EUROPEAN TRANSACTIONS ON TELECOMMUNICATIONS
Euro. Trans. Telecomms. **00**: 1–17 (2008)
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/ett.0000

Variable-length channel coding with noisy feedback

Stark C. Draper[1] and Anant Sahai[2]*

[1]*Dept. of Electrical and Computer Engineering, University of Wisconsin, Madison, 53705.* sdraper@ece.wisc.edu
[2] *Wireless Foundations, Dept. of Electrical Engineering and Computer Sciences, University of California, Berkeley, 94720.* sahai@eecs.berkeley.edu

SUMMARY

It is known that perfect noiseless feedback can be used to improve the reliability of communication systems. We show how to make those gains robust to noise on the feedback link. We focus on feedback links that are themselves discrete memoryless channels. We demonstrate that Forney's erasure-decoding exponent is achievable given any positive capacity feedback channel. We also demonstrate that as the desired rate of communication approaches the capacity of the forward channel, the Burnashev upper bound on the reliability function is achievable given *any* positive-capacity noisy feedback channel. Finally, we demonstrate that our scheme dominates the erasure-decoding exponent at all rates and, for instance, at zero rate can achieve up to three-quarters of Burnashev's zero-rate reliability. This implies that in a shared medium, to maximize the reliability function some degrees of freedom should be allocated to feedback. Copyright © 2008 AEIT

1. Introduction

The availability of feedback in a communication system makes it possible to relax a fixed-block-length constraint and work instead with variable-length codes. Variable-length strategies have the distinct advantage of much better reliability functions, i.e., the trade off between error probability and the duration of transmission. The possibility of improvement may be understood by analogy with lossless source coding. In lossless source coding more likely (typical) sequences are assigned codewords whose description length is roughly equal to the entropy of the source. Such sequences make up most of the probability mass. In contrast, less likely sequences are assigned longer descriptions. While the expected description length is roughly equal to the description length of a typical sequence, zero-error is attained by (rarely) using longer descriptions. Feedback enables a similar *variable-length* paradigm for channel coding.

*Correspondence to: 267 Cory Hall, University of California, Berkeley CA 94720-1770, USA

As an important first example of the improvement possible for channel coding, consider Forney's paper [11] on erasure and list decoding. Forney gives the destination the option to declare an "erasure". When this occurs a retransmission is requested (via the feedback link), resulting in a variable-length code. Roughly, an erasure is declared if one codeword isn't sufficiently more likely than all the rest. (Alternately, in [12] exercise 5.14 an erasure is declared if the observation isn't jointly typical with a single codeword. The resulting reliability function is only slightly lower than Forney's.) The reliability function attained by Forney's scheme is strictly better than the sphere-packing bound that upper bounds the reliability function of fixed-block-length codes without feedback [12] and also of fixed-block-length codes with feedback over output-symmetric channels [6].[†]

As is shown by Burnashev in [2], a further improvement in the reliability function is possible by including the

[†]See [16] for a more detailed review.

source[‡] in the decision whether to retransmit. At the end of an initial transmission period the source knows via noiseless feedback of channel outputs whether the destination's best guess of the transmitted message is correct. If it is correct, the source follows up with a confirmation. Otherwise it sends a denial. In [2], Burnashev upper-bounds the attainable trade off, demonstrating that his achievability is tight.

Both Forney and Burnashev assume the feedback link is noiseless. However, while Forney's approach requires only a single bit be fed back noiselessly per data block, Burnashev requires noiseless output feedback, i.e., a noiseless feedback rate equal to the log-cardinality of the output alphabet. If noiseless feedback is claimed to be useful, it is natural to query whether the improvement persists if the feedback link is noisy, as would be the case in any real-world system. The assumption of noiseless feedback has long been recognized as the Achilles' heel of the information-theoretic study of feedback. In 1973 Bob Lucky stated it dramatically [14]:

> Feedback communications was an area of intense activity in 1968.... A number of authors had shown constructive, even simple, schemes using noiseless feedback to achieve Shannon-like behavior... The situation in 1973 is dramatically different.... The subject itself seems to be a burned out case. ...
>
> In extending the simple noiseless feedback model to allow for more realistic situations, such as noisy feedback channels, bandlimited channels, and peak power constraints, theorists discovered a certain "brittleness" or sensitivity in their previous results.

The goal of this paper is to demonstrate that improvements in reliability due to feedback are not necessarily "brittle" with respect to noise in the feedback link. However, to do this, we must take some care to formulate the problem appropriately. Although we cast our problem as one of point-to-point communications, we draw inspiration from networking. In networking a regular goal is to share a common communication resource among many users. Often the way to do so efficiently is via statistical multiplexing. Consider a time-division multiple-access backhaul network through which each user in some community occasionally wants to route a large data packet. Three important measures of performance are: (a) network efficiency, (b) latency, (c) error probability.

(a) Efficiency – the overall rate of transmission – is of paramount importance to the service provider who owns the backhaul link. Given the expected number of channel uses allocated to the transmission of each packet, and the size of each packet, efficiency is quantified in terms of how close the systems come to the Shannon capacity of the link.
(b) On the other hand, a network user is more concerned with latency. A user would like the time between when the transmission of their packet commences and when it is accepted by the destination to be as small as possible.
(c) In addition to low latency, a user also wants reliability. The service provider must create a network, and implement a communication protocol, that achieves suitably small probabilities of error for all users.

The trade off between efficiency, latency, and error probability, is studied in the information-theoretic limit of large packet payloads. Implicit is the idea that for any user, the time between packets is significantly longer than the acceptable average latency on an individual packet.

This is not the only possible context in which unreliable feedback is interesting. While the above discussion provides a packet-switched type motivation, for certain applications a circuit-switched perspective is more appropriate. In both [9, 19], we examine the case of a point-to-point link dedicated to a single user where packet payloads are of fixed size. Instead of packet size, the expected end-to-end delay is allowed to grow. That setting is appropriate to scenarios where data comes in a continuous stream of small chunks, each associated with an individual deadline (as in streaming media playback or a video conference). Thus, many packets can be "in flight" before the first one is due. That changes the nature of the solution quite dramatically. The difference between [9] and [19] has to do with the nature of the deadlines. In [9] a "soft" deadline is considered. The destination is tolerant of occasional detected erasures and is presumed to be able either to request retransmission or to mask detected errors. In [19] the deadline is "hard". If a packet misses its deadline it is considered to be in error. Each of these distinctions make a difference in the resulting asymptotic behavior.

[‡]In this paper we use the terms "source" and "destination" in preference to the more standard "transmitter" and "receiver". This is because to combat the noise in the feedback link the destination will actively decide what to transmit back to the source. Thus the destination will also have a transmitter and the source a receiver.

1.1. Illustrative results

In Figure 1 we plot the outer hull of reliability functions for the schemes derived in this paper. As points of comparison we plot the sphere-packing bound, the Forney exponent, and the Burnashev bound. The sphere-packing bound is an upper bound on the fixed-block-length error exponent (tight at rates close to capacity). The Forney exponent is a lower bound on the error exponent of decoder-driven variable-length coding. The Burnashev bound is a tight upper bound at all rates.

In the figure, the channel under consideration is a binary symmetric channel (BSC) with crossover probability 0.11. The feedback link is an independent channel with capacity $C_{fb} = 4C$. Any memoryless feedback channel that can support this rate of communication will suffice. The particular details of the reliability function of the feedback channel are not important for the results of this paper, in contrast to those of [17].

We present three schemes in this paper, the contributions of which are indicated by the differently shaded regions in Figure 1. We initially present our "basic" scheme that contains the crucial ideas needed to address noisy feedback. Typically, the reliability function of this scheme dominates at high rates of communication. For instance, this reliability function always approaches capacity at the same slope as Burnashev's bound, regardless of the capacity of the feedback channel. In the basic scheme, the source alone determines whether or not a retransmission is needed.

At lower communication rates, the time required to get the source enough information for it to determine whether or not to retransmit becomes a bottleneck. We immediately improve on the basic scheme by allowing the decoder the option of making the decision whether to retransmit. This decision is made by using an erasure decoder at the destination. The reliability function of this scheme, labelled "basic + erasure" is always at least as large as Forney's, and demonstrates that Forney's reliability function is achievable as long as the capacity of the feedback link is non-zero.

Finally, we develop a hybrid scheme labelled "multiple hash". This scheme trades off longer transmission duration in order to overcome the feedback bottleneck mentioned in the last paragraph. The source is again included in the decision whether to retransmit. In contrast to the erasure option, which dominates when C_{fb} is small, this scheme dominates when $\bar{R}$ is small. When $\bar{R}$ is small, but C_{fb} is too, this scheme reduces to "basic + erasure".

1.2. Outline and notation

The outline of this paper is as follows. In Section 2 we detail relevant background results. In Section 3 we describe the main challenges posed by noisy feedback and describe our solutions. Given any feedback channel that is a discrete memoryless channel (DMC) with positive capacity we develop a scheme whose reliability function converges to Burnashev's as the average rate of communication approaches capacity. This implies that in a communication medium where degrees of freedom must be shared between forward and feedback communication, a reliability function exceeding the sphere packing bound is attainable when some degrees of freedom are used for feedback. We detail the protocols and present the basic analysis thereof in Section 4. Detailed derivations are deferred to the Appendix.

Regarding notation, we use serifed-fonts, e.g., x to indicate sample values, and sans-serif, e.g., x, to indicate random variables. Sample or random vectors, e.g., of length-n, are respectively written as x^n and x^n. An element and a set to which it belongs are denoted, e.g., as $x \in \mathcal{X}$, and the cardinality of the set $\mathcal{X}$ by $|\mathcal{X}|$. We use $\mathbb{Z}_+$ to denote the positive integers.

2. Preliminaries

Definition 1 *A noiseless-feedback variable-length code for a discrete memoryless channel (DMC) with input $\mathcal{X}$ and output $\mathcal{Y}$ is a pair of sets of maps*

$$\mathcal{E} = \{\mathcal{E}_n \ : \ \mathcal{M} \times \mathcal{Y}^{n-1} \to \mathcal{X}\}_{1 \leq n}, \quad (1)$$
$$\mathcal{D} = \{\mathcal{D}_n \ : \ \mathcal{Y}^n \to \{0, 1, 2, \ldots, |\mathcal{M}|\}\}_{1 \leq n}, \quad (2)$$

where $\mathcal{M}$ is the set of messages and decoding to 0 denotes "erasure", i.e., transmission continues.

Definition 2 *The transmission duration Δ, average transmission rate $\bar{R}$, and reliability function $E_{vl}(\bar{R})$ are defined as follows:*

$$\Delta = n \text{ s.t. } \begin{cases} \mathcal{D}_n(y^n) \neq 0, \\ \mathcal{D}_{n'}(y^{n'}) = 0 \text{ for all } n' < n. \end{cases} \quad (3)$$

$$\bar{R} = \frac{\log |\mathcal{M}|}{E\left[\Delta\right]}. \quad (4)$$

$$E_{vl}(\bar{R}) = -\frac{\log \Pr[\text{error}]}{E\left[\Delta\right]}. \quad (5)$$

where $\Pr[\text{error}] = \max_{m \in \mathcal{M}} \Pr[\mathcal{D}(\mathsf{y}^{\Delta}) \neq m | \mathsf{m} = m]$ *and the expectation is taken over the channel noise (and the codebook if a randomized code is used).*

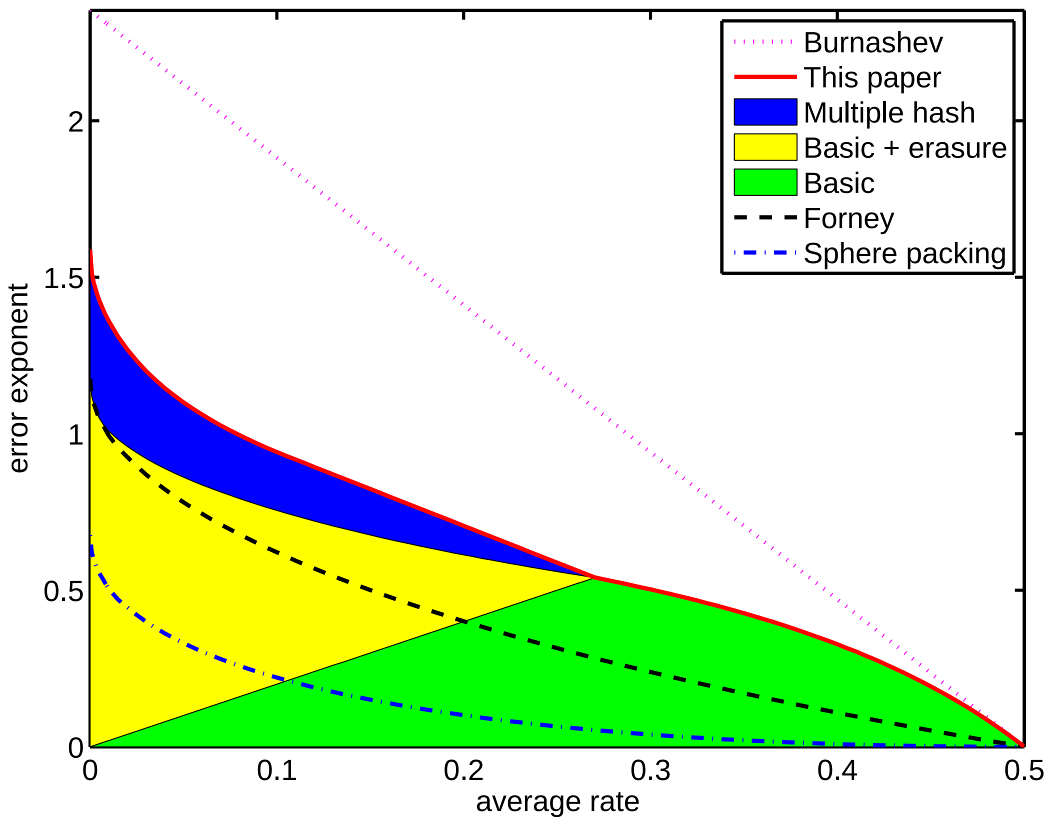

Figure 1. Diagram of the reliability functions for the three schemes presented in this paper. In this example the channel is the BSC with crossover probability 0.11, and the feedback $C_{fb} = 4C$.

In [11] Forney proved the following theorem.

Theorem 1 *There exists an average rate $\bar{R}$ noiseless-feedback variable-length coding strategy over DMC $p(y|x)$ satisfying $E_{vl}(\bar{R}) \geq E_{forn}(\bar{R})$ where*

$$E_{forn}(\bar{R}) = E_{sp}(\bar{R}) + C(1 - \bar{R}/C), \; 0 \leq \bar{R} \leq C, \quad (6)$$

and where $E_{sp}(\cdot)$ is the sphere-packing bound [12] of the channel.

In [7] Telatar provides an erasure decoding rule that leads to a higher reliability function than Forney's. However, for totally symmetric channels his result reduces to Forney's (6). Telatar further demonstrated that his exponent is tight at zero rate. For a universal erasure decoder see [5].

In [2] Burnashev provides the following bound on $E_{vl}(\bar{R})$.

Theorem 2 *For any average rate $\bar{R}$ noiseless-feedback variable-length coding strategy over DMC $p(y|x)$, $E_{vl}(\bar{R}) \leq E_{burn}(\bar{R})$ where*

$$E_{burn}(\bar{R}) = C_1 \left(1 - \frac{\bar{R}}{C}\right). \quad (7)$$

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

$$C_1 = \max_{x_i, x_j} \sum_y p(y|x_i) \log \frac{p(y|x_i)}{p(y|x_j)}. \quad (8)$$

and C is the channel capacity.

To connect Theorems 1 and 2, one should consider Forney's result to be an achievability result for fixed block-length erasure decoding *without* feedback (shown to be tight by Telatar at zero rate). On the other hand, Burnashev's result provides a tight upper bound on fixed block-length erasure decoding *with* feedback.[§] Effectively, in this paper we use erasure decisions to trigger retransmissions via a noisy feedback link. Of course, in certain settings one may not be interested in retransmissions and the erasure-decoding story stands on its own.

As mentioned in the introduction, Forney's scheme tests the likelihoods of the codewords. If the maximum likelihood (ML) codeword is not sufficiently more likely than the rest, the message is retransmitted. The decision whether to retransmit is made by the destination alone. In contrast, Burnashev's approach moves the decision whether to retransmit to the source. It improves upon the Forney reliability by attempting to recognize noise events that cannot be detected at the destination. An example would be when the channel noise moves the observation into the typical noise sphere surrounding a non-transmitted codeword. The source knows the codeword sent and can therefore detect such errors. The source follows up by signaling to the destination that a mistaken decoding decision is pending by transmitting a NAK. Conversely, it ACK-s correct decisions. Since this confirm/deny message is binary its detection can be extremely reliable.

In the place of Burnashev's scheme, we describe a conceptually simpler strategy due to Yamamoto and Itoh [22]. This scheme achieves the Burnashev bound asymptotically in block-length N. The strategy also nicely illustrates the mechanism that improves the reliability function beyond what is possible without feedback. The Yamamoto-Itoh strategy (and Burnashev's also) is a two phase scheme. In the first phase a message is sent to the destination at a rate slightly below capacity, achieving a somewhat small probability of error. For example, in [22] a length-λN block code is used where $0 < \lambda < 1$. Via the feedback, at time λN the source knows the destination's ML estimate or "tentative decision". The source follows up with a confirm/deny (ACK/NAK) signal. This signal is $(1-\lambda)N$ repetitions of the x_{i^*} or the x_{j^*} symbol, respectively (cf. (8)). At the conclusion of the second phase the destination runs a binary hypothesis test to determine whether the confirm/deny signal is an ACK or a NAK. If the result is a NAK, the message is retransmitted (via the noiseless feedback the source is aware of the destination's decision). Otherwise, the destination commits to its tentative decision and the next message is sent.

Undetected errors can occur only if a deny signal is misdetected as a confirm. To achieve the exponent in (7) one skews the binary hypothesis test so that the probability of false alarm (declaring NAK when an ACK is transmitted) 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 [4].

Retransmissions only result from errors on the forward channel or false alarms. As N increases the likelihood of either event can be made as rare as desired. Therefore the probability of retransmission can be driven to zero. This means that the average rate of communication converges to the rate of communication in a single block.

3. Accommodating noisy feedback

The variable-length feedback schemes discussed in Section 2 all assume noiseless feedback. The focus of this paper is on noisy feedback. If we consider the Yamamoto-Itoh scheme from the perspective of noise in the feedback link three challenges arise:

1. How should the source decide whether to ACK or to NAK a message?
2. How do we keep source and destination synchronized?
3. How can we utilize the forward channel efficiently while waiting for feedback information?

The first issue arises because noise in the feedback link prevents the source from knowing exactly what the destination observed. Therefore, after transmitting its messages over the forward channel using a block code, the source does not immediately know whether to follow up with an ACK or a NAK. We address this issue by having the destination transmit an identification code (protected by a second block code) to the source over the noisy feedback link. This indicates to the source the destination's tentative decision and is sent immediately following the data transmission. After decoding the identification code the source decides whether the destination correctly decoded the last data transmission. It then sends its confirm/deny message.

The second issue arises because in practice a code is almost never intended for one-shot use. Rather, it is used repeatedly to encode a (possibly infinite) sequence of messages. For fixed-block-length codes, the decoder can decide which symbol belongs to which message

[§]One sees this by recognizing that for fixed block-lengths the Burnashev bound is an achievable error exponent in an erasures context by using a single-shot Yamamoto-Itoh [22] strategy. Conversely, if any better errors/erasures trade off were possible with feedback, that scheme could be used to implement a retransmission strategy. The result would be a variable-length code with a reliability function that would exceed Burnashev's.

without looking at the symbol values. No synchronization issues arise. However, if the block length is variable the receiver must parse into messages the unending stream of received symbols. In lossless source-coding, for example, the requirement that the code be self-punctuating manifests itself at the single block level through the explicit requirement of unique-decodability or equivalently that the code is prefix-free. For channel coding, the counterpart is that the decoder must be able to decide how to parse the stream of channel outputs based upon the outputs themselves. With noiseless feedback, the requirement can be expressed at the single block level by requiring the block stopping time to be a causal function of the channel outputs.

When the feedback is noisy, it seems hopeless to try to maintain synchronization as an absolute requirement. Unless either the forward or feedback channels have positive zero-undetected-error capacity, the only way absolute synchronization can be achieved is through fixed-block-length codes. Consequently, we relax the design constraint of perfect synchronization at the single (variable-length) block level. Instead, we recognize the potential loss of synchronization to be a source of decoding error, and bound its contribution to the error probability. In addition we introduce a mechanism for reestablishing synchronization once lost. Without such a mechanism the source and destination would always eventually fall out of synch.

To maintain synchronization in the context of Yamamoto-Itoh with noisy feedback, the source must determine how each of its confirm/deny transmissions is decoded by the destination. If the source makes a mistake here, it will fall out of synch with the destination. For example, the source might transmit a new message while the destination is expecting a retransmission. We keep source and destination keep in synch by using a very low-rate anytime code [15, 18] to appraise the source of the destination's sequence of ACK/NAK decisions. With high probability the anytime code keeps the two discussing the same message. Just as importantly, it recovers from any out-of-synch events with increasing probability as time passes.

The third issue arises because while the feedback information is being transmitted along the reverse link, the forward link should not lie idle. If it did those degrees of freedom would be wasted. We interleave messages to maintain an average utilization of the forward link. These messages could either come from different "users" time-sharing the forward link, or could be a succession of messages coming from a single user.

4. Protocols for noisy feedback

In this section we give details of the protocols for noisy feedback. In Section 4.1 we present our basic protocol that addresses the challenges detailed in Section 3. We extend the basic protocol in Sections 4.2 and 4.3 to improve, respectively, performance when the feedback capacity is small and when the communication rate is low. The derivations of protocol performance are given in the Appendix. The underlying ideas and resultant reliability functions are discussed in the main text next.

4.1. Basic protocol

The basic strategy operates across a sequence of length-N time slots. Each time slot is partitioned into three segments of lengths $\gamma\lambda N$, $(1-\lambda)N$, and $(1-\gamma)\lambda N$, respectively, where $0 \leq \lambda, \gamma \leq 1$ are design parameters. This partitioning is illustrated in Figure 2. The first and third segments are used to transmit a length-λN rate-R_{data} block code while the middle segment is used for sending an ACK/NAK signal. At the end of each time slot the destination's tentative decision is its ML message estimate $\hat{m} \in \{1, 2, \ldots, 2^{\lambda N R_{data}}\}$.

The reason for the (possibly) non-contiguous transmission of the block code – in segments one and three – is that the confirm/deny message should be sent as soon as possible. This minimizes the user's transmission duration and increases the error exponent. Say that the feedback channel is far better than the forward channel. Then, the information required by the source for deciding whether to retransmit can be available long before the transmission of the new codeword wraps up. There is no reason to postpone transmission of the confirm/deny message, and hence it may be transmitted mid-time-slot.

An important observation is that the source does not need to know $\hat{m}$ to decide whether to transmit. It only needs to know whether the destination's tentative decision is correct, i.e., if it matches the message sent. This is an identification problem [8, 10, 3]. To address the capacity limitations of the feedback link we encode the tentative decision into an identification code. To deal with the noise on the feedback link the output of the identification code is encoded using a forward error-correcting block code. The resulting codeword is transmitted over the feedback channel in the first $\gamma\lambda N$ channel uses of the next time slot.

For the identification code we use a binning encoder per Slepian-Wolf [21]. We assume the existence of unlimited common randomness between encoder and decoder. This means that from time slot to time slot binning functions are statistically independent. Statistical independence of

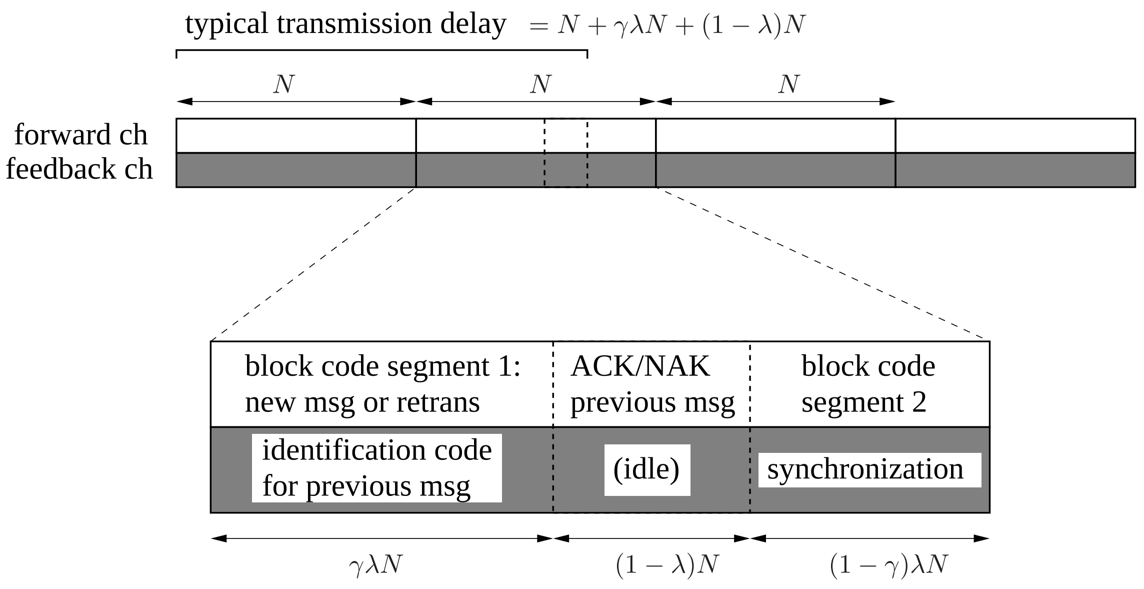

Figure 2. Diagram of interleaving data messages and confirm/deny messages on the forward channel and identification and synchronization messages on the feedback channel. A single data message is transmitted in each length-N time slot. However, the block code is divided into two segments. The feedback link lies idle for the middle $(1-\lambda)N$ channel uses of each time slot.

the binning functions greatly simplifies the error analysis.[¶] The $2^{\lambda N R_{data}}$ messages are randomly partitioned into $2^{\gamma\lambda N R_h}$ bins $\mathcal{B}_1 \dots \mathcal{B}_{2^{\gamma\lambda N R_h}}$. The index k of the bin such that $\hat{m} \in \mathcal{B}_k$ is then encoded and transmitted along the reverse channel using a length-$\gamma\lambda N$ block code. The source decodes the feedback message to $\hat{k}$. For compactness, we refer to the message bin index as the message "hash".

If $m \in \mathcal{B}_{\hat{k}}$ (i.e., if the transmitted message and the tentative decision are in the same bin) the source uses the second segment of the appropriate times slot (see Fig. 2) to transmit the same length-$(1-\lambda)N$ ACK message as is used in Yamamoto-Itoh. Otherwise it transmits the NAK message. If the decoded confirm/deny message is a NAK the destination expects a retransmission. If it is an ACK the destination finalizes its tentative decision, terminating the reception of that message. The duration of transmission is calculated from the time the block code corresponding to the message first begins transmission to the time the destination detects an ACK for that message.

To maintain synchronization the destination uses an anytime code in conjunction with a round-robin scheduling of time slots among L "virtual users". (See [17, 10] for earlier variants of this strategy.) Each data message is considered to come from a particular user. After the transmission of a message, the user waits $(L-1)N$ channel uses for its next transmission opportunity. In the meantime, in the final $(1-\gamma)\lambda N$ channel uses of each time slot the destination is transmitting an anytime code to the source, indicating the ACK/NAK decisions the destination has made to this point. The anytime code takes as input the sequence of the one-bit ACK/NAK decisions for *all* users' messages.

Although not immediately highly reliable, the outcome of any particular ACK/NAK decision is learned by the source with increasing reliability over time. If retransmission opportunities are spaced sufficiently far apart, i.e., if LN is large enough, by the time the first opportunity to retransmit the message arises the source knows with high reliability whether or not it needs to retransmit. In Appendix .1.1 we show that synchronization can be maintained with arbitrary reliability and, once lost, is soon reestablished. The timing of the synchronization protocol is illustrated in Figure 3.

In Appendix .1.1 we bound the error probability of this scheme as $\Pr[\text{error}] <$

$$2^{-(1-\lambda)C_1 N} + 2 \cdot 2^{-\gamma\lambda R_h N} + \Pr[\text{not synched}]. \quad (9)$$

The first term results from a missed NAK. The second term arises from undetected errors resulting from hash

¶It is known that identification codes with unlimited common randomness have infinite identification capacity [13]. This is in contrast with the original results on identification codes [1] that require encoder randomization but assume no randomness is shared with the decoder.

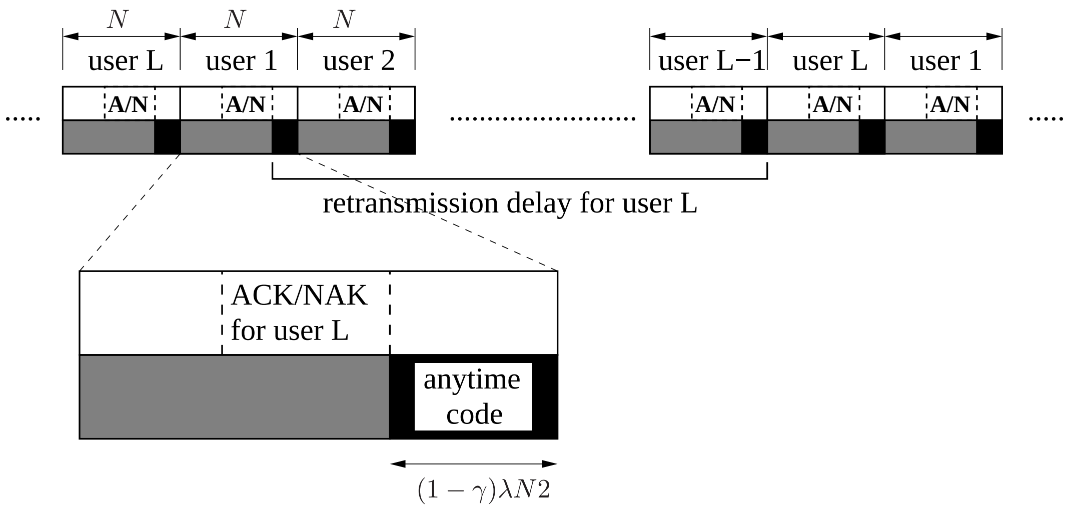

Figure 3. Transmission rotates in a round-robin fashion among L users. Each message is assigned a particular user who must wait till his next time slot to retransmit. By then the bit indicating to the source whether or not the destination received a NAK (and expects a retransmission) has been in the anytime code for $(L-1)(1-\gamma)\lambda N$ channel uses.

collisions. These comes about in two ways. The destination may make a decoding error, but the hash of the (erroneous) tentative decision matches the hash of the transmitted codeword. If this occurs the source would not detect the error. Alternately, while the hashes may not match when initially calculated, a false match may occur because of noise in the feedback link. In such cases the hash corresponding to the ML decoding of the feedback message happens to match that of the true message and the source does not detect the error. The final term is the probability of encoder and decoder being out of synch. As is shown in the appendix, the last term can be made as small as either of the first two resulting in the limiting error exponent

$$\lim_{N\to\infty} -\frac{1}{N}\Pr[\text{error}] = \min\{(1-\lambda)C_1, \gamma\lambda R_h\}. \quad (10)$$

In Appendix .1.2 we also bound the expected transmission duration as being bounded above by

$$E[\Delta] < N + \gamma\lambda N + (1-\lambda)N + \frac{2\epsilon_{FA}LN}{1-2\epsilon_{FA}}(1+o(1)).$$

The false alarm probability ϵ_{FA} is the probability of an ACK being mis-detected as a NAK, while LN is the retransmission delay. As is discussed in the appendix, the false-alarm probability ϵ_{FA} can be made to decrease exponentially in N, leading to an asymptotically approachable expected delay

$$\lim_{N\to\infty} \frac{1}{N}E[\Delta] = 1 + \gamma\lambda + (1-\lambda). \quad (11)$$

Together (10) and (11) define the limiting error exponent for the scheme as

$$\lim_{N\to\infty} \frac{-\log\Pr[\text{error}]}{E[\Delta]} = \frac{\min\{(1-\lambda)C_1, \gamma\lambda R_h\}}{1+\gamma\lambda+(1-\lambda)}. \quad (12)$$

The exponent is increased by using the largest possible R_h. We are limited to $R_h < C_{fb}$ else the rate of the identification code would exceed the capacity of the feedback link, resulting in frequent retransmissions. Set $R_h = C_{fb}$ to maximize the exponent. The reason the reliability function of the feedback channel doesn't come into play is that feedback errors lead to retransmissions with high probability. Only in the extremely rare case of a hash collision (cf. (9) does a feedback error likely lead to a undetected decoding error. Since the objective is to reduce hash collisions we choose R_h to approach C_{fb}.

Of the N channel uses in a time slot a fraction λ are dedicated to transmitting data giving a "per-slot" rate λR_{data}. As is shown in Appendix .1.1 the probability of retransmission is upper bounded by $2\epsilon_{FA}$. Since each transmission is independent the expected data rate

$$\bar{R} > \frac{\lambda R_{data}}{1-2\epsilon_{FA}}. \quad (13)$$

Recall that ϵ_{FA} can be made to decrease to zero exponentially quickly in N. As in Yamamoto-Itoh, the best reliability function is found by using a data rate R_{data} just below capacity. Setting $R_{data} = C$ and $\epsilon_{FA} = 0$ in (13) to evaluate the limit gives $\lambda = \bar{R}/C$ and the resulting reliability function

$$\frac{\min\{(1-\frac{\bar{R}}{C})C_1, \gamma C_{fb}\frac{\bar{R}}{C}\}}{1+\gamma\frac{\bar{R}}{C}+(1-\frac{\bar{R}}{C})} \quad \text{where } \bar{R} \in [0, C]. \tag{14}$$

The reliability function is increasing in γ until the the probability of missed a NAK (the first term in the numerator) equals the probability of hash collision (the second), or the maximum $\gamma = 1$ is reached. Therefore choose

$$\gamma = \min\left\{1, \left[\frac{C}{\bar{R}} - 1\right]\frac{C_1}{C_{fb}}\right\}. \tag{15}$$

We term the "high-rate" region the region such that $\gamma < 1$, i.e., such that

$$\bar{R} > \frac{C}{1+\frac{C_{fb}}{C_1}}. \tag{16}$$

In the high-rate region the bound on the reliability function simplifies to

$$\frac{\left(1-\frac{\bar{R}}{C}\right)C_1}{1+\left(1-\frac{\bar{R}}{C}\right)\left(1+\frac{C_1}{C_{fb}}\right)}. \tag{17}$$

In the high-rate region the difference between the Burnashev exponent and that achieved by this strategy lies in the denominator of (17). As the average rate of communication $\bar{R}$ approaches the capacity of the forward channel C the denominator approaches unity. Therefore, reliability function of this scheme approaches Burnashev's as $\bar{R}$ approaches C. The feedback capacity C_{fb} only impacts the error exponent through a term that behaves quadratically in $(1-\frac{\bar{R}}{C})$. One should note that it is in the high-rate region, where $\gamma < 1$, that splitting the transmission of a message into the first and third segments of its time slot becomes important. Without such an allocation the reliability function would not approach capacity with the same slope as Burnashev's bound. This is a major refinement compared with the earlier scheme presented in [10].

Figure 1 plots the reliability function of the "basic" scheme for a BSC with crossover probability 0.11 and $C_{fb} = 4C$. Note that the reliability function increases as $\bar{R}$ decreases from C to about 0.27, after which the reliability starts to decrease. As we discuss next, this decrease is an artifact of the scheme.

The basic scheme has two deficiencies we address in subsequent sections. First, at low rates the Forney exponent is higher than that of the basic scheme, even though the Forney approach requires only a single (albeit reliable) bit of feedback. In the next section we address this by allowing the destination to demand a retransmission (via erasure decoding) rather than always leaving that decision to the source. Second, the reliability function decreases to zero even as the rate decreases (i.e., as λ goes to zero) because the fraction of the time slot in which the identification code is transmitted (equal to $\gamma\lambda$, see Fig. 2) is decreasing. We remedy this in Section 4.3 by transmitting multiple hashes per message, which allows us to gainfully employ the feedback channel during the periods in which it is idle in the basic scheme (the segment marked "idle" in Fig. 2).

4.2. *Adding an erasure option*

An immediate refinement to the basic solution that has a large effect when the feedback capacity C_{fb} is small comes by allowing the destination an erasure option. At the end of each block transmission instead of maximum likelihood decoding, the destination performs erasure decoding. If the decision turns out "erasure" the destination expects a retransmission regardless of the outcome of the source's confirm/deny message. Erasure decisions are indicated to the source via the synchronization messages. For transmission of a message to end, and therefore for a decoding error to be made, a NAK must not be detected *and* an erasure must not be declared. In Appendix .2 we calculate the effect on the error probability and expected transmission duration. The derivation is only slightly different from the basic protocol.

The result of that analysis is that an erasure-decoding error exponent $E_{forn}(\cdot)$ is added to the numerator of (12) giving $\lim_{N\to\infty} \frac{-\log \Pr[\text{error}]}{E[\Delta]} =$

$$\frac{E_{forn}(R_{data}) + \min\{(1-\lambda)C_1, \gamma\lambda R_h\}}{1+\gamma\lambda+(1-\lambda)}. \tag{18}$$

There are $\lambda N R_{data}$ bits transmitted per time slot. In the limit of large N the retransmission probability goes to zero so $\lim_{N\to\infty} \bar{R} = \lambda R_{data}$. We substitute $\lambda = \bar{R}/R_{data}$ into (18). As before we evaluate the function at $R_h = C_{fb}$ getting

$$\frac{E_{forn}(R_{data}) + \min\{(1-\frac{\bar{R}}{R_{data}})C_1, \gamma\frac{\bar{R}}{R_{data}}C_{fb}\}}{1+\gamma\frac{\bar{R}}{R_{data}}+(1-\frac{\bar{R}}{R_{data}})} \tag{19}$$

where $0 \le \bar{R} \le R_{data} \le C$.

One can immediately see the improvement enabled by this scheme when the feedback rate C_{fb} is small. While

C_{fb} limits the second term in the numerator, it does not effect the contribution of Forney's exponent.

For example, letting $\bar{R} = R_{data}$ (in other words $\lambda = 1$ so that there is no confirm/deny phase) and $\gamma = 0$ gives a scheme that achieves the Forney erasure-decoding error exponent with *any* amount of feedback. The available feedback is used to run the anytime code that maintains synchronization. Since the rate of this synchronization code can be made arbitrarily small, C_{fb} can also be arbitrarily small. The scheme is now totally decoder-driven since the decision of whether or not to retransmit is being encoded into the synch information (of negligible size).

The curve labelled "basic + erasure" in Figure 1 illustrates how the major improvement stemming from the erasure option comes at low rates. However, the feedback channel is still idle a fraction $(1-\lambda)$ of the time (cf. Figure 2). In the next section we make use of these channel uses to further improve the exponent at low rate.

4.3. *Multiple hashes gives a low-rate improvement*

In this section we show how to delay retransmission decisions so as to exploit the idle period of the feedback channel. Instead of necessarily sending the confirm/deny message in the first confirm/deny slot after the completion of data transmission, we send it at the kth where k is a positive integer. This allows us to send multiple hashes per message (and multiple messages per hash). Every hash must match for an ACK to be sent. The key engineering assumption behind this section is that the source has access to multiple messages to transmit even before the first message has been accepted at the destination. This means that the inter-arrival time of messages is shorter than the acceptable average latency on a single message.[‖]

To simplify the analysis, first consider the requirement of the synch messages as N gets large. To this point we have assumed that the entire third segment of each time slot is allocated to synchronization. However, as analysis of Appendix .1.1 shows, and one can deal with a shorter third segment (larger γ) by performing the round-robin scheduling depicted in Fig. 3 to include more users. As N and L grow, the fraction of feedback channel uses allocated to synchronization can be allowed to shrink till, in the limit of large N it is negligible. Therefore, for simplicity of presentation we assume all N channel uses of each time slot can be allocated to hash messages.

[‖]While this moves in the direction of [9], the difference is that here we do not assume that the inter-arrival time of messages is asymptotically insignificant relative to the acceptable latency on a single message. Instead, the two are comparable to each other and related through k.

Under this assumption, each message is hashed $k \geq 1$ time by length-$\gamma\lambda N$ hashes, and $k-1$ times by length-$[(1-\lambda)+(1-\gamma)\lambda]N = [1-\gamma\lambda]N$ hashes. The protocol for $k = 3$ is diagrammed in Fig. 4.

In Appendix .3 we derive the following asymptotically approachable error exponent for the scheme:

$$\frac{E_{forn}(R_{data}) + \min\{C_1(1-\lambda), R_h(\gamma\lambda + k - 1)\}}{k + \gamma\lambda + (1-\lambda)},$$

where $k \in \mathbb{Z}_+$ and $\gamma \in (0,1]$. As in Section 4.2 the exponent is maximized by choosing $R_h = C_{fb}$ and $\lambda = \bar{R}/R_{data}$.

A slightly modified analysis is necessary for $\gamma = 0$. If $\gamma = 0$ each block code is not split across the confirm/deny signal. Referring to Fig. 4, when $\gamma = 0$ the transmission of m_i does not commence at time a, but rather time b. This means that the transmission duration τ is smaller by $(1-\lambda)N$, and the error exponent is

$$\frac{E_{forn}(R_{data}) + \min\{C_1(1-\lambda), (k-1)R_h\}}{k} \tag{20}$$

where $k \in \mathbb{Z}_+, k \geq 2$. For $k = 2$ and $R_{data} = C$ this is the result presented in [10].

To evaluate the best exponent, again set $R_h = C_{fb}$ and $\lambda = \bar{R}/R_{data}$. The resulting exponent is given in (21), where $0 \leq \bar{R} \leq R_{data} \leq C$.

We note that neither the first nor the second term of (21) dominates in all regimes. For instance consider a BSC with crossover probability 0.1 and $C_{fb} = C/4$. In the regime $\bar{R} \in [0.435, 0.48]$ the second term is larger with $k \in [2,3,4]$. While in the regime $\bar{R} \in [0.48, 0.505]$ the first term dominates, and above 0.505 the first term is larger for a short range once again.

The zero-rate exponent is calculable in closed form. Let both $\bar{R}$ and R_{data} go to zero with the ratio $\bar{R}/R_{data}$ also going to zero. The exponent becomes

$$\max_{k \in \mathbb{Z}_+} \frac{E_{forn}(0) + \min\{C_1, k C_{fb}\}}{k+1}. \tag{22}$$

In the case where the feedback channel is sufficiently large, $C_{fb} \geq C_1$, we find the maximum attainable exponent of our scheme to be

$$\frac{1}{2}\left[E_{forn}(0) + C_1\right]. \tag{23}$$

As an example, Telatar [7] proves that the zero-rate error exponent of erasure decoding equals

$$\max_{p_{\mathsf{x}}(\cdot)} \sum_{x} \sum_{x'} \sum_{y} p_{\mathsf{x}}(x) p_{\mathsf{x}}(x') p_{\mathsf{y}|\mathsf{x}}(y|x') \log \frac{p_{\mathsf{y}|\mathsf{x}}(y|x')}{p_{\mathsf{y}|\mathsf{x}}(y|x)}. \tag{24}$$

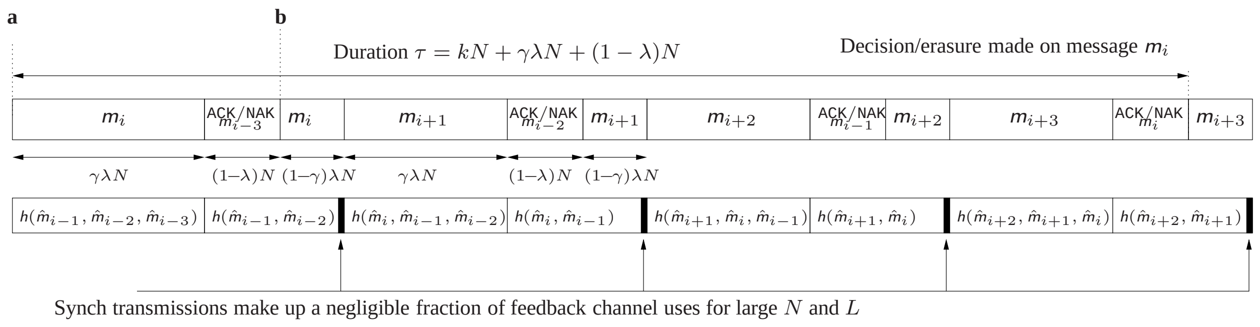
a
b
Duration $\tau = kN + \gamma\lambda N + (1-\lambda)N$
Decision/erasure made on message m_i
m_i
ACK/NAK m_{i-3}
m_{i+1}
ACK/NAK m_{i-2}
m_{i+2}
ACK/NAK m_{i-1}
m_{i+3}
ACK/NAK m_i
$\gamma\lambda N$
$(1-\lambda)N$
$(1-\gamma)\lambda N$
$h(\hat{m}_{i-1}, \hat{m}_{i-2}, \hat{m}_{i-3})$
$h(\hat{m}_{i-1}, \hat{m}_{i-2})$
$h(\hat{m}_i, \hat{m}_{i-1}, \hat{m}_{i-2})$
$h(\hat{m}_i, \hat{m}_{i-1})$
$h(\hat{m}_{i+1}, \hat{m}_i, \hat{m}_{i-1})$
$h(\hat{m}_{i+1}, \hat{m}_i)$
$h(\hat{m}_{i+2}, \hat{m}_{i+1}, \hat{m}_i)$
$h(\hat{m}_{i+2}, \hat{m}_{i+1})$
Synch transmissions make up a negligible fraction of feedback channel uses for large N and L

Figure 4. Diagram of interleaving of user data and NAKs on the forward channel and the corresponding identification coding on the reverse channel.

$$\max_{k\in\mathbb{Z}_+,\gamma\in[0,1]} \max\left\{ \frac{E_{forn}(R_{data}) + \min\{C_1(1-\frac{\bar{R}}{R_{data}}), C_{fb}(\gamma\frac{\bar{R}}{R_{data}} + k - 1)\}}{k + \gamma\frac{\bar{R}}{R_{data}} + (1-\frac{\bar{R}}{R_{data}})}, \frac{E_{forn}(R_{data}) + \min\{C_1(1-\frac{\bar{R}}{R_{data}}), kC_{fb}\}}{k+1} \right\}, \quad (21)$$

where $p_{\mathsf{x}}(\cdot)$ is the user-chosen input distribution and $p_{\mathsf{y}|\mathsf{x}}(\cdot|\cdot)$ is the channel law. For the BSC with cross-over probability p the zero-rate exponent (24) reduces to

$$E_{forn}(0) = 0.5D(p\|1-p) \quad (25)$$

which is *half* Burnashev's zero-rate exponent C_1. Hence for a BSC, if $C_{fb} \geq C_1$ our exponent equals $0.75C_1$. The improvement of the multiple-hashes approach is illustrated in Figure 1.

4.4. Illustrative results

Figure 5 plots the results of this paper for a BSC with crossover probability 0.1 and for two different feedback capacities. We plot our results for the multiple hashing strategy although, in some cases, depending on the C_{fb} and the average communication rate, the scheme will reduce to either the basic scheme or the basic plus erasure.

For $C_{fb} = 5C$ at low rates the multiple hash technique is used. On the other hand, at high rate the strategy reduces to the basic scheme. The break-point between the two occurs at roughly $\bar{R} = 0.24$ where there is a kink in the reliability function. This is analogous to the transition between the two strategies in Figure 1 which occurs at roughly $\bar{R} = 0.27$ (recall that in that plot $C_{fb} = 4C$).

In contrast, when the feedback channel is more noisy, $C_{fb} = C$, each strategy is used in distinct rate regions. When $\bar{R}$ is small (roughly $\bar{R} < 0.2$) the erasure decoder is used and the reliability function matches Forney's. At high rates (roughly $\bar{R} > 0.43$) the basic scheme dominates. In between (roughly $0.2 < \bar{R} < 0.43$) the multiple hash technique is used. The two kinks in the curve in this intermediate region result from the integer effects discussed in (20).

5. Conclusions

In this paper we have shown that the improvements in reliability functions resulting from variable-length coding are not fragile with respect to noisy feedback. We demonstrate three major performance points. First, the performance of the Burnashev bound is approachable even with a noisy feedback link as the rate of communication approaches capacity. Second, the Forney exponent is attainable even if given a tiny feedback capacity. This implies that given a two-way channel whose objective is one-way transmission, the reliability function benefits significantly from allocating some degrees of freedom to feedback. Third, if the feedback link is capable enough, at lower rates we can outperform Forney, sometimes significantly.

One key component at all rates is to use an anytime code to maintain synchronization between source and destination, and to encode jointly the synchronization information to all "users" that share the link to the

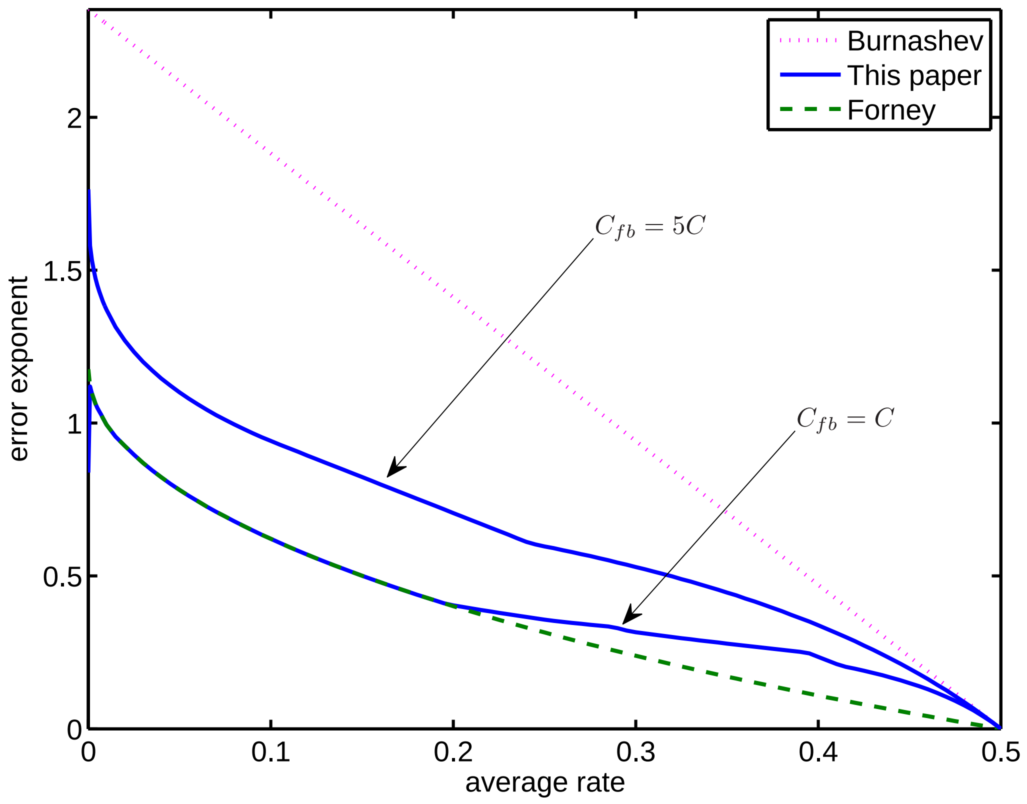

Figure 5. Exponent comparison for a the binary-symmetric channel (BSC) with crossover probability 0.11: Burnashev, Forney, and sample results from this paper. For $C_{fb} = 5C$ at low $\bar{R}$ the multiple hashing technique is used while at high rates the basic technique is used. In contrast, for $C_{fb} = C$ at low rates the erasure decoding option is used, while multiple hashing is used at intermediate rates, and again the basic scheme at high rates.

destination. At high rates, there are two additional key innovations. First, we split the data transmission part of the message into two segments. This allows the destination to make its decoding decision as soon as possible. At the same time we interleave messages to keep the transmission rate high. At lower-rates, we achieve better performance than classic erasure decoding by allowing for a soft combination of Forney-style decoder-driven retransmission and Burnashev-style encoder-driven retransmission. We further improve performance at low rates by sending multiple identification codes per data block. Only if all identification codes match does the source trust the destination's tentative decision.

.1. Basic Protocol

.1.1. Probability of Error We analyze the error probability by decomposing the error events into two root causes. The first cause is an erroneous tentative decision that the system does not NAK. The second cause is non-synchronization of source and destination.

In (26)–(27) we use the following abbreviations to denote the important events. The event "synched" ("not synched") refers to when source and destination are discussing the same (different) messages; "dest err" indicates the destination's tentative decision is incorrect; "NAK not det" means in a given time slot the destination did not detect a NAK (regardless of what was transmitted); "src detects" ("src doesn't detect") indicates that the source does (not) detect (for whatever reason) an erroneous decision; "FB err" ("no FB err") means that the block code that encodes the identification information was incorrectly (correctly) decoded by the source.

$$\begin{aligned}\Pr[\text{error}] &= \Pr[(\text{synched, dest err}, \mathsf{NAK}\text{ not det}) \cup (\text{not synched}, \mathsf{NAK}\text{ not det})] \quad (26)\\
&\le \Pr[\mathsf{NAK}\text{ not det}|\text{synched, dest err}] + \Pr[\text{not synched}]\\
&= \Pr[\mathsf{NAK}\text{ not det, src detects}|\text{synched, dest err}]\\
&\quad + \Pr[\mathsf{NAK}\text{ not det, src doesn't detect}|\text{synched, dest err}] + \Pr[\text{not synched}]\\
&\le \Pr[\mathsf{NAK}\text{ not det}|\text{src detects, synched, dest err}]\\
&\quad + \Pr[\text{src doesn't detect}|\text{synched, dest err}] + \Pr[\text{not synched}]\\
&= \Pr[\mathsf{NAK}\text{ not det}|\text{src detects, synched, dest err}]\\
&\quad + \Pr[\text{src doesn't detect, no FB err}|\text{synched, dest err}]\\
&\quad + \Pr[\text{src doesn't detect, FB err}|\text{synched, dest err}] + \Pr[\text{not synched}]\\
&\le \Pr[\mathsf{NAK}\text{ not det}|\text{src detects, synched, dest err}]\\
&\quad + \Pr[\text{src doesn't detect}|\text{no FB err, synched, dest err}]\\
&\quad + \Pr[\text{src doesn't detect}|\text{FB err, synched, dest err}] + \Pr[\text{not synched}] \quad (27)\end{aligned}$$

We bound (27) as

$$\log \Pr[\text{error}]$$
$$\le \log\left\{2^{-(1-\lambda)NC_1} + 2\times 2^{-\gamma\lambda N R_h} + \Pr[\text{not synched}]\right\} \quad (28)$$
$$\times (1+o(1)). \quad (29)$$

The first term in (28) is the only term that exists in the noiseless setting (e.g., in [22]). The ACK (NAK) sequence is $(1-\lambda)N$ repetitions of the x_i (x_j) symbol determined by the optimization in (8). The destination uses a skewed binary hypothesis test. Stein's Lemma [4] tells us how to control the trade off between false-alarm and missed-detection. We want to minimize the probability of a NAK being mis-detected as an ACK (leading to a decoding error) while keeping a slowly vanishing bound $\epsilon_{FA} > 0$ on the false-alarm probability (an ACK being mis-detected as a NAK, which just leads to a retransmission). For an arbitrarily ϵ_{FA} we can approach a bound of $2^{-C_1(1-\lambda)N}$ on the probability of not detecting a NAK as N gets large. In fact, a slight generalization of Stein's Lemma due to Hoeffding (see, e.g., [5, Exercise 1.2.12]) shows that ϵ_{FA} can be made to decay exponentially in N with negligible effect on the decay of the missed-detection probability as long as the exponent of ϵ_{FA} is suitably small.

Half of the second term in (28) is the probability that the identification code fails. This occurs when the true message is in the same bin as the incorrect tentative decision, a "hash collision". Because the bin index of the true message and the tentative decision are uniformly and independently assigned, this probability equals $2^{-\gamma\lambda N R_h}$, the reciprocal of the number of bins.

The other half of the second term of (28) also results from a hash collision. However, the cause of this collision is different. The hash calculated at the destination may, in fact, not match the hash of the data message. However, because of an error on the feedback link the hash decoded by the source does not match the hash transmitted by the destination. The probability that the correct hash (unfortunately) matches this incorrectly decoded hash is also $2^{-\gamma\lambda N R_h}$.

It is worth commenting on this further. As R_h approaches the capacity of the feedback link, feedback errors occur *much* more frequently than hash collisions. Most of these events lead to retransmissions and not to undetected errors. As with the asymmetric hypothesis test in the confirm/deny phase, this is another point in the protocol where the asymmetry in event probabilities is central to the protocol's success.

In order to bound the third term of (28), as discussed in Sec. 4 we use an anytime code in conjunction with a round-robin scheduling of time slots among L "users". The rate of the anytime code is $\frac{1}{(1-\gamma)\lambda N}$ since each bit being communicated has $(1-\gamma)\lambda N$ feedback channel uses before the next bit arrives. To determine whether to retransmit an old message or transmit new data in a given time slot, the source estimates the destination's sequence of ACK/NAK decisions. If, after $L-1$ time slots the destination's confirm/deny decision – corresponding to the decision made L time slots back – is estimated to be an ACK (and other prior confirm/deny decision estimates remain unchanged) the source transmits a new message, else it retransmits. The most recent confirm/deny message relevant to a transmission/retransmission decision entered the anytime encoder $(L-1)(1-\gamma)\lambda N$ channel uses before this decision is made. The second most recent $(1-\gamma)\lambda N$ channel uses before that, and so forth. From [18, 16], we know that a random anytime code has a probability of incorrectly decoding a confirm/deny message made i time slots ago that decreases exponentially with delay. Let $\xi = 2^{-(1-\gamma)\lambda N E_{any}(1/(1-\gamma)\lambda N)}$. Applying the union bound reveals that the probability that any relevant confirm/deny messages are in error is bounded by

$$\sum_{i=L-1}^{\infty} \xi^i = \xi^{L-1} \frac{1}{1-\xi} \tag{30}$$

where the relevant E_{any} is the random-coding error exponent essentially evaluated at zero-rate since $\frac{1}{(1-\gamma)\lambda N}$ is so small.

If previous confirm/deny decisions are changed then the source re-jigs its transmission schedule to get back into synch with the destination. If the re-jigging extends more than L time slots back, there is likely to have been a burst of incorrect decodings. Such out-of-synch events can be made as rare as desired by choosing L sufficiently large.

It is important to note that error detection and synchronization operate on two distinct time scales. The source needs to detect errors immediately so as to be able to NAK promptly. This time-scale dominates the expected transmission duration. In contrast, retransmissions operate on a much longer time scale. As long as retransmission are suitably rare, their effect on the expected decoding duration is negligible.

Combining (30) with (28) gives

$$\Pr[\text{error}] \leq 2^{-(1-\lambda)NC_1} + 2 \cdot 2^{-\gamma\lambda N R_h} + \xi^{L-1}\frac{1}{1-\xi}. \tag{31}$$

Select L large enough to balance the last term in (31) with the first two. This is always possible if $\lambda > 0$ and $\gamma < 1$. Note also that the resulting L is not a function of N. This means that

$$\lim_{N\to\infty} -\frac{1}{N}\log \Pr[\text{error}] = \min\{(1-\lambda)C_1, \gamma\lambda R_h\}. \tag{32}$$

.1.2. Transmission duration We now bound the expected transmission delay. Transmission begins when a message first shows up at the channel input. Transmission ends when (due to whatever cause) the destination detects an ACK for that message. The time difference between the two events is the transmission delay.

It suffices to think in terms a single user's message stream. Let k be the message number. So $k=1$ is the first message for this user, $k=2$ is the second message, and so on. It is useful to consider time in terms of slots where 1 slot is of length $N+\gamma\lambda N+(1-\lambda)N$ and the slots occur every LN channel uses. Let d_k be the first time slot in which the destination is listening for message k. Because of out-of-synch events this is not necessarily the same time slot in which the source initially transmits that message. We use s_k to be this time slot, i.e., the first time slot in which message k is transmitted by the source. If, for instance, $s_k < d_k$ the source leads the destination. If $s_k > d_k$ the source lags the destination. If $s_k = d_k$ source and destination are in synch. Finally, let a_k be the slot in which message k is accepted by the destination. The natural definition of the transmission delay for message k is $\Delta_k = N+\gamma\lambda N+(1-\lambda)N+LN(a_k - s_k)$.

In the vast majority of cases, the transmission delay as defined above is positive. However, there are very rare occasions when, due to temporary loss of synchronization, the destination accepts a message before the source ever attempted to transmit it. These events are already figured into the error calculation of the previous section. But, to prevent such rare events from skewing the expected transmission duration, we define the transmission duration of the kth message to be

$$\Delta_k = N+\gamma\lambda N+(1-\lambda)N+LN\max(0, a_k - s_k).$$

First observe that $d_{k+1} = a_k + 1$. The destination deterministically increments by one its message count after each acceptance. Observe also that a message is accepted whenever an ACK is detected. Because the channels are assumed to be memoryless, and hash functions and codebooks are re-randomized at every slot, the differences $d_{k+1} - d_k = a_k + 1 - d_k$ are independent geometric random variables and independent of the transmission time.

To understand the probability of detecting an ACK, it is easier to calculate:

$$\begin{aligned}
&\Pr[\textsf{NAK det}]\\
&=\Pr[\textsf{NAK det, dest no err}]+\Pr[\textsf{NAK det, dest err}] && (33)\\
&\le \Pr[\textsf{NAK det, dest no err, no FB err}]\\
&\quad+\Pr[\textsf{NAK det, dest no err, FB err}]\\
&\quad+\Pr[\textsf{dest err}] && (34)\\
&\le \Pr[\textsf{NAK det}|\textsf{dest no err, no FB err}]\\
&\quad+\Pr[\textsf{FB err}]+\Pr[\textsf{dest err}]\\
&\le \epsilon_{FA}+2^{-\gamma\lambda N E_r(R_h)}+2^{-\lambda N E_f(R_{data})} && (35)\\
&\le 2\,\epsilon_{FA}. && (36)
\end{aligned}$$

In (35), $E_f(\cdot)$ and $E_r(\cdot)$ are, respectively, the random coding error exponents of the forward and reverse channels. For (36) we limit $1>\lambda,\gamma>0$, $R<C$, and $R_h<C_{fb}$ where C and C_{fb} are the forward and feedback channel capacities, respectively. Under these conditions, the last two terms of (36) all go to zero as N is increased. Therefore, there exists a block length such that for all larger block lengths (36) holds.

Now, consider an arbitrary message k. The expected delay can be bounded as follows, In (37) we use Bayes rule with two identities. First, $\Pr[\mathsf{a}_k=a|\mathsf{d}_k=d]=\Pr[\mathsf{a}_k-\mathsf{d}_k=a-d|\mathsf{d}_k=d]$, we define $i=a-d$ and note that the length of time the destination listens is a geometrically distributed random variable that is independent of d_k since it only depends on the behavior of the random encoders and forward/feedback channels *after* the destination has started listening for this message.

To go from (37) to (38), we notice that $\max\{0,d+i-s\}\le i+\max\{0,d-s\}$. Let $s=d-l$ where l is the number of time slots the source leads the destination. I.e., the number of time slots between that time slot where the source initially transmits message k and that time slot in which the destination starts listening for message k. To get rid of the max we increase the sum to all l.

For s_k to be l steps out of synch with the destination there must have been at least $|l|$ errors in the anytime decoding at time $s=d-l$. Making sure l is small with high probability is analogous to the problem of tracking unstable processes as described in [15, 20] and is why the anytime code is used. The first error must therefore be at least $|l|$ extra bits ago. By reasoning analogous to (30), for $l\neq 0$ the probability of such an error $\Pr[\mathsf{s}_k=d-l|\mathsf{d}_k=d,\mathsf{a}_k=d+i]\le \xi^{|l|+L-1}\frac{1}{1-\xi}$ since the analysis of the random anytime code doesn't depend on the particular values of the message bits after the desired ones corresponding to $\mathsf{d}_k=d$. So, by the memoryless character of the feedback channel and the disjoint nature of the times lots used for the anytime code, the conditioning on $\mathsf{a}_k=d+i$ is irrelevant.

The derivation continues from (39).

To get to (40), we use that for $|\xi|<1$, the sum $\sum_{l=1}^{\infty}k\xi^k=\xi(1-\xi)^{-2}$ and bound $\frac{l}{i}<l$ for $i\ge 1$. This can be made as close to $1+\gamma\lambda+(1-\lambda)$ as desired by choosing ϵ_{FA} small enough and L large enough. Thus, the expected delay is effectively the length of a single slot.

.2. *Basic protocol with an erasure option*

The analysis of the basic protocol section needs only a slight modification to incorporate an erasure option. In the place of (27) we have (41). The event "undet err" now refers to an undetected error that occurs during erasure-decoding.

The use of an erasure decoding also minimally effects the transmission duration. Retransmission occur either if an erasure is declared or a NAK is detected. We modify (33) as

$$\begin{aligned}
&\Pr[\textsf{NAK det}\cup\textsf{erasure}]\\
&<\Pr[\textsf{NAK det}]+\Pr[\textsf{erasure}]\\
&<3\epsilon_{FA}, && (42)
\end{aligned}$$

where as before ϵ_{FA} can be arbitrarily small. The asymptotically approachable limit on the expected transmission duration therefore remains unchanged, i.e., it equals (11).

.3. *Multiple hashes*

In adapting the error calculation of (41) to multi-block hashing, only the second and third terms inside the parentheses are changed. Respectively, they are replaced by (43) and (44).

The event ≥ 1 FB err is that at least one block-encoded identification code is decoded in error. The constant $(2k-1)$ in (44) occurs because there are $(2k-1)$ hashes transmitted over the feedback link per message. Any of these can be in error. The probability of a missed NAK remains the same since each message is still allocated its own confirm/deny signal. Synchronization can be dealt with as before.

The probability of retransmission remains almost the same. Equation (42) becomes

$$\Pr[\textsf{NAK det}\cup\textsf{erasure}]<(2k+1)\epsilon_{FA}, \qquad (45)$$

$$\begin{aligned}
&\frac{E[\Delta_k]}{N} \\
&= 1 + \gamma\lambda + (1-\lambda) \\
&\quad + L \sum_{d=1}^{\infty} \Pr[d_k = d] \sum_{a=d}^{\infty} \sum_{s=1}^{\infty} \max\{0, a-s\} \Pr[a_k = a, s_k = s | d_k = d] \\
&= + \gamma\lambda + (1-\lambda) \\
&\quad + L \sum_{d=1}^{\infty} \Pr[d_k = d] \sum_{i=0}^{\infty} \sum_{s=1}^{\infty} \max\{0, d+i-s\} \Pr[a_k - d_k = i] \Pr[s_k = s | d_k = d, a_k = d+i] && (37) \\
&\le 1 + \gamma\lambda + (1-\lambda) \\
&\quad + L \sum_{d=1}^{\infty} \Pr[d_k = d] \sum_{i=0}^{\infty} \Pr[a_k - d_k = i] \sum_{s=1}^{\infty} (i + \max\{0, d-s\}) \Pr[s_k = s | d_k = d, a_k = d+i] && (38) \\
&\le 1 + \gamma\lambda + (1-\lambda) \\
&\quad + L \sum_{d=1}^{\infty} \Pr[d_k = d] \sum_{i=0}^{\infty} \Pr[a_k - d_k = i] \sum_{l=-\infty}^{\infty} (i + |l|) \Pr[s_k = d - l | d_k = d, a_k = d+i]. && (39)
\end{aligned}$$

$$\begin{aligned}
&\frac{E[\Delta_k]}{N} \\
&\le 1 + \gamma\lambda + (1-\lambda) + L \sum_{d=1}^{\infty} \Pr[d_k = d] \\
&\quad \times \left(2 \Pr[a_k - d_k = 0] \sum_{l=1}^{\infty} l \frac{\xi^{l+L-1}}{1-\xi} + \sum_{i=1}^{\infty} \Pr[a_k - d_k = i] i \left(1 + 2 \sum_{l=1}^{\infty} \frac{l}{i} \frac{\xi^{l+L-1}}{1-\xi} \right) \right) \\
&< 1 + \gamma\lambda + (1-\lambda) + L \sum_{d=1}^{\infty} \Pr[d_k = d] \\
&\quad \times \left(2\xi^L (1-\xi)^{-3} + \sum_{i=1}^{\infty} \Pr[a_k - d_k = i] i \left(1 + 2\xi^L (1-\xi)^{-3}\right) \right) \\
&\le 1 + \gamma\lambda + (1-\lambda) \\
&\quad + L \left(2\xi^L (1-\xi)^{-3} + \left(1 + 2\xi^L (1-\xi)^{-3}\right) \frac{2\epsilon_{FA}}{1 - 2\epsilon_{FA}} \right). && (40)
\end{aligned}$$

$$\begin{aligned}
\Pr[\text{error}] &= \Pr[\mathsf{NAK}\text{ not det}|\text{synched, undet err}] \Pr[\text{undet err}] + \Pr[\text{not synched}] \\
&\le \Pr[\text{undet err}] \Big(\Pr[\mathsf{NAK}\text{ not det}|\text{src detects, synched, undet err}] \\
&\quad + \Pr[\text{src doesn't detect}|\text{no FB err, synched, undet err}] \\
&\quad + \Pr[\text{src doesn't detect}|\text{FB err, synched, undet err}] \Big) + \Pr[\text{not synched}]. && (41)
\end{aligned}$$

$$\begin{aligned}
&\mathrm{Pr}[\text{src doesn't detect}|\text{no FB err, synched, undet err}] \\
&\qquad\qquad = 2^{-kN\gamma\lambda R_h} 2^{-(k-1)N(1-\gamma\lambda)R_h} = 2^{-\gamma\lambda N R_h} 2^{-NR_h(k-1)}. \qquad (43)\\
&\mathrm{Pr}[\text{src doesn't detect}|\geq 1 \text{ FB err, synched, undet err}] \\
&\qquad\qquad = (2k-1) 2^{-\gamma\lambda N R_h} 2^{-NR_h(k-1)}. \qquad (44)
\end{aligned}$$

because now there are $2k-1$ distinct hashes fed back from destination to source in addition to the probability of false alarm. The limiting normalized expected duration becomes $\lim_{N\to\infty} \frac{1}{N} E[\Delta] = 2 - \lambda(1-\gamma) + (k-1) = k + \gamma\lambda + (1-\lambda)$ resulting in the error exponent

$$\frac{E_{forn}(R_{data}) + \min\{C_1(1-\lambda), R_h(\gamma\lambda + k - 1)\}}{k + \gamma\lambda + (1-\lambda)}. \qquad (46)$$

REFERENCES

1. R. Ahlswede and G. Dueck. Identification via channels. *IEEE Trans. Inform. Theory*, 32:621–629, January 1989.
2. M. V. Burnashev. Data transmission over a discrete channel with feedback. Random transmission time. *Prob. Peredachi Informatsii*, 12(4):10–30, 1976.
3. A. Chawla. Reliability of a Gaussian channel in the presence of Gaussian feedback. Master's thesis, Mass. Instit. of Tech., 2006.
4. T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, 1991.
5. I. Csiszár and J. Körner. *Information Theory, Coding Theorems for Discrete Memoryless Systems*. Akadémiai Kiadó, 1981.
6. 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.
7. İ. E. Telatar. *Multi-Access Communication with Decision Feedback Decoding*. PhD thesis, Mass. Instit. of Tech., 1992.
8. 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*, September 2005.
9. S. C. Draper and A. Sahai. Beating the Burnashev bound using noisy feedback. In *Proc. 44th Allerton Conf. on Communication, Control and Computing*, September 2006.
10. S. C. Draper and A. Sahai. Noisy feedback improves communication reliability. In *Proc. Int. Symp. Inform. Theory*, July 2006.
11. G. D. Forney. Exponential error bounds for erasure, list, and decision feedback schemes. *IEEE Trans. Inform. Theory*, 14:206–220, March 1968.
12. R. G. Gallager. *Information Theory and Reliable Communication*. John Wiley and Sons, 1968.
13. T. S. Han and S. Verdu. New results in the theory of identification via channels. *IEEE Trans. Inform. Theory*, 38:14–25, January 1992.
14. R. W. Lucky. A survey of the communication theory literature: 1968–1973. *IEEE Trans. Inform. Theory*, 19:725–739, November 1973.
15. A. Sahai. *Anytime Information Theory*. PhD thesis, Mass. Instit. of Tech., 2001.
16. A. Sahai. Why block length and delay behave differently if feedback is present. *IEEE Trans. Inform. Theory*, To appear, 2008. Preprint in arXiv.org.
17. 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.
18. A. Sahai and S. K. Mitter. The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link. Part I: scalar systems. *IEEE Trans. Inform. Theory*, 52(8):3369–3395, August 2006.
19. Anant Sahai. Balancing forward and feedback error correction for erasure channels with unreliable feedback. *IEEE Trans. Inform. Theory*, Submitted, 2007. Available in arXiv.
20. Anant Sahai and Sanjoy K. Mitter. Source coding and channel requirements for unstable processes. *IEEE Trans. Inform. Theory*, Submitted, 2006. Preprint in arXiv.org.
21. D. Slepian and J. K. Wolf. Noiseless coding of correlated information sources. *IEEE Trans. Inform. Theory*, 19:471–480, July 1973.
22. 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.