Maximal Noise in Interactive Communication over Erasure Channels and Channels with Feedback

Klim Efremenko
UC Berkeley[*]
klimefrem@gmail.com

Ran Gelles
Princeton University[†]
rgelles@cs.princeton.edu

Bernhard Haeupler
CMU
haeupler@cs.cmu.edu

Abstract

We provide tight upper and lower bounds on the noise resilience of interactive communication over noisy channels with *feedback*. In this setting, we show that the maximal fraction of noise that any robust protocol can resist is 1/3. Additionally, we provide a simple and efficient robust protocol that succeeds as long as the fraction of noise is at most $1/3 - \varepsilon$. Surprisingly, both bounds hold regardless of whether the parties communicate via a binary or an arbitrarily large alphabet. This is contrasted with the bounds of 1/4 and 1/8 shown by Braverman and Rao (STOC '11) for the case of robust protocols over noisy channels without feedback, assuming a large (constant size) alphabet or a binary alphabet respectively.

We also consider interactive communication over *erasure* channels. We provide a protocol that matches the optimal tolerable erasure rate of $1/2 - \varepsilon$ of previous protocols (Franklin et al., CRYPTO '13) but operates in a much simpler and more efficient way. Our protocol works with an alphabet of size 6, in contrast to prior protocols in which the alphabet size grows as $\varepsilon \to 0$. Building on the above algorithm with *fixed* alphabet we are able to devise a protocol that works for *binary* erasure channels, improving the best previously known bound on the tolerable erasure rate from $1/4 - \varepsilon$ to $1/3 - \varepsilon$.

[*]Work done while at Univ. of Chicago.
[†]Work done while a student at UCLA.

1 Introduction

In the interactive communication setting, Alice and Bob are given inputs x and y respectively, and are required to compute and output some function $f(x, y)$ of their joint inputs. To this end, they exchange messages over a channel that may be noisy: up to an ε-fraction of the transmitted bits may be flipped during the communication. Due to the noise, there is a need for error correction and sophisticated coding schemes that will allow the parties to successfully conduct the computation, yet keep the communication complexity small, ideally at most linear in the communication complexity of computing the same function over a noiseless channel (hereinafter, we say that such a scheme has a *constant rate*).

Such coding schemes have been extensively explored, starting with the pioneering work of Schulman [Sch92, Sch96] who gave the first scheme to resist up to a 1/240-fraction of bit flips. Almost two decades later, the question was revisited by Braverman and Rao [BR11] who showed a coding scheme that successfully computes any function, as long as the fraction of corrupted transmissions is at most $1/4 - \varepsilon$. Furthermore, they show that it is impossible to resist noise of 1/4 or more, for a large and natural class of *robust* protocols. In robust protocols both parties are guaranteed to agree whose turn it is to speak at each round, regardless of the noise (see definition in Section 2 below). It should be noted that the above result of $1/4 - \varepsilon$ applies only when the parties send symbols from a large alphabet, whose size is growing with ε going to zero. When the parties are restricted to sending bits, the coding scheme of Braverman and Rao [BR11] tolerates up to a $(1/8 - \varepsilon)$-fraction of bit flips. The question of determining the maximal tolerable noise for binary channels is open.

In this paper we examine different types of communication channels and noise. Specifically, we consider *channels with feedback* and *erasure channels*. In the former it is assumed that after each transmission, the sender learns the (possibly corrupted) symbol received by the other side, i.e., there is noiseless feedback. In the erasure channel case, the noise can turn any symbol into an "erasure" (denoted $\perp$), but it cannot alter the transmission into a different valid symbol. Both erasure channels and channels with feedback have been studied in the classical one-way setting [Sha48, Ber64] albeit typically more from a perspective of optimizing communication rates.

For each of these channels we examine the maximal tolerable noise for interactive communication, both when the parties are restricted to sending bits and in the general case where they are allowed to send symbols from a larger alphabet.

1.1 Our Results

channel type	**alphabet**	**order of speaking**	**lower bound**	**upper bound**
feedback	large	fixed	1/4	1/4
feedback	binary	fixed	1/6	1/6
feedback	large & binary	arbitrary	1/3	1/3
erasure	large	fixed	1/2	1/2
erasure	binary	fixed	1/3	??

Table 1: A summary of the lower and upper bounds for the maximum tolerable error rate for all settings considered in this paper.

Interactive communication over channels with feedback. We completely solve the question of the maximal tolerable noise for robust interactive protocols over channels with feedback, both for the binary alphabet and the large alphabet case. We note that feedback helps with coordinating what party should speak at any point of time and therefore with keeping a protocol robust. Indeed, due to the feedback both parties know the symbols received at the other side and may determine the next party to speak according to their (joint) view, maintaining a consensus regarding the next party to speak. We can therefore refine the class of robust protocols into ones in which the order of speaking is fixed (i.e., a function of the round number) and ones in which the order is arbitrary (i.e., possibly dependent on the noise). We stress that these two subsets of protocols are still robust, and refer the reader to [GHS14, AGS13] for a discussion on *adaptive* (non-robust) protocols.

As a helpful warm-up we first consider protocols with a fixed order of speaking. When the parties are allowed to send symbols from a large alphabet, we show an efficient coding scheme with a constant rate that, for any $\varepsilon > 0$, resists a noise rate of up to $1/4 - \varepsilon$; additionally we show a matching impossibility bound on the fraction of noise. Although the same bounds were already given by [BR11, GH14b] for standard noisy channels, our protocol is considerably simpler while also being computationally efficient. Moreover, while in other schemes the size of the alphabet increases as $\varepsilon \to 0$, in our protocol a *ternary* alphabet suffices. The main idea is the following: the parties exchange messages as in the noiseless protocol, and use the feedback to verify that the messages were received intact. In case of a corrupted transmission, the parties transmit a special symbol '$\leftarrow$' that instructs both parties to rewind the protocol to the step before the corrupted transmission. Building on the above coding scheme we provide a simple and efficient *binary* protocol, that for any $\varepsilon > 0$ resists up to a $(1/6 - \varepsilon)$-fraction of bit flips.

We also prove a matching impossibility bound for binary protocols with fixed order. Since feedback channels are more powerful than standard noisy channels, this impossibility applies also to robust protocols over standard binary noisy channels, narrowing the maximal tolerable noise for this setting to the region $[1/8, 1/6]$.

Theorem 1.1. *There exists a function $f(x, y)$, so that any binary robust interactive protocol cannot compute $f(x, y)$ with probability at least $1 - o(1)$ assuming a $1/6$-fraction of bit-flips.*

Next, we consider robust protocols with arbitrary (noise-dependent) order of speaking over a channel with feedback. In this case the simple idea presented above immediately gives a higher noise-resilience of $1/3$. The reason for this discrepancy in the bounds when we allow the order of speaking to be arbitrary stems from the following issue. When a transmission is corrupted, the *sender* is aware of this event and he will sends a rewind symbol '$\leftarrow$' the next time he speaks. However, when the order of speaking is fixed (say, alternating), the parties 'lose' one slot: while we would like the sender to repeat the transmission that was corrupted, the *receiver* is the next party to speak after the round where the '$\leftarrow$' symbol is sent. If we allow the order of speaking to be arbitrary, we can avoid this excessive round and thus improve the noise resilience.

Translating the above idea to the binary case gives a protocol that resists a noise rate of $1/5 - \varepsilon$. However we can do better. We devise a protocol that resists noise rates below $1/3 - \varepsilon$ for the binary case as well. Here the parties send messages of varying length, consisting of the original information followed by a varying amount of *confirmation bits*. The confirmation bits indicate whether or not the information was corrupted by the adversary. This practically forces the adversary to spend more of his corruption budget per message, or otherwise the receiving party learns about the corruption and simply ignores the message.

Theorem 1.2. *For any $\varepsilon > 0$ and any function $f(x, y)$ there exists a simple and efficient robust coding scheme with constant rate that correctly computes $f(x, y)$ over a binary channel with feedback assuming at most a $(1/3 - \varepsilon)$-fraction of the bits are corrupted.*

It is interesting to mention that, in contrast to all the previous settings, and in contrast to the case of standard (uni-directional) error correction, the size of the alphabet (binary or large) makes no difference to the noise resilience of this setting.

We also provide a matching lower (impossibility) bound of $1/3$ that applies to any alphabet size, and in particular to the binary case. Our bound is similar in spirit to the bound of $1/3$ for adaptive protocols shown in [GHS14].

Theorem 1.3. *There exists a function $f(x, y)$, so that any robust interactive protocol over a channel with feedback cannot compute $f(x, y)$ with probability at least $1 - o(1)$ assuming a $1/3$-fraction of the transmissions are corrupted.*

Interactive communication over erasure channels. In [FGOS13, GHS14] it was shown that the maximal noise over erasure channels when a large alphabet can be used is $1/2 - \varepsilon$. This is trivially tight for protocols with fixed order since by completely erasing the symbols sent by the party talking less prevents any interaction. A similar argument can also be applied to robust protocols as well [GHS14]. When the parties are restricted to using a binary alphabet, it is possible to resist an erasure rate of $1/4 - \varepsilon$ [FGOS13, BR11]. The main drawback of these coding schemes is that they are not efficient.

Here we suggest a coding scheme with constant rate that can tolerate an erasure rate of up to $1/2 - \varepsilon$, yet it is efficient and very simple to implement. Moreover, our "large" alphabet is of size 6, regardless of ε.

Theorem 1.4. *For any $\varepsilon > 0$ and any function $f(x, y)$ there exists a simple and efficient, robust coding scheme with constant rate that correctly computes $f(x, y)$ over an erasure channel with a 6-ary alphabet, assuming at most a $(1/2 - \varepsilon)$-fraction of the bits are corrupted.*

Interestingly, the smaller and fixed alphabet size serves as a stepping stone in devising a protocol that works for *binary* erasure channels. Encoding each symbol of the 6-ary alphabet into a binary string yields a protocol that resists erasures fraction of up to $3/10 - \varepsilon$. Yet, we are able to optimize the above simulation and reduce the size of the alphabet to only 4 symbols. This allows us to encode each symbol in the alphabet using a binary code with a very high relative distance, and obtain a protocol that tolerates a noise rate of $1/3 - \varepsilon$. This improves over the more natural and previously best known bound of $1/4 - \varepsilon$.

Theorem 1.5. *For any $\varepsilon > 0$ and any function $f(x, y)$ there exists a simple and efficient, robust coding scheme with constant rate that correctly computes $f(x, y)$ over a binary erasure channel, assuming at most a $(1/3 - \varepsilon)$-fraction of the bits are corrupted.*

The only lower bound we are aware of is again the trivial bound of $1/2$ [FGOS13, GH14b] which applies even to larger alphabets. We leave determining the optimal rate as an interesting open question.

We summarize our results in Table 1.

1.2 Other Related Work

Maximal noise in interactive communication. As mentioned above, the question of interactive communication over a noisy channel was initiated by Schulman [Sch92, Sch96] who mainly focused on the case of random bit flips, but also showed that his scheme resists an adversarial noise rate of up to $1/240$. Braverman and Rao [BR11] proved that $1/4$ is a tight bound on the noise (for large alphabets), and Braverman and Efremenko [BE14] gave a refinement of this bound, looking at the noise rate separately at each direction of the channel (i.e., from Alice to Bob and from Bob to Alice). For each pair of noise rates, they determine whether or not a coding scheme with a constant rate exists. Another line of work improved the efficiency of coding schemes for the interactive setting, either for random noise [GMS14], or for adversarial noise [BK12, BN13, GH14b].

The protocols of above works are all robust. The discussion about non-robust or *adaptive* protocols was initiated by Ghaffari, Haeupler and Sudan [GHS14, GH14b] and concurrently by Agrawal, Gelles and Sahai [AGS13], giving various notions of adaptive protocols and analyzing their noise resilience. Both the adaptive notion of [GHS14, GH14b] and of [AGS13] are capable of resisting a higher amount of noise than the maximal 1/4 allowed for robust protocols. Specifically, a tight bound of 2/7 was shown in [GHS14, GH14b] for protocols of fixed length; when the length of the protocol may adaptively change as well, a coding scheme that achieves a noise rate of 1/3 is given in [AGS13], yet that scheme does not have a constant rate.

Interactive communication over channels with feedback and erasure channels. To the best of our knowledge, no prior work considers the maximal noise of interactive communication over channels with feedback. Yet, within this setting, the maximal *rate* of coding schemes, i.e., the minimal communication complexity as a function of the error rate, was considered by [Pan13, GH14a] (the rate of coding schemes in the standard noisy channel setting was considered by [KR13, Hae14]).

For erasure channels, a tight bound of 1/2 on the erasure rate of robust protocols was given in [FGOS13]. For the case of adaptive protocols, [AGS13] provided a coding scheme with a constant rate that resists a relative erasure rate of up to $1-\varepsilon$ in a setting that allows parties to remain silent in an adaptive way.

2 Preliminaries

We begin by setting some notations and definitions we use throughout. We sometimes refer to a bitstring $a \in \{0,1\}^n$ as an array $a[0], \dots, a[n-1]$. $a \circ b$ denotes the concatenation of the strings a and b. $\mathsf{prefix}_k(a)$ denotes the first k characters in a string a, and $\mathsf{suffix}_k(a)$ denotes the last k characters in a.

Definition 2.1. *A* feedback channel *is a channel* $\mathsf{CH} : \Sigma \to \Sigma$ *in which at any instantiation noise can alter any input symbol* $\sigma \in \Sigma$ *into any output* $\sigma' \in \Sigma$. *The sender is assumed to learn the (possibly corrupt) output* σ' *via a noiseless feedback channel.*

An erasure channel *is a channel* $\mathsf{CH} : \Sigma \to \Sigma \cup \{\perp\}$ *in which the channel's noise is restricted into changing the input symbol into an erasure symbol* $\perp$.

For both types of channels, the noise rate *is defined as the fraction of corrupted transmissions out of all the channel instantiations.*

Definition 2.2. *We say that an interactive protocol* π *is* robust ([BR11]) *if,*
(1) for all inputs, the protocol runs n rounds; (2) at any round, and given any possible noise, the parties are in agreement regarding the next party to speak.

A fixed order *protocol is one in which condition (2) above is replaced with the following*
(2') there exist some function $g : \mathbb{N} \to \{Alice, Bob\}$ *such that at any round* i, *the party that speaks is determined by* $g(i)$, *specifically, it is independent of the noise.*

Note that any fixed-order protocol is robust, but it is possible that a robust protocol will not have fixed order. In that case we say the robust protocol has *arbitrary* or *noise dependent* order of speaking.

In the following we show how to take any binary alternating (noiseless) protocol, and simulate it over a noisy channel. Note that for any function f there exists a binary alternating (noiseless) protocol π, such that the communication of π is linear in the communication complexity of f, that is,

$$\mathrm{CC}(\pi) = O(\mathrm{CC}(f)).$$

Hence, simulating the above π with communication $O(\mathrm{CC}(\pi))$ has a constant rate, since its communication is linear in $\mathrm{CC}(f)$.

3 Feedback Channels with a Large Alphabet: Upper and Lower Bounds

3.1 Protocols with fixed order of speaking

Let us consider simulation protocols in which the order of speaking is fixed and independent of the inputs the parties hold, and the noise injected by the adversarial channel. We show that $1/4$ is a tight bound on the noise of this case, similar to the case of standard noisy channels (without feedback). We begin with the upper bound, by showing a protocol that correctly simulates π assuming noise level of $1/4 - \varepsilon$. It is interesting to note that the alphabet used by the simulation protocol is independent of ε (cf. [BR11, FGOS13, GH14b, BE14]); specifically, we use a *ternary alphabet*. In addition the simulation is deterministic and efficient.

Theorem 3.1. *For any alternating noiseless binary protocol π of length n, and for any $\varepsilon > 0$, there exists an efficient, deterministic, robust simulation of π over a* feedback channel *using an alphabet of size 3 and a fixed order of speaking, that takes $O_\varepsilon(n)$ rounds and succeeds assuming a maximal noise rate of $1/4 - \varepsilon$.*

Proof. We use a ternary alphabet $\Sigma = \{0, 1, \leftarrow\}$. The simulation works in alternating rounds where the parties run π, and verify via the feedback that any transmitted bit is correctly received at the other side. Specifically, if the received symbol is either a 0 or a 1 the party considers this transmission as the next message of π, and extends the simulated transcript T accordingly. If the received symbol is $\leftarrow$, the party rewinds π by three rounds, that is, the party deletes the last four undeleted symbols of T.[1] Each party, using the feedback, is capable of seeing whether the transcript T held by the other side contains any errors, and if so, it sends multiple $\leftarrow$ symbols until the corrupted suffix is removed. The above is repeated $N = n/4\varepsilon$ times (where $n = |\pi|$), and at the end the parties output T. The protocol is formalized in Algorithm 1.

Note that due to the alternating nature of the simulation, each corruption causes four rounds in which T doesn't extend: (1) the corrupted slot; (2) the other party talks; (3) sending a $\leftarrow$ symbol; (4) the other party talks. After step (4) the simulated transcript T is exactly the same as it was before (1). Also note that consecutive errors (targeting the same party[2]) just increase the amount of $\leftarrow$ the sender should send, so that each additional corruption just extends the recovery process by at most another four rounds. Also note that corrupting a bit into a $\leftarrow$ has a similar effect: after four rounds, T is back to what it was before the corruption: (1) the corrupted slot; (2–4) re-simulating π after three bits of T were deleted.

Therefore, with $1/4 - \varepsilon$ noise, we have at most $4 \cdot (1/4 - \varepsilon)N = N(1 - 4\varepsilon)$ rounds that are used to recover from errors, but do not advance T. Yet, during the rest $4\varepsilon N = n$ rounds T extends correctly and the simulation correctly outputs the entire transcript of π. □

[1] The four symbols removed from T are the received '$\leftarrow$' symbol plus three rounds of π.

[2] consecutive corruptions targeting the other party will be corrected without causing any further delay

Algorithm 1 A fixed-order simulation over a noisy channel with feedback

Input: a binary alternating protocol π of length n, a noise parameter $\varepsilon > 0$, an input value x.

Set $N = \lceil n/4\varepsilon \rceil$, initialize $T \leftarrow \emptyset$; $T^F \leftarrow \emptyset$.
T is the simulated transcript as viewed by the party. We can split T into two substrings corresponding to alternating indices: T^S are the sent characters, and T^R the received characters. Let T^F be the characters received by the other side (as learned via the feedback channel).

for $i = 1$ to N **do**
  if $T^F = T^S$ **then**
    $T \leftarrow T \circ \pi(x \mid T)$ ▷ run one step of π, given the transcript so far is T
    $T^F \leftarrow T^F \circ \langle \text{symbol recorded at the other side} \rangle$
  else
    if sender: send a '$\leftarrow$' symbol; $T \leftarrow T \circ$ '$\leftarrow$'; $T^F \leftarrow T^F \circ \langle \text{symbol recorded at the other side} \rangle$.
    if receiver: extend T according to incoming symbol.
    (Alice is the sender on odd i's and Bob is the sender on the even i's)

  if $\mathsf{sufffix}_1(T^R) =$'$\leftarrow$' or $\mathsf{sufffix}_1(T^F) =$'$\leftarrow$' **then**
    $T \leftarrow \mathsf{prefix}_{|T|-4}(T)$ (also delete the corresponding transmissions in T^F)
Output T.

We continue with proving a lower (impossibility) bound of noise rate 1/4.

Theorem 3.2. *No protocol with a fixed order of speaking can compute $f(x, y) = x, y$ correctly and tolerate an error rate of* $1/4$ *over a* feedback channel.

Proof. The proof of the lower bound is similar to the case of interactive communication over of a standard noisy channel (without feedback) [BR11]. Assume that Alice speaks for R rounds and without loss of generality assume $R \leq N/2$ (note that since the protocol has a fixed order of speaking, the party that speaks in less than half the rounds is independent of the input and noise, and is well defined at the beginning of the simulation). Define $\mathsf{EXP0}$ to be an instance in which Alice holds the input $x = 0$, and we corrupt the first $R/2$ rounds in which Alice talks so that they are the same as what Alice would have sent had she held the input $x = 1$. Define $\mathsf{EXP1}$ to be an instance in which Alice holds the input $x = 1$, and we corrupt the last $R/2$ rounds in which Alice talks so that they are the same as what Alice sends during the same rounds in $\mathsf{EXP0}$.

Note that from Bob's point of view (including his feedback) $\mathsf{EXP0}$ and $\mathsf{EXP1}$ are indistinguishable, thus Bob cannot output the correct x with probability higher than 1/2. In each experiment we corrupt only half of Alice's slots, thus the total noise is at most $R/2 \leq N/4$. □

3.2 Protocols with noise-dependent order of speaking

It is rather clear that the protocol of Theorem 3.1 "wastes" one round (per corruption) only due to the fixed-order of speaking: when a corruption is noticed and a $\leftarrow$ symbol is sent, the parties would have liked to rewind only *two* rounds of π, exactly back to beginning of the round that was corrupted. However, this will change the order of speaking, since that round belongs to the same party that sends the $\leftarrow$ symbol. This suggests that if we lift the requirement of a fixed-order simulation, and allow the protocol to adapt the order of speaking, the simulation will resist up to a fraction 1/3 of noise. In the following we prove that 1/3 is a tight bound on the noise for this case.

We remark that although the protocol is adaptive in the sense that the "order of speaking" may change due to the noise, both parties are always in consensus regarding who is the next party to speak. Indeed, using the feedback channel, both parties learn the symbols received in both sides (i.e., the simulated transcript T held by both sides), and those uniquely determine the next party to speak. Thus, the protocol is robust (Definition 2.2).

Theorem 3.3. *For any alternating noiseless binary protocol π of length n, and for any $\varepsilon > 0$, there exists an efficient, deterministic, robust simulation of π over a* feedback channel *using an alphabet of size 3, that takes $O_\varepsilon(n)$ rounds and succeeds assuming a maximal noise rate of* $1/3 - \varepsilon$.

Proof. We use a ternary alphabet $\Sigma = \{0, 1, \leftarrow\}$. The simulation protocol is similar to Algorithm 1: each party maintains a simulated transcript T, and uses the feedback to verify the other party holds a correct simulated transcript. As long both sides receive the correct bits, in the next round the parties simulate the next step of π given that the transcript so far is T. Otherwise, the party that notices a corruption sends a $\leftarrow$ symbol at the next round assigned to that party. When a $\leftarrow$ symbol is received, the party rewinds π by *two* rounds, that is, the party deletes the last three symbols of T.[3] The next party to speak is determined by $\pi(x \mid T^R, T^F)$; note that $(T^F, T^R)_{\text{Alice}} = (T^R, T^F)_{\text{Bob}}$, thus the the parties progress according the same "view" and are in-synch at all times. The above is repeated $N = n/3\varepsilon$ times (where $n = |\pi|$), and at the end the parties output T.

It is easy to verify that each corruption causes at most three recovery rounds, after which T is restored to its state prior the corruption: (1) the corrupted slot; (2) the other party talks; (3) sending a $\leftarrow$ symbol; After step (3) the simulated transcript T is exactly the same as it was before (1), and the party that spoke at (1) has the right to speak again. Again, note that consecutive errors just increase the amount of $\leftarrow$ symbols the sender should send, so that each additional corruption just extends the recovery process by another three rounds. Also note that corrupting a bit into a $\leftarrow$ has a similar effect: after three rounds, T is back to what it was before the corruption: (1) the corrupted slot; (2–3) re-simulating the two bits of T that were deleted.

Therefore, when the noise level is bounded by $1/3 - \varepsilon$, we have at most $3 \cdot (1/3 - \varepsilon)N = N(1 - 3\varepsilon)$ rounds that are used to recover from errors and do not advance T; yet, during the rest $3\varepsilon N = n$ rounds T extends correctly. Therefore, at the end of the simulation the parties output a transcript of π with a correct prefix of length at least n, thus they successfully simulate π. ☐

Theorem 3.4. *No robust protocol can compute $f(x, y) = x, y$ over a* feedback channel *and tolerate an error rate of* $1/3$.

Proof. The proof is based on ideas from [GHS14] used for proving a lower bound on the noise for adaptive protocols over a standard noisy channel (without feedback).

Consider a protocol of length N, and suppose that on inputs $x = y = 0$, Bob is the party that speaks less times in the first $2N/3$ rounds of the protocol. Recall that due to the feedback, we can assume the parties are always in consensus regarding the party to speak on the next round, so that at every round only a single party talks; thus Bob talks at most $N/3$ times during the first $2N/3$ rounds. Consider the following experiment $\mathsf{EXP1}$ in which $x = 0, y = 1$ however we corrupt Bob's messages during the first $2N/3$ rounds so that they are the same as Bob's messages given $y = 0$. From Alice point of view, the case where Bob holds $y = 0$ and the case where $y = 1$ but all his messages are corrupted to be as if he had $y = 0$, are equivalent. Therefore, with the consensus assumption, in both cases Bob's talking slots are exactly the same, and this strategy corrupts at most $N/3$ messages.

[3]The three symbols removed from T are the received '$\leftarrow$' symbol plus two rounds of π.

Now consider the following experiment EXP0 in which $x = y = 0$, however, during the last $N/3$ rounds of the protocol we corrupt all Bob's messages to be the same as what he sends in EXP1 during the same rounds. Note that due to the adaptiveness of the protocol, it may be that Bob talks in *all* these $N/3$ rounds, but corrupting all of them is still within the corruption budget.

Finally, in both EXP0 and EXP1 Alice's view (messages sent, received and feedback) is the same, implying she cannot output the correct answer with probability higher than 1/2. □

4 Feedback Channels with a Binary Alphabet: Upper and Lower Bounds

We now turn to examine the case of feedback channels with a binary alphabet. As before we begin with the case of a robust simulation where the order of speaking is fixed, and show a tight bound of 1/6 on the noise. We then relax the fixed-order requirement and show that 1/3 is a tight bound on the noise (similar to the case of a large alphabet).

4.1 Protocols with fixed order of speaking

Theorem 4.1. *For any alternating noiseless binary protocol π of length n, and for any $\varepsilon > 0$, there exists an efficient, deterministic, robust simulation of π over a* feedback channel *with a binary alphabet and a fixed order of speaking, that takes $O_\varepsilon(n)$ rounds and succeeds assuming a maximal noise rate of* $1/6 - \varepsilon$.

Proof. In Algorithm 1 we used a special symbol $\leftarrow$ to signal that a transmission was corrupted and the simulated protocol should be rewinded. When the alphabet is binary such a symbol can not be used directly, but we can code it into a binary string. Specifically, we begin by pre-processing the protocol to simulate π so that no party sends two consecutive zeros (cf. [GH14a]). This can easily be done by padding each two consecutive rounds in π by two void rounds where each party sends a '1' (two rounds are needed to keep the padded protocol alternating). Denote by π' the pre-processed protocol, and note that $|\pi'| = 3|\pi|$.

We now simulate π' in a manner similar to Theorem 3.1, that is, the parties communicate in alternating rounds where at each round they send the next bit defined by π' according to the current simulated transcript T. We code the $\leftarrow$ symbol as the string 00, which can never appear as a substring of the correct transcript of π'. Upon receiving a 00 rewind command, the party deletes the last 6 bits of T. Observe that we must remove an even number of bits so that the alternating order of speaking is maintained, although the erroneous bit is only 5 rounds prior to receiving the $\leftarrow$ command, see Figure 1. The simulation is performed for $N = |\pi'|/6\varepsilon = O(|\pi|)$ rounds at the end of which both parties output T.

The analysis is similar to the proof of Theorem 3.1, yet since each corruption needs at most six rounds to recover[4], the maximal amount of noise the protocol can tolerate is $1/6 - \varepsilon$. □

Even more interesting is the fact that the above protocol is the best possible, in terms of the maximal tolerable noise.

Theorem 4.2. *No binary protocol with a fixed order of speaking can compute $f(x, y) = x, y$ over a* feedback channel *and tolerate an error rate of* $1/6$.

[4] At some situations, for instance when a 1 bit is corrupted into a 0, the amount of rounds needed to recover may even be smaller.

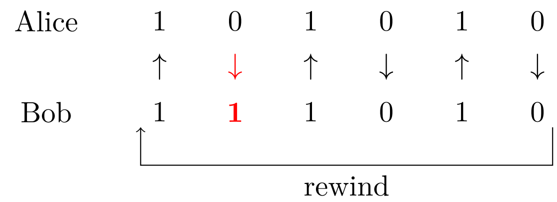

Figure 1: Illustration of rewinding after Alice's bit is corrupted.

Proof. Assume a binary robust protocol π that computes $f(x,y)=x,y$ where x,y belong to some domain of size at least 3; assume $|\pi|=N$, and without loss of generality let Alice be the party that speaks at most $T\le N/2$ times in the protocol. We show an adversarial strategy that corrupts at most 1/3 of Alice's messages and makes Bob's view look the same for two different inputs of Alice.

Assume Alice holds one of three inputs, x_0,x_1,x_2. For a given instance, define $x_0[i],x_1[i],x_2[i]$ to be the i-th bit sent by Alice for the respective inputs. Note that the i-th transmission may depend on the transcript so far. If we fix a transcript up to the round where Alice sends her i-th bit, and look at the next bit that Alice send, $x_0[i],x_1[i],x_2[i]$, we observe that either the bit value is the same for all x_0,x_1,x_2, or it is the same for two of these inputs, and different for the third one. For every $i\le T$, let $\mathsf{maj}(i)=\mathsf{majority}(x_0[i],x_1[i],x_2[i])$, given that previous transmissions are consistent with the adversarial strategy described below, and let $w[1,\dots,T]=\mathsf{maj}(0)\cdots\mathsf{maj}(T)$.

The adversarial strategy only corrupts Alice, so we should describe what is being sent at each one of the T rounds in which Alice has the right of speak. The attack consists of two parts. For any transmission $i\le R$, we corrupt the transmission so it equals $\mathsf{maj}(i)$ (i.e., if Alice sends $\mathsf{maj}(i)$ we leave the transmission intact and otherwise we flip the bit). The number R is set to be the minimal round such that for at least one of x_0,x_1,x_2, the above strategy corrupts exactly $R-2T/3$ bits up to round R (included). It is easy to verify we can always find a round $2T/3\le R\le T$ that satisfies the above: for every x_j the quantity $d(w[1,\dots,R],x_j[1,\dots,R])$ starts at 0, never decreases, and increases at most by one in every round. Furthermore since in the first part at most T corruptions happen over all rounds and all three inputs, at least for one input x_j the quantity $d(w[1,\dots,R],x_j[1,\dots,R])$ will grows from 0 to at most $T/3$. On the other hand, the quantity $R-2T/3$ increases exactly by one in every round and thus goes from $-2T/3$ to $T/3$. Therefore, there exists a round R in which $R-2T/3$ catches up with $d(w[1,\dots,R],x_j[1,\dots,R])$ for some input x_j.

Let x_0 be the input for which the number of corruptions up to round R is $R-2T/3$; note that x_0 minimizes $d(w[1,\dots,R],x_j[1,\dots,R])$, or otherwise one of the other inputs would have satisfied $d(w[1,\dots,R'],x_j[1,\dots,R'])=R'-2T/3$ for some earlier round $R'<R$. We can therefore assume without loss of generality that,

$$d(w[1,\dots,R],x_0[1,\dots,R])\le d(w[1,\dots,R],x_1[1,\dots,R])\le d(w[1,\dots,R],x_2[1,\dots,R]). \quad (1)$$

In the second part (the last $T-R$ of Alice's slots), Eve corrupts the i-th transmission so it equals $x_1[i]$. That is, for x_1 she does nothing and for x_0 she flips the bit as needed. We will not care about x_2 in this part.

First note that from Bob's point of view, the transcripts he sees given that Alice holds x_0 or x_1 are exactly the same. Next, we claim that for both these inputs, the total amount of corruptions is at most $T/3\le N/6$. If Alice holds the input x_0, then the total amount of corruptions is at most

$$d(w[1,\dots,R],x_0[1,\dots,R])+(T-R)\le(R-2T/3)+(T-R)=T/3\le N/6.$$

If Alice holds x_1 then we don't make any corruption during the last $(T-R)$ rounds, and the total amount of corruptions is at most $d(w[1,\dots,R], x_1[1,\dots,R])$. Since w is the majority, at each round i, there exists at most a single input x_j, for which $d(w[i], x_j[i]) = 1$, while for both other inputs $x_{j'}$, the i-th transmitted bit is the same as their majority, $d(w[i], x_{j'}[i]) = 0$. It follows that

$$d(w[1,\dots,R], x_0[1,\dots,R]) + d(w[1,\dots,R], x_1[1,\dots,R]) + d(w[1,\dots,R], x_2[1,\dots,R]) \le R,$$

thus with Eq. (1) and the fact that $d(w[1,\dots,R], x_0[1,\dots,R]) = R - 2T/3$, we have

$$\begin{aligned} R - 2T/3 + 2d(w[1,\dots,R], x_1[1,\dots,R]) &\le R \\ d(w[1,\dots,R], x_1[1,\dots,R]) &\le T/3 \le N/6. \end{aligned}$$

□

4.2 Protocols with noise-dependent order of speaking

When the oder of speaking needs not be fixed, we can improve the protocol of Theorem 4.1 so that after receiving a 00 rewind command, the party deletes only 5 bits of the simulated transcript T. This immediately yields a protocol that resist noise level up to 1/5. However, we can do even better. Using similar ideas from [AGS13], we devise a protocol in which the parties adaptively change the length of the messages they send, forcing the adversary to make more corruptions in order to "change" a message (or otherwise to cause only an "erasure"). We show that in this case resisting noise level of up to 1/3 is possible even when the channel is binary. This bound is tight due to the impossibility shown in Theorem 3.4, which holds also for binary alphabet.

Theorem 4.3. *For any alternating noiseless binary protocol π of length n, and for any $\varepsilon > 0$, there exists an efficient, deterministic, robust simulation of π over a* feedback channel *with binary alphabet, that takes $O_\varepsilon(n)$ rounds and succeeds assuming a maximal noise rate of $1/3 - \varepsilon$.*

The idea of the simulation is the following. The parties exchange messages of varying lengths. Each message consists of three parts: (a) 1 bit of information — the next bit of π; (b) 1 control ("rewind") bit — an indication that the previous message was corrupted and the protocol should be rewinded to the beginning of that message; and (c) $t \ge 1$ confirmation bits, set according to the message received at the other side (according to the feedback): if the information and rewind bits were received intact, the confirmation bits will be '1', or otherwise they will be '0'. The sender keeps sending confirmation bits until one of the following happens:

1. the number of 0-confirmation bits is at least 1/3 of the length of the current message — in this case the message is *unconfirmed* and the protocol rewinds to the beginning of that message (so that the sender has the right of speak again to send the same message).

2. the number of 1-confirmation bits minus the number of 0-confirmation bits is larger than $1/\varepsilon$ — in this case the message is confirmed and the parties either rewind the protocol to the previous message of the sender (if the rewind bit is on), or the next simulated bit is the information bit.

The parties perform the above until a total number of n/ε^2 bits are communicated. We formulate the simulation protocol in Algorithm 2. We now continue to prove Theorem 4.3.

Proof. Assume that the for-loop runs for N times, and for $i = 1,\dots,N$ let m_i be the i-th message sent, of length $|m_i|$. We know that $\sum_i |m_i| = n/\varepsilon^2$.

Divide the messages $m_1,\dots,m_N$ into three disjoint sets:

Algorithm 2 Simulation over a feedback channel with a binary alphabet for noise rates $< 1/3 - \varepsilon$

Input: an alternating binary protocol π of length n, a noise parameter $\varepsilon > 0$, an input value x.

Initialize $T \leftarrow \emptyset$.
$T = (T^S, T^R, T^F)$ is the simulated transcript, separated to sent, received and feedback bits.

1: **for** $i = 1, 2, \ldots$ **do**
2:     **if** $\pi(x \mid T^F, T^R)$ is your turn to speak **then**
3:         $sentLength \leftarrow 2$
4:         $conf_0 \leftarrow 0$, $conf_1 \leftarrow 0$
5:         **if** $T^S = T^F$ **then** ▷ No corruptions are known
6:             $rewind = 0$
7:         **else** ▷ The transcript at the other side is corrupt
8:             $rewind = 1$

9:         $msg \leftarrow \pi(x \mid T^F, T^R) \circ rewind$
10:         send msg
11:         **while** ($conf_0 < sentLength/3$) and ($conf_1 - conf_0 < 1/\varepsilon$) **do**
12:             **if** msg received correctly **then** ▷ verify via the feedback
13:                 send 1
14:             **else**
15:                 send 0
16:             $sentLength \leftarrow sentLength + 1$
17:             $conf_b \leftarrow \mathrm{conf}_b + 1$ ▷ b is the bit received at the other side (learned via the feedback)

18:         **if** $conf_0 \geq sentLength/3$ **then** ▷ message is not confirmed
19:             **continue** (next for loop instance)
20:         **else if** $conf_1 - conf_0 \geq 1/\varepsilon$ **then** ▷ message is confirmed: rewind or advance T according to info/rewind received at other side
21:             **if** ⟨rewind bit recorded at the other side⟩ = 0 **then**
22:                 $T^S \leftarrow T^S \circ \pi(x \mid T^S, T^R)$
23:                 $T^F \leftarrow T^F \circ$ ⟨info bit recorded at the other side⟩
24:             **else if** ⟨rewind bit recorded at the other side⟩ = 1 **then**
▷ Remove from T the last two simulated rounds
25:                 $T^R \leftarrow \mathsf{prefix}_{|T^R|-1}(T^R)$
26:                 $T^S \leftarrow \mathsf{prefix}_{|T^S|-1}(T^S)$
27:                 $T^F \leftarrow \mathsf{prefix}_{|T^F|-1}(T^F)$

28:     **else** ▷ The other party is the speaker at this round
29:         Record msg, and confirmation bits according to the conditions of the while loop on line 11.
30:         If msg unconfirmed (line 18), ignore msg and **continue**.
31:         If msg confirmed (line 20):
either extend T^R (if $rewind = 0$) or delete the suffix bit of T^R, T^S, T^F (if $rewind = 1$).

32:     If more than n/ε^2 bits were communicated, terminate and output T.

- $C = \{m_i \mid m_i \text{ is correct and confirmed}\}$
- $U = \{m_i \mid m_i \text{ is unconfirmed}\}$
- $W = \{m_i \mid m_i \text{ is incorrect, yet confirmed}\}$

It is easy to see that an unconfirmed message $m_i \in U$ has no effect on the simulated transcript as any such message is just ignored. If a message is confirmed, it can either be interpreted as an information bit or as a rewind request, and the simulation is similar to the algorithm of Theorem 3.3 where each one of the symbols $\{0, 1, \leftarrow\}$ is encoded into a longer message. Specifically, similar to Theorem 3.3, after a single incorrect message (i.e., $m_i \in W$) the simulation takes two correct messages (i.e, $m_i \in C$) in order to "recover" from the error, i.e., revert to the state before the corruption. Also here, multiple erroneous messages just linearly accumulate, hence,

Claim 4.4. *The simulation of a protocol π of length n succeeds as long as*

$$|C| - 2|W| \geq n.$$

Next, we bound the length and noise rate of a message.

Claim 4.5. *For any i, $|m_i|$ is bounded by $2 + 3/\varepsilon$.*

Proof. Assume a message reaches length $2 + 3/\varepsilon$, and consider its $3/\varepsilon$ confirmation bits: If $1 + 1/\varepsilon$ of these bits are zeros, then the message is unconfirmed since $(1 + 1/\varepsilon)/(2 + 3/\varepsilon) > 1/3$. Otherwise, there are at most $1/\varepsilon$ zeros and at least $2 + 3/\varepsilon - 1/\varepsilon > 2/\varepsilon$ ones, thus the difference between confirmation zeros and ones exceeds $1/\varepsilon$ and the message is confirmed. □

Note that for a confirmed message, $2 + 1/\varepsilon \leq |m_i| \leq 2 + 3/\varepsilon$, and for an unconfirmed message $3 \leq |m_i| \leq 2 + 3/\varepsilon$. Since the total amount of bits the protocol communicates is n/ε^2, we have

$$|C| + |W| < \frac{n}{\varepsilon}. \tag{2}$$

The specific length of a message relates to the amount of corruption the adversary must make during that message. For messages in C that were eventually confirmed, the noise is any zero-confirmation bits received. Since $conf_1 - conf_0 \geq 1/\varepsilon$ we have that $conf_0 \leq conf_1 - 1/\varepsilon$ and thus $2conf_0 \approx |m_i| - 2 - 1/\varepsilon$ (up to ± 1 due rounding). For unconfirmed messages in U, the message gets unconfirmed as soon as $conf_0 \geq |m_i|/3$. There are two cases: (i) if the information/control bits are correct, then the noise is any 0-confirmation bit, thus $conf_0 \geq |m_i|/3$; (ii) the information/control bits are corrupt, and then the noise is the corruption of the information/control plus any 1-confirmation bit. We have $3conf_0 \geq |m_i| = (2 + conf_0 + conf_1)$ thus $conf_1 \approx \frac{2|m_i|}{3}$ (up to rounding). Finally, when the message is in W, the corruption consists of at least one of the information/control bit and any $conf_1$ received. We have $conf_1 - conf_0 \geq 1/\varepsilon$ thus $conf_1 \geq conf_0 + 1/\varepsilon$ or equivalently $conf_1 \geq (|m_i| - 2 + 1/\varepsilon)/2$.

To summarize, the error rate in each one of the cases is lower bounded by,

$$\begin{array}{ll} m_i \in C & \dfrac{|m_i| - 2 - 1/\varepsilon}{2|m_i|} = \dfrac{1}{2} - \dfrac{2 + 1/\varepsilon}{2|m_i|}, \\ m_i \in U & \dfrac{1}{3}, \\ m_i \in W & \dfrac{|m_i| + 1/\varepsilon}{2|m_i|}. \end{array}$$

Therefore, the global noise rate in any given simulation must be lower bounded by

$$\text{Noise Rate} \geq \frac{\sum_C (|m_i| - 2 - 1/\varepsilon)/2 + \sum_U |m_i|/3 + \sum_W (|m_i| + 1/\varepsilon)/2}{\sum_C |m_i| + \sum_U |m_i| + \sum_W |m_i|}.$$

The noise rate can be written as

$$\begin{aligned}
&\geq \frac{\frac{1}{2}\sum_C |m_i| - \frac{1}{2}|C|(2+\frac{1}{\varepsilon}) + \frac{1}{3}\sum_U |m_i| + \frac{1}{2}\sum_W |m_i| + \frac{1}{2}W \cdot \frac{1}{\varepsilon}}{n/\varepsilon^2} \\
&\geq \frac{1}{3} + \frac{\frac{1}{6}\sum_C |m_i| - \frac{1}{2}|C|(2+\frac{1}{\varepsilon}) + \frac{1}{6}\sum_W |m_i| + \frac{1}{2}W \cdot \frac{1}{\varepsilon}}{n/\varepsilon^2} \\
&\geq \frac{1}{3} + \frac{\frac{1}{6}\sum_C |m_i| - \frac{1}{2\varepsilon}|C| + \frac{1}{6}\sum_W |m_i| + \frac{1}{2\varepsilon}|W|}{n/\varepsilon^2} - O(\varepsilon).
\end{aligned}$$

Now, consider a failed simulation, that is $|C| - 2|W| < n$ (Claim 4.4), or equivalently, $|W| > \frac{1}{2}(|C| - n)$. If $|C| < n$ it is trivial that the error rate is $\geq 1/3 - O(\varepsilon)$. Otherwise, the error rate decreases with the total length of messages in C and W. In the worst case, all such messages are of length $\geq 1/\varepsilon$.

$$\begin{aligned}
&\geq \frac{1}{3} + \frac{\frac{1}{6\varepsilon}|C| - \frac{1}{2\varepsilon}|C| + \frac{1}{12\varepsilon}(|C| - n) + \frac{1}{4\varepsilon}(|C| - n)}{n/\varepsilon^2} - O(\varepsilon) \\
&\geq \frac{1}{3} - O(\varepsilon).
\end{aligned}$$

□

5 Interactive Coding over Erasure Channels

It is already known that 1/2 is a tight bound on the noise rate for interactive communication over erasure channels [FGOS13]. Specifically, that work shows that no protocol can resist an erasure level of 1/2, due to the trivial attack that completely erases one party. Furthermore, for any $\varepsilon > 0$ they give a coding scheme that tolerates an erasure level of $1/2 - \varepsilon$. However, the coding scheme of [FGOS13] has several drawbacks. First, it takes exponential time due assuming *tree codes*, a data structure whose efficient construction is still unknown (see [Sch96, Bra12, MS14, GMS14]). Furthermore, as $\varepsilon \to 0$ and the erasure level approaches 1/2, the tree code needs to be more powerful, which implies the increase of the alphabet size (as a function of ε).

In the following subsection 5.1 we provide a simple, efficient coding scheme for interactive communication over erasure channels in which the alphabet is small (namely, 6-ary), and yet it tolerates erasure rates up to $1/2 - \varepsilon$. Then, in subsection 5.2 we transform this protocol to obtain the best known protocol for binary erasure channels, resisting an erasure level of up to $1/3 - \varepsilon$. It is there where having a fixed, relatively small, alphabet leads to an improved noise tolerance.

5.1 Erasure channels with a "large" alphabet

Theorem 5.1. *For any alternating noiseless binary protocol π of length n, and for any $\varepsilon > 0$, there exists an efficient, deterministic, robust simulation of π over an* erasure channel *with a 6-ary alphabet, that takes $O_\varepsilon(n)$ rounds and succeeds assuming a maximal erasure rate of $1/2 - \varepsilon$.*

The main idea is the following. The parties talk in alternating rounds, in each of which they send a symbol $m \in \mathsf{Info} \times \mathsf{Parity}$ where $\mathsf{Info} = \{0, 1\}$ is the next information bit according to π (given the accepted simulated transcript so far) and $\mathsf{Parity} = \{0, 1, 2\}$ is the parity of the round in π being simulated, modulus 3.

Assume T is the transcript recorded so far, and let $p = (|T| \mod 3)$. If a received m has parity $p + 1$, the receiving party accepts this message and extends T by one bit according to the Info part. Otherwise, or in the case m is erased, the party ignores the message, and re-send its last sent message.

Since messages might get erased, the parties might get "out-of-sync", that is, one party extends its accepted T while the other party does not. However, this discrepancy is limited to a single bit, that is, the length of Alice's T differs from Bob's by at most ± 1. Sending a parity—the length of the current T modulus 3—gives full information on the status of the other side, and allows the parties to regain synchronization. We formalize the above as Algorithm 3.

Algorithm 3 Simulation over an erasure channel with a 6-ary alphabet for erasure rates $< 1/2 - \varepsilon$

Input: an alternating binary protocol π of length n, s noise parameter $\varepsilon > 0$, an input value x.
Initialize: $T \leftarrow \emptyset$, $p \leftarrow 0$, and $m \leftarrow (0, 0)$. Set $N = \lceil n/\varepsilon \rceil$.

1: **for** $i = 1$ to N **do**

2:   **if** Sender **then** ▷ Assume alternating rounds: Alice sends on odd i's, Bob on even i's
3:     **if** your turn to speak according to $\pi(\cdot \mid T)$ **then**
4:       $t_{send} \leftarrow \pi(x \mid T)$
5:       $m \leftarrow (t_{send},\ (p + 1) \bmod 3)$
6:       $T \leftarrow T \circ t_{send}$
7:       $p \leftarrow |T| \mod 3$
8:     send m ▷ Note: when $\pi(\cdot \mid T)$ belongs to other party, send the m stored in memory

9:   **if** Receiver **then**
10:     record $m' = (t_{rec},\ p')$
11:     **if** m' contains no erasures and $p' \equiv p + 1 \mod 3$ **then**
12:       $T \leftarrow T \circ t_{rec}$

13: Output T.

Proof. (Theorem 5.1.) First we set some notations for this proof. For a variable v denote with v_A Alice's instance of the variable (resp. v_B for Bob's instance) and with $v(i)$ the state of the variable at the beginning of the i-th instance of the for-loop. For a string $a = a[0] \cdots a[k]$ denote by $\mathsf{Trim}(a) = a[0] \cdots a[k-1]$ the string obtained by trimming the last index, that is $\mathsf{prefix}_{k-1}(a)$.

Next, we analyze Algorithm 3 and show it satisfies the theorem.

Claim 5.2. *For any $i \leq N$, $||T_A(i)| - |T_B(i)|| \leq 1$.*

Proof. We prove by induction. The base case $i = 1$ trivially holds. Now assume the claim holds up to the beginning of the i-th iteration of the loop. We show that the claim still holds at the end of the i-th iteration.

We consider several cases. (a) $|T_A(i)| = |T_B(i)|$. Then trivially each of T_A, T_B can extend by at most a single bit and the claim still holds. (b) $|T_A(i)| = |T_B(i)| + 1$. Note that this situation is only possible if the $|T_A(i)|$-th round of π is Alice's round: otherwise, in a previous round T_A was of length $|T_B(i)|$ and it increased by one bit at line 12. But for this to happen, it must be that the received parity was $p' = |T_B(i)| + 1$. Yet, T_B never decreases, so such a parity could be sent by Bob only if at that same round, $|T_B| = |T_A| + 1 - 3k$ for some $k \in \mathbb{N}$. But then $|T_B| \leq |T_A| - 2$, which contradicts the induction hypothesis.

Consider round i. If i is odd, Alice just resends $m_A(i)$ form her memory, since according to $T_A(i)$ the next bit to simulate belongs to Bob; she thus doesn't change T_A. Bob might increase $T_B(i)$ since $p_A = p_B + 1$. So the claim still holds for an odd i. If i is even, Bob is re-sending $m_B(i)$ since according to $T_B(i)$ the next bit to simulate belongs to Alice. Yet Alice will not increase her T_A since the parity mismatches. There is no change in T_A, T_B in this case and thus the claim holds. The third case (c) $|T_B(i)| = |T_A(i)| + 1$, is symmetric to (b). ☐

Claim 5.3. *For any $i \leq N$, T_A and T_B hold a correct prefix of the transcript of $\pi(x, y)$.*

Proof. Again, we show this by induction. The claim trivially holds for $i = 1$. Now, assume that at some round i, $T_A(i)$ and $T_B(i)$ are correct. We show they are still correct at round $i + 1$.

Consider round i, and assume without loss of generality, that Alice is the sender in this round. Then, if Alice adds a bit to T_A this bit is generated according to π given the *correct* prefix T_A, and thus this bit is correct. As for the receiver, note that any message that contains erasures is being ignored. Thus, the only possibility for T_B to be incorrect is if the parties are out of sync, and Bob receives a message that does not correspond to his current position in π. However, if Bob accepts the received bit it must be that the received parity satisfies $p' = |T_B(i)| + 1 \mod 3$.

Since $||T_A(i)| - |T_B(i)|| \leq 1$ (claim 5.2), there are two cases here. Either $|T_A| = |T_B|$ which implies that Alice simulated the correct round, and the received bit is indeed the next on in the transcript of π; or $|T_A(i)| \neq |T_B(i)|$, so that Alice sent the message m that was stored in her memory, in which $p' = |T_B(i)| + 1 \mod 3$. It is easy to verify that at any given time, the message saved in the memory is the one that was generated given the transcript $\mathsf{Trim}(T_A)$ (otherwise, a new m must have been generated by Alice). Along with the constraint on the parity, it must be the case that $|T_A(i)| = |T_B(i)| + 1$ which means the stored m is indeed the correct bit expected by Bob. Thus the claim holds in this case as well. ☐

After establishing the above properties of the protocol, we consider how the protocol advances at each round. We begin by exploring the case where the transmission was not erased.

Lemma 5.4. *Assume that no erasure happens during the i-th transmission. Then,*
(a) if $|T_A(i)| = |T_B(i)|$, then $|T_A(i + 1)| = |T_A(i)| + 1 = |T_B(i + 1)|$.
(b) if i is even and $|T_A(i)| < |T_B(i)|$, then $|T_A(i + 1)| = |T_A(i)| + 1$.

Proof. Part (a): trivial from Algorithm 3. Part (b): recall that this situation is only possible if $|T_B(i)| = |T_A(i)| + 1$ and that the $|T_B(i)|$-th round in π belongs to Bob (see the proof of claim 5.2 above). Thus, if i is even, Bob is the sender, and since $\pi(y \mid T_B(i))$ is Alice's round, Bob will retransmit the message $m_B(i)$ stored in his memory; this corresponds to running π given the transcript $\mathsf{Trim}(T_B(i)) = T_A(i)$. The parity in $m_B(i)$ is thus $p_B(i) = p_A(i) + 1$ and Alice accepts this transmission and extends T_A. ☐

A symmetric claim holds for Bob, replacing even and odd rounds, and assuming $|T_B(i)| \leq |T_A(i)|$. Next, consider erased transmissions. These can only cause the parties a discrepancy of a single bit in their simulated transcripts.

Lemma 5.5. *Assume an erasure happens during the i-th transmission.*
If $|T_A(i)| = |T_B(i)|$ then if i is odd $|T_A(i + 1)| = |T_A(i)| + 1$ while $|T_B(i + 1)| = |T_B(i)|$ and if i is even $|T_B(i + 1)| = |T_B(i)| + 1$ while $|T_A(i + 1)| = |T_A(i)|$. Otherwise (i.e., $|T_A(i)| \neq |T_B(i)|$), there is no change in T_A, T_B.

Proof. For the first part of the lemma, assume $j = |T_A(i)| = |T_B(i)|$. Let R be the first round for which $|T_A(R)| = |T_B(R)| = j$, and assume that at the previous round, $|T_A(R-1)| < |T_B(R-1)|$. Since the transcript are of equal length at round R, Alice must have been the receiver at the $R-1$ round, which means that the $(j-1)$-th round of π belongs to Bob. Therefore, the j-th round of π belongs to Alice, who is also the sender of round R (i.e., R is odd), thus she extends her transcript in one bit at this round as well. Note that T never decreases, thus we must have that $R = i$, and the claim holds for this case. Also note that a similar reasoning applies for the edge case of $i = 1$. If at $R-1$ we have $|T_A(R-1)| > |T_B(R-1)|$ instead, the symmetric argument implies that Bob owns the j-th round of π and is also the sender in the R-th round ($R = i$ is even), thus the claim holds for this case as well.

Next, for the second part of the claim, assume that $|T_A(i)| \neq |T_B(i)|$, and without loss of generality assume that $|T_A(i)| < |T_B(i)|$. We know that the gap between the two transcripts is at most 1, thus if there is any change in these transcripts during the i-th rounds, either both parties increase their transcript, or only Alice does. Since the receiver sees an erasure, that party ignores the incoming message and doesn't change his transcript, so it cannot be that both parties extend their transcripts. On the other hand, if $|T_B(i)| = |T_A(i)| + 1$ then the $|T_B(i)|$-th round of π belongs to Bob (see the proof of Claim 5.2), and Alice will increase T_A only if she is the receiver. However, in this case Alice sees an erasure and she ignores the message, keeping her transcript unchanged. □

Lemma 5.4 and Lemma 5.5 lead to the following corollary.

Corollary 5.6. *Any erasure causes at most two rounds in which neither T_A nor T_B extend.*

Thus, if the adversary is limited to making $(1/2 - \varepsilon)N = N/2 - n$ erasures, it must hold that $|T_A(N)| + |T_B(N)| \geq N - 2 \cdot (N/2 - n) = 2n$ and since $||T_A(N)| - |T_B(N)|| \leq 1$, each of them must be of length at least n. This concludes the proof of the Theorem 5.1. □

5.2 Binary erasure channels

By encoding each symbol from our 6-ary alphabet using a binary code containing at least six codewords with maximal relative distance δ_6 we can immediately get a binary scheme that resists erasure rates of up to $\delta_6/2 - \varepsilon$. To our knowledge the maximal distance δ_6 achievable is 6/10 (see [BBM+78]). This leads to the following lemma:

Lemma 5.7. *For any alternating noiseless binary protocol π of length n, and for any $\varepsilon > 0$, there exists an efficient, deterministic, robust simulation of π over a binary* erasure channel*, that takes $O_\varepsilon(n)$ rounds and succeeds assuming a maximal noise rate of at most $3/10 - \varepsilon$.*

Finally, we use the above ideas to devise a binary protocol that resists an adversarial erasure rate of up to $1/3 - \varepsilon$. The idea is to reduce the number of different messages used by the underlying simulaiton: since Algorithm 3 assumes a 6-ary alphabet, the best code has maximal distance $\delta_6 = 6/10$ which, as mentioned, leads to a maximal resilience of $\delta_6/2 - \varepsilon = 3/10 - \varepsilon$. However, were the alphabet in use smaller, say 4-ary, then we could have used better codes with a higher relative distance $\delta_4 = 2/3$ and achieve a maximal resilience of $\delta_4/2 - \varepsilon = 1/3 - \varepsilon$.

In the following we adapt Algorithm 3 to use an alphabet size of at most 4, and obtain the following.

Theorem 5.8. *For any alternating noiseless binary protocol π of length n, and for any $\varepsilon > 0$, there exists an efficient, deterministic, robust simulation of π over a binary* erasure channel*, that takes $O_\varepsilon(n)$ rounds and succeeds assuming a maximal noise rate of at most $1/3 - \varepsilon$.*

Proof. Each message in Algorithm 3 consists of two parts: an information bit, and a parity modulus three. In order to reduce the number of possible messages we introduce a simple pre-processing step that takes an alternating protocol π of length n and converts it into a protocol π' of length $3n$ by padding each two consecutive transmutations of π with two vacuous transmissions (say, of the value 1). That is, if the communication in π is the bitstring $a_1, b_1, a_2, b_2 \dots$, then in π' the parties communicate $a_1, 1, 1, b_1, 1, 1, a_2, 1, 1, \dots$ (recall that the protocol is alternating, thus Alice sends the odd bits, and Bob the even ones).

In the preprocessed π', both parties know that a bit of information lies only in transmissions whose parity is 0 (mod 3). Thus, for the other parities there is no need to send the information bit — it is always 1! This reduces the size of the alphabet in use, specifically, the parties send messages out of the following message space[5]

$$\mathcal{M} = \{0 \times (\text{mod } 0)\ ,\ 1 \times (\text{mod } 0)\ ,\ 1 \times (\text{mod } 1)\ ,\ 1 \times (\text{mod } 2)\}\,.$$

Now that the message space is of size 4 we can encode each message using a binary code of relative distance $\delta_4 = 2/3$. For instance, we can use the code $\{000, 011, 110, 101\}$ (cf. [BBM+78]). Similar to Algorithm 3, the obtained simulation is deterministic, efficient and takes $O_\varepsilon(n)$ rounds. As for its noise resilience, for any $\varepsilon > 0$ the underlying Algorithm 3 can resist up to $1/2 - \frac{3}{2}\varepsilon$ erased messages (Theorem 5.1). Since each message is coded into a binary string, in order to erase a codeword, 2/3 of its bits must be erased. Therefore, in the concatenated algorithm we can resist a maximal erasure rate of $2/3 \cdot (1/2 - \frac{3}{2}\varepsilon) = 1/3 - \varepsilon$. ☐

References

[AGS13] Shweta Agrawal, Ran Gelles, and Amit Sahai. Adaptive protocols for interactive communication. Manuscript, arXiv:1312.4182 (cs.DS), 2013.

[BBM+78] M.R. Best, A.E. Brouwer, F. Jessie MacWilliams, Andrew M. Odlyzko, and Niel J.A. Sloane. Bounds for binary codes of length less than 25. *Information Theory, IEEE Transactions on*, 24(1):81–93, 1978.

[BE14] Mark Braverman and Klim Efremenko. List and unique coding for interactive communication in the presence of adversarial noise. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, FOCS '14, 2014.

[Ber64] Elwyn R. Berlekamp. *Block coding with noiseless feedback.* PhD thesis, Massachusetts Institute of Technology, 1964.

[BK12] Zvika Brakerski and Yael Tauman Kalai. Efficient interactive coding against adversarial noise. *Foundations of Computer Science (FOCS), IEEE Annual Symposium on*, pages 160–166, 2012.

[BN13] Zvika Brakerski and Moni Naor. Fast algorithms for interactive coding. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 443–456, 2013.

[BR11] Mark Braverman and Anup Rao. Towards coding for maximum errors in interactive communication. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC '11, pages 159–166, New York, NY, USA, 2011. ACM.

[Bra12] Mark Braverman. Towards deterministic tree code constructions. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 161–167, New York, NY, USA, 2012. ACM.

[5] In Algorithm 3 the first information bit will actually have parity 1 rather than 0; we can alter $\mathcal{M}$ to have the information bit on parity 1, and the rest remains the same.

[FGOS13] Matthew Franklin, Ran Gelles, Rafail Ostrovsky, and Leonard J. Schulman. Optimal coding for streaming authentication and interactive communication. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013*, volume 8043 of *LNCS*, pages 258–276. Springer Berlin Heidelberg, 2013.

[GH14a] Ran Gelles and Bernhard Haeupler. Capacity of interactive communication over erasure channels and channels with feedback, 2014. submitted.

[GH14b] Mohsen Ghaffari and Bernhard Haeupler. Optimal Error Rates for Interactive Coding II: Efficiency and List Decoding. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, FOCS '14, 2014.

[GHS14] Mohsen Ghaffari, Bernhard Haeupler, and Madhu Sudan. Optimal error rates for interactive coding I: Adaptivity and other settings. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 794–803, New York, NY, USA, 2014. ACM.

[GMS11] Ran Gelles, Ankur Moitra, and Amit Sahai. Efficient and explicit coding for interactive communication. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 768–777, 2011.

[GMS14] Ran Gelles, Ankur Moitra, and Amit Sahai. Efficient coding for interactive communication. *Information Theory, IEEE Transactions on*, 60(3):1899–1913, March 2014. A preliminary version [GMS11] appeared in FOCS '11.

[Hae14] Bernhard Haeupler. Interactive Channel Capacity Revisited. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2014.

[KR13] Gillat Kol and Ran Raz. Interactive channel capacity. In *STOC '13: Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 715–724, New York, NY, USA, 2013. ACM.

[MS14] Cristopher Moore and Leonard J. Schulman. Tree codes and a conjecture on exponential sums. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, ITCS '14, pages 145–154. ACM, 2014.

[Pan13] Denis Pankratov. On the power of feedback in interactive channels. [Online:] `http://people.cs.uchicago.edu/~pankratov/papers/feedback.pdf`, 2013.

[Sch92] Leonard J. Schulman. Communication on noisy channels: a coding theorem for computation. *Foundations of Computer Science, Annual IEEE Symposium on*, pages 724–733, 1992.

[Sch93] Leonard J. Schulman. Deterministic coding for interactive communication. In *STOC '93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 747–756, New York, NY, USA, 1993. ACM.

[Sch96] Leonard J. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996. A preliminary version [Sch93] appeared in STOC '93.

[Sha48] C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal, The*, 27(3):379–423, July 1948.