erasures, incorporating unreliable feedback.

where $\mathbf{\Phi}_\tau(z)$ is the MGF of τ, $\pi_I = \pi\mathbf{P}_0$ is the probability vector of transmitting a new packet, and $\mathbf{1}$ is a column vector of ones.

The average transmission time $\bar{\tau}$ is found by evaluating the first derivative of $\phi_\tau(z)$ at $z = 1$, i.e. $\bar{\tau} = \phi'_\tau(1)$. We define the throughput η as the reciprocal of the average transmission time, i.e., $\eta = 1/\bar{\tau}$, which is indeed a lower bound on the actual throughput $\mathbb{E}[1/\tau]$ due to the convexity of $1/\tau$, $\tau \geq 0$.

Corollary 1. Throughput for memoryless channels. *When both the forward and reverse links are memoryless, $\eta = 1/\mathbf{\Phi}'_\tau(1)$ since $\pi = 1$, $\mathbf{P}_0 = 1 - \epsilon$, and $\phi_\tau(z) = \mathbf{\Phi}_\tau(z)$.*

Proposition 2. Distribution of delay D [39]. *The PGF of the delay D is given as*

$$\phi_D(z) = \frac{\pi_\mathbf{I}\mathbf{\Phi}_D(z)\mathbf{1}}{\pi_\mathbf{I}\mathbf{1}}, \tag{3}$$

where $\mathbf{\Phi}_D(z)$ is the MGF of the delay. The average delay $\bar{D}$ is found by evaluating the first derivative of the PGF $\phi_D(z)$ at $z = 1$, i.e. $\bar{D} = \phi'_D(1)$.

Corollary 2. Average delay for memoryless channels. *When both the forward and reverse links are memoryless, $\bar{D} = \mathbf{\Phi}'_D(1)$ since $\pi = 1$, $\mathbf{P}_0 = 1 - \epsilon$, and $\phi_D(z) = \mathbf{\Phi}_D(z)$.*

In reality, the feedback is lossy and delayed, burst errors occur, and the fluctuations in the RTT can cause a high variability in the delay. To understand these effects, we exploit the three-sigma rule[5]. The 3σ heuristic is justifiable when the distribution of the delay is sub-Gaussian. If there are constants $C > 0$, $v > 0$ such that $\mathbb{P}(D > d) \leq Ce^{-vd^2}$ for every $d > 0$, then the probability distribution of a random variable D is sub-Gaussian. A sub-Gaussian distribution has strong tail decay property since the tails decay at least as fast as the tails of a Gaussian. In this case, the guaranteeable delay $\hat{D}$ of a protocol is upper bounded by the guaranteeable delay of a Gaussian distribution with the same mean and variance as the distribution of D. Later in Sect. VII, we demonstrate via numerical simulations that the tails of the delay distribution are dominated by the tails of a Gaussian distribution.

Definition 1. Guaranteeable delay. *The guaranteeable delay of the ARQ protocol – given that the*

[5] Even for non-normally distributed variables, at least 88.8% of cases should fall within properly calculated three-sigma intervals, which follows from Chebyshev's Inequality. For unimodal distributions, the probability of being within the interval is at least 95% [50].

distribution of the delay is sub-Gaussian – is defined as

$$\hat{D} = \bar{D} + 3\sigma_D, \tag{4}$$

where $\bar{D}$ is the average delay, and σ_D^2 is the variance of the delay, which is calculated as $\sigma_D^2 = \phi''_D(1) + \bar{D} - \bar{D}^2$, where the term $\phi''_D(1)$ is the second derivative of $\phi_D(z)$ evaluated at $z = 1$.

D. Selective-Repeat ARQ

Each packet is independently transmitted and acknowledged. Hence, it suffices to consider a MSFG of SR protocol for a single packet [39]. The HMMs for throughput and delay analysis of uncoded SR ARQ in unreliable feedback are detailed in [39], which is the baseline model for our paper. The flow graphs are illustrated in Fig. 3. In the flow graph for the transmission time, as shown in Fig. 3-(a), every time a packet is transmitted, the branch gain is multiplied by $z\mathbf{P}^{k-1}$ since transmission time is defined as the number of packets transmitted per successful packet. In the delay graph, however, as shown in Fig. 3-(b), every time a packet is transmitted, the branch gain is multiplied by $z^{k-1}\mathbf{P}^{k-1}$ because the RTT is k slots. In these MSFGs, nodes I and O represent the input and output nodes, and nodes A, B, C, G denote the hidden states. The possibilities are:

- **State I.** This state represents transmission of a new packet by the sender. The probability vector of transmitting a new packet is $\pi_I = \pi\mathbf{P}_0$ (see Appendix A).
- **Transition to state A.** After sending a new packet, the transmitter receives a feedback message $k-1$ time slots later. This state is represented by node A. The branch gains for τ and D are
$$\mathsf{BG}_\tau(I \to A) = z\mathbf{P}^{k-1}, \quad \mathsf{BG}_\mathsf{D}(I \to A) = z^{k-1}\mathbf{P}^{k-1}. \tag{5}$$
- **Transition to state O.** If the feedback is an error-free ACK, which occurs with probability $\mathbf{P}_{00}$, or if it is an erroneous ACK but an error-free ACK/NACK is received before timer expiration, which occurs with probability $\sum_{j=0}^{d-1}\mathbf{P}_{01}\mathbf{P}_{x1}^{j}\mathbf{P}_{x0}$, then the system transits to state O and the packet is removed from the system. The branch gains for τ and D are
$$\mathsf{BG}_\tau(A \to O) = \mathbf{P}_{00} + \sum\nolimits_{j=0}^{d-1}\mathbf{P}_{01}\mathbf{P}_{x1}^{j}\mathbf{P}_{x0}, \quad \mathsf{BG}_\mathsf{D}(A \to O) = z\mathbf{P}_{00}. \tag{6}$$
- **Transition to state B.** This state represents retransmission of an erroneous packet. Denote by π_B its probability vector. For the HMM of the throughput, if the feedback is an error-free NACK,

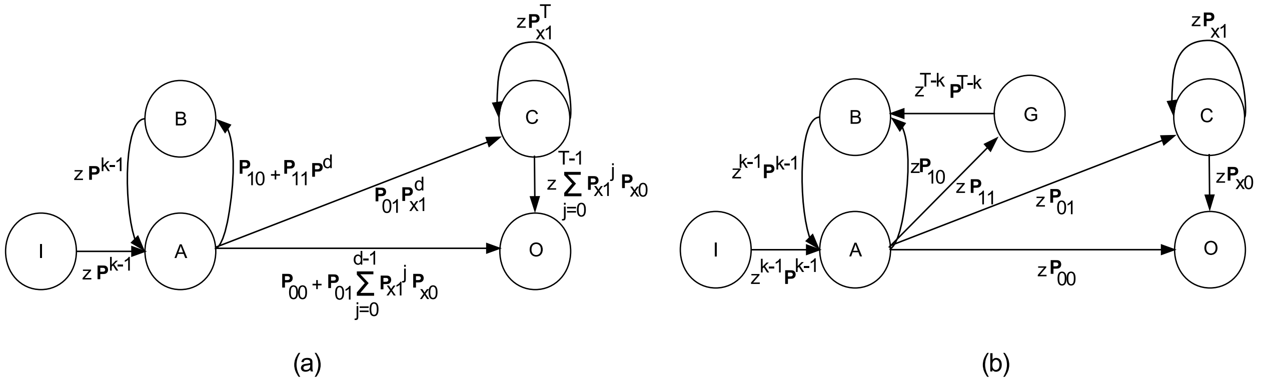

Fig. 3: MSFGs for (a) throughput and (b) delay analysis in unreliable feedback for uncoded ARQ.

with probability $\mathbf{P}_{10}$, or a NACK is lost and the timer expires, with probability[6] $\mathbf{P}_{11}\mathbf{P}^d$, the system goes to state B. Hence, the branch gain for τ is

$$\mathsf{BG}_\tau(A \to B) = \mathbf{P}_{10} + \mathbf{P}_{11}\mathbf{P}^d. \tag{7}$$

In the delay graph however, state B represents only the reception of an error-free NACK at A:

$$\mathsf{BG}_\mathsf{D}(A \to B) = z^{k-1}\mathbf{P}^{k-1}z\mathbf{P}_{10}. \tag{8}$$

The loop between A and B models retransmission of the erroneous packet until correctly received.

- **Transition to state G (delay graph).**[7] If the feedback is an erroneous NACK, the transmitter waits for timeout. The packet will be retransmitted after the timer expires, which involves a delay (transition from G to B). Hence, the branch gain for D is

$$\mathsf{BG}_\mathsf{D}(A \to G) = z^{k-1}\mathbf{P}^{k-1}z\mathbf{P}_{11}z^{T-k}\mathbf{P}^{T-k}. \tag{9}$$

In our delay analysis, we use the following shorthand notation to incorporate the states B and G:

$$\mathsf{BG}_\mathsf{D}(A \to B,\, G) = z\mathbf{P}_{1x}^{\mathrm{D}} = z^k\mathbf{P}^{k-1}\mathbf{P}_{10} + z^T\mathbf{P}^{k-1}\mathbf{P}_{11}\mathbf{P}^{T-k}. \tag{10}$$

- **Transition to state C.** State C represents retransmission of a packet that was correctly received, but with its timer expiring. Denote by π_C its probability vector. If the feedback is an erroneous

[6] When a NACK is lost, the transmitter waits for timeout to retransmit the corresponding packet, which involves a delay of $d = T - k$.

[7] State G is only relevant in the delay graph because losing a NACK causes an additional delay since the sender waits till timeout. However, in the throughput graph, we look at the number of packets transmitted per successful packet. When the forward link is bad, irrespective of whether the NACK is successfully received at the sender or is lost, the packet has to be retransmitted. Therefore, the states G and B can be clumped together into a single state B as shown in Fig. 3-(a).

ACK and the timer expires before receiving any error-free ACKs/NACKs, the system transits to state C, the packet is retransmitted, modeled by the self-loop at state C (the self-loop represents the delay from losing subsequent ACKs/NACKs), and the timeout is reset. The packet is acknowledged when a succeeding ACK/NACK is correctly received. Hence, the branch gains for τ and D are

$$\mathsf{BG}_\tau(A \to C) = \mathbf{P}_{01}\mathbf{P}_{x1}^d, \quad \mathsf{BG}_\mathsf{D}(A \to C) = z\mathbf{P}_{01}. \tag{11}$$

Finally, incorporating the self-loop at C, the branch gains for τ and D from state C and O are

$$\mathsf{BG}_\tau(C \to O) = (\mathbf{I} - z\mathbf{P}_{x1}^T)^{-1} z \sum\nolimits_{j=0}^{T-1} \mathbf{P}_{x1}^j \mathbf{P}_{x0}, \quad \mathsf{BG}_\mathsf{D}(C \to O) = (\mathbf{I} - z\mathbf{P}_{x1})^{-1} z\mathbf{P}_{x0}. \tag{12}$$

Referring to the MSFG for throughput analysis of uncoded ARQ in Fig. 3-(a), a packet will be transmitted only in states I, B, and C. The probability vectors of states I, B, and C are denoted by π_I, π_B, and π_C, respectively. These vectors can be found by solving the following equations [39]:

$$\begin{aligned} \pi_B &= (\pi_I + \pi_B)\mathbf{P}^{k-1}(\mathbf{P}_{10} + \mathbf{P}_{11}\mathbf{P}^{T-k}), \\ \pi_C &= (\pi_I + \pi_B)\mathbf{P}^{k-1}\mathbf{P}_{01}\mathbf{P}_{x1}^{T-k} + \pi_C \mathbf{P}_{x1}^T, \end{aligned} \tag{13}$$

where π satisfies $\pi = \pi_I + \pi_B + \pi_C$, which comes from the fact that the transmitter always has a packet to transmit. Solving for π_I from the system (13), the PGF $\Phi_{\tau_{\mathsf{ARQ}}}(z)$ is derived using (2). We refer the reader to Appendices B and C for the derivations of the MGFs $\mathbf{\Phi}_{\tau_{\mathsf{ARQ}}}(z)$ and $\mathbf{\Phi}_{\mathsf{D}_{\mathsf{ARQ}}}(z)$ [39].

In the following, we investigate various extensions of uncoded ARQ, and analyze the throughput η and the guaranteeable delay $\hat{D}$ by deriving the PGFs using MSFGs. Our analysis is based on the GE channel model. However, to simplify notation and have a better understanding of the main results in Sections III-VI, throughput and delay results are given in closed form for memoryless channels only. Note that one may follow the analysis and obtain the results for the GE channel model, as detailed in Sect. II. Later in Sect. VII, based on the GE channel model, we will provide a numerical comparison of different ARQ protocols in terms of their throughputs and delay guarantees.

IV. Uncoded Hybrid ARQ with Soft Combining

The Hybrid ARQ (HARQ) protocol with soft combining is a repetition-based uncoded transmission scheme, in which incorrectly received packets are stored, and the (re)transmitted packets are combined at the receiver [51]. While it is possible that two given transmissions cannot be independently decoded without error, the combination of the previously erroneously received transmissions may give enough

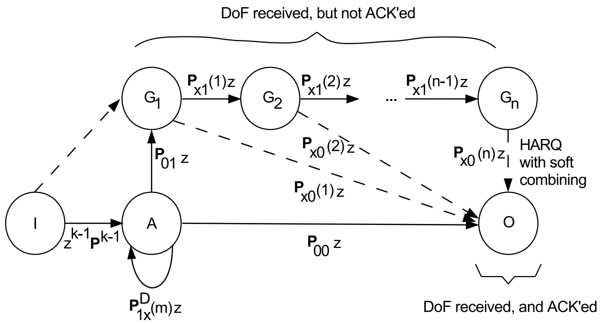

Fig. 4: MSFG for delay analysis of Hybrid ARQ with soft combining at the receiver.

information to be correctly decoded. Hence, this protocol is an improvement on uncoded ARQ in which incorrectly received packets are discarded [39].

We use HARQ with Chase combining such that every (re)transmission contains the same data bits. Because all transmissions are identical, this protocol can be seen as additional repetition coding. Every (re)transmission adds extra energy to the received transmission through an increased E_b/N_0. The receiver uses maximal-ratio combining (MRC) to combine the received bits with the same bits from previous transmission attempts, to successfully decode the transmitted packet. However, since it is a repetition-based data transmission method, it is suboptimal. Different methods, such as incremental redundancy, can be used such that multiple coded packets are generated, each representing the same data packet, and the (re)transmission uses a different coded packet than the previous transmission [52]. Thus, at every (re)transmission the receiver gains extra information.

We illustrate the HMM for the HARQ scheme with soft combining in Fig. 4. The proposed model improves the chance of successful reception at every attempt. Therefore, the erasure rates decrease at each retransmission attempt. We assume that the erasure rate for state G is $\epsilon_G(m) = 0$, and to compute the erasure rate for state B, we exploit the Chase combiner output signal-to-noise ratio (SNR). We have shown in [33, Proposition 1] that the combiner output SNR for a total of m transmissions is SNR$(m) = \rho m$, where $\rho = P_t d^{-\beta}/\sigma^2$ is the received SNR at the receiver per transmission, where P_t is the transmit power, σ^2 is the received noise power, d is the distance between the sender and the receiver, β is the path loss exponent. We then have the following relation:

$$\epsilon_B(m) = \mathbb{P}(\text{SNR}(m) < \gamma) = \mathbb{P}(mP_t d^{-\beta} h/\sigma^2 < \gamma) = \mathbb{P}(h < \gamma P_t^{-1} \sigma^2 d^{\beta}/m) = 1 - e^{-\mu\gamma P_t^{-1}\sigma^2 d^{\beta}/m}.$$

where $\mathrm{SNR}(m)$ is the soft combined SNR at the receiver as a result of m retransmissions, γ is the decoding threshold, and $h \sim \exp(\mu)$ is the channel power gain with parameter μ in the presence of Rayleigh fading. Using the relation $\rho = P_t d^{-\beta}/\sigma^2$ and clumping all the parameters into $\alpha = \mu\gamma/\rho$, we can obtain that the erasure rate for state B satisfies the relation $\epsilon_B(m) = 1 - e^{-\alpha/m}$ as a function of the retransmission attempt m, where the parameter α controls the erasure rate. As $\alpha \to 0$ ($\rho \to \infty$), $\epsilon_B(m) = 0$, and as $\alpha \to \infty$ ($\rho \to 0$), $\epsilon_B(m) = 1$. Hence, as α decreases or m increases, $\epsilon_B(m)$ drops.

On a retransmission attempt m, the branch gains for τ and D for receiving an error-free NACK by the end of the RTT, or receiving an erroneous NACK before the timer expires, equal

$$\begin{aligned}\mathsf{BG}_{\tau}(A \to B) &= \mathbf{P}_{10}(m) + \mathbf{P}_{11}(m)\mathbf{P}^{d},\\ \mathsf{BG}_{\mathsf{D}}(A \to B,\, G) &= z\mathbf{P}^{\mathsf{D}}_{1x}(m) = z^{k}\mathbf{P}^{k-1}\mathbf{P}_{10}(m) + z^{T}\mathbf{P}^{k-1}\mathbf{P}_{11}(m)\mathbf{P}^{T-k},\end{aligned} \tag{14}$$

where $\mathbf{P}_{xy}(m)$ is the composite channel matrix for $X_m^{(c)} = xy$ on attempt m. Note that in the uncoded scheme, the matrices $\mathbf{P}_{x0}(i)$'s and $\mathbf{P}_{x1}(i)$'s do not change with the transmission attempt i.

For the derivation of the MGFs of the transmission and delay times of HARQ with soft combining, reader is referred to Appendix E and Appendix F, respectively. We now present the closed form expressions for throughput and average delay of memoryless channels.

Proposition 3. *The throughput of HARQ for memoryless channels is given by*

$$\begin{aligned}\eta_{\mathsf{HARQ}} = 1\Big/ \Bigg\{ &\sum_{j=0}^{\infty}\Big(\prod_{i=0}^{j}\epsilon(i)\Big)\Bigg[(1-\epsilon(j^*))\epsilon(j^*)\Big(\prod_{i=0}^{d}\epsilon(j^*+i)\Big)\\ &\times\sum_{j=0}^{\infty}(j+1)\prod_{i=1}^{j}\Big(\prod_{l=(i-1)T+1}^{iT}\epsilon(j^*+l)\Big)\sum_{j'=0}^{T-1}\Big(\prod_{i=0}^{j'}\epsilon(j^*+jT+i)\Big)(1-\epsilon(j^*+jT+j'+1))\Bigg]\\ &+\Big(\sum_{j=0}^{\infty}(j+1)\prod_{i=0}^{j}(\epsilon(i))\Big)\Bigg[(1-\epsilon(j^*))^2+(1-\epsilon(j^*))\epsilon(j^*)\sum_{j=0}^{d-1}\Big(\prod_{i=0}^{j}\epsilon(j^*+i)\Big)(1-\epsilon(j^*+j+1))\\ &+(1-\epsilon(j^*))\epsilon(j^*)\Big(\prod_{i=0}^{d}\epsilon(j^*+i)\Big)\sum_{j=0}^{\infty}\prod_{i=1}^{j}\Big(\prod_{l=(i-1)T+1}^{iT}\epsilon(j^*+l)\Big)\\ &\times\sum_{j'=0}^{T-1}\Big(\prod_{i=0}^{j'}\epsilon(j^*+jT+i)\Big)(1-\epsilon(j^*+jT+j'+1))\Bigg]\Bigg\},\end{aligned} \tag{15}$$

where $\epsilon(0) = 1$, $\epsilon(i)$ is the channel erasure rate at retransmission attempt i, and j^* is the required number of forward (re)transmissions for successful decoding.

Proof. See Appendix E. □

Proposition 4. *The average delay of HARQ for memoryless channels is given by [39]*

$$\begin{aligned}\overline{\mathrm{D}}_{\mathsf{HARQ}} &= (1-\epsilon(j^*)) \times \bigg\{ \Big((1-\epsilon(j^*)) + \epsilon(j^*) \sum_{j'=0}^{\infty} \Big(\prod_{i=0}^{j'} \epsilon(j^*+i) \Big) (1-\epsilon(j^*+j'+1)) \Big) \\ &\quad \times \Big(\sum_{j=0}^{\infty} \sum_{i^*=0}^{j} (k\epsilon(i^*)(1-\epsilon(i^*)) + T\epsilon(i^*)^2) \Big(\prod_{i=0\neq i^*}^{j} \epsilon(i) \Big) \Big) \\ &\quad + \Big(\sum_{j=0}^{\infty} \prod_{i=0}^{j} \epsilon(i) \Big) \bigg[k(1-\epsilon(j^*)) + (k+1) \Big(\epsilon(j^*) \sum_{j'=0}^{\infty} \Big(\prod_{i=0}^{j'} \epsilon(j^*+i) \Big) (1-\epsilon(j^*+j'+1)) \Big) \\ &\quad + \epsilon(j^*) \sum_{j'=0}^{\infty} j' \Big(\prod_{i=0}^{j'} \epsilon(j^*+i) \Big) (1-\epsilon(j^*+j'+1)) \bigg] \bigg\}, \end{aligned} \tag{16}$$

where $\epsilon(0) = 1$, $\epsilon(i)$ *is the channel erasure rate at retransmission attempt* i, *and* j^* *is the required number of forward (re)transmissions for successful decoding.*

Proof. See Appendix F. □

Uncoded ARQ is a special case of HARQ with soft combining where $\epsilon(i) = \epsilon$ for all channel uses $i \in \mathbb{Z}^+$. Following Propositions 3 and 4, respectively, we can derive the following compact results.

Proposition 5. *The throughput* η *for uncoded ARQ for memoryless channels is given by*

$$\eta_{\mathsf{ARQ}} = \frac{1-\epsilon}{1+\epsilon^{d+1}(1-\epsilon)/(1-\epsilon^T)}. \tag{17}$$

Proof. See Appendix B. □

Proposition 6. *The average delay for uncoded ARQ for memoryless channels is given by*

$$\overline{\mathrm{D}}_{\mathsf{ARQ}} = k + \frac{\epsilon}{1-\epsilon}(1+T\epsilon) + k\epsilon. \tag{18}$$

Proof. See Appendix C. □

Proposition 7. *The variance of delay for uncoded ARQ for memoryless channels is*

$$\sigma^2_{\mathrm{D}_{\mathsf{ARQ}}} = k^2 \frac{(\epsilon+\epsilon^2+\epsilon^3)}{1-\epsilon} - \frac{k}{1-\epsilon} \Big(1 - 2\epsilon + 2\epsilon^2 + 2T\epsilon^2 \Big(\frac{1-2\epsilon-\epsilon^2}{1-\epsilon} \Big) \Big) + \mathcal{O}(1). \tag{19}$$

Proof. See Appendix D. □

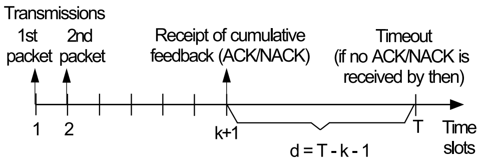
Transmissions
1st packet
2nd packet
Receipt of cumulative feedback (ACK/NACK)
Timeout (if no ACK/NACK is received by then)
1
2
k+1
T
Time slots
d = T - k - 1

Fig. 5: CF ARQ protocol description.

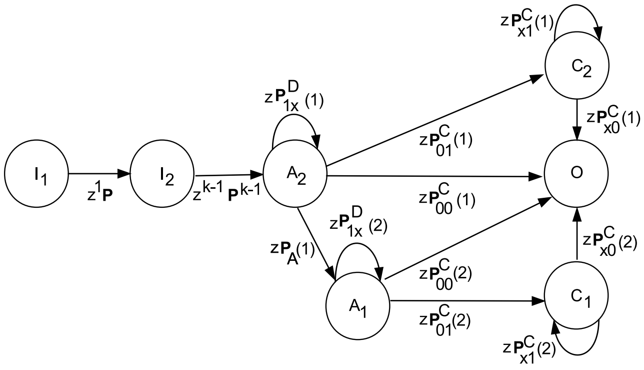

Fig. 6: Matrix-flow graph for delay of CF ARQ.

From (17), throughput $\eta_{\mathsf{ARQ}} = 1/\,\overline{\tau}_{\mathsf{ARQ}}$ is upper bounded by $1-\epsilon$ as $T, d \to \infty$, and there is no such bound for the average delay in (18). From (19), we observe that the standard deviation of delay scales with the RTT k. Hence, RTT causes significant variability in delay.

We next consider a cumulative feedback-based SR ARQ protocol, and analyze its MSFGs.

V. Uncoded ARQ with Cumulative Feedback

We propose a cumulative feedback-based ARQ (CF ARQ) scheme, where the transmitted packets are uncoded. The sender has a coding bucket. When it is ready to send a packet to the receiver, it transmits all the uncoded packets in the bucket to the receiver. The receiver sends a cumulative feedback to indicate the set of successfully received packets in the bucket. If the receiver successfully collects all packets in the coding bucket, and the sender successfully receives the cumulative ACK message, it then purges all the packets in the bucket and moves new packets into the bucket [53].

We consider minimum coding, i.e., with a sliding window of size $M = 2$. While the protocol can be generalized to packet streams with $M > 2$, this is left as future work. In this scheme, the transmitted packet stream is maximum distance separable (MDS) coded, and the feedback acknowledges all correctly received packets, and is cumulative for $M = 2$ coded packets. However, the transmission scheme is repetition-based, i.e. the transmission rate is not adjusted based on the cumulative feedback. The receiver needs both coded packets to reconstruct the transmitted packet stream, i.e., the degrees of freedoms (DoFs) required at the receiver is $N = 2$. We do not assume in-order packet delivery. Thus, the transmitted packets in the coding bucket will be successfully decoded when both of the coded packets are successfully received and ACK'ed by the receiver.

The feedback is cumulative for 2 packets, and it takes $k-1$ time slots between the transmission of the second packet and receipt of the feedback. Hence, the RTT of CF ARQ is $\mathrm{RTT}=k+1$ slots. If the feedback was not cumulative, i.e., the first feedback was received $k-1$ slots after the first packet was transmitted, then the RTT would have been k slots. If the sender does not receive an ACK before the timeout, it retransmits both packets until it receives an ACK. The protocol is shown in Fig. 5.

The combined observation set for CF ARQ with $M=2$ packets is all 3-tuples of $\mathbb{Z}_2=\{0,1\}$, i.e., $\mathcal{X}^{(c)}=\mathbb{Z}_2^3$. For example, $X_t^{(c)}=001$ means that the forward channel is good for both packets and the reverse channel is erroneous, i.e., the ACK for both packets is lost at time t. Since the feedback is cumulative for $M=2$ packets, it is possible that both packets are successfully acknowledged, or they both need to be retransmitted or only one of the packets has to be retransmitted.

The HMM for delay analysis of CF ARQ is shown in Fig. 6. The states I_1 and O are the input and output nodes, respectively, and nodes I_2, A_1, A_2, C_1, C_2 represent the hidden states. States I_1 and I_2 represent transmission of the first new packet and the second packet one time slot later, respectively. The possibilities upon the transmission of $M=2$ coded packets are:

- **Transition to state A_2.** Node A_2 denotes the reception of the first feedback. The coded packets are retransmitted until the forward link is successful and at least one packet is successfully transmitted. The retransmission is modeled by the self-loop at A_2, where the branch gain for delay is

$$\mathsf{BG}_{\mathsf{D}}(A_2\to B_2,\,G_2)=z\mathbf{P}_{1x}^{\mathrm{D}}(1)=z^{\mathrm{RTT}+1}\mathbf{P}^{\mathrm{RTT}}(\mathbf{P}_{10}^{\mathsf{CF}}(1)+z^d\mathbf{P}_{11}^{\mathsf{CF}}(1)\mathbf{P}^d),$$

where $d=T-\mathrm{RTT}$ is the time for timer expiration upon the reception of the first feedback. The branch gain for delay using transition probability matrix $\mathbf{P}_{10}^{\mathsf{CF}}(1)$ for transmitting 2 packets is

$$\mathsf{BG}_{\mathsf{D}}(A_2\to B_2)=z\mathbf{P}_{10}^{\mathsf{CF}}(1)=z(\mathbf{P}_{10}\mathbf{P}_{10}+\mathbf{P}_{10}\mathbf{P}_{01}+\mathbf{P}_{01}\mathbf{P}_{10}),$$

which models the error-free NACK. It combines the different cases such that the feedback is an error-free NACK, i.e., the forward link was bad for both packets and the reverse link was good (first term), or the forward link was bad for either one of the packets only and the reverse link was good (second and third terms). We assume the cumulative feedback is error-free as long as the reverse link is good before the forward transmission is over.

The transition probability matrix $\mathbf{P}_{11}^{\mathsf{CF}}(1)$ is given by

$$\mathbf{P}_{11}^{\mathsf{CF}}(1) = \mathbf{P}_{11}\mathbf{P}_{11} + \mathbf{P}_{11}\mathbf{P}_{10} + \mathbf{P}_{10}\mathbf{P}_{11},$$

which models the erroneous NACK. It combines the cases in which the feedback is an erroneous NACK, i.e., the forward link was bad for both packets and the reverse link was also bad.

In CF ARQ, unless both packets are successfully acknowledged, we always need retransmissions. Hence, it is suboptimal. Furthermore, the erasure rate of CF ARQ is not the same as the erasure rate of uncoded ARQ. For example, for the case of symmetric memoryless channels, the relationship between the erasure rate for CF ARQ with $M = 2$ packets, i.e. $\epsilon_{\rm CF}$, and of the erasure rate of the uncoded ARQ in [39], i.e. ϵ, can be computed as $\epsilon_{\rm CF} = \sqrt{\epsilon^4 + 2\epsilon^3(1-\epsilon)}$. Hence, $\epsilon_{\rm CF} \geq \epsilon^2$.

- **Transition to state A_1.** When the first feedback is received at node A_2, if the number of DoFs acknowledged by the receiver equals 1, then the system transits to state A_1. The matrix

$$\mathsf{BG}_{\mathsf{D}}(A_2 \to A_1) = z\mathbf{P}_{A}^{\mathsf{CF}}(1) = z(\mathbf{P}_{00}\mathbf{P}_{10} + \mathbf{P}_{10}\mathbf{P}_{00} + \mathbf{P}_{11}\mathbf{P}_{01} + \mathbf{P}_{01}\mathbf{P}_{11} + \mathbf{P}_{00}\mathbf{P}_{11} + \mathbf{P}_{11}\mathbf{P}_{00})$$

denotes the branch gain and $\mathbf{P}_{A}^{\mathsf{CF}}(1)$ is the transition probability matrix from A_2 to A_1. Hence, if the system goes into state A_1, the additional number of DoFs required by the receiver is 1, i.e., only one packet needs to be retransmitted, which is modeled by the self-loop at A_1, where

$$\mathsf{BG}_{\mathsf{D}}(A_1 \to B_1,\, G_1) = z\mathbf{P}_{1x}^{\rm D}(2) = (z\mathbf{P})^{\rm RTT-1}(z\mathbf{P}_{10}^{\mathsf{CF}}(2) + z\mathbf{P}_{11}^{\mathsf{CF}}(2)z^{d+1}\mathbf{P}^{d+1}),$$

where the probability matrices $\mathbf{P}_{10}^{\mathsf{CF}}(2)$ and $\mathbf{P}_{11}^{\mathsf{CF}}(2)$ model the error-free and the erroneous NACK, respectively. At node A_1, since only one packet is retransmitted, the transition probability matrices satisfy $\mathbf{P}_{xy}^{\mathsf{CF}}(2) = \mathbf{P}_{xy}$, where $\mathbf{P}_{xy}$'s, $x, y \in \{0, 1\}$ are same as the ones for uncoded ARQ in [39].

- **Transition to state O.** If $N = 2$ DoF's are received, the stream can be successfully decoded. If $N = 2$ DoF's are acknowledged (with probability $\mathbf{P}_{00}^{\mathsf{CF}}(1) = \mathbf{P}_{00}\mathbf{P}_{00}$), the system transits to O.
- **Transition to state C_2.** If $N = 2$ DoF's are received, but the feedback is an erroneous ACK (with probability $\mathbf{P}_{01}^{\mathsf{CF}}(1) = \mathbf{P}_{01}\mathbf{P}_{01} + \mathbf{P}_{01}\mathbf{P}_{00} + \mathbf{P}_{00}\mathbf{P}_{01}$, where the branch gain for delay satisfies $\mathsf{BG}_{\mathsf{D}}(A_2 \to C_2) = z\mathbf{P}_{01}^{\mathsf{CF}}(1)$), then the system transits to C_2, where the sender waits till it receives an error-free ACK/NACK, modeled by the self-loop at C_2.
- **Transition to state C_1.** If $N = 2$ DoF's are received, but only one packet is successfully acknowledged and the feedback for the other packet is an erroneous ACK (with probability

$\mathbf{P}_{01}^{\mathsf{C}}(2)$, where $\mathsf{BG}_{\mathsf{D}}(A_1 \to C_1) = z\mathbf{P}_{01}^{\mathsf{CF}}(2)$), then the system transits to C_1, where the sender waits till it receives an error-free ACK/NACK. This is modeled by the self-loop at C_1.

The success and error probability matrices on the reverse channel for CF ARQ are given as

$$\mathbf{P}_{x0}^{\mathsf{CF}}(n) = \mathbf{P}_{00}^{\mathsf{CF}}(n) + \mathbf{P}_{10}^{\mathsf{CF}}(n), \quad \mathbf{P}_{x1}^{\mathsf{CF}}(n) = \mathbf{P}_{01}^{\mathsf{CF}}(n) + \mathbf{P}_{11}^{\mathsf{CF}}(n),\ n \in \{1, 2\},$$

respectively, given the transition probabilities, where $n-1$ is the number of DoFs acknowledged by the receiver, i.e., $2-(n-1)$ DoFs are needed at the receiver.

For the proposed scenario, both $\tau_{\mathsf{CF\text{-}ARQ}}$ and $\overline{\mathrm{D}}_{\mathsf{CF\text{-}ARQ}}$ are random variables with positive integer outcomes. The matrix gain of the graph in Fig. 6 is calculated using the basic simplification rules.

For the derivation of the MGFs of the transmission and delay times of CF ARQ, see Appendix G and Appendix H. Using the PGFs, the throughput $\eta_{\mathsf{CF\text{-}ARQ}}$, which is the reciprocal of the average value of $\tau_{\mathsf{CF\text{-}ARQ}}$, and the average value of $D_{\mathsf{CF-ARQ}}$, i.e., $\overline{\mathrm{D}}_{\mathsf{CF\text{-}ARQ}}$, can be calculated. We now present the closed form expressions for throughput and average delay of the memoryless channels.

Proposition 8. *The throughput for CF ARQ for memoryless channels is given by*

$$\eta_{\mathsf{CF\text{-}ARQ}} \approx \frac{2(1-\epsilon)}{1+\alpha_{\mathsf{CF}}(\epsilon)^{-1}\epsilon^d(1-\epsilon)/(1-\epsilon^T)}, \tag{20}$$

where $d = T-k$, and $\alpha_{\mathsf{CF}}(\epsilon)$ is given by

$$\alpha_{\mathsf{CF}}(\epsilon) = \frac{1+3\epsilon-2\epsilon^2+20\epsilon^3-18\epsilon^4+28\epsilon^5-60\epsilon^6+72\epsilon^7-40\epsilon^8+8\epsilon^9}{2(2-\epsilon)(1-\epsilon+4\epsilon^2-2\epsilon^3)(1+\epsilon-2\epsilon^2+2\epsilon^3)},$$

where it can be easily verified that $\eta_{\mathsf{CF\text{-}ARQ}} \geq \eta_{\mathsf{ARQ}}$.

Proof. See Appendix G. □

Proposition 9. *The average delay of CF ARQ for memoryless channels is given by*

$$\overline{\mathrm{D}}_{\mathsf{CF\text{-}ARQ}} = k+1+(2k+8)\epsilon-(3k+11)\epsilon^2+(6T+10k+26)\epsilon^3+\mathcal{O}(\epsilon^4), \quad \epsilon \to 0. \tag{21}$$

Proof. See Appendix H. □

Comparing this with the average delay of uncoded ARQ, we observe that $\overline{\mathrm{D}}_{\mathsf{CF\text{-}ARQ}} - \overline{\mathrm{D}}_{\mathsf{ARQ}} = 1 + (k+7)\epsilon - (3k+T+12)\epsilon^2 + \mathcal{O}(\epsilon^3)$ as $\epsilon \to 0$, which is due to the cumulative feedback. On the other hand, when ϵ is large, $\overline{\mathrm{D}}_{\mathsf{CF\text{-}ARQ}}$ becomes less than $\overline{\mathrm{D}}_{\mathsf{ARQ}}$, as we demonstrate in Sect. VII.

Note that the MGFs for CF ARQ given in Appendix G and Appendix H with $M = N = 1$ is equivalent to the MGFs of uncoded ARQ given in [39].

VI. Coded ARQ

In this section, we propose a Coded ARQ scheme, where the transmitted packets are coded. The coding scheme is similar to the generation-based random linear network coding in [54]. The sender has a coding bucket, and when it is ready to send a packet to the receiver, it produces a coded packet by forming a random linear combination of all the packets in the bucket. The encoded packet is then transmitted to the receiver. The receiver sends a cumulative feedback to indicate the set of successfully received encoded packets in the coding bucket. If the receiver successfully collects a sufficient number of encoded packets to decode all packets in the coding bucket, and the sender successfully receives the cumulative ACK message, it then purges the successfully ACK'ed encoded packets in the coding bucket and partially updates the coding bucket by moving new packets.

We consider minimum coding, i.e., with a sliding window of size $M = 2$. Different from uncoded ARQ, HARQ with soft combining, and CF ARQ, the transmission scheme is adaptive, i.e. the transmission rate is adjusted based on the cumulative feedback for $M = 2$ MDS coded packets in the transmitted packet stream. The receiver needs both coded packets to reconstruct the transmitted packet stream, i.e., the DoFs required at the receiver is $N = 2$. We do not assume in-order packet delivery. Therefore, the transmitted packets in the bucket will be successfully decoded when both of the coded transmitted packets are successfully received and ACK'ed by the receiver. While the model can be extended to $M > 2$ using a recursion, the state space scales exponentially, and the analysis becomes prohibitively complicated without any additional insights. Therefore, it is left as future work.

The combined observation set for Coded ARQ with $M = 2$ packets is $\mathcal{X}^{(c)} = \mathbb{Z}_2^3$. For example, $X_t^{(c)} = 001$ means that the forward channel is good for both packets and the reverse channel is erroneous, i.e., the ACK for $M = 2$ packets is lost. The HMM for the delay of Coded ARQ is shown in Fig. 7. Similar to previous models, I_1 and O are the input and output nodes, and other nodes are the hidden states, and I_1 and I_2 represent transmission of the first new packet and the second packet one time slot later, respectively. The possibilities upon the transmission of the 2 packets are:

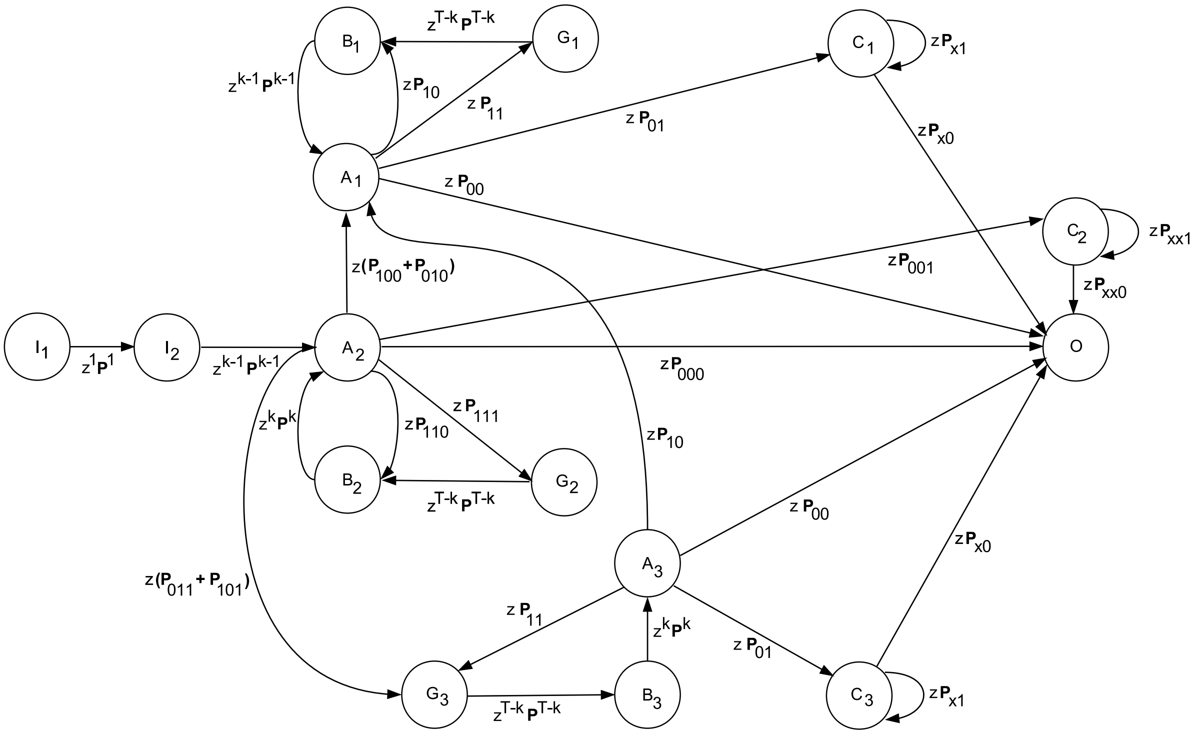

Fig. 7: Matrix-flow graph for delay analysis of SR ARQ in unreliable feedback with coding.

- **Transition to state** A_2**.** After sending the new packets ($M = 2$), the transmitter receives a feedback message $k-1$ time slots later. This state is represented by node A_2.
- **Transition to state** O**.** If the feedback is an error-free ACK (with probability $\mathbf{P}_{000} = \mathbf{P}_{0x}\mathbf{P}_{00}$), then the system transits to state O.
- **Transition to state** B_2**.** If the feedback is a successful NACK for both packets (with probability $\mathbf{P}_{110} = \mathbf{P}_{1x}\mathbf{P}_{10}$), then the system transits to state B_2, where both packets have to be retransmitted.
- **Transition to state** C_2**.** If the feedback is an erroneous ACK (with probability $\mathbf{P}_{001} = \mathbf{P}_{0x}\mathbf{P}_{01}$) and the timer expires before receiving any error-free ACKs/NACKs, the system will transit to state C_2, the packets will be retransmitted, and the timeout will be reset. The packets will then be acknowledged when a succeeding ACK/NACK is correctly received.
- **Transition to state** A_1**.** If only one of the packets is successfully transmitted and the feedback is an error-free ACK (with probability $\mathbf{P}_{100} + \mathbf{P}_{010} = \mathbf{P}_{1x}\mathbf{P}_{00} + \mathbf{P}_{0x}\mathbf{P}_{10}$), the system goes to state A_1. This state is equivalent to the state A for the uncoded ARQ model as shown in Fig. 3-(b). Hence, the rest of the analysis follows from the uncoded ARQ analysis in [39].

- **Transition to state G_2.** Node G_2 indicates that a NACK is lost (with probability $\mathbf{P}_{111} = \mathbf{P}_{1x}\mathbf{P}_{11}$), both packets are lost, and the transmitter waits for timeout (node B_2). See also Fig. 3.
- **Transition to state G_3.** Node G_3 indicates that a NACK is lost, but only one of the packets is successfully transmitted and the other one is lost, (with probability $\mathbf{P}_{011} + \mathbf{P}_{101} = \mathbf{P}_{0x}\mathbf{P}_{11} + \mathbf{P}_{1x}\mathbf{P}_{01}$), and the transmitter waits for timeout (node B_3). Node A_3 denotes the retransmission of both packets, and the receiver only needs one of the packets. Therefore, once the system goes to state A_3, the rest of the analysis follows from the uncoded ARQ analysis in [39].

In this paper, since we use tiny codes, i.e. sliding window by coding with just 2 packets, the available redundancy rate in terms of the packets in the encoding window is 50%. However, we do a finer-grained control over the redundancy rate via the feedback which is cumulative. This can be observed from Fig. 7. For example, if the CF acknowledges the successful reception of 1 packet only, i.e., the system transits to state A_1, then the rate is adaptively adjusted to retransmit 1 packet only. On the other hand, if only 1 packet is successfully transmitted and the CF is lost, then the system has a transition to G_3, and then to A_3 that represents the retransmission of both packets while the receiver only needs one of the packets. In this case, upon the successful reception of the CF in the succeeding time slots, the system either transits to state A_1, i.e. 1 packet has to be retransmitted again, or to state 0, i.e. no retransmission is required. Therefore, the redundancy rate of the model is not always 50%, and a finer-grained control is provided through the feedback.

The matrix gain of the graph in Fig. 7 can be calculated using the basic simplification rules. For the derivation of the MGFs of the transmission and delay times, hence characterization of the throughput and delay performance, of Coded ARQ, see Appendices I and J, along with the relations (2) and (3). We now present closed form expressions for throughput and delay of memoryless channels.

Proposition 10. *The throughput for Coded ARQ for memoryless channels is given by*

$$\eta_{\textsf{C-ARQ}} = \frac{(1-\epsilon)}{\alpha_C(\epsilon) + \epsilon^{d+1}(1-\epsilon)\beta_C(\epsilon)/(1-\epsilon^T)}, \tag{22}$$

where $\alpha_C(\epsilon) = (1+\epsilon+7\epsilon^2/2-\epsilon^3/2-3\epsilon^4+\epsilon^6)/(1+\epsilon)^2$, $\beta_C(\epsilon) = 1/2+\epsilon^2(1-\epsilon)$, *and* $d = T-k$.

Proof. See Appendix I. □

Note that in (22), it is easy to show that $3/4 \leq \alpha_C(\epsilon) \leq 1$, and $1/2 \leq \beta_C(\epsilon) < 13/20$. Therefore, we can conclude that $\eta_{\textsf{C-ARQ}}$ is always higher than $\eta_{\textsf{ARQ}}$ in (17) for any given T, d. Furthermore, $\eta_{\textsf{C-ARQ}}$ is upper bounded by $(1-\epsilon)/\alpha_C(\epsilon)$ as $T, d \to \infty$. This implies that for memoryless systems, with minimum coding, it is possible to achieve a gain of more than 30% compared with uncoded ARQ. The gain becomes higher if the channel has memory, as demonstrated in Sect. VII.

Proposition 11. *The average delay of the Coded ARQ for memoryless channels is given by*

$$\overline{\mathrm{D}}_{\textsf{C-ARQ}} = k + 1 + 3\epsilon + (2T + 5k + 4)\epsilon^2 + (T - 7k + 5)\epsilon^3 + \mathcal{O}(\epsilon^4), \quad \epsilon \to 0. \tag{23}$$

Proof. See Appendix J. □

The variance of delay for Coded ARQ is derived next.

Proposition 12. *The variance of delay for Coded ARQ for memoryless channels is*

$$\sigma^2_{\mathrm{D}_{\textsf{C-ARQ}}} = k^2\epsilon^2(1 + \epsilon - 16\epsilon^2) + k\epsilon^2(5 - 31\epsilon + 43\epsilon^2 - T\epsilon^2(4 - 6\epsilon + 2\epsilon^2)) + \mathcal{O}(1), \quad \epsilon \to 0. \tag{24}$$

Proof. See Appendix K. □

Comparing (23) with the average delay of uncoded ARQ, we observe that $\overline{\mathrm{D}}_{\textsf{C-ARQ}} - \overline{\mathrm{D}}_{\textsf{ARQ}} = 1 + (3-k-1)\epsilon + (T+5k+3)\epsilon^2 + \mathcal{O}(\epsilon^3)$ as $\epsilon \to 0$, which is due to the cumulative feedback. On the other hand, when ϵ is large, $\overline{\mathrm{D}}_{\textsf{C-ARQ}}$ becomes comparable to $\overline{\mathrm{D}}_{\textsf{ARQ}}$, as we demonstrate in Sect. VII. However, Coded ARQ always provides better delay guarantees than uncoded ARQ. This provides insights in designing systems that are robust to the RTT fluctuations.

We next numerically evaluate the performance of the different ARQ protocols and outline the advantages of cumulative feedback and Coded ARQ over uncoded ARQ protocols.

VII. Numerical Results

We evaluate the performance of the SR ARQ schemes outlined in Sects. III-D-VI by computing the MGFs of transmission and delay times via the MSFG approach detailed in Sect. III, and provide a comparison of ARQ, HARQ, CF ARQ, and Coded ARQ schemes with feedback erasures. We also include the simulation results[8] to validate our analytical models. The parameters for the numerical

[8] The source code for simulation and analysis is available at github.com/deryam/TinyCodesforDelayGuarantees.

results are selected as follows. The RTT[9] is $k = \{5, 10\}$ time slots, timeout is $T = \{8, 15\}$ slots when $k = 5$, and $T = \{16, 30\}$ slots when $k = 10$, and we have the same parameters $\epsilon_B = 1$, $\epsilon_G = 0$, and r for both the forward and reverse links, hence the same erasure rate, such that the proportion of the time spent in G and B can be computed using the stationary probabilities, given the erasure rate ϵ. The performance metrics are the throughput η, the average delay $\bar{D}$, i.e. the per packet delay for uncoded ARQ and HARQ, and the delay corresponding to the transmission of $M = 2$ packets in CF ARQ and Coded ARQ, and the guaranteeable delay $\hat{D}$ versus ϵ for varying RTT k, timeout T and r. Unless otherwise specified, solid (Coded ARQ), dash-dot (CF ARQ), dashed (HARQ), dotted (ARQ) curves denote the analytical results, and unfilled circles denote the simulation results of this paper.

We next investigate the reliability of the protocols via numerically investigating the tail distribution of the delay. We illustrate the delay tail behavior in terms of the complementary cumulative distribution function (CCDF) in Fig. 8 on a logarithmic scale, for the ARQ and HARQ protocols, and the CF ARQ and Coded ARQ protocols with $M = 2$ packets. We demonstrate what these distributions look like both for memoryless and Gilbert-Elliott channels for $\epsilon = 0.5$, $T = 15$, $k = 5$. From these semilogarithmic plots, it is clear that the tail decays exponentially. Thus, even though we did not prove analytically, the distribution of the delay is shown to be sub-Gaussian since $\mathbb{P}(D > d) \leq e^{-vd^2}$ for $v = 3 \times 10^{-4}$ and every $d > 0$ as shown in marked curves in Fig. 8. Hence, the guaranteeable delay $\hat{D}$ of a protocol is upper bounded by the guaranteeable delay of a Gaussian distribution with the same first and second order parameters as D. Given that URLLC has different delay and reliability requirements, ranging from 10^{-5} to 10^{-9}, we now discuss about guaranteed delays under different reliability requirements. Exploiting the sub-Gaussian behavior of the delay, a reliability requirement as high as $1 - 10^{-9}$ is guaranteed when we have that $\mathbb{P}(D \leq \hat{D}) \geq 1 - e^{-v\hat{D}^2}$ for some $v > 0$. Equivalently, the guaranteeable delay satisfies $\hat{D} \geq \sqrt{\frac{9 \log(10)}{v}}$. This result can be improved significantly when ϵ is smaller.

The throughput and delay of the different ARQ protocols in the Markov channel for $r = 0.3$ are shown in Figs. 9 and 10-11, respectively, for $k = \{5, 10\}$, for different values of T. The baseline model is the uncoded ARQ scheme of [39]. For the HARQ scheme with soft combining at the receiver, $\epsilon_B(m) = 1 - e^{-\alpha/m}$ on a retransmission attempt m, where we assume $\alpha = 10\epsilon$, which is high, hence

[9] The slot duration should be adjusted according to the transmission protocol. For example, if the transmission rate is 10 Mbits/sec, it takes 1 ms to transmit 10,000 bits over the channel. In that case, the RTT of $k = 5$ time slots will be equivalent to 1 ms.

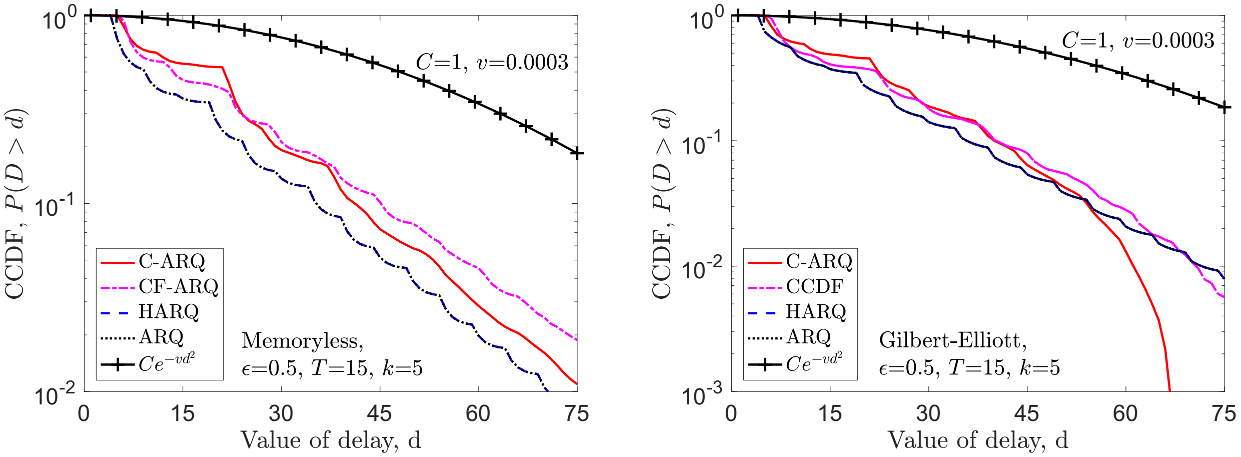

Fig. 8: Delay CCDF for different ARQ schemes for $\epsilon = 0.5$, $T = 15$, $k = 5$: (Left) Memoryless channel. (Right) Gilbert-Elliott channel, for $r = 0.3$.

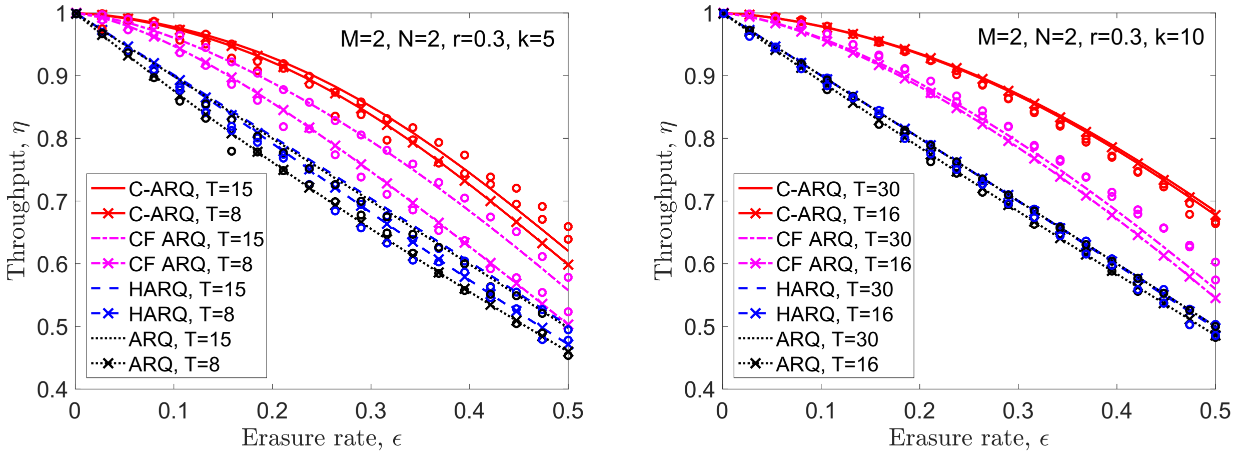

Fig. 9: Throughput η versus erasure rate ϵ, in Markov errors for $r = 0.3$, and $k = 5$ and $k = 10$.

the erasure rate of state B. The HARQ scheme slightly improves the delay compared to the uncoded scheme, however its throughput is similar. In the Coded ARQ, more packets can be reliably transmitted even when the packet loss rate ϵ is large. As ϵ increases, throughput of Coded ARQ scheme decays more slowly than the other schemes because coding can compensate the packet losses. Hence, fewer retransmissions are required. Furthermore, delay is significantly lower than the uncoded ARQ schemes. In all ARQ models, when the timeout T increases, both throughput and delay are higher.

The simulation and analytical results agree for throughput and delay, as in [39], except when the erasure or burst rates are high, or when RTT is comparable to timeout. Therefore, we did not include simulations for $r = 0.1$. The throughput and delay of the protocols in the Markov channel for $r = 0.1$

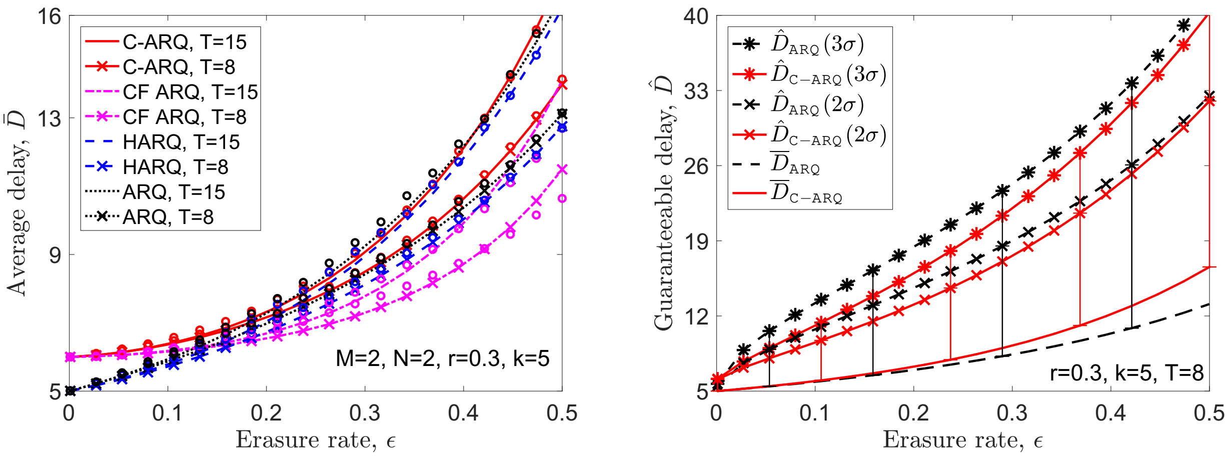

Fig. 10: (Left) Average delay $\bar{D}$ versus erasure rate ϵ for various ARQ schemes. (Right) Guaranteeable delay $\hat{D}$ versus erasure rate ϵ for uncoded and Coded ARQ schemes for $T = 8$, in Markov errors for $r = 0.3$, and $k = 5$.

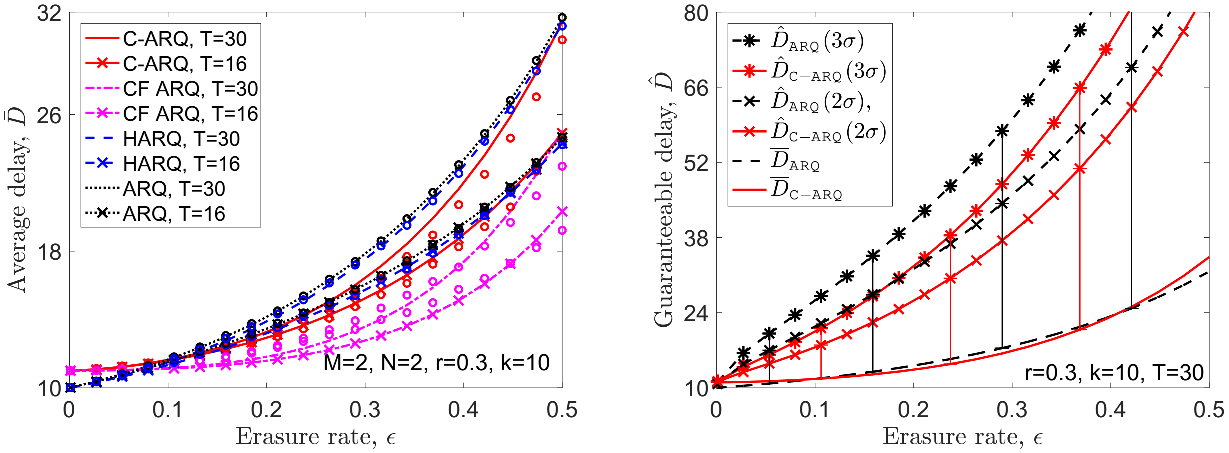

Fig. 11: (Left) Average delay $\bar{D}$ versus erasure rate ϵ for various ARQ schemes. (Right) Guaranteeable delay $\hat{D}$ versus erasure rate ϵ for uncoded and Coded ARQ schemes for $T = 30$, in Markov errors for $r = 0.3$, and $k = 10$.

are shown in Figs. 12 and 13-14, respectively, for $k = \{5, 10\}$ for different T. Comparing these results with the ones for $r = 0.3$, the delay is higher and the throughput is lower for uncoded ARQ.

In CF ARQ and Coded ARQ with $M = 2$ packets, since $\mathrm{RTT} = k + 1$, the delay gap between the coded and uncoded schemes at $\epsilon = 0$ is 1 slot. Although the gap is indeed very small for small coding bucket sizes M, it also means that when the erasure rate is low, CF ARQ and Coded ARQ have higher average delays compared to the uncoded ARQ models. For CF ARQ, the throughput is higher than the throughput of the uncoded ARQ and HARQ with Chase combining since the feedback is cumulative, which decreases the number of packets being transmitted per a successful packet. Still, since CF ARQ is redundant in terms of transmissions, we do not have significant throughput gains compared to the

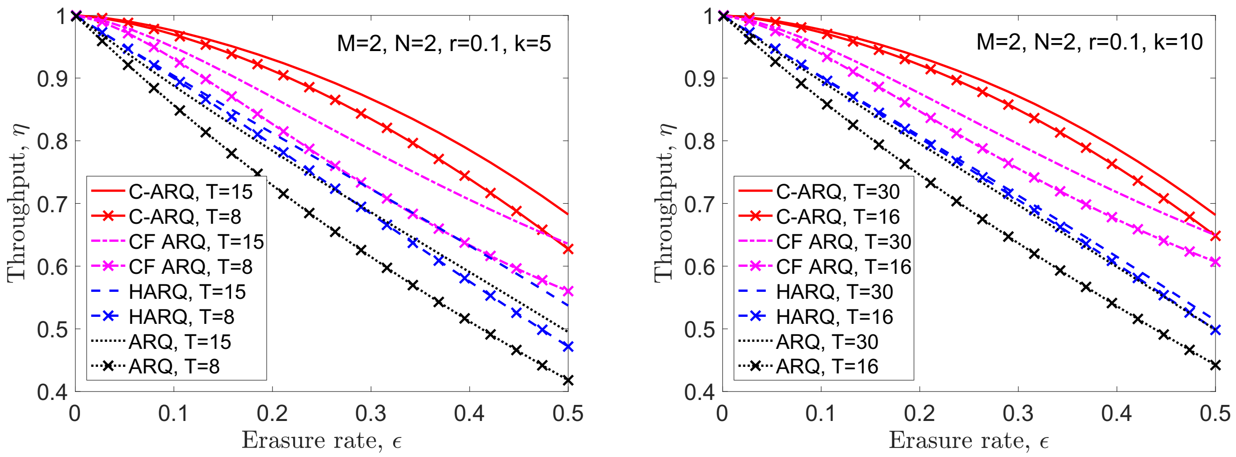

Fig. 12: Throughput η versus erasure rate ϵ, in Markov errors for $r = 0.1$, and $k = 5$ and $k = 10$.

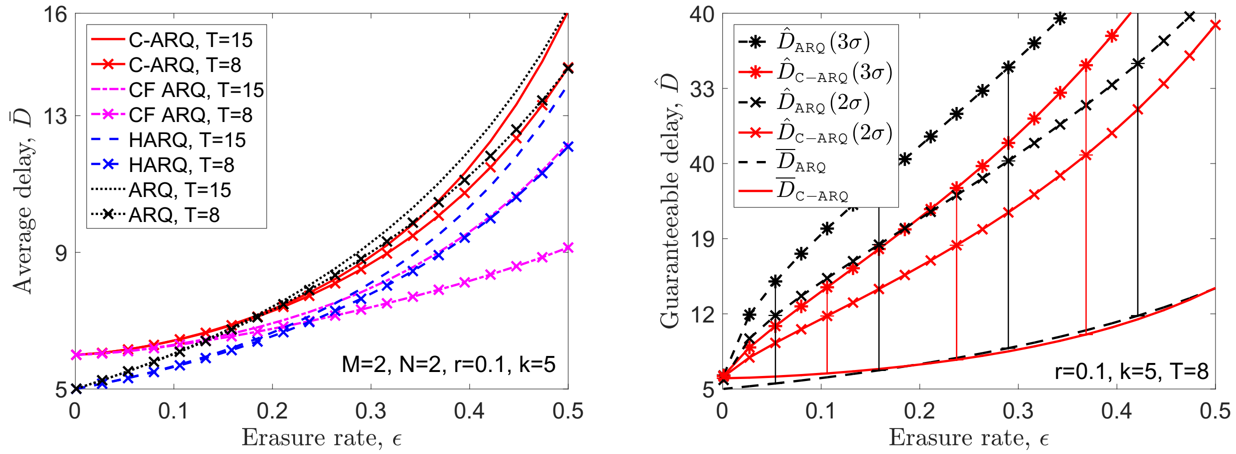

Fig. 13: (Left) Average delay $\bar{D}$ versus erasure rate ϵ for various ARQ schemes. (Right) Guaranteeable delay $\hat{D}$ versus erasure rate ϵ for uncoded and Coded ARQ schemes for $T = 8$, in Markov errors for $r = 0.1$, and $k = 5$.

uncoded ARQ. However, its average delay is lower than the average delays of the uncoded ARQ, HARQ and Coded ARQ under moderate or high erasures. Since this scheme has higher redundancy, it can achieve a better delay performance even under bursty errors. For Coded ARQ, the throughput is always higher than the throughput of the CF ARQ because the transmission rate is adapted based on the feedback received. However, Coded ARQ provides a higher average delay than CF ARQ.

From Figures 9 to 14, we can observe that the higher the error burst ($r = 0.1$), the lower the uncoded SR ARQ throughput in noisy feedback [39]. For uncoded ARQ and HARQ, the sensitivity of throughput to timeout T increases as r decreases, hence the throughput becomes very low when the timeout T is very small. When $r = 0.1$, for Coded ARQ, throughput becomes more sensitive to

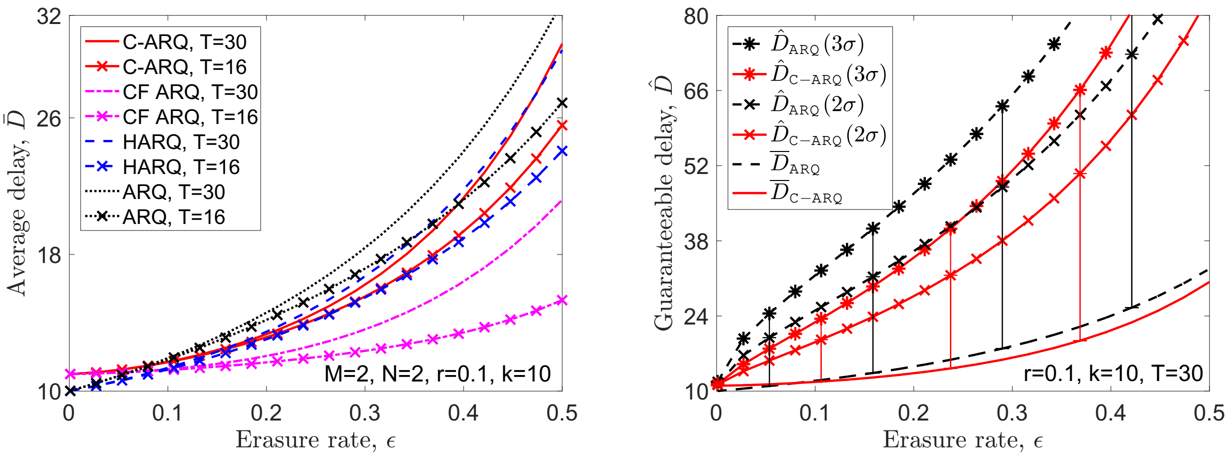

Fig. 14: (Left) Average delay $\bar{D}$ versus erasure rate ϵ for various ARQ schemes. (Right) Guaranteeable delay $\hat{D}$ versus erasure rate ϵ for uncoded and Coded ARQ schemes for $T = 30$, in Markov errors for $r = 0.1$, and $k = 10$.

T, and it is possible to achieve significantly higher throughputs by increasing T.

As the burst rate increases, the average delay is higher for uncoded ARQ, HARQ and Coded ARQ. However, for both uncoded ARQ and HARQ models and Coded ARQ, the sensitivity of delay to timeout T decreases as r decreases, hence the variability of delay with timeout T becomes less important under burst errors. For CF ARQ, however, when $r = 0.1$, the sensitivity of the average delay to T increases, and delay can be made significantly lower by keeping T small.

We next investigate the guaranteeable delay with respect to erasure rate, ϵ, for different values of r, k, T only for the uncoded ARQ and Coded ARQ. The guaranteeable delay $\hat{D}$ is illustrated in Figs. 10 and 11 for $r = 0.3$, and in Figs. 13 and 14 for $r = 0.1$, respectively, for different sets of k and T.

We observe that average and guaranteeable delay for both uncoded and Coded ARQ increases in k and T. Although the Coded ARQ scheme might have a higher average delay $\bar{D}$ than the uncoded ARQ scheme, the guaranteeable delay $\hat{D}$ for Coded ARQ is lower than the one for uncoded ARQ. By increasing the timeout, the gap between the guaranteeable delays of both models can be made smaller.

Comparing Figs. 10 and 11 with Figs. 13 and 14, we observe that under burst erasures ($r = 0.1$), delay guarantees become more prevalent. Coded ARQ is also more reliable under high erasure rates. Therefore, we have the benefit of coding when we have burst or high rate of erasures. Coded ARQ is more guaranteeable across statistics, and hence is more stable.

Our findings on various ARQ schemes suggest that the following design insights should enable more robust design for two-way erasure channels for wireless networks:

- Sensitivity of throughput and delay to timeout T and RTT k increases under burst errors.
- Uncoded ARQ is very sensitive to error bursts. Therefore, the higher the burst rate, the lower its throughput and the higher its delay is.
- CF ARQ provides significantly less average delay (when the erasure rate is high) than uncoded ARQ, HARQ and Coded ARQ, and a higher throughput than uncoded ARQ and HARQ, but its throughput performance is not as good as Coded ARQ.
- Coded ARQ can provide throughput gains up to about 40% more than the baseline uncoded ARQ.
- Coded ARQ has higher average delay but lower variability. Guaranteeable delay for Coded ARQ is lower than the guaranteeable delay for uncoded ARQ.
- Coding has benefits under burst errors or higher erasure rates. Coded ARQ is more predictable across statistics, hence is more stable. This can help design robust systems when RTT is unreliable.

While the analysis is conducted only for tiny codes, this is the regime where substantial throughput gains can be achieved via coding. While the proposed technique becomes prohibitively complex to analyze for general coding window sizes due to the excessive number of hidden states, an exact analysis is possible for small window sizes using recursive formulas. Furthermore, using a network simulator, the technique can easily be extended to general window sizes. This can help understand the tradeoffs between the coding window size and system parameters along with the channel variations.

VIII. Conclusions

We leveraged signal-flow techniques and coding theory to provide delay guarantees in SR ARQ. For tiny codes, we analyzed the distributions of transmission and delay times of HARQ with soft combining, cumulative-feedback based ARQ and Coded ARQ. Contrasting the performance of HARQ with soft combining and CF ARQ with the uncoded ARQ scheme, we demonstrated their gains in terms of throughput and delay. For the given parameter setting, the CF ARQ scheme can provide a significant reduction in average delay, and a better throughput compared to the uncoded cases. Coded ARQ can provide gains up to about 40% in throughput, and lower guaranteeable delay than the one for uncoded ARQ. This strategy also requires less feedback than that required by uncoded ARQ.

The insights can be applied to the design of mission-critical communications and industrial control for critical control messaging, which will be important use cases of 5G with ultra-reliability and ultra-low latency. Extensions include the optimization of the erasure coded schemes with minimal encoding and decoding complexity for asymmetric and bursty channels, and their code rate, and the development of more sophisticated coding schemes, such as sequential MDS or convolutional codes, Reed-Solomon codes for better FEC. Possible future directions also include the extension of the tiny coding scheme to long codes. Extending Coded ARQ to larger window sizes, we can investigate the scaling between the bucket size, the RTT and the DoFs required at the receiver, which will pave the way for protocol design with desirable throughput-delay tradeoffs.

Appendix

A. Transition Probability Matrices for the Gilbert-Elliott Channel and the Combined Channel

The state-transition matrix both for the forward and $\mathbf{P}$ for the reverse channels is denoted by $\mathbf{P}$. Since $\mathbf{P}$ is a stochastic matrix, $\mathbf{P}^n\mathbf{1} = \mathbf{1}$ for $n \geq 1$. The stationary vector of the state-transition matrix π satisfies $\pi\mathbf{1} = 1$ and $\pi\mathbf{P} = \pi$. The combined state-transition matrix for the symmetric GE channels equals the Kronecker product of $\mathbf{P}$ with itself, i.e., $\mathbf{P}^{(c)} = \mathbf{P} \otimes \mathbf{P}$, which is given by

$$\mathbf{P}^{(c)} = \begin{bmatrix} (1-q)^2 & q(1-q) & q(1-q) & q^2 \\ r(1-q) & (1-q)(1-r) & qr & q(1-r) \\ r(1-q) & qr & (1-q)(1-r) & q(1-r) \\ r^2 & r(1-r) & r(1-r) & (1-r)^2 \end{bmatrix}. \tag{25}$$

Let $\mathbf{P}_0$ and $\mathbf{P}_1$, respectively, be the success and the error probability matrices of an HMM. Both for the forward and reverse links, we have that $\mathbf{P}_0^{(f)} = \mathbf{P}_0^{(r)} = \mathbf{P}_0$ and $\mathbf{P}_1^{(f)} = \mathbf{P}_1^{(r)} = \mathbf{P}_1$, where

$$\mathbf{P}_0 = \mathbf{P} \cdot \mathrm{diag}\{\mathbf{1} - \boldsymbol{\epsilon}\} = \begin{bmatrix} 1-q & q \\ r & 1-r \end{bmatrix} \begin{bmatrix} 1-\epsilon_G & 0 \\ 0 & 1-\epsilon_B \end{bmatrix} = \begin{bmatrix} (1-q)(1-\epsilon_G) & q(1-\epsilon_B) \\ r(1-\epsilon_G) & (1-r)(1-\epsilon_B) \end{bmatrix},$$

$$\mathbf{P}_1 = \mathbf{P} \cdot \mathrm{diag}\{\boldsymbol{\epsilon}\} = \begin{bmatrix} 1-q & q \\ r & 1-r \end{bmatrix} \begin{bmatrix} \epsilon_G & 0 \\ 0 & \epsilon_B \end{bmatrix} = \begin{bmatrix} (1-q)\epsilon_G & q\epsilon_B \\ r\epsilon_G & (1-r)\epsilon_B \end{bmatrix}.$$

The probability vector of transmitting a new packet is $\pi_I = \pi\mathbf{P}_0$. Given the erasure rates $\boldsymbol{\epsilon} = [\epsilon_G, \epsilon_B]$, and $\epsilon = \pi\boldsymbol{\epsilon}^\intercal$, we have $\pi_I\mathbf{1} = \pi\mathbf{P}_0\mathbf{1} = 1 - \pi\mathbf{P}\boldsymbol{\epsilon}^\intercal = 1 - \epsilon$, and $(\pi - \pi_I)\mathbf{1} = \pi\mathbf{P}_1\mathbf{1} = \pi\mathbf{P}\boldsymbol{\epsilon}^\intercal = \epsilon$. The combined observation probabilities are given by the following 4×4 matrices:

$$\mathbf{P}_{00}^{(c)} = \begin{bmatrix} (1-\epsilon_G)^2\bar{q}^2 & (1-\epsilon_B)(1-\epsilon_G)q\bar{q} & (1-\epsilon_B)(1-\epsilon_G)q\bar{q} & (1-\epsilon_B)^2q^2 \\ (1-\epsilon_G)^2r\bar{q} & (1-\epsilon_B)(1-\epsilon_G)\bar{q}\bar{r} & (1-\epsilon_B)(1-\epsilon_G)qr & (1-\epsilon_B)^2q\bar{r} \\ (1-\epsilon_G)^2r\bar{q} & (1-\epsilon_B)(1-\epsilon_G)qr & (1-\epsilon_B)(1-\epsilon_G)\bar{q}\bar{r} & (1-\epsilon_B)^2q\bar{r} \\ (1-\epsilon_G)^2r^2 & (1-\epsilon_B)(1-\epsilon_G)r\bar{r} & (1-\epsilon_B)(1-\epsilon_G)r\bar{r} & (1-\epsilon_B)^2\bar{r}^2 \end{bmatrix},$$

$$\mathbf{P}_{01}^{(c)} = \begin{bmatrix} \epsilon_G(1-\epsilon_G)\bar{q}^2 & \epsilon_B(1-\epsilon_G)q\bar{q} & \epsilon_G(1-\epsilon_B)q\bar{q} & \epsilon_B(1-\epsilon_B)q^2 \\ \epsilon_G(1-\epsilon_G)r\bar{q} & \epsilon_B(1-\epsilon_G)\bar{q}\bar{r} & \epsilon_G(1-\epsilon_B)qr & \epsilon_B(1-\epsilon_B)q\bar{r} \\ \epsilon_G(1-\epsilon_G)r\bar{q} & \epsilon_B(1-\epsilon_G)qr & \epsilon_G(1-\epsilon_B)\bar{q}\bar{r} & \epsilon_B(1-\epsilon_B)q\bar{r} \\ \epsilon_G(1-\epsilon_G)r^2 & \epsilon_B(1-\epsilon_G)r\bar{r} & \epsilon_G(1-\epsilon_B)r\bar{r} & \epsilon_B(1-\epsilon_B)\bar{r}^2 \end{bmatrix},$$

$$\mathbf{P}_{10}^{(c)} = \begin{bmatrix} \epsilon_G(1-\epsilon_G)\bar{q}^2 & \epsilon_G(1-\epsilon_B)q\bar{q} & \epsilon_B(1-\epsilon_G)q\bar{q} & \epsilon_B(1-\epsilon_B)q^2 \\ \epsilon_G(1-\epsilon_G)r\bar{q} & \epsilon_G(1-\epsilon_B)\bar{q}\bar{r} & \epsilon_B(1-\epsilon_G)qr & \epsilon_B(1-\epsilon_B)q\bar{r} \\ \epsilon_G(1-\epsilon_G)r\bar{q} & \epsilon_G(1-\epsilon_B)qr & \epsilon_B(1-\epsilon_G)\bar{q}\bar{r} & \epsilon_B(1-\epsilon_B)q\bar{r} \\ \epsilon_G(1-\epsilon_G)r^2 & \epsilon_G(1-\epsilon_B)r\bar{r} & \epsilon_B(1-\epsilon_G)r\bar{r} & \epsilon_B(1-\epsilon_B)\bar{r}^2 \end{bmatrix},$$

$$\mathbf{P}_{11}^{(c)} = \begin{bmatrix} \epsilon_G^2\bar{q}^2 & \epsilon_B\epsilon_G q\bar{q} & \epsilon_B\epsilon_G q\bar{q} & \epsilon_B^2q^2 \\ \epsilon_G^2r\bar{q} & \epsilon_B\epsilon_G\bar{q}\bar{r} & \epsilon_B\epsilon_G qr & \epsilon_B^2q\bar{r} \\ \epsilon_G^2r\bar{q} & \epsilon_B\epsilon_G qr & \epsilon_B\epsilon_G\bar{q}\bar{r} & \epsilon_B^2q\bar{r} \\ \epsilon_G^2r^2 & \epsilon_B\epsilon_G r\bar{r} & \epsilon_B\epsilon_G r\bar{r} & \epsilon_B^2\bar{r}^2 \end{bmatrix},$$

where we used the shorthand notation $\bar{q} = 1 - q$ and $\bar{r} = 1 - r$.

Transition Probability Matrices for the Symmetric Memoryless Channel. Since the memoryless channel has only one state, $\mathbf{P} = 1$ and its combined state-transition matrix is $\mathbf{P}^{(c)} = \mathbf{P} \otimes \mathbf{P} = 1$. Hence, for memoryless channels with a symmetric erasure rate ϵ, we have $\mathbf{P}_0 = (1 - \epsilon)$, and $\mathbf{P}_1 = \epsilon$. Thus, the observation probabilities are $\mathbf{P}_{00} = (1 - \epsilon)^2$, $\mathbf{P}_{01} = (1 - \epsilon)\epsilon$, $\mathbf{P}_{10} = \epsilon(1 - \epsilon)$, and $\mathbf{P}_{11} = \epsilon^2$.

B. Proof of Proposition 5

From (5)-(7), (11) and (12), the MGF of the transmission time for uncoded ARQ is given by [39]

$$\begin{aligned}\mathbf{\Phi}_{\tau_{\mathsf{ARQ}}}(z) &= z\mathbf{P}^{k-1}(\mathbf{I} - z\mathbf{P}_{10}\mathbf{P}^{k-1} - z\mathbf{P}_{11}\mathbf{P}^{T-1})^{-1} \\ &\times \left[\mathbf{P}_{00} + \mathbf{P}_{01}\sum_{j=0}^{d-1}\mathbf{P}_{x1}^{j}\mathbf{P}_{x0} + \mathbf{P}_{01}\mathbf{P}_{x1}^{d}(\mathbf{I} - z\mathbf{P}_{x1}^{T})^{-1} z\sum_{j=0}^{T-1}\mathbf{P}_{x1}^{j}\mathbf{P}_{x0}\right]. \qquad (26)\end{aligned}$$

Computing the derivative of $\mathbf{\Phi}_{\tau}(z)$ at $z = 1$, we have

$$\begin{aligned}\mathbf{\Phi}_{\tau_{\mathsf{ARQ}}}{}'(1) &= \mathbf{P}^{k-1}(\mathbf{I} - \mathbf{P}_{10}\mathbf{P}^{k-1} - \mathbf{P}_{11}\mathbf{P}^{T-1})^{-1} \\ &\quad \left\{\left[\mathbf{I} + (\mathbf{P}_{10}\mathbf{P}^{k-1} + \mathbf{P}_{11}\mathbf{P}^{T-1})(\mathbf{I} - \mathbf{P}_{10}\mathbf{P}^{k-1} - \mathbf{P}_{11}\mathbf{P}^{T-1})^{-1}\right]\right. \\ &\quad \times \left[\mathbf{P}_{00} + \mathbf{P}_{01}\sum_{j=0}^{d-1}\mathbf{P}_{x1}^{j}\mathbf{P}_{x0} + \mathbf{P}_{01}\mathbf{P}_{x1}^{d}(\mathbf{I} - \mathbf{P}_{x1}^{T})^{-1}\sum_{j=0}^{T-1}\mathbf{P}_{x1}^{j}\mathbf{P}_{x0}\right] \\ &\quad \left. + \mathbf{P}_{01}\mathbf{P}_{x1}^{d}(\mathbf{I} - \mathbf{P}_{x1}^{T})^{-1}\mathbf{P}_{x1}^{T}(\mathbf{I} - \mathbf{P}_{x1}^{T})^{-1}\sum_{j=0}^{T-1}\mathbf{P}_{x1}^{j}\mathbf{P}_{x0} + \mathbf{P}_{01}\mathbf{P}_{x1}^{d}(\mathbf{I} - \mathbf{P}_{x1}^{T})^{-1}\sum_{j=0}^{T-1}\mathbf{P}_{x1}^{j}\mathbf{P}_{x0}\right\},\end{aligned}$$

where we use the identity $\frac{d}{dz}(\mathbf{I} - \mathbf{A}z)^{-1} = (\mathbf{I} - \mathbf{A}z)^{-1}\mathbf{A}(\mathbf{I} - \mathbf{A}z)^{-1}$ for a square matrix $\mathbf{A}$.

Considering the case of memoryless channel, the mean transmission time is expressed as:

$$\begin{aligned}\overline{\tau}_{\mathsf{ARQ}} &= (1 - \epsilon(1-\epsilon) - \epsilon^{2})^{-1}\left\{\left[1 + (\epsilon(1-\epsilon) + \epsilon^{2})(1 - \epsilon(1-\epsilon) - \epsilon^{2})^{-1}\right]\right. \\ &\quad \times \left[(1-\epsilon)^{2} + (1-\epsilon)\epsilon\sum_{j=0}^{d-1}\epsilon^{j}(1-\epsilon) + (1-\epsilon)\epsilon\epsilon^{d}(1-\epsilon^{T})^{-1}\sum_{j=0}^{T-1}\epsilon^{j}(1-\epsilon)\right] \\ &\quad \left. + (1-\epsilon)\epsilon\epsilon^{d}(1-\epsilon^{T})^{-1}\epsilon^{T}(1-\epsilon^{T})^{-1}\sum_{j=0}^{T-1}\epsilon^{j}(1-\epsilon) + (1-\epsilon)\epsilon\epsilon^{d}(1-\epsilon^{T})^{-1}\sum_{j=0}^{T-1}\epsilon^{j}(1-\epsilon)\right\} \\ &= (1-\epsilon)^{-1} + \epsilon^{d+1}(1-\epsilon^{T})^{-1}.\end{aligned}$$

Using this along with the relation $\eta_{\mathsf{ARQ}} = 1/\overline{\tau}_{\mathsf{ARQ}}$, the final expression for throughput can be obtained.

C. Proof of Proposition 6

From (5), (6), (8), (9), (11) and (12), the MGF of the delay for uncoded ARQ is given by [39]

$$\mathbf{\Phi}_{\mathsf{D}_{\mathsf{ARQ}}}(z) = z^{k-1}\mathbf{P}^{k-1}(\mathbf{I} - z^{k}\mathbf{P}_{10}\mathbf{P}^{k-1} - z^{T}\mathbf{P}_{11}\mathbf{P}^{T-1})^{-1} \times \left[z\mathbf{P}_{00} + z^{2}\mathbf{P}_{01}(\mathbf{I} - z\mathbf{P}_{x1})^{-1}\mathbf{P}_{x0}\right]. \qquad (27)$$

Computing the derivative of $\mathbf{\Phi}_{\mathsf{D_{ARQ}}}(z)$ at $z=1$, we have

$$\begin{aligned}\mathbf{\Phi}_{\mathsf{D_{ARQ}}}{}'(z) = &\big((k-1)\mathbf{P}^{k-1}(\mathbf{I}-\mathbf{A}(1))^{-1} + \mathbf{P}^{k-1}(\mathbf{I}-\mathbf{A}(1))^{-1}\mathbf{A}'(1)(\mathbf{I}-\mathbf{A}(1))^{-1}\big)\\ &\times \Big[\mathbf{P}_{00} + \mathbf{P}_{01}(\mathbf{I}-\mathbf{P}_{x1})^{-1}\mathbf{P}_{x0}\Big]\\ &+\mathbf{P}^{k-1}(\mathbf{I}-\mathbf{A}(1))^{-1} \times \Big[\mathbf{P}_{00} + 2\mathbf{P}_{01}(\mathbf{I}-\mathbf{P}_{x1})^{-1}\mathbf{P}_{x0} + \mathbf{P}_{01}\mathbf{C}(1)\mathbf{P}_{x0}\Big],\end{aligned}$$

where $\mathbf{A}(z) = z^k\mathbf{P}_{10}\mathbf{P}^{k-1} + z^T\mathbf{P}_{11}\mathbf{P}^{T-1}$, we have $\mathbf{A}'(z) = kz^{k-1}\mathbf{P}_{10}\mathbf{P}^{k-1} + Tz^{T-1}\mathbf{P}_{11}\mathbf{P}^{T-1}$, and $\mathbf{C}(z) = (\mathbf{I}-z\mathbf{P}_{x1})^{-1}\mathbf{P}_{x1}(\mathbf{I}-z\mathbf{P}_{x1})^{-1}$. Hence, the mean delay equals $\overline{\mathsf{D}}_{\mathsf{ARQ}} = \frac{1}{1-\epsilon}\pi\mathbf{P}_0\mathbf{\Phi}'_D(1)\mathbf{1}$.

Considering the case of memoryless channel, the mean delay is expressed as

$$\overline{\mathsf{D}}_{\mathsf{ARQ}} = \mathbf{\Phi}'_D(1) = k + \frac{\epsilon}{1-\epsilon}(1+T\epsilon) + k\epsilon.$$

D. Proof of Proposition 7

The second derivative of $\mathbf{\Phi}_{\mathsf{D_{ARQ}}}(z)$ at $z=1$ for uncoded ARQ can be computed as

$$\begin{aligned}\mathbf{\Phi}_{\mathsf{D_{ARQ}}}{}''(1) = &\big((k-1)(k-2)\mathbf{P}^{k-1}(\mathbf{I}-\mathbf{A}(1))^{-1} + 2(k-1)\mathbf{P}^{k-1}\mathbf{B}(1) + \mathbf{P}^{k-1}\mathbf{B}'(1)\big)\\ &\times\Big[\mathbf{P}_{00} + \mathbf{P}_{01}(\mathbf{I}-\mathbf{P}_{x1})^{-1}\mathbf{P}_{x0}\Big]\\ &+2((k-1)\mathbf{P}^{k-1}(\mathbf{I}-\mathbf{A}(1))^{-1} + \mathbf{P}^{k-1}\mathbf{B}(1)) \times \Big[\mathbf{P}_{00} + 2\mathbf{P}_{01}(\mathbf{I}-\mathbf{P}_{x1})^{-1}\mathbf{P}_{x0} + \mathbf{P}_{01}\mathbf{C}(1)\mathbf{P}_{x0}\Big]\\ &+\mathbf{P}^{k-1}(\mathbf{I}-\mathbf{A}(1))^{-1} \times \Big[2\mathbf{P}_{01}(\mathbf{I}-\mathbf{P}_{x1})^{-1}\mathbf{P}_{x0} + 4\mathbf{P}_{01}\mathbf{C}(1)\mathbf{P}_{x0} + \mathbf{P}_{01}\mathbf{C}'(1)\mathbf{P}_{x0}\Big],\end{aligned}$$

where $\mathbf{A}(z)$ and $\mathbf{C}(z)$ are defined in Appendix C, and $\mathbf{B}(z) = (\mathbf{I}-\mathbf{A}(z))^{-1}\mathbf{A}'(z)(\mathbf{I}-\mathbf{A}(z))^{-1}$. Given these functions, we can compute their derivatives as follows:

$$\mathbf{A}'(z) = kz^{k-1}\mathbf{P}_{10}\mathbf{P}^{k-1} + Tz^{T-1}\mathbf{P}_{11}\mathbf{P}^{T-1},$$

$$\mathbf{A}''(z) = k(k-1)z^{k-2}\mathbf{P}_{10}\mathbf{P}^{k-1} + T(T-1)z^{T-2}\mathbf{P}_{11}\mathbf{P}^{T-1},$$

$$\mathbf{B}'(z) = 2(\mathbf{I}-\mathbf{A}(z))^{-1}\mathbf{A}'(z)(\mathbf{I}-\mathbf{A}(z))^{-1}\mathbf{A}'(z)(\mathbf{I}-\mathbf{A}(z))^{-1} + (\mathbf{I}-\mathbf{A}(z))^{-1}\mathbf{A}''(z)(\mathbf{I}-\mathbf{A}(z))^{-1},$$

$$\mathbf{C}'(z) = 2(\mathbf{I}-z\mathbf{P}_{x1})^{-1}\mathbf{P}_{x1}(\mathbf{I}-z\mathbf{P}_{x1})^{-1}\mathbf{P}_{x1}(\mathbf{I}-z\mathbf{P}_{x1})^{-1}.$$

For the case of memoryless channel, simplifying above expressions, we have that $\mathbf{A}(z) = z^k\epsilon(1-\epsilon)+z^T\epsilon^2$, and $\mathbf{B}(z) = (1-\mathbf{A}(z))^{-2}\mathbf{A}'(z)$, and $\mathbf{C}(z) = \epsilon(1-z\epsilon)^{-2}$ and $\mathbf{C}'(z) = 2\epsilon^2(1-z\epsilon)^{-3}$. Using the simplified functions, we obtain that $\Big[\mathbf{P}_{00}+\mathbf{P}_{01}(\mathbf{I}-\mathbf{P}_{x1})^{-1}\mathbf{P}_{x0}\Big] = 1-\epsilon$, $\Big[\mathbf{P}_{00}+2\mathbf{P}_{01}(\mathbf{I}-\mathbf{P}_{x1})^{-1}\mathbf{P}_{x0}+\mathbf{P}_{01}\mathbf{C}(1)\mathbf{P}_{x0}\Big] = 1$, and $\Big[2\mathbf{P}_{01}(\mathbf{I}-\mathbf{P}_{x1})^{-1}\mathbf{P}_{x0}+4\mathbf{P}_{01}\mathbf{C}(1)\mathbf{P}_{x0}+\mathbf{P}_{01}\mathbf{C}'(1)\mathbf{P}_{x0}\Big] = 2(1-\epsilon)^{-1}\epsilon$. Hence, using the MGF of D, the variance of the delay for uncoded ARQ is given as follows:

$$\begin{aligned}
\sigma^2_{\mathsf{D}_{\mathsf{ARQ}}} &= \frac{1}{1-\epsilon}\pi\mathbf{P}_0\mathbf{\Phi}_{\mathsf{D}_{\mathsf{ARQ}}}{}''(1)\mathbf{1} + \overline{\mathsf{D}}_{\mathsf{ARQ}} - \overline{\mathsf{D}}_{\mathsf{ARQ}}{}^2 \\
&= \Big\{\Big((k-1)(k-2)(1-\epsilon(1-\epsilon)-\epsilon^2)^{-1} + 2(k-1)\mathbf{B}(1) + \mathbf{B}'(1)\Big)(1-\epsilon) \\
&+ 2\Big((k-1)(1-\epsilon(1-\epsilon)-\epsilon^2)^{-1} + \mathbf{B}(1)\Big) + (1-\epsilon(1-\epsilon)-\epsilon^2)^{-1}2(1-\epsilon)^{-1}\epsilon\Big\} + \bar{D} - \bar{D}^2 \\
&= -3k + 2 - k^2\epsilon^2 + \frac{\epsilon^4}{(1-\epsilon)^3}2T^2 + \frac{\epsilon}{(1-\epsilon)^2}(2 + 4kT\epsilon^2 + T^2\epsilon + T\epsilon - \epsilon - 2T\epsilon^2 - T^2\epsilon^3) \\
&+ \frac{1}{1-\epsilon}(k^2\epsilon + 2k - 2 - k\epsilon + \epsilon + 2T\epsilon^2 + 2k^2\epsilon^2 - 2kT\epsilon^2 - 2k\epsilon^2 - 2Tk\epsilon^3).
\end{aligned}$$

Hence, the final result can be given in the form of (19).

E. *Proof of Proposition 3*

The MGF of the transmission time in the case of no coding with HARQ combining is given by

$$\begin{aligned}
\mathbf{\Phi}_{\tau_{\mathsf{HARQ}}}(z) = z\mathbf{P}^{k-1}\sum_{j=0}^{\infty}\prod_{i=0}^{j}(z\mathbf{P}_{10}(i)\mathbf{P}^{k-1} + z\mathbf{P}_{11}(i)\mathbf{P}^{T-1})\Big[\mathbf{P}_{00}(j^*) \\
+ \mathbf{P}_{01}(j^*)\sum_{j=0}^{d-1}\Big(\prod_{i=0}^{j}\mathbf{P}_{x1}(j^*+i)\Big)\mathbf{P}_{x0}(j^*+j+1) + \mathbf{P}_{01}(j^*)\Big(\prod_{i=0}^{d}\mathbf{P}_{x1}(j^*+i)\Big) \\
\times\sum_{j=0}^{\infty}z^j\prod_{i=1}^{j}\Big(\prod_{l=(i-1)T+1}^{iT}\mathbf{P}_{x1}(j^*+l)\Big)z\sum_{j'=0}^{T-1}\Big(\prod_{i=0}^{j'}\mathbf{P}_{x1}(j^*+jT+i)\Big)\mathbf{P}_{x0}(j^*+jT+j'+1)\Big],
\end{aligned} \tag{28}$$

which is a generalization of uncoded ARQ in [39]. In the above, j^* denotes attempt index where the forward link is successful, i.e. the initial j^*-1 attempts were not successful, $\mathbf{P}_{10}(i)$ and $\mathbf{P}_{11}(i)$ denote the probabilities of receiving an error-free NACK and an erroneous NACK on attempt $i \geq 0$, respectively, and $\mathbf{P}_{x0}(i)$ and $\mathbf{P}_{x1}(i)$ denote the success and failure probability matrices on attempt $i \in \{1,\ldots,d-1\}$, respectively. Furthermore, for notational convenience, we let $\mathbf{P}_{x1}(0) = \mathbf{I}$.

The value of $\overline{\tau}_{\mathsf{HARQ}}$ can be found using the relation (2), then computing the reciprocal of its first derivative. Simplifying above expression using the relations for the transition probability matrices for the symmetric memoryless channel, given in Appendix A, the value of throughput $\eta_{\mathsf{HARQ}} = \overline{\tau}_{\mathsf{HARQ}}{}^{-1}$ for the memoryless channel is given by (15).

F. Proof of Proposition 4

The MGF of the delay in the case of (uncoded) HARQ with soft combining is

$$\mathbf{\Phi}_{\mathsf{D}_{\mathsf{HARQ}}}(z) = z^{k-1}\mathbf{P}^{k-1}\sum_{j=0}^{\infty}\prod_{i=0}^{j}(z^{k}\mathbf{P}_{10}(i)\mathbf{P}^{k-1} + z^{T}\mathbf{P}_{11}(i)\mathbf{P}^{T-1}) \\ \times\left[z\mathbf{P}_{00}(j^{*}) + z^{2}\mathbf{P}_{01}(j^{*})\sum_{j'=0}^{\infty}z^{j'}\prod_{i=0}^{j'}\mathbf{P}_{x1}(j^{*}+i)\mathbf{P}_{x0}(j^{*}+j'+1)\right]. \tag{29}$$

Average delay $\overline{\mathsf{D}}_{\mathsf{HARQ}}$ can be found using the relation (3), then computing its first derivative. Simplifying above expressions, average delay $\overline{\mathsf{D}}_{\mathsf{HARQ}}$ for the memoryless channel is given by (16).

G. Proof of Proposition 8

For CF ARQ, the MGF of the transmission time for $M = 2$ packets is given by

$$\mathbf{\Phi}_{\tau_{\mathsf{CF\text{-}ARQ}}}(z) = z\mathbf{P}^{k}\Big[(\mathbf{I}-\mathbf{P}^{\mathcal{T}}_{1x}(1))^{-1}\mathbf{A}^{\mathsf{CF}}_{1}(z) + \mathbf{P}^{\mathsf{CF}}_{A}(1)\prod\nolimits_{i=1}^{2}(\mathbf{I}-\mathbf{P}^{\mathcal{T}}_{1x}(i))^{-1}\mathbf{A}^{\mathsf{CF}}_{2}(z)\Big], \tag{30}$$

where

$$\mathbf{P}^{\mathcal{T}}_{1x}(1) = z(\mathbf{P}^{\mathsf{CF}}_{10}(1) + \mathbf{P}^{\mathsf{CF}}_{11}(1)(\mathbf{P}^{2})^{d})\mathbf{P}^{k+1}, \quad \mathbf{P}^{\mathcal{T}}_{1x}(2) = z(\mathbf{P}^{\mathsf{CF}}_{10}(2) + \mathbf{P}^{\mathsf{CF}}_{11}(2)\mathbf{P}^{d-1})\mathbf{P}^{k},$$

and the matrix $\mathbf{A}^{\mathsf{CF}}_{n}(z)$ that gives the gain of the transition from the state $A^{\mathsf{CF}}_{2-(n-1)}$ is computed as

$$\mathbf{A}^{\mathsf{CF}}_{n}(z) = \mathbf{P}^{\mathsf{CF}}_{00}(n) + \mathbf{P}^{\mathsf{CF}}_{01}(n)\Big[\sum\nolimits_{i=1}^{d_n}\mathbf{P}^{\mathsf{CF}}_{x1}(n)^{i-1}\mathbf{P}^{\mathsf{CF}}_{x0}(n) \\ +\mathbf{P}^{\mathsf{CF}}_{x1}(n)^{d_n}(\mathbf{I}-z\mathbf{P}^{\mathsf{CF}}_{x1}(n)^{T})^{-1}z\sum\nolimits_{i=0}^{T-1}\mathbf{P}^{\mathsf{CF}}_{x1}(n)^{i}\mathbf{P}^{\mathsf{CF}}_{x0}(n)\Big], \quad n = \{1,2\},$$

where $d_n = d-(n-1)$. The PGF of the transmission time of CF ARQ for $M = 2$ coded packets is computed as $\Phi_{\tau_{\mathsf{CF\text{-}ARQ}}}(z) = \pi_I\,\mathbf{\Phi}_{\tau_{\mathsf{CF\text{-}ARQ}}}(z)\mathbf{1}/(\pi_I\mathbf{1})$ using the MGF $\mathbf{\Phi}_{\tau}(z)$ in (30), where $\mathbf{1}$ is a column vector of ones, π_I is the probability vector of state I, and equals $\pi_I = \pi\mathbf{P}_0$. Finally, the throughput is the reciprocal of the derivative of $\Phi_{\tau_{\mathsf{CF\text{-}ARQ}}}(z)$ at $z = 1$, i.e., $\eta_{\mathsf{CF\text{-}ARQ}} = 1/\,{\Phi_{\tau_{\mathsf{CF\text{-}ARQ}}}}'(1)$.

For CF ARQ with $M = 2$, using the matrix-flow graph for throughput analysis (similar to the graph shown in Fig. 6 for delay), and using the basic simplification rules, the MGF of $\tau_{\mathsf{CF\text{-}ARQ}}$, i.e.,

$\mathbf{\Phi}_{\tau_{\mathsf{CF\text{-}ARQ}}}(z)$, can be computed as

$$\begin{aligned}
&\mathbf{\Phi}_{\tau_{\mathsf{CF\text{-}ARQ}}}(z)\\
&= \mathbf{P}^{T-d-1}(\mathbf{I} - P_{1x,T1})^{-1}\Big[\mathbf{P}_{01}^{\mathsf{CF}}(1)\big(\mathbf{P}_{x1}^{\mathsf{CF}}(1)\big)^{d}(\mathbf{I} - \big(\mathbf{P}_{x1}^{\mathsf{CF}}(1)\big)^{T})^{-1}\Big(\sum_{n=0}^{T-1}\big(\mathbf{P}_{x1}^{\mathsf{CF}}(1)\big)^{n}\Big)\mathbf{P}_{x0}^{\mathsf{CF}}(1)\\
&+ \mathbf{P}_{00}^{\mathsf{CF}}(1) + \Big(\sum_{n=1}^{d}\mathbf{P}_{01}^{\mathsf{CF}}(1)\big(\mathbf{P}_{x1}^{\mathsf{CF}}(1)\big)^{n-1}\mathbf{P}_{x0}^{\mathsf{CF}}(1)\Big)\\
&+ \mathbf{P}_{0}^{*}(\mathbf{I} - \mathbf{P}_{1x,T2}(1))^{-1}\Big(\mathbf{P}_{00} + \Big(\sum_{n=1}^{d-1}\mathbf{P}_{01}\mathbf{P}_{x1}^{n-1}\mathbf{P}_{x0}\Big) + \mathbf{P}_{01}\mathbf{P}_{x1}^{d-1}(\mathbf{I} - \mathbf{P}_{x1}^{T})^{-1}\Big(\sum_{n=0}^{T-1}\mathbf{P}_{x1}^{n}\Big)\mathbf{P}_{x0}\Big)\Big]\\
&+ \mathbf{P}^{T-d-1}(\mathbf{I} - \mathbf{P}_{1x,T1}(1))^{-1}\mathbf{P}'_{1x,T1}(1)(\mathbf{I} - \mathbf{P}_{1x,T1}(1))^{-1}\Big[\mathbf{P}_{01}^{\mathsf{CF}}(1)\big(\mathbf{P}_{x1}^{\mathsf{CF}}(1)\big)^{d}(\mathbf{I} - \big(\mathbf{P}_{x1}^{\mathsf{CF}}(1)\big)^{T})^{-1}\\
&\Big(\sum_{n=0}^{T-1}\big(\mathbf{P}_{x1}^{\mathsf{CF}}(1)\big)^{n}\Big)\mathbf{P}_{x0}^{\mathsf{CF}}(1) + \mathbf{P}_{00}^{\mathsf{CF}}(1) + \Big(\sum_{n=1}^{d}\mathbf{P}_{01}^{\mathsf{CF}}(1)\big(\mathbf{P}_{x1}^{\mathsf{CF}}(1)\big)^{n-1}\mathbf{P}_{x0}^{\mathsf{CF}}(1)\Big)\\
&+ \mathbf{P}_{0}^{*}(\mathbf{I} - \mathbf{P}_{1x,T2}(1))^{-1}\Big(\mathbf{P}_{00} + \Big(\sum_{n=1}^{d-1}\mathbf{P}_{01}\mathbf{P}_{x1}^{n-1}\mathbf{P}_{x0}\Big) + \mathbf{P}_{01}\mathbf{P}_{x1}^{d-1}(\mathbf{I} - (\mathbf{P}_{x1}^{\mathsf{CF}}(2))^{T})^{-1}\Big(\sum_{n=0}^{T-1}\mathbf{P}_{x1}^{n}\Big)\mathbf{P}_{x0}\Big)\Big]\\
&+ P^{T-d-1}(\mathbf{I} - \mathbf{P}_{1x,T1}(1))^{-1}\Big[\mathbf{P}_{01}^{\mathsf{CF}}(1)\big(\mathbf{P}_{x1}^{\mathsf{CF}}(1)\big)^{d}(\mathbf{I} - \big(\mathbf{P}_{x1}^{\mathsf{CF}}(1)\big)^{T})^{-1}\big(\mathbf{P}_{x1}^{\mathsf{CF}}(1)\big)^{T}(\mathbf{I} - \big(\mathbf{P}_{x1}^{\mathsf{CF}}(1)\big)^{T})^{-1}\\
&\times\Big(\sum_{n=0}^{T-1}\big(\mathbf{P}_{x1}^{\mathsf{CF}}(1)\big)^{n}\Big)\mathbf{P}_{x0}^{\mathsf{CF}}(1) + \mathbf{P}_{01}^{\mathsf{CF}}(1)\big(\mathbf{P}_{x1}^{\mathsf{CF}}(1)\big)^{d}(\mathbf{I} - \big(\mathbf{P}_{x1}^{\mathsf{CF}}(1)\big)^{T})^{-1}\Big(\sum_{n=0}^{T-1}\big(\mathbf{P}_{x1}^{\mathsf{CF}}(1)\big)^{n}\Big)\mathbf{P}_{x0}^{\mathsf{CF}}(1)\\
&+ \mathbf{P}_{0}^{*}(\mathbf{I} - \mathbf{P}_{1x,T2}(1))^{-1}\mathbf{P}'_{1x,T2}(1)(\mathbf{I} - \mathbf{P}_{1x,T2}(1))^{-1}\Big(\mathbf{P}_{00} + \Big(\sum_{n=1}^{d-1}\mathbf{P}_{01}\mathbf{P}_{x1}^{n-1}\mathbf{P}_{x0}\Big)\\
&+ \mathbf{P}_{01}(\mathbf{P}_{x1})^{d-1}(\mathbf{I} - (\mathbf{P}_{x1})^{T})^{-1}\Big(\sum_{n=0}^{T-1}(\mathbf{P}_{x1})^{n}\Big)\mathbf{P}_{x0}\Big)\\
&+ \mathbf{P}_{0}^{*}(\mathbf{I} - \mathbf{P}_{1x,T2}(1))^{-1}\Big(\mathbf{P}_{01}\mathbf{P}_{x1}^{d-1}(\mathbf{I} - \mathbf{P}_{x1}^{T})^{-1}\mathbf{P}_{x1}^{T}(\mathbf{I} - \mathbf{P}_{x1}^{T})^{-1}\Big(\sum_{n=0}^{T-1}\mathbf{P}_{x1}^{n}\Big)\mathbf{P}_{x0}\\
&+ \mathbf{P}_{01}(\mathbf{P}_{x1})^{d-1}(\mathbf{I} - (\mathbf{P}_{x1})^{T})^{-1}\Big(\sum_{n=0}^{T-1}\mathbf{P}_{x1}^{n}\Big)\mathbf{P}_{x0}\Big)\Big],
\end{aligned}$$

where we note that $\mathbf{P}_{xy}^{\mathsf{CF}} = \mathbf{P}_{xy}^{\mathsf{CF}}(1)$ and $\mathbf{P}_{xy} = \mathbf{P}_{xy}^{\mathsf{CF}}(2)$ for $x, y \in \{0, 1\}$, and

$$\mathbf{P}_{0}^{*} = \mathbf{P}_{00}\mathbf{P}_{10} + \mathbf{P}_{10}\mathbf{P}_{00} + \mathbf{P}_{00}\mathbf{P}_{01} + \mathbf{P}_{01}\mathbf{P}_{00} + \mathbf{P}_{00}\mathbf{P}_{11} + \mathbf{P}_{11}\mathbf{P}_{00}$$

$$\mathbf{P}_{1x,T1}(z) = z(\mathbf{P}_{10}^{\mathsf{CF}}(1) + \mathbf{P}_{11}^{\mathsf{CF}}(1)\mathbf{P}^{d}\mathbf{P}^{d})\mathbf{P}^{T-d}$$

$$\mathbf{P}_{1x,T2}(z) = z(\mathbf{P}_{10} + \mathbf{P}_{11}\mathbf{P}^{d-1})\mathbf{P}^{T-d-1}.$$

The value of $\overline{\tau}_{\mathsf{CF\text{-}ARQ}}$ for CF ARQ can be computed using the relation $\overline{\tau}_{\mathsf{CF\text{-}ARQ}} = \frac{1}{1-\epsilon}\pi_{I}\,\mathbf{\Phi}_{\tau_{\mathsf{CF\text{-}ARQ}}}(z)\mathbf{1}$.

Simplifying above expressions, the value of throughput $\eta_{\mathsf{CF\text{-}ARQ}} = 2\overline{\tau}_{\mathsf{CF\text{-}ARQ}}{}^{-1}$ for CF ARQ for $M = 2$ for the case of memoryless channel is computed as

$$\eta_{\mathsf{CF\text{-}ARQ}} = \frac{(1+\epsilon-2\epsilon^2+2\epsilon^3)/(2-\epsilon)}{\alpha_{\mathsf{CF}}(\epsilon)-\beta_{\mathsf{CF}}(\epsilon,T,d)+\epsilon^d(1-\epsilon)/(1-\epsilon^T)},$$

where $d = T - k$ and

$$\alpha_{\mathsf{CF}}(\epsilon) = \frac{1+3\epsilon-2\epsilon^2+20\epsilon^3-18\epsilon^4+28\epsilon^5-60\epsilon^6+72\epsilon^7-40\epsilon^8+8\epsilon^9}{2(2-\epsilon)(1-\epsilon+4\epsilon^2-2\epsilon^3)(1+\epsilon-2\epsilon^2+2\epsilon^3)},$$

$$\beta_{\mathsf{CF}}(\epsilon,T,d) = \frac{(1-\epsilon)^2(1-2\epsilon+4\epsilon^2)\epsilon^d(2-\epsilon)^{d-1}(1-2\epsilon+2\epsilon^2)^{d-1}}{2(1-\epsilon+4\epsilon^2-2\epsilon^3)(\epsilon^T(2-\epsilon)^T(1-2\epsilon+2\epsilon^2)^T-1)}.$$

For $\epsilon \leq 0.5$, we have that $\alpha_{\mathsf{CF}}(\epsilon) \in [0.25, 0.8]$ and $\beta_{CF}(\epsilon,T,d) = 0$ for all T, d when $\epsilon = 0$ and $\beta_{CF}(\epsilon,T,d) \to 0$ as $T \to \infty$ for all ϵ, and $\beta_{CF}(\epsilon,T,d) \approx 0$ for any finite $T > k$. Using the relation $\frac{(\alpha_{\mathsf{CF}}(\epsilon)-\beta_{\mathsf{CF}}(\epsilon))(1-\epsilon)}{(1+\epsilon-2\epsilon^2+2\epsilon^3)/(2-\epsilon)} \approx 0.5$ when $T > k$, we have the final expression in (20).

H. Proof of Proposition 9

For CF ARQ, the MGF of the delay for $M = 2$ packets is given by

$$\mathbf{\Phi}_{\mathsf{D}_{\mathsf{CF\text{-}ARQ}}}(z) = z^k\mathbf{P}^k\Big[(\mathbf{I}-\mathbf{P}^{\mathrm{D}}_{1x}(1))^{-1}\mathbf{B}^{\mathsf{CF}}_1(z) + z\mathbf{P}^{\mathsf{CF}}_A(1)\prod\nolimits_{i=1}^{2}(\mathbf{I}-\mathbf{P}^{\mathrm{D}}_{1x}(i))^{-1}\mathbf{B}^{\mathsf{CF}}_2(z)\Big], \tag{31}$$

where $\mathbf{B}^{\mathsf{CF}}_n(z)$ for $n \in \{1, 2\}$ can be computed using relation

$$\mathbf{B}^{\mathsf{CF}}_n(z) = z\mathbf{P}^{\mathsf{CF}}_{00}(n) + z\mathbf{P}^{\mathsf{CF}}_{01}(n)(\mathbf{I}-z\mathbf{P}^{\mathsf{CF}}_{x1}(n))^{-1}z\mathbf{P}^{\mathsf{CF}}_{x0}(n).$$

The PGF of delay of CF ARQ for $M = 2$ coded packets can be computed as $\Phi_{\mathsf{D}_{\mathsf{CF\text{-}ARQ}}}(z) = \pi_I\,\mathbf{\Phi}_{\mathsf{D}_{\mathsf{CF\text{-}ARQ}}}(z)\mathbf{1}/(\pi_I\mathbf{1})$ using the MGF $\mathbf{\Phi}_D(z)$ in (31). Finally, the average delay will be the derivative of $\Phi_{\mathsf{D}_{\mathsf{CF\text{-}ARQ}}}(z)$ at $z = 1$, i.e., $\overline{\mathrm{D}}_{\mathsf{CF\text{-}ARQ}} = \Phi_{\mathsf{D}_{\mathsf{CF\text{-}ARQ}}}{}'(1)$.

Average delay $\overline{\mathrm{D}}_{\mathsf{CF\text{-}ARQ}}$ for CF ARQ for $M = 2$ is given by

$$\begin{aligned}\overline{\mathrm{D}}_{\mathsf{CF\text{-}ARQ}} = \frac{1}{1-\epsilon}\pi_I\Big\{&k\mathbf{P}^k(\mathbf{I}-\mathbf{P}_{1x,D1})^{-1}\Big(\mathbf{P}^{\mathsf{CF}}_{00}(1)+\mathbf{P}^{\mathsf{CF}}_{01}(1)(\mathbf{I}-\mathbf{P}^{\mathsf{CF}}_{x1}(1))^{-1}\mathbf{P}^{\mathsf{CF}}_{x0}(1)\\
&+\mathbf{P}^*_0(\mathbf{I}-\mathbf{P}_{1x,D2})^{-1}(\mathbf{P}_{00}+\mathbf{P}_{01}(\mathbf{I}-\mathbf{P}_{x1})^{-1}\mathbf{P}_{x0})\Big)\\
&+\mathbf{P}^k(\mathbf{I}-\mathbf{P}_{1x,D1})^{-1}\mathbf{P}'_{1x,D1}(\mathbf{I}-\mathbf{P}_{1x,D1})^{-1}\Big(\mathbf{P}^{\mathsf{CF}}_{00}(1)+\mathbf{P}^{\mathsf{CF}}_{01}(1)(\mathbf{I}-\mathbf{P}^{\mathsf{CF}}_{x1}(1))^{-1}\mathbf{P}^{\mathsf{CF}}_{x0}(1)\end{aligned}$$

$$+\mathbf{P}_0^*(\mathbf{I}-\mathbf{P}_{1x,D2})^{-1}(\mathbf{P}_{00}+\mathbf{P}_{01}(\mathbf{I}-\mathbf{P}_{x1})^{-1}\mathbf{P}_{x0})\Big)+\mathbf{P}^k(\mathbf{I}-\mathbf{P}_{1x,D1})^{-1}$$

$$\Big(\mathbf{P}_{00}^{\mathsf{CF}}(1)+2\mathbf{P}_{01}^{\mathsf{CF}}(1)(\mathbf{I}-\mathbf{P}_{x1}^{\mathsf{CF}}(1))^{-1}\mathbf{P}_{x0}^{\mathsf{CF}}(1)+\mathbf{P}_{01}^{\mathsf{CF}}(1)(\mathbf{I}-\mathbf{P}_{x1}^{\mathsf{CF}}(1))^{-1}\mathbf{P}_{x1}^{\mathsf{CF}}(1)(\mathbf{I}-\mathbf{P}_{x1}^{\mathsf{CF}}(1))^{-1}\mathbf{P}_{x0}^{\mathsf{CF}}(1)$$

$$+\mathbf{P}_0^*(\mathbf{I}-\mathbf{P}_{1x,D2})^{-1}(\mathbf{P}_{00}+\mathbf{P}_{01}(\mathbf{I}-\mathbf{P}_{x1})^{-1}\mathbf{P}_{x0})$$

$$+\mathbf{P}_0^*(\mathbf{I}-\mathbf{P}_{1x,D2})^{-1}\mathbf{P}'_{1x,D2}(\mathbf{I}-\mathbf{P}_{1x,D2})^{-1}(\mathbf{P}_{00}+\mathbf{P}_{01}(\mathbf{I}-\mathbf{P}_{x1})^{-1}\mathbf{P}_{x0})$$

$$+\mathbf{P}_0^*(\mathbf{I}-\mathbf{P}_{1x,D2})^{-1}(\mathbf{P}_{00}+2\mathbf{P}_{01}(\mathbf{I}-\mathbf{P}_{x1})^{-1}\mathbf{P}_{x0}+\mathbf{P}_{01}(\mathbf{I}-\mathbf{P}_{x1})^{-1}\mathbf{P}_{x1}(\mathbf{I}-\mathbf{P}_{x1})^{-1}\mathbf{P}_{x0})\Big)\Big\}\mathbf{1},$$

where

$$\mathbf{P}_{1x,D1}=z^{T-d}\mathbf{P}^{T-d}(z\mathbf{P}_{10}^{\mathsf{CF}}(1)+z\mathbf{P}_{11}^{\mathsf{CF}}(1)z^d\mathbf{P}^d)$$

$$\mathbf{P}_{1x,D2}=z^{T-d-1}\mathbf{P}^{T-d-1}(z\mathbf{P}_{10}+z\mathbf{P}_{11}z^{d+1}\mathbf{P}^{d+1}).$$

Simplifying above expressions, the exact expression for average delay $\overline{\mathrm{D}}_{\mathsf{CF\text{-}ARQ}}$ for CF ARQ for $M=2$ for the case of memoryless channel is given by

$$\overline{\mathrm{D}}_{\mathsf{CF\text{-}ARQ}}=(1+k+\epsilon(8+2k)-\epsilon^2(9+k)+2\epsilon^3(27+13k+3T)-\epsilon^4(49+49k+3T)$$

$$+2\epsilon^5(71+60k+5T)-\epsilon^6(393+283k-83T)+4\epsilon^7(213+132k-62T)-2\epsilon^8(780+475k-271T)$$

$$+4\epsilon^9(489+314k-200T)-2\epsilon^{10}(757+521k-365T)+4\epsilon^{11}(171+129k-101T)$$

$$-4\epsilon^{12}(41+35k-31T)+16\epsilon^{13}(1+k-T))/(1+\epsilon^2+6\epsilon^3-12\epsilon^4+12\epsilon^5-4\epsilon^6)^2.$$

Using this, the final expression can be derived as $\epsilon\to 0$ as given in (21).

I. Proof of Proposition 10

For Coded ARQ, the MGF of the transmission time (for $M=2$ packets) is given by

$$\boldsymbol{\Phi}_{\tau_{\mathsf{C\text{-}ARQ}}}(z)=z^2\mathbf{P}^k(\mathbf{I}-\mathbf{g}_2^{\mathsf{C}}(z))^{-1}\times$$

$$\Big[[(\mathbf{P}_{100}+\mathbf{P}_{010})+(\mathbf{P}_{011}+\mathbf{P}_{101})(z\mathbf{P}^T+(\mathbf{I}-\mathbf{g}_3^{\mathsf{C}}(z))^{-1}-\mathbf{I})\mathbf{P}_{10}](\mathbf{I}-\mathbf{g}_1^{\mathsf{C}}(z))^{-1}[\mathbf{A}_{00}^{\mathsf{C}}+\mathbf{A}_{01}^{\mathsf{C}}]$$

$$+\mathbf{A}_{000}^{\mathsf{C}}+\mathbf{A}_{001}^{\mathsf{C}}+(\mathbf{P}_{011}+\mathbf{P}_{101})(z\mathbf{P}^T+(\mathbf{I}-\mathbf{g}_3^{\mathsf{C}}(z))^{-1}-\mathbf{I})[\mathbf{A}_{00}^{\mathsf{C}}+\mathbf{A}_{01}^{\mathsf{C}}]\Big], \qquad (32)$$

where the functions $\mathbf{g}_1^{\mathsf{C}}(z)$, $\mathbf{g}_2^{\mathsf{C}}(z)$ and $\mathbf{g}_3^{\mathsf{C}}(z)$ are the branch gains for the self-loops at states A_1, A_2 and A_3 as shown in Fig. 7 respectively, and are given by

$$\begin{aligned}
\mathbf{g}_1^{\mathsf{C}}(z) &= (\mathbf{P}_{10} + \mathbf{P}_{11}\mathbf{P}^{T-k})z\mathbf{P}^{k-1}, \\
\mathbf{g}_2^{\mathsf{C}}(z) &= (\mathbf{P}_{110} + \mathbf{P}_{111}\mathbf{P}^{T-k})z\mathbf{P}^{k}, \\
\mathbf{g}_3^{\mathsf{C}}(z) &= \mathbf{P}_{11}\mathbf{P}^{T}z,
\end{aligned} \tag{33}$$

and the matrices in (32) are given by

$$\begin{aligned}
\mathbf{A}_{00}^{\mathsf{C}} + \mathbf{A}_{01}^{\mathsf{C}} &= \left[\mathbf{P}_{00} + \mathbf{P}_{01}\Big(\sum_{n=0}^{T-k-1}\mathbf{P}_{x1}^{n}\Big)\mathbf{P}_{x0}\right] + \left[\mathbf{P}_{01}\mathbf{P}_{x1}^{T-k}(\mathbf{I} - z\mathbf{P}_{x1}^{T})^{-1}z\Big(\sum_{n=0}^{T-1}\mathbf{P}_{x1}^{n}\Big)\mathbf{P}_{x0}\right], \\
\mathbf{A}_{000}^{\mathsf{C}} + \mathbf{A}_{001}^{\mathsf{C}} &= \left[\mathbf{P}_{000} + \mathbf{P}_{001}\Big(\sum_{n=0}^{T-k-1}\mathbf{P}_{xx1}^{n}\Big)\mathbf{P}_{xx0}\right] + \left[\mathbf{P}_{001}\mathbf{P}_{xx1}^{T-k}(\mathbf{I} - z\mathbf{P}_{xx1}^{T})^{-1}z\Big(\sum_{n=0}^{T-1}\mathbf{P}_{xx1}^{n}\Big)\mathbf{P}_{xx0}\right].
\end{aligned}$$

The value of $\overline{\tau}_{\mathsf{C\text{-}ARQ}}$ for Coded ARQ for $M = 2$ for the case of memoryless channel is given by

$$\begin{aligned}
\overline{\tau}_{\mathsf{C\text{-}ARQ}} = \frac{1}{1-\epsilon}\pi_I\Big\{&2\mathbf{P}^{k}(\mathbf{I} - \mathbf{g}_2^{\mathsf{C}}(1))^{-1}(\mathbf{B}_1^{\mathsf{C}}(1) + \mathbf{B}_2^{\mathsf{C}}(1) + \mathbf{B}_3^{\mathsf{C}}(1) + \mathbf{B}_4^{\mathsf{C}}(1)) \\
&+ \mathbf{P}^{k}(\mathbf{I} - \mathbf{g}_2^{\mathsf{C}}(1))^{-1}(\mathbf{g}_2^{\mathsf{C}})'(1)(\mathbf{I} - \mathbf{g}_2^{\mathsf{C}}(1))^{-1}(\mathbf{B}_1^{\mathsf{C}}(1) + \mathbf{B}_2^{\mathsf{C}}(1) + \mathbf{B}_3^{\mathsf{C}}(1) + \mathbf{B}_4^{\mathsf{C}}(1)) \\
&+ \mathbf{P}^{k}(\mathbf{I} - \mathbf{g}_2^{\mathsf{C}}(1))^{-1}((\mathbf{B}_1^{\mathsf{C}})'(1) + (\mathbf{B}_2^{\mathsf{C}})'(1) + (\mathbf{B}_3^{\mathsf{C}})'(1) + (\mathbf{B}_4^{\mathsf{C}})'(1))\Big\}\mathbf{1},
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{B}_1^{\mathsf{C}}(z) &= ((\mathbf{P}_{100} + \mathbf{P}_{010}) + (\mathbf{P}_{011} + \mathbf{P}_{101})(z\mathbf{P}^{T} + (\mathbf{I} - \mathbf{g}_3^{\mathsf{C}}(z))^{-1} - \mathbf{I})\mathbf{P}_{10})(\mathbf{I} - \mathbf{g}_1^{\mathsf{C}}(z))^{-1} \\
&\quad \times ((\mathbf{P}_{00} + \mathbf{P}_{01}\Big(\sum_{n=0}^{T-k-1}\mathbf{P}_{x1}^{n}\Big)\mathbf{P}_{x0}) + (\mathbf{P}_{01}\mathbf{P}_{x1}^{T-k}(\mathbf{I} - z\mathbf{P}_{x1}^{T})^{-1}z\Big(\sum_{n=0}^{T-1}\mathbf{P}_{x1}^{n}\Big)\mathbf{P}_{x0})) \\
\mathbf{B}_2^{\mathsf{C}}(z) &= \mathbf{P}_{000} + \mathbf{P}_{001}\Big(\sum_{n=0}^{T-k-1}\mathbf{P}_{xx1}^{n}\Big)\mathbf{P}_{xx0} \\
\mathbf{B}_3^{\mathsf{C}}(z) &= \mathbf{P}_{001}\mathbf{P}_{xx1}^{T-k}(\mathbf{I} - z\mathbf{P}_{xx1}^{T})^{-1}z\Big(\sum_{n=0}^{T-1}\mathbf{P}_{xx1}^{n}\Big)\mathbf{P}_{xx0} \\
\mathbf{B}_4^{\mathsf{C}}(z) &= (\mathbf{P}_{011} + \mathbf{P}_{101})(z\mathbf{P}^{T} + (\mathbf{I} - \mathbf{g}_3^{\mathsf{C}}(1))^{-1} - \mathbf{I}) \\
&\quad \times ((\mathbf{P}_{00} + \mathbf{P}_{01}\Big(\sum_{n=0}^{T-k-1}\mathbf{P}_{x1}^{n}\Big)\mathbf{P}_{x0}) + (\mathbf{P}_{01}\mathbf{P}_{x1}^{T-k}(\mathbf{I} - z\mathbf{P}_{x1}^{T})^{-1}z\Big(\sum_{n=0}^{T-1}\mathbf{P}_{x1}^{n}\Big)\mathbf{P}_{x0})),
\end{aligned}$$

where $\mathbf{g}_i^{\mathsf{C}}(z)$'s for $i \in \{1, 2, 3\}$ are given in (33). Finally, simplifying above expressions, throughput $\eta_{\mathsf{C\text{-}ARQ}} = 2\overline{\tau}_{\mathsf{C\text{-}ARQ}}{}^{-1}$ for the memoryless channel is given by (22).

J. Proof of Proposition 11

For Coded ARQ, the MGF of the delay (for $M = 2$ packets) is given by

$$\begin{aligned}\mathbf{\Phi}_{\mathsf{D}_{\text{C-ARQ}}}(z) &= z^k\mathbf{P}^k(\mathbf{I}-\mathbf{f}_2^{\mathsf{C}}(z))^{-1}\times\\ &\Big[[z(\mathbf{P}_{100}+\mathbf{P}_{010})+z(\mathbf{P}_{011}+\mathbf{P}_{101})(z^T\mathbf{P}^T+(\mathbf{I}-\mathbf{f}_3^{\mathsf{C}}(z))^{-1}-\mathbf{I})z\mathbf{P}_{10}](\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(z))^{-1}\mathbf{B}_{00}^{\mathsf{C}}\\ &+\mathbf{B}_{000}^{\mathsf{C}}+z(\mathbf{P}_{011}+\mathbf{P}_{101})(z^T\mathbf{P}^T+(\mathbf{I}-\mathbf{f}_3^{\mathsf{C}}(z))^{-1}-\mathbf{I})\mathbf{B}_{00}^{\mathsf{C}}\Big], \qquad (34)\end{aligned}$$

where the functions $\mathbf{f}_1^{\mathsf{C}}(z)$, $\mathbf{f}_2^{\mathsf{C}}(z)$ and $\mathbf{f}_3^{\mathsf{C}}(z)$ are the branch gains for the self-loops at states A_1, A_2 and A_3 as shown in Fig. 7 respectively, and are given by

$$\begin{aligned}\mathbf{f}_1^{\mathsf{C}}(z) &= (z\mathbf{P}_{10}+z\mathbf{P}_{11}z^{T-k}\mathbf{P}^{T-k})z^{k-1}\mathbf{P}^{k-1},\\ \mathbf{f}_2^{\mathsf{C}}(z) &= (z\mathbf{P}_{110}+z\mathbf{P}_{111}z^{T-k}\mathbf{P}^{T-k})z^{k}\mathbf{P}^{k},\\ \mathbf{f}_3^{\mathsf{C}}(z) &= z\mathbf{P}_{11}z^{T}\mathbf{P}^{T}, \qquad (35)\end{aligned}$$

and the matrices in (34) are given by

$$\begin{aligned}\mathbf{B}_{00}^{\mathsf{C}} &= z\mathbf{P}_{00}+z\mathbf{P}_{01}(\mathbf{I}-z\mathbf{P}_{x1})^{-1}z\mathbf{P}_{x0},\\ \mathbf{B}_{000}^{\mathsf{C}} &= z\mathbf{P}_{000}+z\mathbf{P}_{001}(\mathbf{I}-z\mathbf{P}_{xx1})^{-1}z\mathbf{P}_{xx0}.\end{aligned}$$

Average delay $\overline{\mathsf{D}}_{\text{C-ARQ}}$ for Coded ARQ for $M = 2$ for the case of memoryless channel is given by

$$\begin{aligned}\overline{\mathsf{D}}_{\text{C-ARQ}} &= \frac{1}{1-\epsilon}\pi_I\Big\{k\mathbf{P}^k(\mathbf{I}-\mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1)+\mathbf{A}_3^{\mathsf{C}}(1)+\mathbf{A}_4^{\mathsf{C}}(1))\\ &+\mathbf{P}^k(\mathbf{I}-\mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{f}_2^{\mathsf{C}})'(1)(\mathbf{I}-\mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1)+\mathbf{A}_3^{\mathsf{C}}(1)+\mathbf{A}_4^{\mathsf{C}}(1))\\ &+\mathbf{P}^k(\mathbf{I}-\mathbf{f}_2^{\mathsf{C}}(1))^{-1}\Big((\mathbf{A}_1^{\mathsf{C}})'(1)(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1)\\ &+\mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{f}_1^{\mathsf{C}})'(1)(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1)\\ &+\mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{A}_2^{\mathsf{C}})'(1)+(\mathbf{A}_3^{\mathsf{C}})'(1)+(\mathbf{A}_4^{\mathsf{C}})'(1)\Big)\Big\}\mathbf{1},\end{aligned}$$

where

$$\begin{aligned}\mathbf{A}_1^{\mathsf{C}}(z) &= z(\mathbf{P}_{100}+\mathbf{P}_{010})+z(\mathbf{P}_{011}+\mathbf{P}_{101})(z^T\mathbf{P}^T+(\mathbf{I}-\mathbf{f}_3^{\mathsf{C}}(z))^{-1}-\mathbf{I})z\mathbf{P}_{10}\\ \mathbf{A}_2^{\mathsf{C}}(z) &= z\mathbf{P}_{00}+z\mathbf{P}_{01}(\mathbf{I}-z\mathbf{P}_{x1})^{-1}z\mathbf{P}_{x0}\end{aligned}$$

$$\mathbf{A}_3^{\mathsf{C}}(z) = z\mathbf{P}_{000} + z\mathbf{P}_{001}(\mathbf{I} - z\mathbf{P}_{xx1})^{-1}z\mathbf{P}_{xx0}$$

$$\mathbf{A}_4^{\mathsf{C}}(z) = z(\mathbf{P}_{011} + \mathbf{P}_{101})(z^T\mathbf{P}^T + (\mathbf{I} - \mathbf{f}_3^{\mathsf{C}}(z))^{-1} - \mathbf{I})(z\mathbf{P}_{00} + z\mathbf{P}_{01}(\mathbf{I} - z\mathbf{P}_{x1})^{-1}z\mathbf{P}_{x0}),$$

where $\mathbf{f}_i^{\mathsf{C}}(z)$'s are given in (35). The average delay for the memoryless channel is computed as

$$\overline{\mathrm{D}}_{\mathsf{C\text{-}ARQ}} = (4\epsilon + k + \epsilon k + 2T\epsilon^2 + 3T\epsilon^3 - T\epsilon^4 + 4T\epsilon^5 + 2T\epsilon^6 - 4T\epsilon^7 + 4\epsilon^2 k - 3\epsilon^3 k$$
$$- \epsilon^4 k - 4\epsilon^5 k - 2\epsilon^6 k + 4\epsilon^7 k + 6\epsilon^2 + 5\epsilon^3 - 6\epsilon^4 - 4\epsilon^5 + 2\epsilon^6 + 1)/((1-\epsilon)(1+\epsilon)^2).$$

K. Proof of Proposition 12

The second derivative of $\mathbf{\Phi}_{\mathsf{D_{C\text{-}ARQ}}}(z)$ at $z = 1$ for Coded ARQ can be computed as

$$\mathbf{\Phi}_{\mathsf{D_{C\text{-}ARQ}}}{}''(1) = k(k-1)\mathbf{P}^k(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1) + \mathbf{A}_3^{\mathsf{C}}(1) + \mathbf{A}_4^{\mathsf{C}}(1))$$
$$+ k\mathbf{P}^k(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{f}_2^{\mathsf{C}}(1))'(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1) + \mathbf{A}_3^{\mathsf{C}}(1) + \mathbf{A}_4(1))$$
$$+ k\mathbf{P}^k(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}((\mathbf{A}_1^{\mathsf{C}}(1))'(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2(1) + \mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{f}_1^{\mathsf{C}}(1))'(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1)$$
$$+ \mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{A}_2^{\mathsf{C}}(1))' + (\mathbf{A}_3^{\mathsf{C}}(1))' + (\mathbf{A}_4^{\mathsf{C}}(1))')$$
$$+ k\mathbf{P}^k(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{f}_2^{\mathsf{C}}(1))'(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1) + \mathbf{A}_3^{\mathsf{C}}(1) + \mathbf{A}_4^{\mathsf{C}}(1))$$
$$+ \mathbf{P}^k(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{f}_2^{\mathsf{C}}(1))'(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{f}_2^{\mathsf{C}}(1))'(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1)$$
$$+ \mathbf{A}_3^{\mathsf{C}}(1) + \mathbf{A}_4^{\mathsf{C}}(1)) + \mathbf{P}^k(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{f}_2^{\mathsf{C}}(1))'(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1) + \mathbf{A}_3^{\mathsf{C}}(1) + \mathbf{A}_4^{\mathsf{C}}(1))$$
$$+ \mathbf{P}^k(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{f}_2^{\mathsf{C}}(1))''(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1) + \mathbf{A}_3^{\mathsf{C}}(1) + \mathbf{A}_4^{\mathsf{C}}(1))$$
$$+ \mathbf{P}^k(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{f}_2^{\mathsf{C}}(1))'(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{f}_2^{\mathsf{C}}(1))'(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1) + \mathbf{A}_3^{\mathsf{C}}(1)$$
$$+ \mathbf{A}_4^{\mathsf{C}}(1)) + \mathbf{P}^k(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{f}_2^{\mathsf{C}}(1))'(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}((\mathbf{A}_1^{\mathsf{C}}(1))'(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1)$$
$$+ \mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{f}_1^{\mathsf{C}}(1))'(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1) + \mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{A}_2^{\mathsf{C}}(1))' + (\mathbf{A}_3^{\mathsf{C}}(1))' + (\mathbf{A}_4^{\mathsf{C}}(1))')$$
$$+ k\mathbf{P}^k(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}((\mathbf{A}_1^{\mathsf{C}}(1))'(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1) + \mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{f}_1^{\mathsf{C}}(1))'(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1)$$
$$+ \mathbf{A}_1^{\mathsf{C}}(1)\mathbf{I}/(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))(\mathbf{A}_2^{\mathsf{C}}(1))' + (\mathbf{A}_3^{\mathsf{C}}(1))' + (\mathbf{A}_4^{\mathsf{C}}(1))')$$
$$+ \mathbf{P}^k(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}(\mathbf{f}_2^{\mathsf{C}}(1))'(\mathbf{I} - \mathbf{f}_2^{\mathsf{C}}(1))^{-1}((\mathbf{A}_1^{\mathsf{C}}(1))'(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1)$$
$$+ \mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{f}_1^{\mathsf{C}}(1))'(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1) + \mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I} - \mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{A}_2^{\mathsf{C}}(1))' + (\mathbf{A}_3^{\mathsf{C}}(1))' + (\mathbf{A}_4^{\mathsf{C}}(1))')$$

$$
\begin{aligned}
&+ \mathbf{P}^k(\mathbf{I}-\mathbf{f}_2^{\mathsf{C}}(1))^{-1}((\mathbf{A}_1^{\mathsf{C}}(1))''(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1) + (\mathbf{A}_1^{\mathsf{C}}(1))'(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{f}_1^{\mathsf{C}}(1))'(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1)\\
&+ (\mathbf{A}_1^{\mathsf{C}}(1))'(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{A}_2^{\mathsf{C}}(1))' + (\mathbf{A}_1^{\mathsf{C}}(1))'(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{f}_1^{\mathsf{C}}(1))'(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1)\\
&+ \mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{f}_1^{\mathsf{C}}(1))'(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{f}_1^{\mathsf{C}}(1))'(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1)\\
&+ \mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}})^{-1}(\mathbf{f}_1^{\mathsf{C}}(1))''(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1) + \mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{f}_1^{\mathsf{C}}(1))'(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1)\\
&+ \mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{f}_1^{\mathsf{C}}(1))'(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{f}_1^{\mathsf{C}}(1))'(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}\mathbf{A}_2^{\mathsf{C}}(1)\\
&+ \mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}})^{-1}(\mathbf{f}_1^{\mathsf{C}}(1))'(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{A}_2^{\mathsf{C}}(1))' + (\mathbf{A}_1^{\mathsf{C}}(1))'(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{A}_2^{\mathsf{C}}(1))'\\
&+ \mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{f}_1^{\mathsf{C}}(1))'(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{A}_2^{\mathsf{C}}(1))' + \mathbf{A}_1^{\mathsf{C}}(1)(\mathbf{I}-\mathbf{f}_1^{\mathsf{C}}(1))^{-1}(\mathbf{A}_2^{\mathsf{C}}(1))'' + (\mathbf{A}_3^{\mathsf{C}}(1))'' + (\mathbf{A}_4^{\mathsf{C}}(1))''),
\end{aligned}
$$

where $\mathbf{A}_j^{\mathsf{C}}(1)$, $j \in \{1,2,3,4\}$, are given in Appendix J, and $\mathbf{f}_i^{\mathsf{C}}(1)$'s for $i \in \{1,2,3\}$ are given in (35).

Using the MGF of delay, we can compute the variance of the delay for memoryless channels as

$$
\begin{aligned}
\sigma^2_{\mathrm{D}_{\textsf{C-ARQ}}} &= \frac{\epsilon^2 k^2}{(1+\epsilon)^4}(1+5\epsilon-6\epsilon^2-\epsilon^3-5\epsilon^4+10\epsilon^5+28\epsilon^6+20\epsilon^7+12\epsilon^8-16\epsilon^9-16\epsilon^{10})\\
&- \frac{2\epsilon^4 Tk}{(1-\epsilon)(1+\epsilon)^3}(2+\epsilon-5\epsilon^2+10\epsilon^3-16\epsilon^4+12\epsilon^5-12\epsilon^6-16\epsilon^7+16\epsilon^8)\\
&+ \frac{\epsilon^2 k}{(1-\epsilon)(1+\epsilon)^4}(5-16\epsilon-40\epsilon^2-52\epsilon^3+3\epsilon^4+60\epsilon^5+96\epsilon^6+32\epsilon^7-72\epsilon^8-24\epsilon^9+16\epsilon^{10})\\
&+ \frac{\epsilon^2 T^2}{(1-\epsilon)^2(1+\epsilon)^2}(2+\epsilon-8\epsilon^2+13\epsilon^3+7\epsilon^4-30\epsilon^5+20\epsilon^6+20\epsilon^7-52\epsilon^8+48\epsilon^9-16\epsilon^{10})\\
&+ \frac{\epsilon^2 T}{(1-\epsilon)^2(1+\epsilon)^3}(4+5\epsilon-21\epsilon^2-3\epsilon^3+27\epsilon^4-24\epsilon^5-52\epsilon^6+104\epsilon^7+8\epsilon^8-56\epsilon^9+16\epsilon^{10})\\
&+ \frac{\epsilon}{(1-\epsilon)^2(1+\epsilon)^4}(3+4\epsilon-4\epsilon^2-33\epsilon^3-29\epsilon^4+47\epsilon^5+100\epsilon^6-8\epsilon^7-72\epsilon^8+4\epsilon^9+16\epsilon^{10}-4\epsilon^{11}).
\end{aligned}
$$

Hence, as $\epsilon \to 0$, the variance of delay can be approximated as in (24).

References

[1] P. Popovski *et al.*, "Wireless access for ultra-reliable low-latency communication: Principles and building blocks," *IEEE Network*, vol. 32, no. 2, pp. 16–23, Mar. 2018.

[2] G. P. Fettweis, "The tactile internet: Applications and challenges," *IEEE Veh. Technol. Mag.*, vol. 9, no. 1, p. 6470, Mar. 2014.

[3] [Online]. Available: http://urllc2018.executiveindustryevents.com/Event/

[4] M. Bennis, M. Debbah, and H. V. Poor, "Ultra-reliable and low-latency wireless communication: Tail, risk and scale," *arXiv preprint arXiv:1801.01270*, Jan. 2018.

[5] X. Lin *et al.*, "The sky is not the limit: LTE for unmanned aerial vehicles," *IEEE Commun. Mag.*, vol. 56, pp. 204–10, Apr. 2018.

[6] A. Iera *et al.*, "The internet of things," *IEEE Wireless Commun.*, vol. 17, no. 6, pp. 8–9, Dec. 2010.

[7] "3GPP TSG RAN WG1 Meeting 87," Nov. 2016.

[8] "3GPP TS 23.725 Study on enhancement of URLLC supporting in 5GC," 3GPP, Tech. Rep., Mar. 2018.

[9] H. S. Dhillon, H. Huang, and H. Viswanathan, "Wide-area wireless communication challenges for the Internet of Things," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 168–174, Feb. 2017.

[10] J. G. Andrews *et al.*, "What will 5G be?" *IEEE Journ. on Sel. Areas in Comm.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[11] 3GPP, "Service requirements for the 5G system," 3rd Generation Partnership Project (3GPP), TS 22.261, Jun. 2017.

[12] S. R. Khosravirad and H. Viswanathan, "Analysis of feedback error in Automatic Repeat reQuest," *arXiv preprint arXiv:1710.00649*, Oct. 2017.

[13] J. Lu *et al.*, "A network coding based hybrid ARQ algorithm for wireless video broadcast," *Science China Information Sciences*, vol. 54, no. 6, pp. 1327–1332, 2011.

[14] "3GPP TS 36.212; evolved universal terrestrial radio access (E-UTRA); multiplexing and channel coding," Tech. Rep., Dec. 2016.

[15] M. Karzand and D. J. Leith, "Low delay random linear coding over a stream," in *Proc., IEEE Allerton*, Sep. 2014, pp. 521–528.

[16] J. Lieb, "Complete MDP convolutional codes," *arXiv preprint arXiv:1712.08767*, Dec. 2017.

[17] M. Luby *et al.*, "Practical loss-resilient codes," in *Proc., ACM Symp. Theory of Comput.*, New York, NY, USA, 1997, pp. 150–159.

[18] Y. Zaki *et al.*, "Adaptive congestion control for unpredictable cellular networks," *ACM SIGCOMM Computer Commun. Review*, vol. 45, no. 4, pp. 509–522, Aug. 2015.

[19] S. Zander and G. Armitage, "Minimally-intrusive frequent round trip time measurements using synthetic packet-pairs," in *Proc., IEEE Conf. Local Computer Netw.*, Oct. 2013, pp. 264–267.

[20] L. Keller, E. Drinea, and C. Fragouli, "Online broadcasting with network coding," in *in Proc. of NetCod*, Jan. 2008.

[21] S. Katti *et al.*, "XORs in the air: Practical wireless network coding," *ACM SIGCOMM Computer Commun. Review*, vol. 36, no. 4, pp. 243–254, Sep. 2006.

[22] J. K. Sundararajan, D. Shah, and M. Médard, "ARQ for network coding," in *Proc., IEEE ISIT*, Jul. 2008.

[23] C. Fragouli *et al.*, "On feedback for network coding," in *Proc., IEEE Annu. Conf. Inf. Sciences and Systems*, Mar. 2007.

[24] "3GPP TR 36.877; LTE device to device (D2D) proximity services (ProSe)," 3GPP, Tech. Rep., Mar. 2015.

[25] M. Tömösközi *et al.*, "On the delay characteristics for point-to-point links using random linear network coding with on-the-fly coding capabilities," in *Proc., European Wireless*, May 2014.

[26] G. Joshi, Y. Kochman, and G. W. Wornell, "On playback delay in streaming communication," in *Proc., IEEE ISIT*, Jul. 2012.

[27] E. Martinian, "Dynamic information and constraints in source and channel coding," Ph.D. dissertation, Cambridge, MA, USA, Sep. 2004.

[28] S. V. Hanly and D. N. C. Tse, "Multiaccess fading channels. ii. delay-limited capacities," *IEEE Trans. Inf. Theory*, vol. 44, no. 7, pp. 2816–2831, Nov. 1998.

[29] M. Sybis *et al.*, "Channel coding for ultra-reliable low-latency communication in 5G systems," in *Proc., IEEE VTC*, Sep. 2016.

[30] H. Al-Zubaidy, J. Liebeherr, and A. Burchard, "Network-layer performance analysis of multihop fading channels," *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 204–217, Feb. 2016.

[31] E. Bastug *et al.*, "Big data meets telcos: A proactive caching perspective," *J. Commun. and Netw.*, vol. 17, pp. 549–57, Dec. 2015.

[32] G. Pocovi *et al.*, "On the impact of multi-user traffic dynamics on low latency communications," in *Proc., Intl. Symp. Wireless Commun. Systems*, Sep. 2016, pp. 204–208.

[33] D. Malak, H. Huang, and J. G. Andrews, "Throughput maximization for delay-sensitive random access communication," *Trans. Wireless Commun.*, vol. 18, no. 1, pp. 709–23, Jan. 2019.

[34] A. Anand and G. de Veciana, “Resource allocation and HARQ optimization for URLLC traffic in 5G wireless networks,” *arXiv preprint arXiv:1804.09201*, Apr. 2018.

[35] L. Dai *et al.*, “Non-orthogonal multiple access for 5G: solutions, challenges, opportunities, and future research trends,” *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 74–81, Sep. 2015.

[36] H. Xing *et al.*, “Optimal throughput fairness tradeoffs for downlink non-orthogonal multiple access over fading channels,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 3556–3571, Jun. 2018.

[37] A. Eryilmaz, A. Ozdaglar, and M. Médard, “On delay performance gains from network coding,” in *Proc., IEEE Annu. Conf. Inf. Sciences and Systems*, Mar. 2006, pp. 864–870.

[38] D. Lun *et al.*, “An analysis of finite-memory random linear coding on packet streams,” in *Proc., IEEE WiOpt*, Apr. 2006.

[39] K. Ausavapattanakun and A. Nosratinia, “Analysis of selective-repeat ARQ via matrix signal-flow graphs,” *IEEE Trans. Commun.*, vol. 55, no. 1, pp. 198–204, Jan. 2007.

[40] E. N. Gilbert, “Capacity of a burst-noise channel,” *Bell System Technical Journal*, vol. 39, pp. 1253–1265, 1960.

[41] L. E. Baum and T. Petrie, “Statistical inference for probabilistic functions of finite state markov chains,” *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. 1554–1563, Dec. 1966.

[42] D. Malak, M. Médard, and E. Yeh, “ARQ with cumulative feedback to compensate for burst errors,” in *Proc., IEEE Globecom*, Dec. 2018.

[43] M. Luby *et al.*, “Forward error correction (FEC) building block,” 2002.

[44] S. J. Mason and H. J. Zimmermann, *Electronic, Circuits, Signals, and Systems*. New York: Wiley, 1960.

[45] R. A. Howard, *Dynamic Probabilistic Systems*. Courier Corporation, 1971.

[46] D. L. Lu and J. F. Chang, “Analysis of ARQ protocols via signal flow graphs,” *IEEE Trans. Commun.*, vol. 37, no. 3, pp. 245–51, Mar. 1989.

[47] ——, “Performance of ARQ protocols in nonindependent channel errors,” *IEEE Trans. Commun.*, vol. 41, pp. 721–30, May 1993.

[48] Y. J. Cho and C. K. Un, “Performance analysis of ARQ error controls under Markovian block error pattern,” *IEEE Trans. Commun.*, vol. 42, no. 2-4, pp. 2051–2061, Feb. - Apr. 1994.

[49] W.-K. Chen, “The use of matrix signal-flow graph in state-space formulation of feedback theory,” in *Proc., IEEE Int. Symp. Circuits and Systems*, Jun. 1991, pp. 1005–1008.

[50] F. Pukelsheim, “The three sigma rule,” *American Statistician*, vol. 48, p. 8891, 1994.

[51] E. Dahlman *et al.*, *3G Evolution – HSPA and LTE for Mobile Broadband (2 ed.). Academic Press*. Elsevier Science Publishing Co Inc, 2008.

[52] K. R. Sachin *et al.*, “A review of Hybrid ARQ in 4G LTE,” *IJARIIE*, vol. 1, no. 3, pp. 160–165, 2015.

[53] W. Zeng *et al.*, “Joint coding and scheduling optimization in wireless systems with varying delay sensitivities.” in *Proc., Annu. IEEE Commun. Society Conf. Sensor, Mesh and Ad Hoc Commun. and Networks (SECON)*, Jun. 2012, pp. 416–424.

[54] T. Ho *et al.*, “A random linear network coding approach to multicast,” *IEEE Trans. Inf. Theory*, vol. 52, pp. 4413–30, Oct. 2006.