Noisy feedback improves communication reliability

Stark C. Draper and Anant Sahai
Dept. of EECS, University of California, Berkeley, CA, 94720
{sdraper,sahai}@eecs.berkeley.edu

***Abstract*— We show how to exploit a noisy feedback link to implement high-reliability communication. We specify a variable-length coding strategy that achieves the error exponent (in delay) of erasure decoding using any noisy feedback channel which has a positive zero-rate random coding error exponent. Building on this result, we give a second approach that, depending only on the capacity of the feedback link, achieves an error exponent up to half of the Burnashev exponent—the maximum exponent that can be achieved with a noiseless feedback link. The resulting exponent can be far larger than the exponent of erasure decoding, particularly at rates close to capacity.**

I. Introduction

Feedback is present in many deployed communication systems and yet the theory of feedback has made little impact on communications in practice. In part this is because the models of instantaneous and noiseless feedback often assumed in theory do not match the delayed and noisy feedback usually available in practice. In this paper we look at one of the promised benefits of feedback—large increases in the reliability of communications over discrete memoryless channels (DMCs). We show that, to some degree, this benefit can still be realized even when the feedback link is itself a noisy channel.

Essentially feedback makes it possible to increase the reliability of communication over a DMC because it enables the use of variable-length transmission. If the system detects that the channel noise has been atypically bad for the duration of transmission, it refrains from decoding until the channel appears to have improved. In contrast, in systems without feedback the duration of transmission must be fixed. Even if the channel noise is bad for the entire duration, the decoder has no option but to decode.

We compare coding strategies based on their reliability functions (or error exponents), calculated in terms of the expected decoding delay. This is the length of time from when the source begins transmitting a message to when the destination makes a final decision. This function is

$$-\log \Pr[\text{error}]/E\left[t\right]. \tag{1}$$

In (1) $\Pr[\text{error}]$ is the probability of erroneous decoding whenever decoding finally occurs and t is the duration of transmission. For block coding t is deterministic (the block length). For the variable-length strategies discussed in this paper t is random. It depends on the channel noise.

Data is transmitted over a "forward" discrete memoryless channel $p_{y_f|x_f}(y|x)$. For every use of the forward channel, a symbol can be sent by the destination back to the source along a noisy "reverse" link $p_{y_r|x_r}(y|x)$. Our problem can be considered one of coding for the two-way channel where the rate of communication in the reverse direction is zero. The reverse link's only purpose is to facilitate communication across the forward link. We concentrate on the case where the rate of communication on the forward link $R < C_f = \max_{p_{x_f}} I(x_f; y_f)$ is greater than the capacity of the reverse link $C_r = \max_{p_{x_r}} I(x_r; y_r)$.

In this paper we combine ideas developed in [5] and [3] to design two communication strategies. First, in Section II we show how to implement an erasure decoding scheme introduced by Forney in settings where the feedback channel is noisy. The reliability function (1) evaluated for Forney's scheme (for noiseless feedback) greatly exceeds that which can be achieved by block coding. We show how to maintain the same reliability even when the feedback channel is very noisy. However, the error exponent of erasure decoding is not the maximum exponent that can be achieved when noiseless feedback is available. That is known as the Burnashev exponent. In Section III we develop a second strategy that gets closer to the Burnashev exponent. When the capacity of the feedback link is large enough, an error exponent of up to half of Burnashev's can be achieved. In Section IV we compare the two protocols. We show that, when communicating close to capacity, the strategy developed in Section III always dominates.

II. Erasure decoding with noisy feedback

In [4] Forney describes an erasure-decoding scheme. The scheme transmits a length-N block code and, if the maximum likelihood (ML) codeword isn't sufficiently more likely than the rest of the codewords (a parameter controls what is meant by "more") the decoder requests a retransmission. Retransmission events are termed erasures. Since retransmission requests require only a single bit of feedback, this scheme is particularly well-matched to implementation. The main challenge raised by noisy feedback is to make sure that the source and destination stay synchronized. If the source has moved on to the next data block while the destination expects a retransmission, the system will fail catastrophically.

We use an anytime code to maintain synchronization. [5] Anytime codes have the property that the reliability of any particular information bit is delay-dependent. Information bits are encoded sequentially, and the probability of decoding any given bit in error decays exponentially in the time since that bit entered the encoder. As an example, an infinite constraint-length time-varying convolutional code can have this property.

While not achieving the Burnashev exponent, the error exponent in erasure decoding (with noiseless feedback) far exceeds the sphere-packing bound, which bounds what is possible without feedback. Forney shows that for totally symmetric channels[1] an achievable lower bound on the error exponent is

$$E_{forn}(R) = E_{sp}(R) + C_f\left(1 - R/C_f\right), \quad 0 \leq R \leq C_f, \tag{2}$$

where $E_{sp}(\cdot)$ is the sphere-packing bound of the forward channel. Taking into account retransmissions, the average communication rate is $\bar{R} = NR/E[t]$. The random decoding time t is an integer multiple of N that depends on the number of retransmissions, which itself is a geometrically distributed random variable. As long as retransmissions are sufficiently rare (which they are as long as $R < C_f$) then $E[t] \simeq N$ and $\bar{R} \simeq R$.

We now specify a strategy that deals with the synchronization issues that arise when the feedback channel is noisy, and that achieves an error exponent equal to (2).

A. *Erasure coding with noisy feedback*

Our strategy works in parallel on the forward and reverse links. The forward channel is used for data transmission. The reverse link is used to send confirm/deny messages (ACKs or NAKs). In order to keep forward channel utilization high while getting a long delay on the confirmation messages (which makes it possible to preserve synchronization), we cycle transmission through W "users". Each user gets a time slot of N channel uses of the forward channel every WN channel uses. If a user needs to retransmit it must first wait $(W-1)N$ channel uses for its next time slot. See [5] for a discussion of a similar round-robin strategy.

[1]Forney's bound applies to other channels as well, but has a slightly more complex form as the capacity-achieving input distribution need not also maximize the sphere-packing bound at all rates.

Consider the first user and say that each of the W users has already sent $i-1$ messages (some of which may have been retransmissions). Let $m_{1,i} \in \{1, \ldots, 2^{NR}\}$ be the first user's ith message where $R < C_f$. The source uses a standard length-N block code to transmit this message in the $W(i-1)+1$th time slot, extending from channel use $WN(i-1)+1$ till channel use $WN(i-1)+N$. The destination uses erasure decoding, decoding to $\hat{m}_{1,i} \in \{\emptyset, 1, 2, \ldots, 2^{NR}\}$. The symbol $\emptyset$ indicates an erasure. If $\hat{m}_{1,i} \neq \emptyset$ the destination settles on $\hat{m}_{1,i}$ as its estimate. If $\hat{m}_{1,i} = \emptyset$ the destination requests a retransmission.

Confirmation messages are coded on the reverse link using an anytime code. This code takes as input the vector of messages estimates $\hat{m}_{w,i}$ for $w \in \{1, \ldots W\}$ made thus far. Since one new message estimate is made each N channel uses and only a single bit needs to be fed back per estimate (indicating an ACK or a NAK), the rate of the anytime code is $1/N$. At the beginning of the wth user's ith time slot the transmitter makes an estimate of all the previous confirm/deny messages for this user's stream to decide on the number j of ACKs the decoder has produced. It then sets message $m_{w,i}$ equal to the $j+1$st message in that user's sequence of messages. By cycling through the W users the minimal delay built up on any of the ACK/NAK messages that the source needs to estimate is $(W-1)N$. By revisiting these decisions at each transmission time the transmitter is guaranteed eventually to resynchronize whenever it gets out of sync with the receiver.

B. *Analysis*

We now analyze the scheme's reliability and efficiency. Its reliability equals the probability of a decoding error once the destination settles on its final message estimate. Its efficiency equals the average communication rate $\bar{R} = NR/E[t]$. To analyze these quantities we first understand the performance of the constituent erasure and anytime codes.

For erasure coding Forney shows that for any positive bound δ on the probability of erasure, and for the block length N large enough, the probability of decoding error can be made to decay exponentially as 2^{-NE} for any $E < E_{forn}(R)$. The error probability of the confirm/deny messages is determined by the error exponent of the anytime code. The anytime code has an error exponent $E_{any}(1/N)$ where $1/N$ is the rate of the synchronization messages. The exponent $E_{any}(\cdot)$ is lower bounded by the standard random coding exponent.

The destination makes an error if one of two events occurs. The first is that an error is made during erasure decoding. This event has probability

$$2^{-NE_{forn}(R)}. \tag{3}$$

The second event occurs when encoder and decoder get out of synchronization. This can happens if any of the confirmation messages for a given user is mis-decoded. The most recent confirmation message for a given user enters the anytime encoder $(W-1)N$ channel uses before the user needs that message. The second most recent message entered the anytime encoder WN channel before that, and so forth. Using the union bound we upper bound the probability that any of these messages are decoded in error as

$$\sum_{i=1}^{\infty} 2^{-(iW-1)NE_{any}(1/N)} = \frac{2^{-(W-1)NE_{any}(1/N)}}{1-2^{-WNE_{any}(1/N)}}. \tag{4}$$

Letting N grow large and setting (3) equal to (4) allows us to solve for W, the number of users we need to cycle through to maintain the Forney exponent under noisy feedback.

$$W = \left\lceil \frac{E_{forn}(R)}{E_{any}(0)} + 1 \right\rceil. \tag{5}$$

We now calculate the expected duration of transmission which will give the average communication rate. If decoding occurs after the first transmission of a data block, the duration of communication is N. If there is a single retransmission the duration is $(W+1)N$, if two $(2W+1)N$, and so forth. Since the probability of retransmission is bounded by δ, the expected transmission length $E[t]$ can be bounded as

$$E[t] < N + \delta WN/(1-\delta). \tag{6}$$

The second term can be made as small as desired by picking δ suitably small. Therefore, the average rate $\bar{R} > R/(1+\delta W/(1-\delta))$ can be made to approximate R.

We now show that the reliability function for this strategy equals Forney's. With the choice of W as in (5) the probability of decoding error is given by (3). Substituting this error probability together with $E[t]$ from (6) into (1) with $\bar{R} \simeq R$ gives $E_{forn}(\bar{R})$ as the reliability function.

Note that the capacity C_r of the reverse channel did not enter into the discussion (it does effect the choice of W in (5)). As long as the reverse channel has a positive random coding error exponent at zero rate then $E_{forn}(\bar{R})$ is achievable.

III. Getting to half the Burnashev exponent

As mentioned in the introduction, the error exponent (2) is not the maximum that can be achieved with feedback under an expected block-length constraint. This is because in erasure decoding only the destination is involved in decisions whether to retransmit and certain types of decoding errors are not detectable by the destination. By involving the source (who knows what was transmitted) in the decision whether to retransmit, a larger exponent can be achieved. Burnashev [1] gives both an upper bound on this exponent, and a scheme that achieves that upper bound. His bound is

$$E_{burn}(\bar{R}) = C_1\left(1 - \bar{R}/C_f\right), \quad 0 \le \bar{R} \le C_f, \tag{7}$$

where C_1 is defined as

$$C_1 = \max_{x_i, x_j} \sum_y p_f(y|x_i) \log[p_f(y|x_i)/p_f(y|x_j)].$$

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

To deal with noisy feedback we build on a scheme devised by Yamamoto and Itoh. This scheme achieves the Burnashev bound when the feedback channel is noiseless. Similar to Forney's, Yamamoto and Itoh's scheme also works in blocks of length N, with possible retransmissions. In the first λN channel uses, where $0<\lambda<1$, the source uses a rate-R block code to transmit the data. In Yamamoto and Itoh's setting, because of the noiseless feedback, at the end of the block code the source knows whether the destination's ML codeword is correct. The source then uses the remaining $(1-\lambda)N$ channel uses in the block to transmit a binary message that either confirms or denies the correctness of the destination's ML codeword. If the destination detects a NAK it expects a retransmission. If it detects an ACK the source moves on to the next message. Via the noiseless feedback the source stays synchronized with the destination. Errors only occur when a NAK is mis-detected as an ACK.

The ACK (NAK) codeword is $(1-\lambda)N$ repetitions of the x_{i^*} (x_{j^*}) symbol. For the binary hypothesis test, the destination uses a decoding rule tuned to achieve a fixed bound $\epsilon > 0$ on the probability of a false alarm, i.e., of an ACK being mis-detected as a NAK. This probability determines the probability of retransmission. Under this constraint the rule is tuned to minimize the probability of missed detection, i.e., of the NAK message being mis-detected as a ACK. Since this event determines the probability of error we want it to be very small. A direct application of Stein's Lemma [2] bounds this probability by $2^{-(1-\lambda)NC_1}$.

In our context the reverse link is not noiseless. We therefore encode our feedback messages. If $R > C_r$

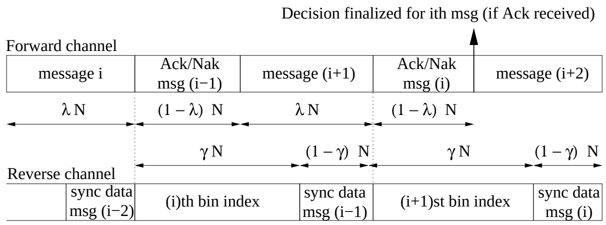

Fig. 1. Diagram of interleaving of different users' data and confirm/deny messages on the forward channel with their hash and sync messages on the reverse channel.

we cannot reliably transmit enough codewords on the reverse link to uniquely index each of the destination's possible decisions. Using an idea developed in [3] we instead send back a random hash of the decision. Each hash corresponds to a subset, or bin, of messages. In contrast to Forney's scheme where we only had to deal with synchronization issues, we now also have to deal with coding errors on the reverse link and with hash collisions—when an incorrect decision and the true message share the same hash.

Formally, our strategy work in four stages operating sequentially along the forward and reverse links. In the first stage a length-λN block code, $0 < \lambda < 1$ is used to transmit a message $m \in \{1, 2, \ldots 2^{\lambda N R}\}$ across the forward channel. The destination decodes to the most likely message $\hat{m}$. The $2^{\lambda N R}$ messages are randomly partitioned into $M_{fb} = 2^{\gamma N R_{fb}}$ bins $\mathcal{B}_1 \ldots \mathcal{B}_{M_{fb}}$ where $0 < \gamma < 1$. Next, the index k of the bin such that $\hat{m} \in \mathcal{B}_k$ is encoded and transmitted along the reverse channel using a length-γN block code. The source decodes this message to $\hat{k}$. If $m \in \mathcal{B}_{\hat{k}}$ (equivalently we say if $\hat{k} = B(m)$) the source transmits the ACK message along the forward channel using a length-$(1-\lambda)N$ repetition code (as in Yamamoto-Itoh). Otherwise it transmits the NAK message. Finally, an anytime code is used for the last $(1-\gamma)N$ channel uses of the reverse channel to inform the source whether the destination understood the confirm/deny message as an ACK or as a NAK. Just as in Section II we interleave the transmissions of W users to ensure that the source and destination stay synchronized. See Figure 1 for a diagram of the phases of transmission and their interleaving between users.

The expected duration of transmission is determined by the number of retransmissions. There are three events that can lead to a retransmission. The first is when the destination makes a decoding error ($\hat{m} \neq m$), the source learns of the mistake via feedback, and successfully transmits a NAK. The probability of this event is upper bounded by $2^{-\lambda N E_f(R)}$ where $E_f(\cdot)$ is the random coding error exponent of the forward channel. The second event occurs when the destination does not make a decoding error ($\hat{m} = m$), the source sends the NAK message, but that message is mistakenly decoded as an ACK. As in Yamamoto-Itoh, the probability of this event can be upper bounded by a constant ϵ. The third event also occurs when the destination does not make a decoding error ($\hat{m} = m$), but where there is an error on the feedback channel so the source thinks the destination has made an error ($\hat{k} \neq B(m)$). The probability of this event is upper bounded by $2^{-\gamma N E_r(R_{fb})}$ where $E_r(\cdot)$ is the random coding error exponent of the reverse channel. Union bounding these three events bounds the probability of retransmission by

$$2^{-\lambda N E_f(R)} + \epsilon + 2^{-\gamma N E_r(R_{fb})}. \tag{8}$$

As long as $R < C_f$ and $R_{fb} < C_r$ then as N grows large, the probability of retransmission is dominated by ϵ. Since ϵ can be set to any positive constant, for N large enough we can bound the probability of retransmission by any positive constant δ.

We next bound the expected duration of transmission. Before making a decision the destination must have received the initial transmission, fed back its encoded hash, and received the confirm/deny message. Thus, the shortest possible transmission duration is $2N$. See Figure 1. In comparison with erasure decoding, the extra delay of N is the cost of bringing the encoder into the decision whether to retransmit. If there is a single retransmission the duration is $(W+2)N$, if two the duration is $(2W+2)N$, and so forth. Since the probability of retransmission is bounded by δ, the expected transmission length $E[t]$ can be bounded as

$$E[t] < 2N + \delta W N/(1-\delta). \tag{9}$$

The second term can be made as small as desired by picking δ suitably small. Since two messages are interleaved every $2N$ channel uses, the average rate is

$$\bar{R} = \frac{2\lambda N R}{E[t]} > \frac{\lambda R}{1 + \frac{\delta W}{2(1-\delta)}}. \tag{10}$$

Since δ can be set arbitrarily small, $\bar{R}$ can be made to approximate λR.

To determine the reliability of our scheme we first enumerate the events that can lead to decoding errors. The first occurs when the destination makes a decoding error ($\hat{m} \neq m$) but the source thinks it has not since $\hat{k} = B(m)$. We term this a hash collision and bound its probability below. The second event is that the destination has made a decoding error, the source realizes this since $\hat{k} \neq B(m)$, but the NAK message is mis-decoded as an ACK. This is the same error event as in Yamamoto-Itoh, and is upper bounded by $2^{-(1-\lambda)NC_1}$. Finally, as in Section II there can be synchronization errors. We bound these as in (4) where the anytime code works across blocks of length $(1-\gamma)N$ rather than N and, to account for the hash feedback, there is an extra time-slot delay before the sync data enters the anytime code (see Fig. 1).

We now bound the probability of hash collisions. These occur when $\hat{k} = B(m)$ but $\hat{m} \neq m$. This can happen either because of decoding errors made at the destination (i.e., when $\hat{m} \neq m$) or decoding errors made on the feedback message (i.e., when $\hat{k} \neq B(\hat{m})$).

$$\begin{aligned}
&\Pr[\hat{k} = B(m)|\hat{m} \neq m] = \\
&\Pr[\hat{k} = B(m)|\hat{k} = B(\hat{m}), \hat{m} \neq m] \Pr[\hat{k} = B(\hat{m})] \\
&+ \Pr[\hat{k}=B(m)|\hat{k}\neq B(\hat{m}), \hat{m}\neq m] \Pr[\hat{k}\neq B(\hat{m})] \quad (11)\\
&\leq 2^{-\gamma N R_{fb}} \Pr[\hat{k}=B(\hat{m})] + 2^{-\gamma N R_{fb}} \Pr[\hat{k}\neq B(\hat{m})] \quad (12)\\
&= 2^{-\gamma N R_{fb}}. \quad (13)
\end{aligned}$$

In (11) we expand the collision events in terms of whether the feedback message was decoded correctly. Because the hashing is done independently of the code design in (12) the probability that a randomly picked bin equals the one the true codeword is in is equal to $2^{-\gamma N R_{fb}}$.

Combining the probability of a hash collision from (13) with the other error events gives the following bound on the probability of decoding error:

$$2^{-\gamma N R_{fb}} + 2^{-(1-\lambda) N C_1} + \frac{2^{-(W-2)(1-\gamma) N E_{any}\left(\frac{1}{(1-\gamma)N}\right)}}{1 - 2^{-W(1-\gamma) N E_{any}\left(\frac{1}{(1-\gamma)N}\right)}}. \quad (14)$$

To minimize (14) first note that as N gets large, by setting $W = \lceil (1-\lambda)C_1/(1-\gamma)E_{any}(1/(1-\gamma)N) + 2 \rceil$ the third term equals the second. The error probability is therefore controlled by the trade off between the first two terms. To minimize the first term, we want to choose R_{fb} to be as large as possible. However, from (8) we know that we need $R_{fb} < C_r$ to keep the number of retransmissions small. Hence we set R_{fb} to be just below C_r. For the same reason we set R to be just below C_f, and also so as to maximize $\bar{R}$ in (10).

Inspecting (14), with $R_{fb} \simeq C_r$ the error probability is $2^{-(1-\lambda)NC_1}$ as long as $\gamma C_r \geq (1-\lambda)C_1$. By choosing W (and therefore N) suitably large, we can set γ arbitrarily close to one. In the limit of $\gamma = 1$ the first two terms (hashing error and missed detection of a NAK) balance as long as $\lambda \geq (C_1 - C_r)/C_1$. Since $R \simeq C_f$, and $\bar{R} \simeq \lambda R \simeq \lambda C_f$ from (9), we express this condition in terms of the average rate:

$$\bar{R} \geq (1 - C_r/C_1) C_f. \quad (15)$$

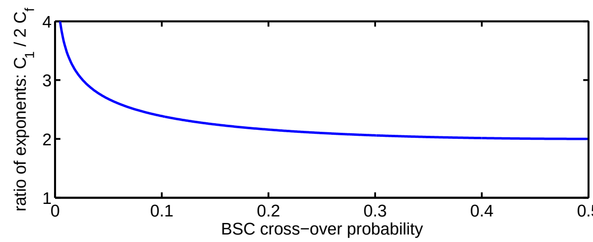

Fig. 2. Ratio of the reliability functions of hashing versus erasure decoding for $\bar{R} \geq (1 - C_r/C_1)C_f$ and $E_{sp}(\bar{R}) \simeq 0$.

Finally, since $E[t] \simeq 2N$, if (15) is satisfied the reliability function (1) is

$$-\frac{\log 2^{-(1-\lambda)NC_1}}{E[t]} \simeq \frac{(1 - \frac{\bar{R}}{C_f})C_1 N}{2N} = \frac{E_{burn}(\bar{R})}{2}.$$

For $\bar{R} < (C_1 - C_r)C_f/C_1$ the first term in (14) dominates, and the reliability function is $C_r/2$. Putting these two regimes together gives

$$-\log \Pr[\text{error}]/E[t] \simeq \min\left\{E_{burn}(\bar{R})/2, C_r/2\right\}.$$

IV. Discussion

In Section II we showed that a reliability function of $E_{forn}(\bar{R})$ is achievable as long as the feedback channel has a non-zero random (more generally expurgated) coding exponent at zero rate. In Section III we showed that for communication rates satisfying $\bar{R} \geq (1 - C_r/C_1)C_f$, a reliability function of $E_{burn}(\bar{R})/2$ is achievable. In general the ratio of the two functions is

$$\frac{0.5 C_1 (1 - \bar{R}/C_f)}{E_{sp}(\bar{R}) + C_f(1 - \bar{R}/C_f)}.$$

For rates close to capacity $E_{sp}(\bar{R}) \simeq 0$, and the ratio simplifies to $C_1/2C_f$. In Figure 2 we plot this ratio for a binary symmetric channel with cross-over probability less than $1/2$. For the whole range $C_1/2C_f \geq 2$, so the hashing scheme gives an error exponent that is at least twice as large as erasure decoding.

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] 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 Communication, Control and Computing*, Sept. 2005.

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

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