ARQ with Cumulative Feedback to Compensate for Burst Errors

Derya Malak and Muriel Médard
RLE, Massachusetts Institute of Technology
Cambridge, MA 02139 USA
Email: {deryam, medard}@mit.edu

Edmund M. Yeh
ECE Dept., Northeastern University
Boston, MA 02115, USA
Email: eyeh@ece.neu.edu

Abstract—We propose a cumulative feedback-based ARQ (CF ARQ) protocol for a sliding window of size 2 over packet erasure channels with unreliable feedback. We exploit a matrix signal-flow graph approach to analyze probability-generating functions of transmission and delay times. Contrasting its performance with that of the uncoded baseline scheme for ARQ, developed by Ausavapattanakun and Nosratinia, we demonstrate that CF ARQ can provide significantly less average delay under bursty feedback, and gains up to about 20% in terms of throughput. We also outline the benefits of CF ARQ under burst errors and asymmetric channel conditions. The protocol is more predictable across statistics, hence is more stable. This can help design robust systems when feedback is unreliable. 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.

I. Introduction

Ultra reliability and low latency in 5G are key factors for many applications ranging from industrial automation, tactile Internet, remote healthcare, public safety, to mission-critical communications such as autonomous driving and wearable computing devices [1]–[3]. 5G will need to support a round-trip time (RTT) of about 1 millisecond, along with necessary overheads for resource allocation and access in 5G networks. Such severe latency constraints introduce a plethora of challenges in terms of the protocol stack design, control/user plane, and the core network [4].

Repetition of a packet over non-deterministic channel conditions, and the use of forward error correction (FEC) codes help repair the loss of the packets. Feedback packets are used to request FEC retransmission for increasing the reliability in packet delivery. The role of feedback is to increase data channel efficiency by limiting the repetitions. However, coding and feedback have been difficult to blend.

Reliable communication over a packet erasure channel can be achieved using Automatic Repeat reQuest (ARQ), when there is full feedback [5]. This simple scheme achieves 100% throughput, in-order delivery and the lowest possible packet delay, and it is composable across links. However, when the network is lossy, i.e., with no idealized feedback, link-by-link ARQ cannot achieve the capacity of a general network.

In the literature, the achievable rate has been optimized using acknowledgments and coding, under the condition that each received packet is either useless or can be immediately decoded by the destination [6]. Feedback and coding over a broadcast erasure channel have been combined in [7] to optimize decoding delay when perfect feedback is available from the receivers. An extension of ARQ for coded networks has been proposed in [5] 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 [8]. The proposed scheme in [5] is robust to delayed or imperfect feedback. None of these examples jointly investigate the delay and throughput when the feedback is imperfect.

For schemes requiring feedback, it is generally assumed that feedback is lossless (perfect) and instantaneous (delay-free) [5], [8], [9]. Inevitable feedback channel impairments may cause unreliability in packet delivery. Burst errors might occur, which can impede the stability. The situation becomes worse under round-trip time (RTT) fluctuations along with the delayed feedback. To the best of our knowledge, the effect of unreliable feedback has not been captured before.

In this paper, we investigate the effect of unreliable feedback in packet erasure channels. Erasure errors can occur in both the forward and reverse channels. However, an acknowledgment (ACK) cannot be decoded as a negative acknowledgment (NACK), and vice versa. Building on the uncoded baseline scheme proposed in [10], we propose a SR ARQ scheme under a cumulative feedback-based ARQ (CF ARQ) scheme in order to investigate the role of feedback. We investigate how much we can gain with cumulative feedback and how to compensate the forward errors with cumulative feedback. Contrasting the throughput and delay performance of CF ARQ with the uncoded ARQ in [10], we demonstrate that with a sliding window of size 2, CF ARQ can provide gains up to 18% in terms of throughput. Cumulative feedback also has benefits under burst errors or high erasure rates.

II. Channel Model

We have a point-to-point channel model consisting of a sender and a receiver. In the forward link, the sender attempts to transmit a packet to the receiver, and upon the successful reception of the packet, in the reverse link, the receiver acknowledges the sender by transmitting a feedback. We use

a Gilbert-Elliott (GE) model[1] [12], which is a special case of hidden Markov models (HMMs), both for the forward and reverse channels. The status of a transmission at time t is a random variable taking values in $\mathcal{X} = \{0, 1\}$, where 0 denotes an error-free packet, and 1 means the packet is erroneous. This binary-state Markov process S_t, with probability transition matrix $\mathbf{P}$, has states G (good) and B (bad), i.e. $\mathcal{S} = \{G, B\}$, with $\boldsymbol{\epsilon} = [\epsilon_G, \epsilon_B]$ where ϵ_G and ϵ_B are the probabilities of transmitting a packet in error in the respective states. The GE channel X_t, driven by S_t, is characterized by $\{\mathcal{S}, \mathcal{X}, \mathbf{P}, \boldsymbol{\epsilon}\}$.

The channel state information is not available at the transmitter and the receiver. Hence, the transmitter does not know the status of a transmission (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 taking values in $\mathcal{X} = \{0, 1\}$. Similarly, the receiver does not know the status of the reverse link, but it observes the status of a transmission at time t, which is a Bernoulli random variable taking values in $\mathcal{X} = \{0, 1\}$. The transmitter and receiver do not observe the process $\mathcal{X}^{(c)}$. However, for the GE channel, given the channel state at time $t - 1$, the joint probabilities of channel state and observation at time t can be computed using the state-transition probabilities. For a GE channel, the state-transition matrix is

$$\mathbf{P} = \begin{bmatrix} 1-q & q \\ r & 1-r \end{bmatrix}, \tag{1}$$

where the first and second rows correspond to states G and B. The erasure rate is $\epsilon = \pi \boldsymbol{\epsilon}^{\mathsf{T}}$, where $\pi = [\pi_G, \ \pi_B]$ is the stationary vector of $\mathbf{P}$, which is found by solving $\pi \mathbf{P} = \pi$ and $\pi \mathbf{1} = 1$. Note that $1/r$ represents the average error burst. Hence, 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 given as

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

which can be collected into a matrix of transition probabilities $\mathbf{P}_1 = \mathbf{P} \cdot \mathrm{diag}\{\boldsymbol{\epsilon}\}$. Similarly, define $\mathbf{P}_0 = \mathbf{P} \cdot \mathrm{diag}\{\mathbf{1} - \boldsymbol{\epsilon}\}$. The entries in matrices $\mathbf{P}_0$ and $\mathbf{P}_1$ are state-transition probabilities when viewed jointly with the conditional channel observations [10]. Hence, the HMM is characterized by $\{\mathcal{S}, \mathcal{X}, \mathbf{P}_0, \mathbf{P}_1\}$.

In practice, data packets and acknowledgments typically have different lengths and different coding levels. Therefore, the erasure rates $\boldsymbol{\epsilon}$ and the parameters r and q of the forward and reverse channels are not necessarily the same, which is accounted in our model. Denote by $\mathbf{P}^{(f)}$ and $\mathbf{P}^{(r)}$ the state-transition matrices for the forward and reverse channels, respectively. The forward link $\{\mathcal{S}^{(f)}, \mathcal{X}^{(f)}, \mathbf{P}_0^{(f)}, \mathbf{P}_1^{(f)}\}$ and the reverse link $\{\mathcal{S}^{(r)}, \mathcal{X}^{(r)}, \mathbf{P}_0^{(r)}, \mathbf{P}_1^{(r)}\}$ are mutually independent.

For the GE channel, the probability matrices for the forward and reverse channels $\mathbf{P}_0^{(f)}$ and $\mathbf{P}_1^{(r)}$ are given as

$$\mathbf{P}_0^{(f)} = \mathbf{P}^{(f)} \cdot \mathrm{diag}\{\mathbf{1} - \boldsymbol{\epsilon}^{(f)}\} = \begin{bmatrix} \bar{q}^{(f)}\bar{\epsilon}_G^{(f)} & q^{(f)}\bar{\epsilon}_B^{(f)} \\ r^{(f)}\bar{\epsilon}_G^{(f)} & \bar{r}^{(f)}\bar{\epsilon}_B^{(f)} \end{bmatrix},$$

[1]A general finite-state Markov model can be used to represent a physical channel with fading. The received signal-to-noise ratio can be partitioned into a finite number of states, corresponding to different channel qualities [11].

$$\mathbf{P}_1^{(r)} = \mathbf{P}^{(r)} \cdot \mathrm{diag}\{\boldsymbol{\epsilon}^{(r)}\} = \begin{bmatrix} \bar{q}^{(r)}\epsilon_G^{(r)} & q^{(r)}\epsilon_B^{(r)} \\ r^{(r)}\epsilon_G^{(r)} & \bar{r}^{(r)}\epsilon_B^{(r)} \end{bmatrix}$$

using the shorthand notation $\bar{q} = 1 - q$, $\bar{r} = 1 - r$, $\bar{\epsilon}_G = 1 - \epsilon_G$ and $\bar{\epsilon}_B = 1 - \epsilon_B$. We can similarly compute $\mathbf{P}_1^{(f)}$ and $\mathbf{P}_0^{(r)}$.

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. For example, $X_t^{(c)} = 10$ 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 $(p_{ij}^{(f)}\epsilon_j^{(f)}) \cdot (p_{km}^{(r)}\epsilon_m^{(r)})$. In compact notation, we have $\mathbf{P}_{ij}^{(c)} = \mathbf{P}_i^{(f)} \otimes \mathbf{P}_j^{(r)}$ for $X_t^{(c)} = ij$, where $\otimes$ is the Kronecker product of matrices and $i, j = 0, 1$. For the GE channel, the combined observation probabilities are given by the following 4×4 matrices: $\mathbf{P}_{00}^{(c)} = \mathbf{P}_0^{(f)} \otimes \mathbf{P}_0^{(r)}$, $\mathbf{P}_{01}^{(c)} = \mathbf{P}_0^{(f)} \otimes \mathbf{P}_1^{(r)}$, $\mathbf{P}_{10}^{(c)} = \mathbf{P}_1^{(f)} \otimes \mathbf{P}_0^{(r)}$, and $\mathbf{P}_{11}^{(c)} = \mathbf{P}_1^{(f)} \otimes \mathbf{P}_1^{(r)}$. The combined state-transition matrix for the GE channel, i.e., $\mathbf{P}^{(c)}$, is a 4×4 matrix that is given by the Kronecker product of $\mathbf{P}^{(f)}$ and $\mathbf{P}^{(r)}$, i.e. $\mathbf{P}^{(c)} = \mathbf{P}^{(f)} \otimes \mathbf{P}^{(r)}$.

In the rest of the paper, we will drop the superscript $^{(c)}$ and denote the 4×4 observation probability matrices by $\mathbf{P}_{00}$, $\mathbf{P}_{01}$, $\mathbf{P}_{10}$ and $\mathbf{P}_{11}$. We also let $\mathbf{P}_{0x} = \mathbf{P}_{00} + \mathbf{P}_{01}$ and $\mathbf{P}_{1x} = \mathbf{P}_{10} + \mathbf{P}_{11}$ be the probability matrices of success and error in 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 matrices of success and error in the reverse channel, respectively. Furthermore, we let the matrices $\mathbf{P}, \mathbf{P}_0, \mathbf{P}_1$ denote the 4×4 composite channel matrices.

III. ARQ WITH CUMULATIVE FEEDBACK

We propose a cumulative feedback-based ARQ (CF ARQ) scheme with coding for data transmission. It is an extension of the slotted SR ARQ, which 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 final delivery. Assume that all packets are available at the sender prior to transmission, the receiver does not have buffer overflows, and there is a synchronous transmission from the sender to the receiver.

We consider minimum coding, i.e., with a sliding window of size $M = 2$. The protocol can easily be generalized to packet streams with $M > 2$, which is out of the scope of the current paper. This scheme differs from the uncoded ARQ in [10] in the sense that the transmitted packet stream is MDS coded, and the feedback 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 freedom (DoF) required at the receiver is $N = 2$. We do not assume in-order packet delivery. Hence, the transmitted stream will be successfully decoded when both of the coded packets are successfully received and acknowledged by the receiver.
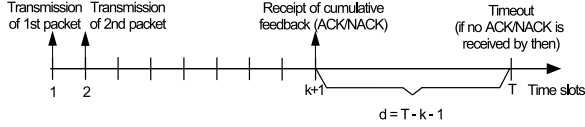
Fig. 1: CF ARQ protocol description.

Fig. 2: Matrix-flow graph for delay of CF ARQ in imperfect feedback.

The feedback, i.e. ACK and NACK messages sent by the receiver indicating if it has correctly received a data packet, acknowledges all correctly received packets, and is cumulative for $M = 2$ coded packets. After the start of transmission (I), it takes $k - 1$ time slots between the transmission of the second packet and receipt of its feedback. Therefore, the round-trip time (RTT) of CF ARQ is $\mathrm{RTT} = k + M - 1 = k + 1$ slots. If the feedback was not cumulative, i.e., the first feedback was received $k - 1$ slots after the transmission of the first packet, then the RTT would have been k slots. A timeout mechanism is used at the transmitter to achieve reliable data transmission. When a packet stream is (re)transmitted, the timeout is set to T that is greater than RTT. If the sender does not receive an acknowledgment before the timeout, it retransmits the packets until it receives an acknowledgment. Hence, we do not have an upper bound on the maximum number of retransmissions.

The ACK/NACK sent in each slot. The packet whose ACK is lost will be acknowledged by subsequent ACKs/NACKs. If the succeeding ACKs/NACKs are successfully received before timer expiration, the packet will not be retransmitted. If the timeout expires and no ACK is received, the packet will be retransmitted. When a packet is lost and its NACK is received, the packet will be retransmitted immediately. If the NACK is also lost, the packet will be retransmitted after the timer expires. The transmission protocol is illustrated in Fig. 1.

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 transmitted or only one of the packets has to be retransmitted.

Hidden Markov model (HMM) is a statistical Markov process with unobserved states. Although the state is not directly observed, the output dependent on the state can be observed. Thus, under unreliable channel conditions, the analysis of ARQ protocol is possible using HMMs. The analysis of finite-state HMMs can be streamlined using flow graphs. Scalar-flow graphs have been used to find the probability-generating functions (PGFs) of transmission and delay times [13]–[15].

HMMs can be analyzed by labeling the branches of scalar-flow graphs with observation probability matrices. The nodes of the flow graphs correspond to the states of the transmitter. The input node (I) represe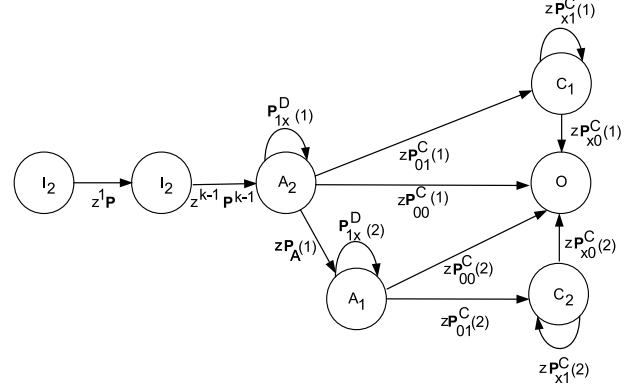nts the start of transmission, and the output node (O) represents correct reception of acknowledgment. Other nodes represent intermediate states. Upon the start of transmission, the transmitter goes from one state to the other. A state transition is accompanied with a certain value for the random variable X, and a probability p, which together appear in the branch gain 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 is equivalent to $\mathbb{E}[z^X]$, the PGF for X. Flow graphs with matrix branch transmissions and vector node values are called matrix signal-flow graphs (MSFGs) [10]. The matrix gain of the graph is calculated using the basic equivalences known as parallel, series, and self-loop. The matrix-generating function (MGF) $\mathbf{\Phi}(z)$ gives the input-output relationship for the matrix-flow graph. Then, the PGF is calculated by pre- and postmultiplications of row and column vectors, respectively.

The HMM for delay analysis of CF ARQ is shown in Fig. 2. The states I and O are the input and output nodes, and nodes A_1, A_2, C_1, C_2 represent the hidden states. 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

$$\mathbf{P}_{1x}^{\mathrm{D}}(1) = z^{\mathrm{RTT}}\mathbf{P}^{\mathrm{RTT}}(z\mathbf{P}_{10}^{\mathrm{C}}(1) + z\mathbf{P}_{11}^{\mathrm{C}}(1)z^d\mathbf{P}^d),$$

where $d = T - \mathrm{RTT}$ is the residual time for timer expiration upon transmission. Upon the reception of the first feedback, the transition probability matrix $\mathbf{P}_{10}^{\mathrm{C}}(1)$ for the transmission of $M = 2$ packets is given by

$$\mathbf{P}_{10}^{\mathrm{C}}(1) = \mathbf{P}_{10}\mathbf{P}_{10} + \mathbf{P}_{10}\mathbf{P}_{01} + \mathbf{P}_{01}\mathbf{P}_{10},$$

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 $\mathbf{P}_{11}^C(1)$ is given by

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

which models the erroneous NACK feedback. It combines the different cases such that the forward link was bad for both packets and the reverse link (CF) 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 ϵ_{CF} for CF ARQ with $M = 2$ packets, and the erasure rate ϵ of the uncoded ARQ in [10] is computed as $\epsilon_{CF} = \sqrt{\epsilon^4 + 2\epsilon^3(1-\epsilon)}$. Hence, $\epsilon_{CF}(1) \geq \epsilon^2$.

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

$$\mathbf{P}_A(1) = \mathbf{P}_{00}\mathbf{P}_{10} + \mathbf{P}_{10}\mathbf{P}_{00}$$

denotes 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. The packet retransmission at A_1 is modeled by the self-loop, where

$$\mathbf{P}_{1x}^D(2) = (z\mathbf{P})^{\text{RTT}-1}(z\mathbf{P}_{10}^C(2) + z\mathbf{P}_{11}^C(2)z^{d+1}\mathbf{P}^{d+1}),$$

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

- **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 $\mathbf{P}_{00}^C(1) = \mathbf{P}_{00}\mathbf{P}_{00}$), the system transits to state O.

- **Transition to state C_1.** If $N = 2$ DoF's are received, but the feedback is an erroneous ACK (with probability $\mathbf{P}_{01}^C(1) = \mathbf{P}_{01}\mathbf{P}_{01} + \mathbf{P}_{01}\mathbf{P}_{00} + \mathbf{P}_{00}\mathbf{P}_{01}$), then the system transits to C_1, where the sender waits till it receives an error-free ACK/NACK, modeled by the self-loop at C_1.

- **Transition to state C_2.** 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 $\mathbf{P}_{01}^C(2)$), then the system transits to C_2, where the sender waits till it receives an error-free ACK/NACK.

Given the transition probabilities, the success and error probability matrices in the reverse channel for CF ARQ are:

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

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

The transmission time (τ) is the number of frames being transmitted per a successful frame, and the delay time (D) is the time from when a frame is first transmitted to when its ACK is received. Under the given model, both τ and D are random variables with positive integer outcomes. The matrix gain of the graph in Fig. 2 can be calculated. Using the PGF, the average values for τ and D can be calculated and the throughput is the reciprocal of τ. We next compute the MGF of the transmission time for $M = 2$ packets.

Proposition 1. *For CF ARQ, the MGF of τ is given by*

$$\mathbf{\Phi}_\tau(z) = z\mathbf{P}^k\Big[(\mathbf{I} - \mathbf{P}_{1x}^T(1))^{-1}\mathbf{A}_1(z)$$
$$+ \mathbf{P}_A(1)\prod\nolimits_{i=1}^2(\mathbf{I} - \mathbf{P}_{1x}^T(i))^{-1}\mathbf{A}_2(z)\Big], \qquad (2)$$

where

$$\mathbf{P}_{1x}^T(1) = z(\mathbf{P}_{10}^C(1) + \mathbf{P}_{11}^C(1)(\mathbf{P}^2)^d)\mathbf{P}^{k+1}$$
$$\mathbf{P}_{1x}^T(2) = z(\mathbf{P}_{10}^C(2) + \mathbf{P}_{11}^C(2)\mathbf{P}^{d-1})\mathbf{P}^k.$$

and the matrix $\mathbf{A}_n(z)$ for $n = \{1, 2\}$ that gives the gain of the transition from the state $A_{2-(n-1)}$ can be computed as

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

where $d_n = d - (n - 1)$.

The PGF of τ of CF ARQ for $M = 2$ packets is computed as $\Phi_\tau(z) = \pi_I\mathbf{\Phi}_\tau(z)\mathbf{1}/(\pi_I\mathbf{1})$ using the MGF $\mathbf{\Phi}_\tau(z)$ in (2), where $\mathbf{1}$ is a column vector of ones, and $\pi_I = \pi\mathbf{P}_0$ is the probability vector of state I. The throughput η is the reciprocal of the derivative of $\Phi_\tau(z)$ at $z = 1$, i.e., $\eta = 1/\Phi_\tau'(1)$.

We next compute the MGF of the delay for $M = 2$ packets.

Proposition 2. *For CF ARQ, the MGF of the delay is*

$$\mathbf{\Phi}_D(z) = z^k\mathbf{P}^k\Big[(\mathbf{I} - \mathbf{P}_{1x}^D(1))^{-1}\mathbf{B}_1(z)$$
$$+ z\mathbf{P}_A(1)\prod\nolimits_{i=1}^2(\mathbf{I} - \mathbf{P}_{1x}^D(i))^{-1}\mathbf{B}_2(z)\Big], \qquad (3)$$

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

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

The PGF of delay of CF ARQ for $M = 2$ coded packets can be computed as $\Phi_D(z) = \pi_I\mathbf{\Phi}_D(z)\mathbf{1}/(\pi_I\mathbf{1})$ using the MGF $\mathbf{\Phi}_D(z)$ in (3). Finally, the average delay will be the derivative of $\Phi_D(z)$ at $z = 1$, i.e., $\bar{D} = \Phi_D'(1)$.

In Sect. IV, we numerically evaluate the throughput and average delay for the GE channel model detailed in Sect. II and using the generating functions derived in Sect. III.

IV. NUMERICAL RESULTS

We numerically investigate the throughput η and average delay $\bar{D}$ of the point-to-point GE channel. Our objective is to understand the impacts of feedback and cumulative feedback (CF) under less reliable and bursty channel conditions.

First assume that the forward and reverse channels have the same erasure rates, i.e., $\epsilon^{(f)} = \epsilon^{(r)} = \epsilon$. In Fig. 3, we fix the burst rate of the forward channel, i.e., $r^{(f)}$, and vary the burst rate of the reverse channel, i.e., $r^{(r)}$, and vice versa. As timeout T increases, both the throughput η and the average delay $\bar{D}$
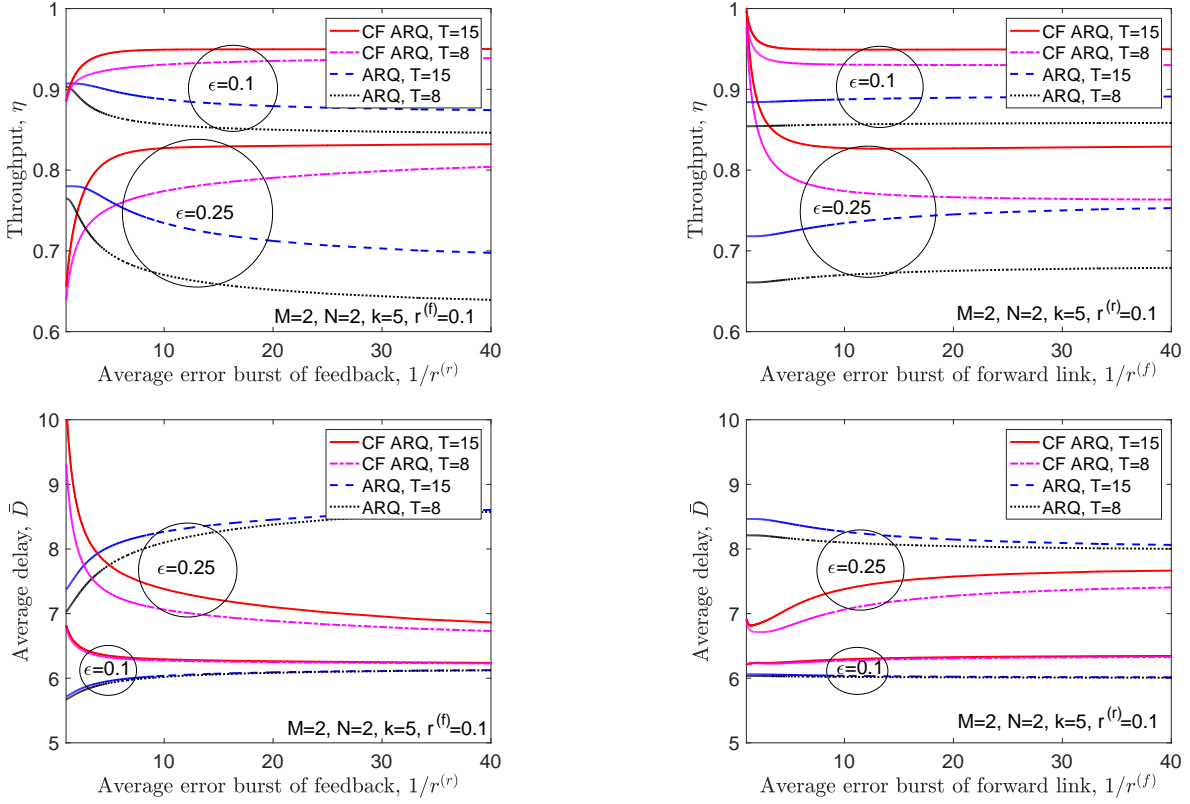
Fig. 3: Throughput and average delay under different burst models given fixed erasure rates: $\epsilon^{(f)} = \epsilon^{(r)} = \epsilon$.

increase. As ϵ increases, it is clear that η gets lower and $\bar{D}$ increases both for uncoded ARQ and CF ARQ. Sensitivity of η to timeout T also increases under burst errors (low r). If the feedback erasures are bursty, η decreases with feedback delay. We also observe that η of CF ARQ is higher than η of uncoded ARQ. The difference becomes significant when the feedback is bursty and ϵ is high. When ϵ is small, since the RTTs of CF ARQ and uncoded ARQ are $k + 1$ and k, respectively, $\bar{D}$ of CF ARQ is higher. However, $\bar{D}$ of CF ARQ is smaller when the feedback is bursty and ϵ is high. Hence, CF ARQ is more robust to burst errors in the feedback. When there is perfect feedback with $\epsilon^{(r)} = 0$, uncoded ARQ has lower $\bar{D}$ and higher η than CF ARQ. Throughput and delay performance of both schemes degrade as the feedback loss $\epsilon^{(r)}$ increases. However, CF ARQ outperforms uncoded ARQ under feedback loss.

In Fig. 4, we investigate the role of asymmetry such that either the forward channel is more robust to erasures, i.e., $\epsilon^{(f)} = 0.1$ and $\epsilon^{(r)} = 0.5$, or vice versa. We keep $r^{(f)}$ fixed and increase $r^{(r)}$. When the forward channel is more robust to erasures, η is higher both for uncoded ARQ and CF ARQ, and $\bar{D}$ is significantly less compared to the case when the reverse channel is more robust. In this case, CF ARQ provides significantly better throughput than uncoded ARQ. Thus, even if the forward and reverse channels are asymmetric, CF ARQ performs better than uncoded ARQ under bursty feedback.

From Figs. 3 and 4, we see that erasures of forward channel

$\epsilon^{(f)}$ scale the throughput, and feedback erasures $\epsilon^{(r)}$ change the shape of the throughput. Hence, forward erasures $\epsilon^{(f)}$ significantly degrade the throughput of both uncoded ARQ and CF ARQ, and dominate the performance of throughput. Still, CF ARQ throughput gap from uncoded ARQ is higher when $\epsilon^{(r)}$ is high. In terms of delay, CF ARQ is more stable than uncoded ARQ when $\epsilon^{(f)}$ increases, and less stable when $\epsilon^{(r)}$ increases. Still, CF ARQ is more stable when $\epsilon^{(r)}$ is high.

We next investigate the robustness of CF ARQ to forward erasures. Letting $\epsilon^{(r)} = 0.1$, we illustrate η and $\bar{D}$ of uncoded ARQ and CF ARQ in Fig. 5 for different sets of burst rates. From the plots, we see that CF ARQ provides a higher η, and even more forward erasures can be compensated (up to $\Delta\epsilon^{(f)} = 0.12$) with CF for $M = 2$ packets if the feedback channel is more bursty without sacrificing $\bar{D}$. CF ARQ can provide a gain of 18% in terms of throughput.

Uncoded ARQ is very sensitive to error bursts. The higher the burst rate, the lower its throughput is and the higher its delay is. To compensate for the forward erasures, CF can be used, which can provide significantly less delay, and better throughput. Similarly, when the feedback erasures dominate, performance of CF is much better in terms of throughput, and CF can provide reductions in delay under bursty feedback.

V. CONCLUSIONS

We proposed a cumulative feedback-based ARQ with a sliding window of size 2, and computed the MGFs of transmission
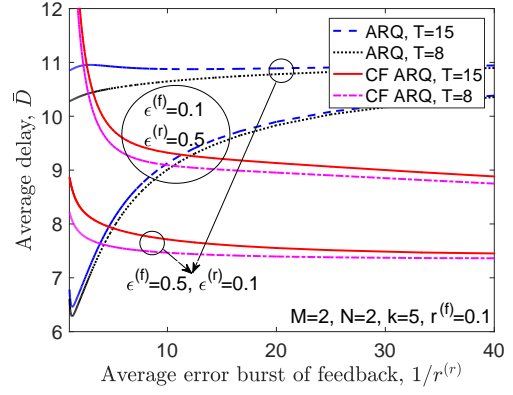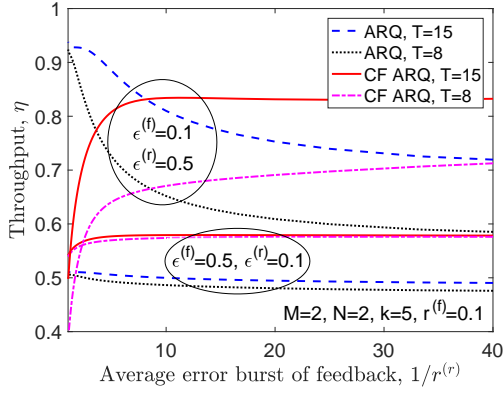
Fig. 4: Throughput and average delay given $r^{(f)} = 0.3$ and different erasure rates: $\epsilon^{(f)} \neq \epsilon^{(r)}$.
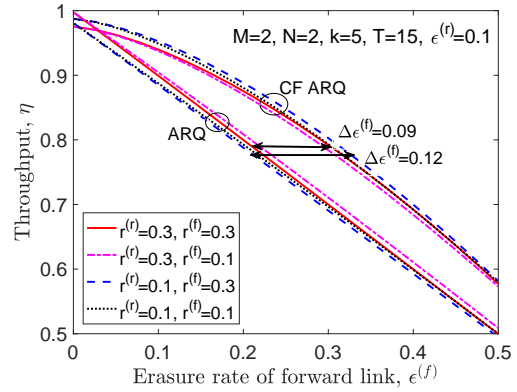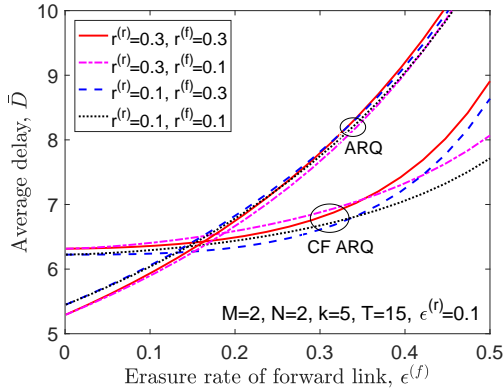

Fig. 5: Average delay and throughput as function of $\epsilon^{(f)}$, under different burst rates.

and delay times. Contrasting its performance with uncoded ARQ, we demonstrated its robustness under burst errors. The following insights should enable more robust design for packet erasure channels with imperfect and bursty feedback:

- At high erasures, CF ARQ provides considerably low average delay and high throughput than uncoded ARQ.
- CF ARQ has benefits under burst errors or higher erasure rates in the reverse channel. It is more predictable across statistics, hence is more stable. This can help design robust systems when feedback is unreliable.

Incorporating FEC, the transmission rate can be adaptively adjusted with the cumulative feedback for multiple coded packets. Extensions hence include the study of different coded schemes, and the throughput achievable with coding. While the analysis is prohibitively complex for larger window sizes with excessive number of hidden states, the technique can easily be evaluated for general window sizes using a network simulator. This can help understand the scalings between the window size and system parameters. This will pave the way for protocol design for 5G with desirable throughput-delay tradeoffs.

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

[4] 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.

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

[6] S. Katti *et al.*, "XORs in the air: Practical wireless network coding," in *Proc., Sigcomm*, 2006.

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

[8] C. Fragouli, D. Lun, M. Médard, and P. Pakzad, "On feedback for network coding," in *Proc., IEEE Annual Conference on Information Sciences and Systems*, Mar. 2007, pp. 248–252.

[9] P. Pakzad, C. Fragouli, and A. Shokrollahi, "Coding schemes for line networks," in *Proc., IEEE ISIT*, 2005.

[10] 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.

[11] Q. Zhang and S. A. Kassam, "Finite-state Markov model for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 47, no. 11, Nov. 1999.

[12] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell System Technical Journal*, vol. 42, pp. 1977–1997, 1963.

[13] D. L. Lu and J. F. Chang, "Performance of ARQ protocols in nonindependent channel errors," *IEEE Trans. Commun.*, vol. 41, no. 5, pp. 721–730, May 1993.

[14] 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.

[15] D.-L. Lu and J.-F. Chang, "Analysis of ARQ protocols via signal flow graphs," *IEEE Trans. Commun.*, vol. 37, no. 3, pp. 245–251, Mar. 1989.