Beating Burnashev in delay with noisy feedback

Stark C. Draper and Anant Sahai

***Abstract*— We show how to use a noisy feedback link to yield high-reliability streaming data communications. We demonstrate a strategy that can maintain the best known reliability function (error exponent) in delay, even when the feedback link is noisy. Only the capacity of the feedback channel need be sufficiently large for this result to hold. More detailed characterization of the feedback link, e.g., its error exponent, is not required. We discuss the architectural implications of our result, which are different from those drawn from block-coding paradigms.**

I. Introduction

Many communication applications such as control-over-networks or streaming media have stringent delay requirements. These application-layer demands are often in conflict with physical-layer requirements for long block-lengths needed to ensure communication reliability. It has long been known that feedback can help alleviate these tensions and make the trade-off between error probability and delay much more manageable [7] [6] [9] [12]. These studies fall under the rubric of error exponents. Previous studies have focused on noiseless feedback where the transmitter can "look over the shoulder of the receiver." In this paper we show how these improved trade-offs can often be maintained even when there is noise on the feedback link.

In Fig. 1 we diagram the end-to-end system issues that motivate this work. At a high level, data is produced by the application layer at the source, bits are packetized into messages that are encoded into codewords, these packets are enqueued, and then transmitted over the channel. Once decoded the data is consumed at the destination. In all there are three sources of delay: packetization (waiting for enough data to form a message), queuing (caused by a back-up in the transmission process), and service (the transmission time).

In this paper we focus on the service time delay from transmitter to receiver. In regimes where the best trade-off between reliability and delay is achieved, service time dominates the end-to-end delay. We review results for block and variable-length coding before focusing in on streaming data. Streaming data systems are designed to communicate to the destination a sequence of packets that can be realized in real time at the source. We develop a coding strategy that is appropriate for streaming data and show that its reliability function is far larger than the Burnashev exponent, which is the largest exponent achievable by variable-length schemes. We then show how the same error exponent can be maintained as long as the capacity of the feedback link is larger than the exponent.

S. Draper and A. Sahai are with the Dept. of Electrical Engineering and Computer Science, University of California, Berkeley, 94720. {sdraper, sahai}@eecs.berkeley.edu

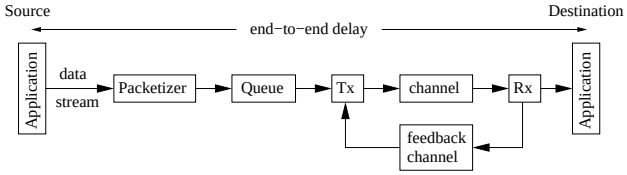

Fig. 1. End-to-end delay is a fundamental metric of system performance for streaming data systems.

II. Preliminaries

A. Fixed-length block coding without (and with) feedback

We start by summarizing results for block coding without feedback. When feedback is not available, the duration of transmission must be fixed to ensure that encoder and decoder stay synchronized. This is the familiar coding paradigm of Shannon [13]. Let W be a stationary discrete memoryless channel (DMC) with input alphabet $\mathcal{X}$ and output alphabet $\mathcal{Y}$.

Definition 1: A rate-R fixed-length-N block encoder-decoder pair $(\mathcal{E}_N, \mathcal{D}_N)$ is a pair of maps

$$\mathcal{E}_N : \mathcal{M} = \{1, 2, \ldots, 2^{NR}\} \to \mathcal{X}^N, \tag{1}$$

$$\mathcal{D}_N : \mathcal{Y}^N \to \hat{\mathcal{M}} = \{1, 2, \ldots, 2^{NR}\}. \tag{2}$$

An error exponent E is said to be achievable at rate R if there exists a family of coding strategies (of increasing length N) such that

$$\lim_{\epsilon \to 0} -\frac{\log \epsilon}{N} \geq E, \tag{3}$$

where $\epsilon = \Pr[\hat{m} \neq m]$ is the probability that the message estimate $\hat{m} \in \hat{\mathcal{M}}$ is incorrect at the decoding time N and where $m \in \mathcal{M}$.

The error exponent of this class of codes is upper-bounded by the sphere-packing bound:

$$E_{sp}(R) = \max_{P} \min_{V:I(P,V)\leq R} D(V\|W|P), \qquad (4)$$

where P is the input distribution and V characterizes the channel behavior. As shown by Dobrushin [3], (4) continues to upper bound the error exponent of fixed-length block-coding over symmetric channels even when feedback is present. An extension of the bound to arbitrary DMCs with feedback is given in [8].

B. Variable-length block coding with feedback

If feedback is available, and variable-length coding is allowed, a far larger exponent than the sphere-packing bound can be achieved. Let W be a stationary DMC with input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$, and with causal instantaneous noiseless feedback.

Definition 2: An average-rate-$\bar{R}$ variable-length block encoder-decoder pair $(\mathcal{E}_{vl}, \mathcal{D}_{vl})$ is a pair of maps

$$\mathcal{E}_{vl} = \{\mathcal{E}_n \ : \ \mathcal{M} \times \mathcal{Y}^{n-1} \to \mathcal{X}\}_{n\geq 1}, \qquad (5)$$

$$\mathcal{D}_{vl} = \{\mathcal{D}_n \ : \ \mathcal{Y}^n \to \mathcal{M} \cup \{0\}\}_{n\geq 1}, \qquad (6)$$

together with a random decision time t, defined as the first n such that $\mathcal{D}_n(y^n) \neq 0$, that satisfies

$$\frac{\log|\mathcal{M}|}{E[t]} \geq \bar{R}, \qquad (7)$$

and where $\mathcal{M} = \{1, 2, \ldots, M\}$ is the set of messages.

An error exponent E is said to be achievable if there exists a family of coding strategies (of increasing message set size $|\mathcal{M}|$) such that

$$\lim_{\epsilon\to 0} -\frac{\log\epsilon}{E[t]} \geq E \qquad (8)$$

where $\epsilon = \Pr[\hat{m} \neq m]$ is the probability that the message is decoded incorrectly at the decision time t.

Burnashev [1] shows that an achievable upper bound on the error exponent in this setting is

$$E_b(\bar{R}) = C_1\left(1 - \frac{\bar{R}}{C}\right), \qquad (9)$$

where the constant C_1 is defined by the two "most distinguishable" input symbols as

$$C_1 = \max_{x_i, x_j} \sum_{y} W(y|x_i) \log \frac{W(y|x_i)}{W(y|x_j)}.$$

Let x_{i^*} and x_{j^*} denote the two symbols that yield C_1.

C. Streaming transmission with feedback

Streaming transmission schemes concern the sending of a sequence of messages $m_1, m_2, m_3, \ldots$ where $m_j \in \mathcal{M}_j = \{1, 2, \ldots M_j\}$. At any time the message estimate $\hat{m}_j \in \hat{\mathcal{M}}_j = \{0, 1, \ldots M_j\}$. The number of messages that have entered the encoder by channel use n is denoted $J_{enc}[n]$ and the decoder's estimate of this number is $\hat{J}_{enc}[n]$. The count $J_{enc}[n]$ is also a function of the outputs y^{n-1}. We suppress this dependence for notational simplicity. Let W be a stationary DMC with input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$, and with causal instantaneous noiseless feedback.

Definition 3: An average-rate-$\bar{R}$ streaming encoder-decoder pair $(\mathcal{E}_{seq}, \mathcal{D}_{seq})$ is a set of maps

$$\mathcal{E}_{seq} = \left\{\mathcal{E}_n \ : \ \prod_{j=1}^{J_{enc}[n]} \mathcal{M}_j \times \mathcal{Y}^{n-1} \to \mathcal{X}\right\}_{n\geq 1}, \qquad (10)$$

$$\mathcal{D}_{seq} = \left\{\mathcal{D}_n \ : \ \mathcal{Y}^n \to \prod_{j=1}^{\hat{J}_{enc}[n]} \hat{\mathcal{M}}_j\right\}_{n\geq 1}, \qquad (11)$$

and where with high probability $\mathcal{D}_{seq}$ satisfies

$$\lim_{n\to\infty} \frac{\sum_{j=1}^{\hat{J}_{enc}[n]} 1[\hat{m}_j \neq 0] \log|\mathcal{M}_j|}{n} \geq \bar{R}. \qquad (12)$$

Let t'_j denote the *first* time that $J_{enc}[t'_j] \geq j$ and let t''_j denote the *first* channel use such that $\hat{m}_j \neq 0$. Then, an error exponent E is said to be achievable if there exists a family of coding strategies (of increasing message set sizes $|\mathcal{M}_j|$) such that

$$\lim_{\epsilon\to 0} -\frac{\log\epsilon_j}{E[t''_j - t'_j]} \geq E \qquad \text{for all } j, \qquad (13)$$

where $\epsilon_j = \Pr[\hat{m}_j \neq m_j]$ is the probability that the jth message is decoded incorrectly.

In [9], [10] two strategies are given for streaming communications over a binary symmetric channel (BSC) with cross-over probability p. Their results are encapsulated in the following theorem.

Theorem 1: Consider a BSC with cross-over probability $p \leq 0.5$. Then, a streaming encoder/decoder pair exists that can achieve any average-rate and error-exponent pair $(\bar{R}, E)$ that satisfy the following:

$$\bar{R} < H_B(q * p) - H_B(p), \qquad (14)$$

$$E < D(q * p\|p), \qquad (15)$$

where $0.5 \leq q \leq 1$ is a design parameter, and $q*p = q(1-p) + (1-q)p$ (we use $*$ to denote binary convolution).

Remark: The value $q * p = q(1-p) + (1-q)p$ parameterizes the output distribution of a BSC when the input is Bernoulli-q. While (14)-(15) isn't the form of

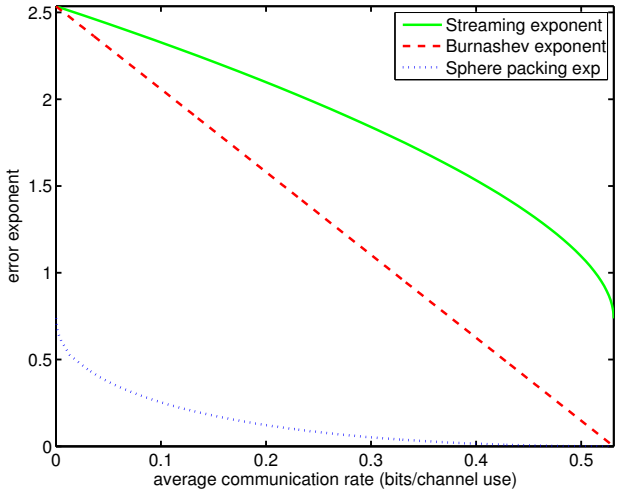

Fig. 2. Exponent comparison for streaming, Burnashev (variable-length block code), and sphere-packing (fixed-length block code).

the rate and exponent pair given in [9], it is a quick exercise to check that the expressions match. We later give a clear operational interpretation to this form.

D. Comparison of exponents

In Fig. 2 we compare the error exponents for a binary symmetric channel with cross-over probability 0.1. The sphere-packing bound is an upper bound on the error exponent (tight at rates close to capacity) of coding without feedback, the Burnashev upper bound is tight at all rates, and the streaming exponent is a lower (achievable) bound on the exponent.

There is a clear intuition why the combination of feedback and variable-length codes makes possible a major improvement in the reliability function given by the sphere-packing bound. The sphere-packing bound (4) tells us the probability, $2^{-ND(V\|W|P)}$, that over the block length N the channel displays a behavior V that does not support the communication rate, $I(P,V) < R$. With feedback the transmitter can detect when the channel behaves badly, i.e., when $I(P,V) < R$. It can then attempt to signal the decoder to delay decoding until channel behavior improves. Since bad behavior is exponentially rare, $2^{-ND(V\|W|P)}$, decoding is only very rarely delayed. Therefore huge improvements in reliability can be made at negligible cost in average rate.

The reason why streaming achieves a higher exponent than Burnashev is more subtle. We develop intuition by first understanding how the Burnashev exponent is achieved, and then point out what is missing when that solution is applied to streaming data.

Both the coding strategy used by Burnashev, and a later strategy by Yamamoto and Itoh [14] that also achieve the Burnashev exponent, are two phase schemes. In the first phase a strategy is used to communicate just below capacity at a somewhat small error probability (e.g., in [14] a fixed-length feedback-free block code, per Def. 1, is used). Via the feedback the source knows the destination's message estimate and also knows when the destination thinks the first phase has ended (if that time is not fixed). The source then follows up with a confirm/deny signal, the all-x_{i^*} or the all-x_{j^*} signal, respectively. If the destination detects a deny, the source retransmits the message and the system ignores the first transmission. If the destination detects a confirm, the system moves onto the next message.

An error can only occur if a deny signal is mis-detected as a confirm. Retransmissions occur either through detected denials or false alarms (detecting a denial when a denial was not sent). To achieve (9) one skews the binary hypothesis test so that the probability of false alarm is bounded by a small constant while the probability of missed detection is driven as small as possible. The best exponent is found via an application of Stein's Lemma [2].

Figure 3 diagrams the usage of the forward channel when a Burnashev-type scheme is used to transmit a sequence of messages. We make two observations. First, since almost all messages are received correctly in their first transmission, almost all confirm/deny signals are confirms. Confirm signals are therefore quite predictable and so the degrees-of-freedom corresponding to these channel uses are not used efficiently. To improve this situation, in Sec. IV we make confirms implicit and only make denials explicit. This latter strategy achieves the streaming exponent of Fig. 2.

The second aspect of Fig. 3 to note is an architectural one. The probability of erroneous decoding given by (9) is $2^{-NC_1(1-\bar{R}/C)}$. To make this probability as small as possible, we choose N as large as possible. However, because many uses of streaming data are delay limited (as opposed to file transfers), the application-layer latency constraint puts an upper bound on N. We reason that N should be on the order of the latency constraint to minimize error probability while meeting the constraint. A different principle emerges in the streaming context.

III. Main Results: Streaming transmission with active feedback

In this section we state our main result for streaming transmission over a channel with noisy feedback. Let W and W_r be a pair of stationary DMCs with input alphabets $\mathcal{X}$ and $\mathcal{X}_r$, and output alphabet $\mathcal{Y}$ and $\mathcal{Y}_r$,

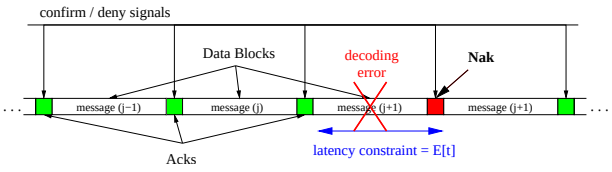

Fig. 3. Using a Burnashev-type strategy for multiple messages means that almost all confirm/deny messages are confirms. This is an inefficient use of channel uses.

respectively. Communication from source to destination is along the forward channel W. Feedback from destination to source is along the reverse channel W_r. The capacity of the forward channel is $C = \max_P I(P, W)$ and that of the reverse channel $C_r = \max_{P_r} I(P_r, W_r)$. Channel uses are assumed synchronized so that n simultaneously indexes a use of the forward and a use of the reverse channel.

As in Def. 3, the source transmits a sequence of messages $m_1, m_2, m_3, \ldots$ to the destination where $m_j \in \mathcal{M}_j = \{1, 2, \ldots M_j\}$. At any time the message estimate $\hat{m}_j \in \hat{\mathcal{M}}_j = \{0, 1, \ldots M_j\}$. The number of messages that have entered the encoder by channel use n is denoted $J_{enc}[n]$ and the decoder's estimate is $\hat{J}_{enc}[n]$.

Definition 4: An average-rate-$\bar{R}$ streaming encoder / feedback-encoder / decoder triplet $(\mathcal{E}_{seq}, \mathcal{F}_{fb}, \mathcal{D}_{seq})$ with active feedback is a set of maps

$$\mathcal{E}_{fb} = \left\{ \mathcal{E}_n \ : \ \prod_{j=1}^{J_{enc}[n]} \mathcal{M}_j \times \mathcal{Y}_r^{n-1} \to \mathcal{X} \right\}_{n \geq 1}, \quad (16)$$

$$\mathcal{F}_{fb} = \left\{ \mathcal{E}_n \ : \ \mathcal{Y}^n \to \mathcal{X}_r \right\}_{n \geq 1}, \quad (17)$$

$$\mathcal{D}_{fb} = \left\{ \mathcal{D}_n \ : \ \mathcal{Y}^n \to \prod_{j=1}^{\hat{J}_{enc}[n]} \hat{\mathcal{M}}_j \right\}_{n \geq 1}, \quad (18)$$

and where with high probability $\mathcal{D}_{fb}$ satisfies

$$\lim_{n \to \infty} \frac{\sum_{j=1}^{\hat{J}_{enc}[n]} 1[\hat{m}_j \neq 0] \log |\mathcal{M}_j|}{n} \geq \bar{R}. \quad (19)$$

Remark: This definition enlarges on that of Def. 3 though the addition of the feedback encoder (17). Since the destination decides what to feed back, this is termed "active" feedback. When the feedback channel is noiseless and such that we can set $y_r = x_r = y$, we can return to the "passive" noiseless feedback setting of Def. 3.

The error exponent is defined exactly as in the noiseless case (13).

Our main result demonstrates that the exponent of streaming transmission presented in Thm. 1 can be maintained under noisy feedback.

Theorem 2: Let the forward channel be a BSC with cross-over probability p, and the reverse channel be any channel with capacity C_r. Then, a streaming encoder/feedback-encoder/decoder triplet exists that can achieve any average-rate and error-exponent pair $(\bar{R}, E)$ that satisfies the following inequalities:

$$\bar{R} < H_B(q * p) - H_B(p), \quad (20)$$

$$E < \min\{D(q * p \| p), C_r\}, \quad (21)$$

where $0.5 \leq q \leq 1$ is a design parameter, and $q * p = q(1-p) + (1-q)p$.

Remark: Contrasting Thm. 2 with Thm. 1 shows that in the streaming context noisy feedback need not lead to any reduction in the error exponent. There is no loss in the exponent as long as the capacity of the feedback channel exceeds the streaming error exponent of the forward channel. Note that detailed characteristics of the feedback channel (such as its error exponent) do not enter this theorem.

IV. Beating Burnashev in delay

In this section we build towards our general communication protocol. We first describe the protocol in the special case of a binary-symmetric forward channel and noiseless output feedback, Def. 3. In this context the scheme is a version of Kudryashov [10]. Subsequently we generalize to noisy feedback.

A. BSC with noiseless feedback

The strategy presented in this section achieves the rate/exponent tradeoff given by (14) and (15) in Thm. 1. It further gives a clear operational interpretation to that trade-off.

The strategy operates across a sequence of "time slots", each comprising N channel uses. Each time slot is used to transmit a new message, retransmit a message, or to send the denial signal. The functionality of each time slot is therefore not pre-determined. At the end of each slot the destination first decides whether the time slot contained a data message or a NAK. In the former case it makes a tentative decision equal to its maximum likelihood (ML) estimate of the transmitted message. That tentative decision will be finalized Δ time slots later if, in the ensuing time, the destination does not detected a NAK. Therefore, the minimum transmission time of any message is $(1 + \Delta)N$.

Via the feedback link the source monitors the destination's decisions and immediately learns of any decoding errors. As soon as a tentative decoding error occurs the source starts to transmit the denial signal. This is a sequence of NAKs of maximum length ΔN. We discuss below how to design the NAK signal. A diagram of the protocol is depicted in Fig. 4.

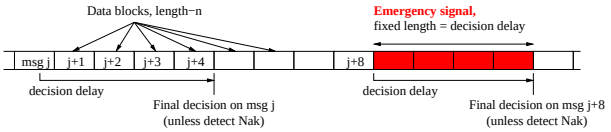

Fig. 4. In a streaming context, there are past messages and future messages. Unlike in Burnashev, a single message is not considered in isolation from the rest of the system. In this context it is very advantageous to make confirm signals implicit, and only explicitly send denial (NAK) signals. By doing this channel uses are more efficiently used, and the error exponent can be boosted massively.

We now describe a number of components of the protocol in more detail. At any time the destination has an estimate $\hat{j} \leq \hat{J}_{enc}[n]$ of the packet being transmitted in that time slot. The source knows $\hat{j}$ because of the noiseless feedback.

1) Data is communicated using length-N constant-q composition codes [1] of rate $R = H_B(q * p) - H_B(p) - \epsilon$ where $0.5 \leq q \leq 1$. The NAK signal is the all-zeros sequence.
2) If the destination does not detect a NAK in a given time slot, it increments its sequence number from $\hat{j}$ to $\hat{j} + 1$.
3) If, instead, at the end of any time slot, the destination detects a NAK, it interprets this as indicating that one of the last $\Delta+1$ messages was decoded in error. It decrements its sequence number by $\Delta+1$ (from $\hat{j}$ to $\hat{j}-\Delta$) and deletes the respective earlier tentative decisions. Retransmissions take priority over other enqueued messages.
4) When a NAK is detected by the decoder (even if incorrectly), the source backs up the index $j \leq J_{enc}[n]$ of the message it is transmitting by $\Delta+1$ and retransmits.
5) Whenever the destination makes a decoding error the source immediately learns of it via the feedback. The source then starts transmitting the emergency denial signal. The source continues to transmit the denial signal until it is detected or up to the maximum denial length of Δ time slots.

In order for the strategy to work, it is crucial that the denial signal is easily distinguished from any message. To ensure this property the denial sequence is designed to be far removed from all codewords. A sense of the geometry is shown in Fig. 5. Codewords are depicted as being on the surface of the sphere (in the case of the BSC the sphere's surface corresponds to the set of composition-q sequences) while the NAK is the all-zeros sequence, located at the center of the sphere.

[1]For all codewords $\mathbf{x}[m]$ where $m \in \{1, 2, \ldots 2^{NR}\}$, $(1/N)\sum_{i=1}^{N} 1[x_i[m] = 1] = q$ where $1[\cdot]$ is the indicator function.

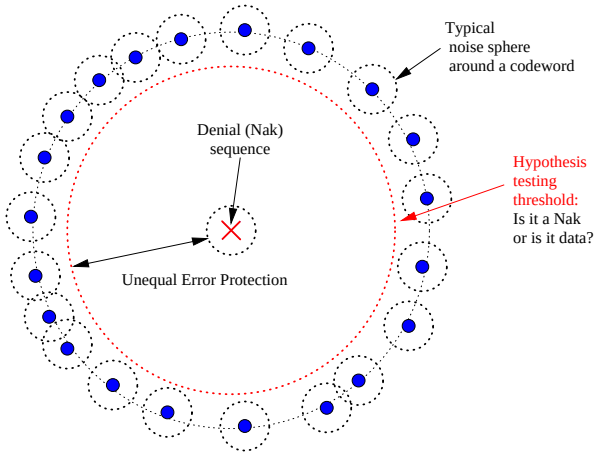

Fig. 5. The NAK sequence is designed to be distant from any codeword. The binary hypothesis test–whether the last time slot contained a message or a NAK–is skewed to make the probability of missed detection of a NAK extremely small. At the same time the decision regions for messages must contain the union of the typical noise spheres of all codewords to keep retransmissions rare.

When deciding whether or not the current time slot is a NAK, the destination skews its hypothesis test. It designs the test to make the probability of missed detection of a NAK extremely small while keeping the probability of false alarm (which just leads to a retransmission) bounded, but small. This aspect of the design is akin to the skewed hypothesis test in Burnashev. The difference is that the decision is now between the NAK sequence and *any* codeword, not between the NAK sequence and a *particular* confirmation signal. Fig. 5 gives a sense of this unequal error protection.

In addition to achieving a much larger exponent for positive rates, the architectural implications of this strategy are far different from those drawn from Burnashev-type approaches at the end of Sec. II. In streaming we choose each time slot to be significantly shorter than the latency constraint so that the denial signal can extend across many time slots. Information is treated more as a flow (in small chunks) than as blocks (very large chunks). This shift to shorter block lengths makes it possible to maintain the same exponent even when the feedback channel is noisy.

1) Error bound: An error occurs only if a denial signal is missed. To decide whether a NAK was sent, at the conclusion of each time slot the destination calculates the composition of the N output symbols. If a data message was transmitted, the expected fraction of ones is $q * p = q(1-p) + (1-q)p$. If a NAK was sent the expected fraction of ones is p. The gap between the two output compositions means that the hypothesis test

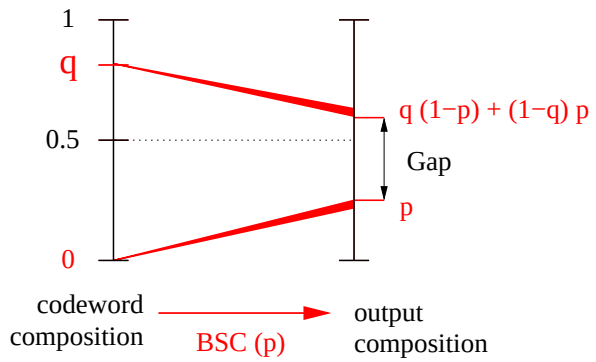

Fig. 6. The gap in output distributions means a highly discriminating hypothesis test can be made.

can be biased to make it extremely unlikely to miss a NAK. The gap is depicted in Fig. 6. We design the test so that the probability of false alarm (false detection of a NAK) is bounded above by an arbitrarily small constant $\epsilon_{FA} > 0$. This keeps retransmissions rare. Using Stein's lemma [2, Thm. 12.8.1] we can then say that in the limit of large N the probability of missed detection of a NAK in a single time slot can be upper bounded by $2^{-ND(q*p\|p)}$. Since times slots are tested independently and codewords are chosen independently, the probability of missing a NAK in each of the (maximum denial signal length of) Δ times slots is

$$2^{-N\Delta D(q(1-p)+p(1-q)\|p)}. \tag{22}$$

When a denial signal is messed, the system keeps NAK-ing to correct the missed-NAK decisions made in subsequent time slots.

2) Service time bound: There are three events that can lead to the retransmission of a message that has been transmitted. First, the message itself may be decoded in error. Second, one of the subsequent $\Delta-1$ messages may be decoded in error so that the message may be caught in one of their NAK windows. Third, the destination may falsely detect a NAK within one of the subsequent Δ time slots. The probability of the union of these events is upper bounded by δ where

$$\delta = \epsilon_{ML} + (\Delta - 1)\epsilon_{ML} + \Delta\epsilon_{FA}, \tag{23}$$

and where ϵ_{ML} is the probability of erroneous tentative decoding. The constant δ can be made arbitrarily small if $R < H_B(q*p) - H_B(p)$ by choosing N large enough.

To lower bound the average rate, we assume that every NAK signal extends to its maximum duration of Δ time slots. Messages received correctly after the first transmission take N channel uses to send. A message received correctly after k retransmissions takes at most $N + k(1+\Delta)N$ channel uses to send. The probability of retransmission is upper bounded by δ. We upper bound the expected number of transmissions of a particular message by the expectation of a geometrically distributed random variable with parameter δ. This upper bounds the expected number of channel uses dedicated to the transmission of a particular message by $N + \delta(1+\Delta)N/(1-\delta)$. Any average transmission rate $\bar{R}$ such that

$$\bar{R} < \frac{R}{1 + \frac{\delta}{1-\delta}(1+\Delta)} \tag{24}$$

is therefore achievable.

To bound the expected transmission delay note that whenever a retransmission event occurs the additional delay (excepting further retransmissions) is upper bounded by $1+\Delta$ time slots. If a message is accepted after a single transmission, the total delay is $(1+\Delta)N$. If k retransmissions occur, the total delay is upper bounded by $(1+k)(1+\Delta)N$. Using the bound on probability of retransmission yields the upper bound on the expected transmission time,

$$E[t] \leq \frac{(1+\Delta)N}{1-\delta}. \tag{25}$$

Plugging (22) and (25) into (13) gives the following lower bound on the achievable exponent

$$(1-\delta)\frac{\Delta}{1+\Delta}D(q*p\|p). \tag{26}$$

For long delays Δ any pair $(\bar{R}, E)$ that satisfies

$$\bar{R} < H_B(q(1-p) + (1-q)p) - H_B(p),$$
$$E < D(q(1-p) + (1-q)p\|p),$$

is achievable, where $0.5 \leq q \leq 1$. As q approaches 0.5, the upper bound on $\bar{R}$ approaches the capacity of this channel $C_f = 1 - H_B(p)$, and the error exponent approaches

$$\frac{1}{2}\log\frac{1}{4p(1-p)}.$$

The error exponent as a function of rate is the same as the dash-dotted "streaming exponent" plotted in Figure 2 The limit of error exponents as rate approaches capacity ($q \to 1/2$) is strictly positive. At lower rates the codebook composition q is biased toward having more ones than zeros. This increases the distance from the all-zero NAK sequence to any of the codewords.

B. Noisy feedback

The challenge posed by the addition of noise to the feedback link is two fold. First, we need a mechanism by which the source can detect when the destination is in error. Second, since the source cannot "look over

the shoulder" of the destination it cannot automatically keep synchronized. Thinking in terms of the noiseless strategy, the source does not immediately know whether the destination has detected a NAK or a non-NAK.

To detect decoding errors at the source the destination transmits a hash of its tentative decisions to the source in each time slot. The hash message is protected with a length-γN block code, per Def. 1. If the decoded hash matches the hash of the corresponding messages transmitted by the source, data transmission continues, otherwise the system drops into emergency mode. This is an extension of an idea originally developed in [4] for low-rate noiseless feedback. The main modification is that the random hash is calculated with respect to the product of the last $\tilde{\Delta}$ tentative decisions. The length of this window grows linearly in the decoding delay to balance the probability of hash collision with the probability of a missed detection of a denial signal.

To maintain synchronization the destination uses the last $(1-\gamma)N$ channel uses in each time slot to indicate to the source whether the binary hypothesis test for the last time slot yielded a NAK or a codeword. This sequence of single bits of information is protected by an anytime code. Although not immediately highly reliable, the outcome of the NAK/non-NAK hypothesis test in any particular time slot is learned by the source with increasing reliability over time. If retransmission opportunities are spaced sufficiently far apart in time, by the time the first opportunity to retransmit the message arises the source knows with high reliability whether or not it needs to retransmit.

We implement this method of maintaining synchronization through a round-robin scheduling of time slots among L "users". (See [11], [5] for earlier variants of this strategy.) Each message is assigned to a particular user. After the initial transmission of the message, the user must wait $(L-1)N$ channel uses for the next time slot in which it can transmit. In order to determine what to transmit, the source needs to estimate the destination's sequence of NAK or non-NAK decisions. The operation of the source in estimating these decisions can be visualized through a synchronization table, as shown in Fig. 7. This table indicates the source's best estimate of the destination's sequence of NAK/non-NAK decisions. Whenever the source marks an entry as a NAK that means that the source plans to retransmit the packet sent in that time slot and those in the $(\Delta+1)$ time slots just before. The table is updated at the end of every time slot. Transmitting a NAK takes priority over a retransmission, and retransmissions take priority over transmitting new data. By picking the number of users

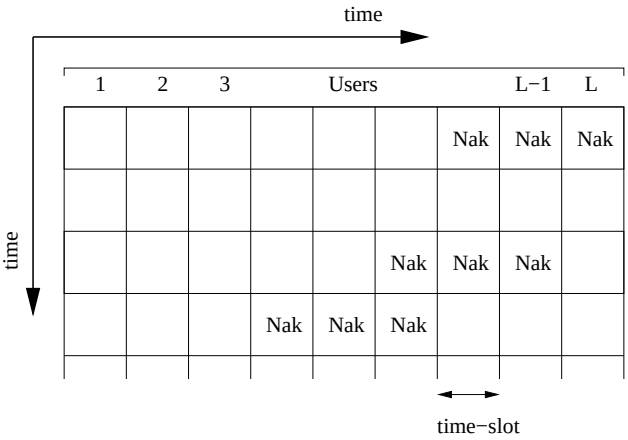

Fig. 7. Round-robin scheduling. Time runs left-to-right and top-to-bottoms (as reading a book). The denial sequences are of length 3. Denial signals take precedence over new data.

L such that $L \gg \Delta$, by the time the first opportunity to retransmit arises the source is very sure whether or not to retransmit.

Error detection and synchronization operate on two distinct time scales. The source must detect errors relatively promptly so as to be able to NAK decoding errors promptly. On the other hand, the source need know whether to retransmit a message much less quickly. As long as retransmissions are suitably rare, the effect on average decoding delay of long retransmission delays can be negligible.

We now summarize the components of the scheme. First we outline the operation of the source, then that of the destination. The source transmitter is always in one of two states: data transmission mode or emergency signaling mode.

1) **Transmit mode**:
   a) At the beginning of each time slot the source decodes the feedback message transmitted by the destination.
   b) The decoded fed-back hash is compared to the correct value of that hash known by the source.
   c) If the hashes match, the source transmits the next data message in the time slot, chosen according to the round-robin schedule described above.
   d) If the hashes do not match, the transmitter enters emergency mode and starts transmitting NAKs.
2) **Emergency mode**:
   a) If the source transmitter has been in emergency mode for Δ time slots, it switches

back into transmit mode.

b) If the source has been in emergency mode for less than Δ time slots it transmits a NAK in the next time slot and remains in emergency mode. (Note that while in emergency mode the source transmitter ignores the fed-back hash messages because it knows that with high probability there is a decoding error in the window the hash is calculated over.)

3) **Synchronization**: At the end of each time slot the source uses its latest observation of the anytime code to update it's estimate of the destination's sequence of NAK/non-NAK decisions.

The destination makes a compound hypothesis test at the end of each time slot.

1) **Synchronization**: The destination tests whether the last time slot is a NAK or a data message.
   a) The result of this hypothesis test (NAK or not NAK) is a single bit, inputted into the anytime code used for synchronization.
   b) The anytime code is fed back using the last $(1-\gamma)N$ reverse channel uses of each slot.
2) **Error Detection (Hashing)**: If the last time slot is determined to be a data message, the decoder makes its ML estimate of the message.
   a) The destination calculates the hash of the product of the last $\tilde{\Delta}$ tentative messages–a NAK is counted as message zero.
   b) The result of this hash is transmitted to the source using the first γN uses of the reverse channel in the next time slot, and protected by a standard length-γN block code.
   c) We set $\tilde{\Delta} = \Delta$. This gives the source as many chances as possible to detect the error before the decision is finalized–see Fig. 8. At the same time, the hash window length $\tilde{\Delta}$ is finite so that emergency mode can end. The source does not pay attention to the feedback messages while in emergency mode (see above), but by choosing $\tilde{\Delta}$ appropriately (equal to Δ), when emergency mode ends the hash window no longer includes the (likely erroneous) tentative decision that first triggered a hash mismatch.
3) **Final Decision**: If no NAKs have been detected in the last $\Delta + 1$ time slots, the tentative decision made about the message sent $\Delta+2$ time slots ago is finalized. At this point the message is served up to the destination's application layer.

1) Error bound: In the noiseless feedback setting the only error mechanism is a missed NAK. In the noisy setting there are additional sources of error. Before transmitting a NAK the source must first learn of the erroneous tentative decision. Undetected errors can slow this process. These errors occur when the destination's tentative decision is incorrect, but the source does not detect the error because the decoded feedback hash and the hash of the correct messages match. We term this a hash "collision". A second source of error is when source and destination get out of synchronization.

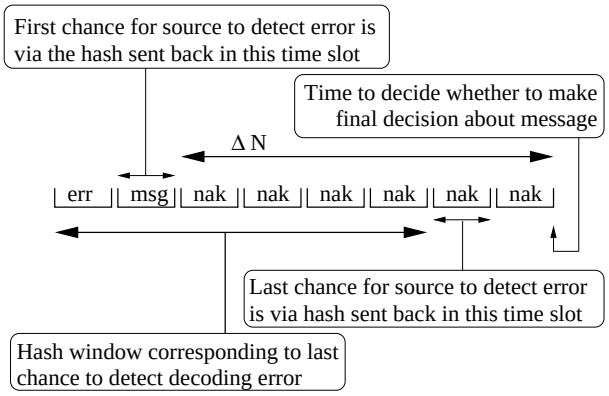

Fig. 8. Timing of the elements of the strategy.

To bound the probability of error, we describe the operation of the feedback hash in full detail. The hash is calculated across the last $\tilde{\Delta}$ tentative decisions $(\hat{m}_{j-\tilde{\Delta}+1}, \ldots, \hat{m}_j)$. The $2^{\tilde{\Delta}NR}$ possible message combinations are randomly partitioned into $M_r = 2^{\gamma NR_r}$ bins $\mathcal{B}_1 \ldots \mathcal{B}_{M_r}$. Next, the index k of the bin such that $(\hat{m}_{j-\tilde{\Delta}+1}, \ldots, \hat{m}_j) \in \mathcal{B}_k$ is encoded and transmitted along the reverse channel using a length-γN block code. The source decodes this message to $\hat{k}$. If $h(m_{j-\tilde{\Delta}+1}, \ldots, m_j) \notin \mathcal{B}_{\hat{k}}$ (equivalently we say if $\hat{k} \neq B(m_{j-\tilde{\Delta}+1}, \ldots, m_j)$) the source enters emergency mode and begins transmitting the denial sequence in the next time slot. Otherwise it sends its next message.

The probability of error is the probability that by the end of the $(\Delta+1)$st time slot after a tentative decoding error occurs, the destination has not learned that it has made an error. We write this event as $\bar{E}^{\text{dest}}_{(\Delta+1)N}$. To bound the probability of this event we define E^{src}_{kN} to be the event that the source first learns of a tentative decoding error kN channel uses after the erroneous tentative decision is made. We further define $\bar{E}^{\text{src}}_{kN}$ to be the event that the source still has not learned of the error by time kN. The error is bounded as $\Pr\left\{\bar{E}^{\text{dest}}_{(\Delta+1)N}\right\} =$

$$\Pr\left\{\bigcup_{k=1}^{\Delta} E^{\text{src}}_{kN} \bigcup \bar{E}^{\text{src}}_{\Delta N} \bigcap \bar{E}^{\text{dest}}_{(\Delta+1)N}\right\} \tag{27}$$

$$
\begin{aligned}
&\le \sum_{k=1}^{\Delta} \Pr\{E_{kN}^{\text{src}}\} \Pr\{\bar{E}_{(\Delta+1)N}^{\text{dest}} \mid E_{kN}^{\text{src}}\} \\
&+ \Pr\{\bar{E}_{\Delta N}^{\text{src}}\} \Pr\{\bar{E}_{(\Delta+1)N}^{\text{dest}} \mid \bar{E}_{\Delta N}^{\text{src}}\} \\
&\le \sum_{k=1}^{\Delta+1} \Pr\{\bar{E}_{(k-1)N}^{\text{src}}\} 2^{-(\Delta+1-k)ND(q*p\|p)} \qquad (28) \\
&= \sum_{k=1}^{\Delta+1} 2^{-(k-1)N\gamma R_r} 2^{-(\Delta+1-k)ND(q*p\|p)} \\
&= \sum_{k=1}^{\Delta+1} 2^{-\Delta N\left[\frac{k-1}{\Delta}\gamma R_r + (1-\frac{k-1}{\Delta})D(q*p\|p)\right]} \\
&\le \sum_{k=1}^{\Delta+1} 2^{-\Delta N \min_\lambda \left\{\lambda\gamma R_r + (1-\lambda)D(q*p\|p)\right\}} \\
&= \sum_{k=1}^{\Delta+1} 2^{-\Delta N \min\left\{\gamma R_r, D(q*p\|p)\right\}} \\
&= (\Delta+1) 2^{-\Delta N \min\{\gamma R_r, D(q*p\|p)\}}. \qquad (29)
\end{aligned}
$$

In (27) the index starts at $k = 1$ because it takes at least one time-slot for the source to learn of a decoding error. During this time slot the initial hash is fed back–see Fig. 8. In (28) the first factor is the probability that the first $k-1$ hashes are all collisions. The source is forced to make a decision on the fed back block code. If any of the fed back messages is decoded in error the probability of a hash collision (leading to a still-undetected error) is the reciprocal of the number of codewords, i.e., $2^{-N\gamma R_r}$. This is the same as the probability of hash collision when the fed back message is decoded correctly.

While (27)–(29) analyzes the probability of error when the system starts out operating correctly, and then drops into emergency mode, there is a second possibility. It is possible that part-way through an emergency NAK transmission, the destination will mistakenly think that a time slot actually corresponds to a data message rather than a NAK. In effect the destination mis-detects the end of the denial signal. Say that this occurs at the lth time-slot of the (length-Δ) denial transmission where $l \le \Delta - 1$. The probability that this mistake isn't itself NAK-ed is the probability that the remaining time-slots of the on-going denial signal are all mis-detected as data messages and that the message is still not NAK-ed in time even after the on-going denial signal concludes and the source starts checking hashes again. The probability that the remaining $(\Delta - l)$ NAK are all missed is

$$2^{-(\Delta-l)ND(q*p\|p)}. \qquad (30)$$

The probability that after the conclusion of the denial signal the error is not detected and NAK-ed by the $(\Delta+1)$st time slot after the mistaken hypothesis test is just (29) evaluated with $\Delta = (\Delta+1) - (\Delta - l) = l + 1$. Putting these together gives a probability equal to

$$
\begin{aligned}
&2^{-(\Delta-l)ND(q*p\|p)} (l+2) 2^{-(l+1)N\min\{\gamma R_r, D(q*p\|p)\}} \\
&\le (l+2) 2^{-(\Delta+1)N\min\{\gamma R_r, D(q*p\|p)\}},
\end{aligned}
$$

which is smaller than the probability in (29).

Now we bound the probability that any of the synchronization messages are in error. When first considering whether to retransmit a particular data message, the most recent string of relevant NAK/non-NAK decisions entered the anytime encoder between $(L-1)$ and $(L-\Delta-1)$ time slots before. These $\Delta+1$ decisions can trigger a retransmission while later NAKs cannot. The second most recent string of decisions entered the anytime encoder L time slots before that, and so forth. We upper bound this probability with $(\Delta+1)$ times the error probability of the least reliable bit in each string, i.e.,

$$
\begin{aligned}
&(\Delta+1) \sum_{i=1}^{\infty} 2^{-(iL-\Delta-2)(1-\gamma)NE_{any}\left(\frac{1}{(1-\gamma)N}\right)} \\
&= \frac{2^{-(L-\Delta-2)(1-\gamma)NE_{any}\left(\frac{1}{(1-\gamma)N}\right)}}{1 - 2^{-L(1-\gamma)NE_{any}\left(\frac{1}{(1-\gamma)N}\right)}}. \qquad (31)
\end{aligned}
$$

Letting N grow large and setting (31) equal to (29) allows us to solve for L, the number of users we need to cycle through to maintain the exponent under noisy feedback.

$$L = \left\lceil \frac{\Delta \min\{\gamma R_r, D(q*p\|p)\}}{(1-\gamma)E_{any}\left(\frac{1}{(1-\gamma)N}\right)} + \Delta + 2 \right\rceil. \qquad (32)$$

2) Service time bound: The probability of retransmission can be bounded in almost the same way as in the noiseless setting (23). The two additional sources of retransmissions comes from errors on the reverse link. An erroneous feedback transmission usually leads to a non-match in the hashes, triggering a denial and a subsequent retransmission. As long as $R_r < C_r$ then for N large enough, this probability can be made arbitrarily small. The second source of retransmissions comes from mis-synchronization. As just discussed in the context of the overall error probability (32), this probability can be made arbitrarily small. Thus, the probability of retransmission can again be bounded by any positive constant δ'.

Given the bound δ' on the probability of retransmission, the average communication rate is bounded just as in (24). Thus any average communication rate such that

$$\bar{R} < \frac{R}{1 + \frac{\delta'}{1-\delta'}(\Delta+2)}$$

is achievable.

We now bound the expected duration of transmission. Before making a final decision the destination must have received the data transmission and not detected a NAK in any of the ensuing $\Delta + 1$ time slots. If there are no retransmissions, the duration of communication is $(\Delta + 2)N$, if there are k retransmissions it is $(kL + \Delta + 2)N$. The expected duration of transmission $E[t]$ can be bounded as

$$E[t] < (\Delta + 2)N + \frac{\delta'}{1 - \delta'} LN. \tag{33}$$

Substituting the probability of error from (29) and the expected duration from (33) into the definition of the streaming error exponent (13) gives

$$\frac{\Delta \min\{\gamma R_r, D(q * p \| p)\}}{(\Delta + 2) + \frac{\delta'}{1-\delta'} L}$$

The best rate/reliability tradeoff is made by examining the limits when $\Delta \to \infty$, $\delta' \to 0$, but $\delta' \cdot \Delta \to 0$, also $\gamma \to 0$ (so $L \to \infty$), but $\Delta \cdot L \to 0$, and finally $R_r \to C_r$ from below, which keeps δ' small.

In these limits any pair $(\bar{R}, E)$ that satisfies

$$\bar{R} < H_B(q(1-p) + (1-q)p) - H_B(p), \tag{34}$$

$$E < \min\{C_r, D(q(1-p) + (1-q)p \| p)\}, \tag{35}$$

is achievable, where $0.5 \leq q \leq 1$.

C. Discussion

In Fig. 9 we plot the achievable tradeoff for our scheme over a BSC with cross-over probability 0.1, and a feedback link capacity $C_r = 1.5$. In the low-rate region hash collisions dominate. In the high-rate region missed denial signals dominate, the same error event as in the noiseless case. Respectively, these are the first and second terms of (35). For comparison we also plot the Burnashev and sphere-packing bounds, and the tradeoff achieved by erasure decoding paired with single-bit feedback as explored by Forney in [6]. In earlier work [5] we have shown that using techniques related to those presented herein, half the Burnashev exponent is achievable (with a hard-limit from hashing errors at $C_r/2$) while Forney's exponent may be maintained with any positive-capacity feedback channel.

In our current research we are working to show that the rate/reliability trade off encapsulated in Thm. 1, equations (14) and (15), is the best achievable. In addition to some preliminary work, the fact that the scheme we present herein is far different from Horstein's [9], and yet the rate/reliability trade off achieved is the same, makes us optimistic that this trade off is fundamental.

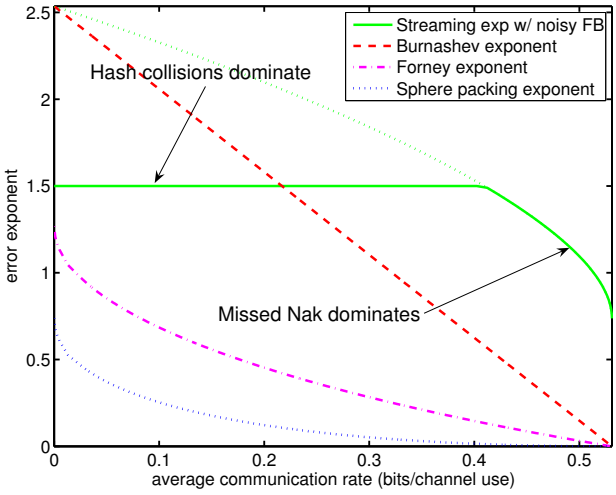

Fig. 9. Effect of noise on streaming feedback exponent. For comparison the Burnashev and sphere-packing bounds are plotted as well as Forney's exponent (variable-length single-bit feedback).

REFERENCES

[1] M. V. Burnashev. Data transmission over a discrete channel with feedback. Random transmission time. *Problems of Information Transmission*, 12(4):10–30, 1976.

[2] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, 1991.

[3] R. L. Dobrushin. An asymptotic bound for the probability error of information transmission through a channel without memory using the feedback. *Problemy Kibernetiki*, 8:161–168, 1963.

[4] S. C. Draper, K. Ramchandran, B. Rimoldi, A. Sahai, and D. Tse. Attaining maximal reliability with minimal feedback via joint channel-code and hash-function design. In *Proc. 43rd Allerton Conf. on Commun. Control and Computing*, September 2005.

[5] S. C. Draper and A. Sahai. Noisy feedback improves communication reliability. In *Proc. Int. Symp. Inform. Theory*, July 2006.

[6] G. D. Forney. Exponential error bounds for erasure, list, and decision feedback schemes. *IEEE Trans. Inform. Theory*, 14:206–220, March 1968.

[7] R. G. Gallager. *Information Theory and Reliable Communication*. John Wiley and Sons, 1968.

[8] E. A. Haroutunian. Lower bound for error probability in channels with feedback. *Problemy Peredachi Imformatsii*, 13(2):36–44, 1977.

[9] M. Horstein. Sequential transmission using noiseless feedback. *IEEE Trans. Inform. Theory*, 1963.

[10] B. D. Kudryashov. Message transmission over a discrete channel with noiseless feedback. *Problemy Peredachi Imformatsii*, 16(1):3–13, 1967.

[11] A. Sahai and T. Şimşek. On the variable-delay reliability function of discrete memoryless channels with access to noisy feedback. In *IEEE Inform. Theory Workshop, San Antonia, Texas*, 2004.

[12] J. P. M. Schalkwijk and T. Kailath. A coding scheme for additive noise channels with feedback – Part I: No bandwidth constraint. *IEEE Trans. Inform. Theory*, 12:172–182, April 1966.

[13] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423, 623–656, 1948.

[14] H. Yamamoto and K. Itoh. Asymptotic performance of a modified Schalkwijk-Barron scheme for channels with noiseless feedback. *IEEE Trans. Inform. Theory*, pages 729–733, November 1979.