Electronic Colloquium on Computational Complexity, Report No. 22 (2020)

Interactive Error Resilience Beyond $\frac{2}{7}$

Klim Efremenko* Ben-Gurion University

Gillat Kol† Princeton University

Raghuvansh R. Saxena‡ Princeton University

Abstract

Interactive error correcting codes can protect interactive communication protocols against a constant fraction of adversarial errors, while incurring only a constant multiplicative overhead in the total communication. What is the maximum fraction of errors that such codes can protect against?

For the non-adaptive channel, where the parties must agree in advance on the order in which they communicate, Braverman and Rao prove that the maximum error resilience is $\frac{1}{4}$ (STOC, 2011). Ghaffari, Haeupler, and Sudan (STOC, 2014) consider the *adaptive* channel, where the order in which the parties communicate may not be fixed, and give a clever protocol that is resilient to a $\frac{2}{7}$ fraction of errors. This was believed to be optimal.

We revisit this result, and show how to overcome the $\frac{2}{7}$ barrier. Specifically, we show that, over the adaptive channel, every two-party communication protocol can be converted to a protocol that is resilient to $\frac{7}{24} > \frac{2}{7}$ fraction of errors with only a constant multiplicative overhead to the total communication. The protocol is obtained by a novel implementation of a feedback mechanism over the adaptive channel.

*klimefrem@gmail.com
†gillat.kol@gmail.com
‡rrsaxena@princeton.edu

ISSN 1433-8092

1 Introduction

We study the *error resilience* of *interactive coding* schemes [Sch96]: what is the maximum[1] $\theta > 0$, such that every two-party communication protocol can be simulated over a channel that corrupts θ fraction of the communicated symbols adversarially, and the simulation blows-up the communication by only a constant multiplicative factor?

One answer is that this maximum θ is $\frac{1}{4}$. It was observed in [BR11] that even the simple *exchange* task, where the communicating parties wish to exchange their inputs, cannot be performed when $\theta \geq \frac{1}{4}$. The reason is that there is a party, say Alice, that 'speaks' in no more than half of the rounds. Let Alice's input be x. The adversary can corrupt the first half of the messages sent by Alice to make it look like her input is some $x' \neq x$. If this happens, Bob cannot tell whether Alice has input x or x'. To prove that *any* task can be performed even when the noise rate approaches $\frac{1}{4}$, [BR11] constructed a beautiful simulation protocol, building on the groundbreaking work of [Sch96].

The impossibility result described above assumes the 'standard' communication complexity model, where parties take turns communicating. Thus, the order of communication in the protocol is determined in advance, and is independent of the parties' inputs and randomness, and of the messages they received previously[2]. Since the order of turns cannot be adapted after an execution commences, these protocols are called *non-adaptive* (*a.k.a. oblivious* or *static*).

A surprising result by Ghaffari, Haeupler, and Sudan [GHS14], shows that this assumption is not a mere technicality[3]. Specifically, they give an insightful scheme for encoding any (noiseless) protocol into a protocol resilient to $\frac{2}{7}$ fraction of adversarial corruptions, in the model where the order of transmissions is not predetermined. This model is a realistic generalization of the standard model, and is inspired by popular models in distributed computing abstracting wireless communication systems (*e.g.*, *radio networks*).

One key idea behind their error resilient protocol, is that it "dynamically" allocates more resources to the party that was corrupted more, thus circumventing the above lower bound. This scheme was believed to be tight. In fact, [GHS14] contains a proof of optimality that was later found to contain a vulnerability.

1.1 Our Results

In this work, we show that the maximum error resilience of adaptive interactive coding protocols is strictly greater than $\frac{2}{7}$.

[1] Actually, supremum.

[2] Indeed, the above impossibility result assumed that the identity of the party that communicates less is known is advance.

[3] We mention that [BR11] raise this question in their paper.

Theorem 1.1. *Let P be a two-party communication protocol*[4]*. For every $\theta < \frac{7}{24}$, there exists an adaptive protocol P' that simulates P and is resilient to θ fraction of adversarial noise. Moreover, the length of P' is linear in the length of P, and its alphabet set is of constant size (that depends on θ).*

Prior to our work, it was not even known if the $\frac{2}{7}$ barrier can be crossed with a protocol P' of *arbitrary length*. It can be shown that $\frac{2}{7}$ is the optimal error resilience that can be obtained by some "natural" families of interactive coding protocols (*i.e.*, symmetric or alternating protocols), even when waiving the constraint on the encoding length. Indeed, the protocol we design has a unique structure (see subsection 1.2 and section 2).

We mention that the best known upper bound on the error resilience of adaptive interactive coding schemes is $\frac{1}{3} = \frac{8}{24}$ [GHS14]. Pinning down the optimal value is an extremely intriguing question, and we conjecture that this value is strictly between $\frac{7}{24}$ and $\frac{8}{24}$.

1.2 Techniques

As a starting point, we observe that, had the adaptive channel been equipped with a *feedback* mechanism, its error resilience could have been improved (*cf.* [EGH16]). Inspired by this observation, we design a novel, error resilient, *online*, feedback mechanism over the (standard) adaptive channel, called the *tagging* mechanism. While not providing full fledged feedback, this tagging mechanism does allow us to send enough feedback to break the $\frac{2}{7}$ barrier. To make the mechanism work, we develop tools allowing the parties to estimate the number of corruptions injected at any given point in the execution of the protocol.

As explained in section 2, the tagging mechanism requires the simulation protocol to exhibit some 'non-standard' features: one inherent property is that Alice's and Bob's sides of the protocol are different – Bob is sending the feedback and Alice is receiving it. Another, perhaps surprising, source of asymmetry between the parties is that they transmit at *different rates*: Alice is transmitting at a higher rate at the beginning of the protocol, while Bob is transmitting at a higher rate later in the protocol. It can be shown that this kind of asymmetry between the rates is *required* in order to break the $\frac{2}{7}$ error resilience barrier, and symmetric schemes à la [GHS14] cannot be used. Our protocol also differs from other simulation protocols in the interactive coding literature in its decoding function, as in some cases, the parties' output is *not* the one that was seen most often.

We believe that our approach of implementing feedback over channels with no feedback via tagging, as well as our error measuring tools, can be adapted to study related problems.

1.3 Additional Related Works

The works most related to our paper are [BR11] and [GHS14] mentioned above.

[4]We assume, without loss of generality, that P is deterministic and non-adaptive, as every randomized protocol is a distribution over deterministic protocols, and as every adaptive protocol can be converted into a non-adaptive protocol by doubling the length of the protocol.

Since the study of coding for interactive communication was initiated by Schulman [Sch92, Sch93, Sch96], numerous works have been published in this area [GMS11, BR11, Bra12, KR13, BE17, BKN14, GMS14, GHK+18, EGH16, BGMO17, *e.g.*]. For a great survey of this field, see [Gel17]. It is, by now, well known that adaptive models can be much more powerful than non-adaptive models [Hae14, GH14, GHS14, AGS16, HV17, EKS18].

The question of maximum error resilience was also studied for feedback and erasure channels [EGH16, FGOS15, SW17, HSV18], and for a different adaptive model (interesting in its own right) where collisions do not occur [AGS16]. The works of [HKV15, WQC17, Ber64] show that the maximal error rate of the message transfer problem (one-way communication) can be improved in the presence of (even partial and noisy) feedback. Feedback was shown to also allow the construction of interactive codes with better rates [Pan13, GH17].

2 Overview of Our Error Resilient Protocol

In this section, we give an overview of our interactive coding scheme, highlighting our main ideas. Recall that our scheme takes a protocol P and converts it to a protocol P' that is resilient to $7/24$ fraction of adversarial errors, while only incurring constant multiplicative overhead.

2.1 The Adaptive Model

The (noisy) *adaptive model*, suggested by [GHS14], assumes that each party is equipped with a device that can either receive or transmit in every communication round, and that an adversary can corrupt some fraction of the rounds, called the '*corruption budget*'.

If, in some round, exactly one party transmits, and the adversary decides not to corrupt the transmission, then the other party receives the transmitted message. In the case where both parties decide to transmit, neither gets a message, as neither was listening for one. Finally, if both parties decide to receive, then since there is no one transmitting, nothing can be assumed about the received messages. This means that these messages are controlled by the adversary, and are not 'charged' to its 'corruption budget'.

While it seems pessimistic to assume that the messages received by the parties when they are both receiving are determined by the adversary, this is crucial in order to avoid 'signaling'. For example, if we were to assume that such simultaneous receives can be detected, then each party can use this to signal their input to the other party. For a formal definition of this model, see section subsection 3.2, and for additional motivation for the definition, see [Hae14, GHS14].

2.2 The [GHS14] Protocol

In their paper, [GHS14] show that the adaptive model can be used to get improved error resilience. Specifically, they show that for *every* communication task, there is a non-adaptive

protocol performing the task in the presence of any adversary whose corruption budget is less than $\frac{2}{7}$. For simplicity, we next describe the [GHS14] protocol for the restricted bit exchange task, where each party has a private input bit and both parties wish to know both bits.

The protocol consists of $7N$ rounds, for some $N > 0$. In the first $6N$ rounds, the parties take turns (alternate) in sending their bits to each other. We call these $6N$ rounds the 'non-adaptive part' of the protocol, as the order of transmissions is known in advance.

The reason for a non-adaptive part of length $6N$, is that after $6N$ rounds, *at least one of the parties is sure* about the bit of the other party. This is because the adversary corrupts $< \frac{2}{7} \cdot 7N = 2N$ messages, thus if a party receives the same value $2N$ times, it must be the correct value. Since one of the parties receives $< 2N/2 = N$ corrupted messages out of a total of $6N/2 = 3N$, this party receives the correct message $> 2N$ times and is thus sure about the value of the other party's bit.

The last N rounds of the [GHS14] protocol form its adaptive part, and in these rounds only the sure party transmits, while the other party listens. It can be shown that this suffices to make the unsure party sure. Note that since it is not known in advance which of the parties will be sure at the beginning of the adaptive part, these rounds are adaptive.

We mention that having at least one of the parties sure at the end of the non-adaptive part helps synchronize the parties and prevents the case where both parties listen in the same round in the adaptive part. Such rounds are particularly harmful as the received messages are adversarial. Furthermore, this property was believed to be necessary. Our proof shows otherwise.

2.3 The Power of Feedback

Our simulation protocol is quite involved, but as a starting point we observe that even a small amount of *feedback* can substantially improve error resilience in the adaptive setting. The following numerical toy example illustrates how.

Example. Let us shorten the non-adaptive part of the [GHS14] bit exchange protocol to $\frac{16}{3}N$ instead of $6N$ rounds, but also assume that at the end of this part Alice 'magically' knows if Bob is sure[5]. We claim that this extra information allows the parties to finish the protocol with only $\frac{4}{3}N$ additional rounds, yielding a protocol with $\frac{16}{3}N + \frac{4}{3}N = 6\frac{2}{3}N$ rounds, thereby getting a higher error resilience (as the corruption budget is still $2N$).

First, suppose that Bob is not sure after $\frac{16}{3}N$ rounds and Alice knows this. Then, no value was received by Bob $\geq 2N$ times, implying that $\geq \frac{8}{3}N - 2N = \frac{2}{3}N$ messages from Alice to Bob were corrupted. Since the adversary corrupts a total of $< 2N$ messages, $< 2N - \frac{2}{3}N = \frac{4}{3}N$ messages from Bob to Alice, out of $\frac{8}{3}N$, were corrupted. This means that the majority value received by Alice is correct and she can be sure of that.

We have thus shown that at least one of the parties is sure after the non-adaptive part. If this party transmits their input $\frac{4}{3}N$ additional times, the unsure party receives a total of

[5]Note that if the communication is over the adaptive channel with feedback, Alice can easily know if Bob is sure by checking if he received $\geq 2N$ uncorrupted messages.

$\frac{8}{3}N + \frac{4}{3}N = 4N$ messages, and the majority value must be correct as the total number of corruptions is $< 2N$.

2.4 Implementing the Feedback

The example shows that if Bob can send some feedback information about how sure he is along with his input, then Alice can use this information to be sure faster, improving the error resilience. Can we implement feedback in channels with no "built-in" feedback mechanism, like the adaptive channel? Our work gives an affirmative answer, at least to some extent.

The first challenge that one encounters when sending feedback over our channel, is that, like all the other messages in the channel, this feedback can also be corrupted by the adversary. This can be extremely harmful, as it may create rounds where both parties listen and thus receive adversarially chosen messages that do not count towards the adversary's corruption budget. This will be the case, if for instance, in the above example, Bob is unsure after the non-adaptive part, but Alice thinks that he is sure.

One way of making the feedback reliable is to send it in multiple rounds so that even if some of these rounds are corrupted, Alice can still decode some meaningful information out of the remaining rounds. It can be shown that waiting until the end of the non-adaptive part and then repeating the feedback either results in an insufficient number of repetitions or in wasted rounds, hurting the error resilience. This poses the following challenge: how can Bob start giving feedback about his sureness level at the end of the non-adaptive part, way *before* the execution reaches the end of the non-adaptive part?

Our online feedback mechanism. Our solution to this challenge is an '*online*' feedback mechanism, implemented through the addition of 'tags'. At a high level, if Bob 'believes' that the last message he received from Alice was corrupted, he will 'tag'[6] his next message to Alice to reflect that.

In more detail, to decide whether to tag, Bob maintains a lower bound, α, on the number of errors from Alice to Bob in the non-adaptive part. For the case of bit exchange, α would be the minimum between the number of times Bob receives 0 and the number of times Bob receives 1. This is a lower bound on the number of corruptions from Alice to Bob, as either Alice has the input 1 and all the 0s are due to corruptions or Alice has the input 0, and all the 1s are due to corruptions. Bob tags the next message he sends if the number of messages he already tagged is smaller than his current α.

Owing to the (non-trivial) fact that α is a good lower bound on the number of corruptions from Alice to Bob, tagging α messages is a way of communicating to Alice whether Bob is sure at the end of the non-adaptive part. Furthermore, it is a resilient way, as Bob basically transmits α in unary, thus to make Alice believe in a very different value of α, the adversary will need to spend many corruptions tagging (or removing tags from) many of Bob's messages.

[6] Tagging is implemented by simply adding a Boolean value to the message, indicating whether the message is meant to be tagged.

Alice's weighted decoding procedure. With the tagging procedure described above, Alice may receive two types of messages throughout the protocol, namely, those that are tagged and those that are not. If Alice receives many tags, she assumes that the value α for Bob was large, and therefore, that many of her messages were corrupted. Consequently, she will try to transmit more often to make Bob sure of her input.

What if the adversary corrupts Bob's messages and creates 'fake' tags? This scenario is especially problematic as the adversary can change the content of an untagged message sent by Bob, as well as tag it, with a single corruption. This fake tag may cause Alice to transmit an unneeded extra message during the adaptive part. Thus, one corruption effectively causes two different rounds to be useless.

To negate the adversary's desire of adding tags to effectively enlarge his corruption budget, in our protocol, Alice discounts tagged messages. This is done by assigning tagged messages a 'weight' that is lower than the weight of untagged messages. Specifically, two tagged messages counts as one untagged message. Alice then outputs the message with the highest weight, instead of the message that was received the most frequently. In particular, perhaps surprisingly, in some cases *Alice does not output the most common message she received*. Deciding what weight a tagged message should have requires careful consideration, as it needs to discourage the adversary from adding fake tags while still giving sufficient weight to messages that were legitimately tagged.

2.5 Different Rates for Alice and Bob

Above, we claimed that the tagging mechanism allows for reliable feedback as it spreads Bob's error count α across many rounds. We next show that α is still not spread enough: consider the implementation of the protocol in the example in subsection 2.3 using our tagging mechanism, and suppose that Bob is unsure at the end of the non-adaptive part. As explained in the example, this implies that $\geq \frac{8}{3}N - 2N = \frac{2}{3}N$ messages from Alice to Bob were corrupted. Assume that our mechanism works well and Bob indeed detects the corruptions and tags $\frac{2}{3}N$ of his messages. By investing another $\frac{2}{3}N$ corruptions (for a total of $\frac{4}{3}N < 2N$ corruptions), the adversary can remove these tags, making Alice believe that Bob is sure and thus listen instead of transmit.

A simple solution to the above problem is to spread α across even more rounds, *e.g.*, have Bob tag 2α messages. However, what if the adversary chooses to corrupt, say, the last t messages transmitted by Alice in the non-adaptive part? In this case, even if Bob detects those errors, he can only tag his last t messages in the non-adaptive part.

Asymmetric rates. To cope with this problem, we unveil another key idea in our construction: *asymmetric rates*. We have Bob transmit more towards the end of the non-adaptive part. Specifically, Bob transmits two messages for every message sent by Alice, and

thus can tag two messages for every corruption he detects[7].

A consequence of having Bob send more messages towards the end of the non-adaptive part, is that he will be transmitting fewer messages near the beginning of the non-adaptive part. This is because the total number of messages sent by the parties should be roughly the same, otherwise the adversary will target the messages sent by the party that sends fewer messages. Indeed, we divide the non-adaptive part of our protocol into two "stages", and have Alice transmit at a higher rate in the first stage (stageNA-1), and Bob transmit at a higher rate in stageNA-2.

We mention that *asymmetry in rate can be shown to be necessary*: if the parties alternate for long enough near the beginning of the protocol, then $\frac{2}{7}$ is the maximum error resilience achievable for general interactive tasks.

2.6 Our Protocol's Structure

Recall that our simulation scheme takes a protocol P and converts it to a protocol P' that is resilient to $\frac{7}{24}$ fraction of adversarial errors. Let x, y be inputs for the parties in P. The protocol P' consists of $24CM$ communication rounds, where C is constant and M is roughly the length of an execution of P with inputs x, y.

At an extremely high level, the non-adaptive part of P' simulates the execution of P with inputs x, y, a total of $20C$ times using an *interactive list-decodable code* that tolerates a very high error rate. Each such execution is called an 'iteration' and consists of M communication rounds. Let π^* be the correct transcript of P when it is run with inputs x, y. In iteration i, each party obtains a (possibly different) candidate[8] for π^*: Alice obtains π_i^A and Bob obtains π_i^B. In addition to executing P, in every iteration the parties also exchange the sets of all candidates they obtained so far.

The first $14C$ iterations in the non-adaptive part are called 'stageNA-1', and in these iterations Alice transmits 4 messages for every 3 messages transmitted by Bob. The next $6C$ iterations in the non-adaptive part are called 'stageNA-2', and in these iterations Bob transmits 2 messages for every message transmitted by Alice. See Figure 1.

On the other hand, the adaptive part (stageA) consists of the communication of $4CM$ messages, where Alice transmits in the first a rounds, for some a, and listens in the rest. Bob, in stageA, listens for $\leq a$ rounds and transmits in the rest. The party that transmits, sends their current list of candidates for π^*. The number a is calculated by Alice using the number of tagged messages she received, and Bob calculates a lower bound on a using the number of tagged messages he sent. Like the [GHS14] protocol, P' is designed to prevent rounds where both parties listen simultaneously.

[7]Note that since P may be highly interactive, we cannot afford to have only one of the parties speak for a long period of time. In particular, we cannot have only Bob speaking towards the end of the non-adaptive part and have to maintain a sufficient level of alternations.

[8]Since we use an interactive list-decodable code, the parties actually obtain a set of candidates in every iteration. We ignore this point in this sketch.

2.7 New Attack: Preying On the Weak

While asymmetric rates are necessary, they are a necessary evil (at least when it comes to highly interactive communication tasks), as they allow for new attacks: near the beginning of the simulation, when Bob is speaking less often, the adversary can corrupt all of Bob's messages to Alice using a small number of corruptions. If the task requires interaction between Alice and Bob, then all of Alice's messages sent in this part are now irrelevant. Thus, the adversary is able to effectively corrupt more messages than the number of errors it invests.

To deal with this problem, Bob maintains a variable k that captures the number of times that the adversary corrupted the communication from him to Alice when he was the quieter party. This is done as follows: recall that our resilient protocol P' simulates the execution of the original protocol P many times. The adversary may corrupt the communication from Bob to Alice in iteration i to make π_i^A incorrect, but π_i^A will always be consistent with Alice's input (meaning that if Bob's messages are as in π_i^A, Alice's messages will also be as in π_i^A). Observe that π^* is the only transcript that is consistent with both parties' inputs. Thus, since $\pi_i^A \neq \pi^*$, then in iteration $i+1$, when Bob receives π_i^A, he can tell that the adversary corrupted his communication to Alice in round i.

Recall that α is Bob's estimate of the number of corrupted messages he received from Alice and that k is his estimate of the number of corrupted messages that Alice received. In our protocol, Bob's actions depend on both α and k. Generally, if k is large, then Bob tends to be unsure, even if α is slightly smaller. Capturing the correct relation between these two error measures is another challenge.

3 Notation and Formal Problem Definition

3.1 Notation

We use '$\circ$' to denote string concatenation. We use 'ϵ' to denote the empty string (string consisting of zero symbols).

For $a, b \in \mathbb{Z}$, we define the sets $[a] = \{1, 2, \ldots, a\}$, $[a : b] = \{a, a+1, \ldots, b\}$, $(a : b] = \{a+1, a+2, \ldots, b\}$. Similarly, we define the sets $(a : b)$ and $[a : b)$.

Let s be a string. We denote by s_i the i^{th} symbol in s. We denote by $s_{\leq i}$ the string consisting of the first i symbols in s. Similarly, we define $s_{<i}$, $s_{\geq i}$, $s_{>i}$.

Ties are broken lexicographically in all arg max. For simplicity of notation, we sometimes omit floor and ceiling signs, *e.g.*, write a/b instead of $\lfloor a/b \rfloor$.

We use small letters as names for numerical variables. We use capital letters as names for sets and string variables. We use small Greek letters to denote protocol transcripts, where π is typically a transcript for the original protocol P and τ is a transcript for the simulation protocol. We use capital Greek letters to denote sets of transcripts, *e.g.,* Π is a set of transcripts π for the original protocol P. For an (adaptive or non-adaptive) protocol P,

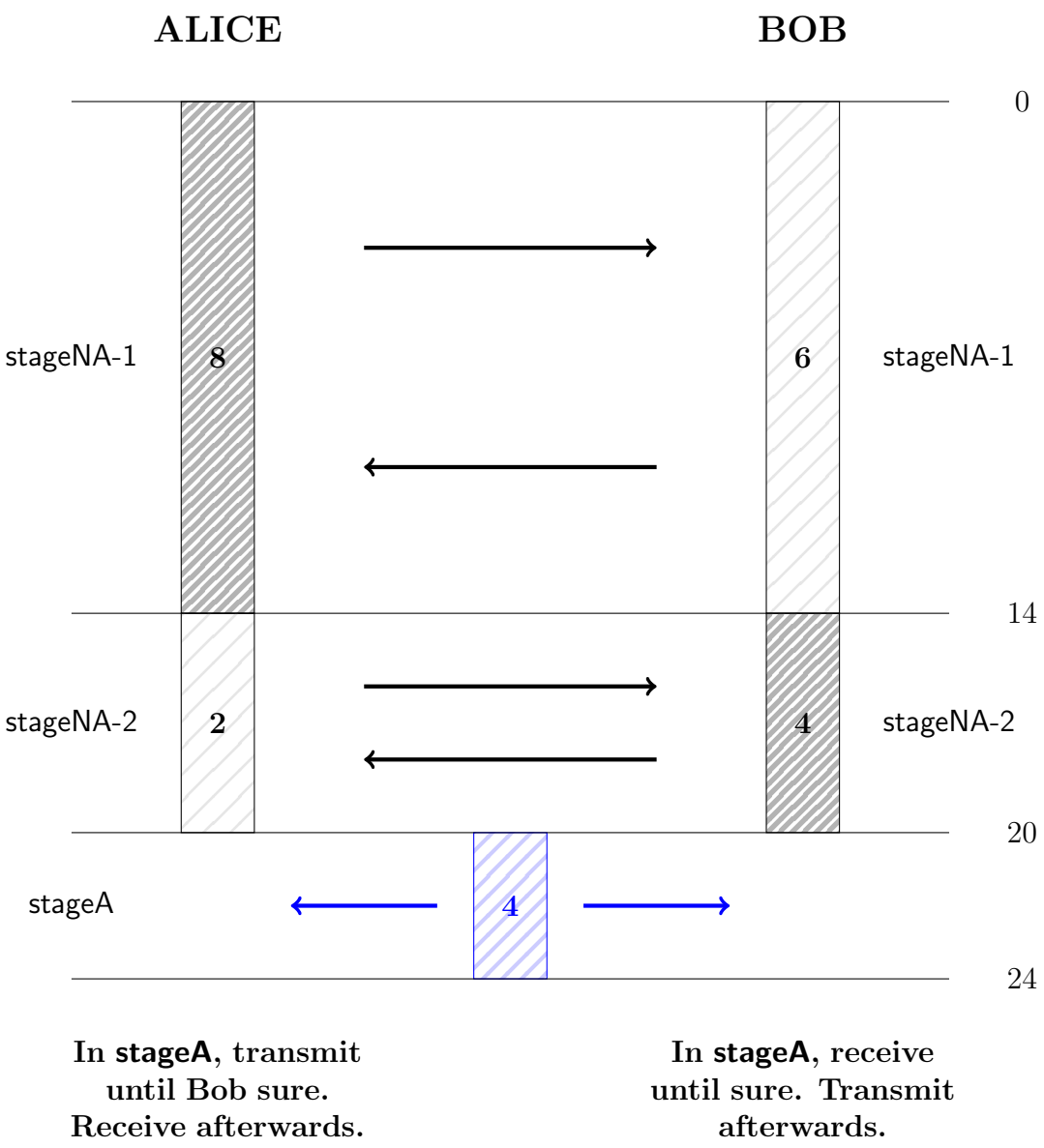

Figure 1: The 3 stages in our $\frac{7}{24}$ error resilient protocol with their relative lengths and rates.

we denote by P^A and P^B Alice's and Bob's sides of the protocol.

3.2 The Adaptive Model

Adaptive protocols. We describe the adaptive model used in the paper. The model was suggested and studied by [GHS14].

Let Γ be a set that does not contain the special symbol λ. A (deterministic) *adaptive protocol, P, over* Γ, is defined by an integer $T \in \mathbb{N}$ corresponding to length, inputs sets $\mathcal{X}, \mathcal{Y}$, output sets $\mathcal{X}', \mathcal{Y}'$, message computing functions f^A, f^B, and output functions out^A, out^B. These functions are of the types

$$f^A : \mathcal{X} \times (\Gamma \cup \{\lambda\})^* \to \Gamma \cup \{\lambda\}, \qquad f^B : \mathcal{Y} \times (\Gamma \cup \{\lambda\})^* \to \Gamma \cup \{\lambda\},$$

$$out^A : \mathcal{X} \times (\Gamma \cup \{\lambda\})^T \to \mathcal{X}', \qquad out^B : \mathcal{Y} \times (\Gamma \cup \{\lambda\})^T \to \mathcal{Y}'.$$

The protocol P is viewed as a communication protocol between two parties, Alice and Bob. We describe an *execution* of this protocol from Alice's perspective. Bob's protocol is analogous. Alice is assumed to be holding an input $x \in \mathcal{X}$. At any point in the execution of the protocol, Alice has a partial *received transcript* π^A. At the beginning of the execution, π^A is set to ϵ, the empty string. The execution proceeds in rounds. In round i, Alice computes $f^A(x, \pi^A)$. If $f^A(x, \pi^A) = \lambda$, we say that Alice *receives* (or *listens*) in round i. Otherwise, we say that Alice *transmits* (or *talks*). If Alice is receiving in round i, she gets a symbol $\pi^A_i \in \Gamma$. She appends this to the transcript π^A and continues executing the protocol. If Alice is transmitting in round i, she sends a symbol $a_i \in \Gamma$, appends λ to the transcript π^A and continues executing the protocol. At the end of the protocol, Alice outputs $out^A(x, \pi^A)$.

Transcripts. Let Γ be a set and let P be an adaptive protocol over Γ. Let $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $\pi^A, \pi^B \in (\Gamma \cup \{\lambda\})^T$. We say that π^A, π^B are *legal received transcripts* for P on inputs x, y, if for every $i \in [T]$, $\pi^A_i = \lambda$ if and only if $f^A(x, \pi^A_{<i}) \neq \lambda$, and, similarly, $\pi^B_i = \lambda$ if and only if $f^B(y, \pi^B_{<i}) \neq \lambda$. Observe that the tuple (x, y, π^A, π^B) describes an execution of P, in the sense that given this tuple, one can generate the view of both parties (or, more generally, compute any value known to the parties) at every point in the execution.

For all that follows, let $x \in \mathcal{X}$, $y \in \mathcal{Y}$, $\pi^A, \pi^B \in (\Gamma \cup \{\lambda\})^T$ be such that π^A, π^B are legal received transcripts for P on x, y.

Types of rounds. We define the set $\mathsf{TR}(x, y, \pi^A, \pi^B) \subseteq [T]$ to be the set of all rounds i such that Alice transmitted and Bob chose to receive in round i of the execution of P described by (x, y, π^A, π^B). Formally, $i \in \mathsf{TR}(x, y, \pi^A, \pi^B)$ if $\pi^A_i = \lambda$ and $\pi^B_i \neq \lambda$. The sets RT and RR are defined analogously.

Corruptions. Let $i \in \mathsf{RT}(x, y, \pi^A, \pi^B)$. We say that the adversary *corrupted Alice* (or, corrupted the message from Bob to Alice) in round i of the execution of P described

by (x, y, π^A, π^B), if the symbol sent by Bob in round i is different than the symbol received by Alice in round i. Formally, if $f^B(y, \pi^B_{<i}) \neq \pi^A_i$. We define the corruption of Bob similarly.

For $\theta \geq 0$, we say that π^A, π^B contain θ *fraction of corruptions* (with respect to the protocol P and inputs x, y), if the total fraction of rounds in which the adversary corrupted either Alice or Bob is θ.

3.3 The Protocol Simulation Problem

Let P and P' be adaptive protocols, and let the output functions of P' be out^A and out^B. We say that P' *simulates* P and is resilient to θ fraction of adversarial noise, if for any inputs x, y for P and every legal received transcripts τ^A, τ^B for P' on inputs x, y that contain at most θ fraction of corruptions, it holds that $out^A(x, \tau^A)$, $out^B(y, \tau^B)$ are legal transcripts for P on x, y that contain 0 fraction of corruptions.

4 Protocol Preliminaries

4.1 Transcripts

For all that follows, fix $\theta < \frac{7}{24}$ and let $\varepsilon = 10^{-5}\left(\frac{7}{24} - \theta\right)$. Fix a (non-adaptive) deterministic two-party communication protocol P to be simulated, and fix the inputs x and y for Alice and Bob.

The correct transcript π^*. Let $\mathcal{P}$ be the set of all possible transcripts of P. For inputs x', y', let $\pi^*(x', y') \in \mathcal{P}$ be the transcript of P when it is run on x' and y' and there are no corruptions (the "*correct*" transcript). We omit the inputs (x', y') when they are equal to (x, y).

Consistent transcripts. We say that $\pi \in \mathcal{P}$ is *consistent* with Bob's input y if, conditioned on Alice's messages being as they are in π, Bob's messages are also as they are in π. Similarly, we define consistency with Alice's input x.

Let $\Pi \subseteq \mathcal{P}$. Define $\text{consist}^B(\Pi) \subseteq \Pi$ to be the set of all $\pi \in \Pi$ that are consistent with Bob's input. Define $\text{consist}^A(\Pi)$ analogously. Observe that if $\Pi \subseteq \text{consist}^B(\mathcal{P})$ is a set of transcripts that are consistent with Bob's input, and $\pi^* \in \Pi$, then $\text{consist}^A(\Pi) = \{\pi^*\}$, else, $\text{consist}^A(\Pi) = \emptyset$. If the value of $\text{consist}^A(\Pi)$ or $\text{consist}^B(\Pi)$ is singleton, say $\text{consist}^A(\Pi) = \{\pi\}$, we sometimes shorten this to $\text{consist}^A(\Pi) = \pi$.

4.2 Distance Functions

Let $n \in \mathbb{N}$ and let Σ be a non-empty set. Let $C, W \in \Sigma^n$ and let $T \in \{\texttt{true}, \texttt{false}\}^n$. Let

$$\text{agree}(C, W) = |\{j \in [n] : \ C_j = W_j\}|,$$

$$\mathrm{agree}_T(C, W) = |\{j \in [n] : \ (T_j = \texttt{true}) \wedge (C_j = W_j)\}|.$$

Observe that for every T, C, W, it holds that $\mathrm{agree}_T(C, W) \leq \mathrm{agree}(C, W)$. For a boolean vector T, let $\overline{T}$ be a vector of the same length as T defined by $\overline{T}_j = \texttt{true} \iff T_j = \texttt{false}$.

4.3 List-Decodable Error Correcting Codes

Our simulation protocol uses the following interactive and non-interactive (standard) list-decodable error correcting codes.

4.3.1 (Standard) List-Decodable Codes

We use the following result about standard list-decodable codes (see [GI03, Gur06]).

Lemma 4.1. *There exists a constant c and a set Σ such that for all $n > 0$ if $n' = \lfloor n/c \rfloor$, there exists an encoding function $C_n : \{0,1\}^{\leq n'} \to \Sigma^n$ and a decoding function $\mathrm{listdec}_n(\cdot)$ such that the following hold:*

1. *For every $U \neq U' \in \{0,1\}^{\leq n'}$ it holds that $\mathrm{agree}(C_n(U), C_n(U')) \leq \varepsilon n$.*
2. *For every $W \in \Sigma^n$, it holds that $\mathrm{listdec}_n(W) \subseteq \{0,1\}^{\leq n'}$ and $|\mathrm{listdec}_n(W)| = O(1/\varepsilon^3)$. Furthermore, if $\mathrm{agree}(C_n(U), W) \geq \varepsilon n$ for some $U \in \{0,1\}^{\leq n'}$, then $U \in \mathrm{listdec}_n(W)$.*

We note that, in fact, there are functions C_n and $\mathrm{listdec}_n$ that can be computed efficiently. Nonetheless, as we will not need these functions to be efficient, for us, the codes in the lemma can be obtained from any standard error correcting code.

4.3.2 Interactive List-Decodable Codes

Let c and Σ be as promised by Lemma 4.1. We show the following result about interactive list-decodable codes.

Lemma 4.2. *There exists a constant $c' = 200! \cdot \frac{c}{\epsilon^6}$ and a set $\Sigma' \supseteq \Sigma$ such that for all $r^A, r^B \in [10]$, there exists a (non-adaptive) protocol $P_{r^A;r^B}$ and a functions $\mathrm{listdec}^A_{r^A;r^B}(\cdot)$, $\mathrm{listdec}^B_{r^A;r^B}(\cdot)$ such that:*

1. *The protocol $P_{r^A;r^B}$ has alphabet Σ' and satisfies $\mathsf{CC}(P_{r^A;r^B}) = c' \cdot \mathsf{CC}(P)$.*
2. *In the protocol $P_{r^A;r^B}$, Alice sends r^A messages for every r^B messages sent by Bob,* i.e., *the fraction of messages sent by Alice is $\frac{r^A}{r^A+r^B}$.*
3. *Let x', y' be inputs to Alice and Bob and $\alpha, \beta \in [0, 1)$ be such that $\alpha + \beta < 1 - \epsilon$. Let τ^A, τ^B be such that, after an execution of $P_{r^A;r^B}$ with inputs x', y' in which the adversary corrupts at most α fraction of Alice's transmissions and at most β fraction of Bob's transmissions, Alice has the transcript τ^A and Bob has the transcript τ^B.*

Then, we have $\text{listdec}^A_{r^A:r^B}(\tau^A) \subseteq \text{consist}^A(\mathcal{P})$, $\text{listdec}^B_{r^A:r^B}(\tau^B) \subseteq \text{consist}^B(\mathcal{P})$ *are sets of size at most* $O(1/\varepsilon)$ *such that* $\text{listdec}^A_{r^A:r^B}(\tau^A) \cap \text{listdec}^B_{r^A:r^B}(\tau^B) = \pi^*(x', y')$.

We note that τ^A *(resp.* τ^B*) above has both, the symbols sent by Alice (resp. Bob) and the symbols received by Alice (resp. Bob).*

Proof. Let $c'' < 10^5$ be the constant in Theorem 1 of [BE17]. Let P_{pad} be the protocol obtained by padding P to length $\frac{2c'}{c''(r^A+r^B)} \cdot \mathsf{CC}(P)$. Let Q be the list decoding protocol with alphabet Σ' for P_{pad} promised by Theorem 1 of [BE17]. Observe that $\mathsf{CC}(Q) = \frac{2c'}{r^A+r^B} \cdot \mathsf{CC}(P)$. We note that the scheme in [BE17] also works with "soft symbols", that is, in the case where the parties receive a distribution over symbols in each round (as opposed to a "regular" symbol), see Section 12 in their paper.

Alice's side of the protocol $P_{r^A:r^B}$ is given in Algorithm 1. Bob's side of the protocol is similar. Observe that for every 2 symbols exchanged in Q, the parties exchange $r^A + r^B$ symbols in $P_{r^A:r^B}$. Thus, $\mathsf{CC}(P_{r^A:r^B}) = c' \cdot \mathsf{CC}(P)$ and part one and part two hold. In the protocol, we denote by $Q^A(x, \tau)$ the symbol sent by Alice when running Q, given input x and prior (distributional) received transcript τ (assuming that it is Alice's turn to transmit). The decoding functions $\text{listdec}^A_{r^A:r^B}(\cdot)$, $\text{listdec}^B_{r^A:r^B}(\cdot)$ for $P_{r^A:r^B}$ is obtained from the decoding functions of [BE17] and removing the padding.

Algorithm 1 The protocol $P^A_{r^A:r^B}$

1: $\tau \leftarrow \epsilon$.
2: **for** $j \in [\mathsf{CC}(Q)]$ **do**
3: *If Alice transmits in round* j *of* Q*:* Transmit $Q^A(x, \tau)$, r^A times.
4: *If Bob transmits in round* j *of* Q*:* If $r^B \neq 0$, receive r^B symbols $a_1, \ldots, a_{r^B}$. Set $\tau = \tau \circ \mathcal{D}^A$, where $\mathcal{D}^A$ is the distribution over symbols induced by the messages received by Alice in this round. That is, for all a, $\mathcal{D}^A(a) = |\{i \in [r^B] : a_i = a\}|/r^B \in \{0/r^B, 1/r^B, \ldots, r^B/r^B\}$ is the fraction of messages received by Alice that were equal to a.
5: **end for**

□

5 The $\frac{7}{24}$ Error Resilient Protocol

5.1 Definitions

Length. Our simulation protocol consists of $24/\varepsilon$ executions of a subroutine called `Chunk` (referred to as "iterations"), such that $\mathsf{CC}(\texttt{Chunk}) = c' \cdot \mathsf{CC}(P) = M$, say, where c' is the constant from Lemma 4.2. Thus, the number of messages exchanged during our simulation protocol is $24N$, where $N = \frac{M}{\varepsilon}$.

Functions. The protocol uses the following simple functions:

$$\begin{aligned}
\texttt{AliceSure}(k) &= 7N - k - 11\varepsilon N,\\
\texttt{StartTag}_1(k) &= \frac{28}{3\varepsilon} + \frac{14k}{9\varepsilon N},\\
\texttt{StartTag}_2(k) &= \frac{11}{\varepsilon} + \frac{k}{\varepsilon N},\\
\texttt{BobReceives}(k,i) &= 3N - \frac{k}{3} + (i - 20/\varepsilon)\cdot \varepsilon N.
\end{aligned}$$

Variables. For simplicity, we use the same variable names in the protocols for Alice and Bob, if the variables have similar roles. Some of the variables are 'global' variables, shared by the simulation protocol and the subroutine `Chunk`, and are updated by `Chunk`.

Next, we give a short description of each of the variables used by the protocol. When we use the notation var_i^A, we describe the value of the variable var in Alice's protocol at the end of iteration i. If there is a variable in Bob's protocol with a similar role, we list var_i^A and var_i^B together. We omit the superscript A when var is not present in Bob's protocol (and vice versa). The notation var_i^A, var_i^B is also used in the analysis.

$\boldsymbol{\Pi_i^A}$ ($\boldsymbol{\Pi_i^B}$): The list of all candidates for the correct transcript π^*, obtained by Alice by the end of iteration i.

$\boldsymbol{\ell_i^A}$ ($\boldsymbol{\ell_i^B}$): The number of rounds in iteration i in which Alice listens.

$\boldsymbol{\tau_i^A}$ ($\boldsymbol{\tau_i^B}$): The transcript received by Alice for the i^{th} execution of the interactive list-decodable code simulating P.

$\boldsymbol{W_i^A}$ ($\boldsymbol{W_i^B}$): The encoding of Π_{i-1}^B received by Alice in iteration i.

$\boldsymbol{T_i^A}$: A boolean vector of length ℓ_i^A. For every round in iteration i in which Alice listens, the vector indicates whether the received message is tagged (messages in the adaptive part are assumed to not be tagged).

$\boldsymbol{t_i^B}$: The total number of tagged messages transmitted by Bob by the end of iteration i.

$\boldsymbol{\alpha_i}$: A lower bound on the total number of errors from Alice to Bob by the end of iteration i, maintained *only* by Bob.

$\boldsymbol{k_i^A}$ ($\boldsymbol{k_i^B}$): An estimate of the total number of corruptions from Alice to Bob by the end of iteration i, maintained by Alice.

$\boldsymbol{\Delta_i}$: $k_i^B - k_{i-1}^B$.

$\boldsymbol{m_i^A}$ ($\boldsymbol{m_i^B}$): The number of rounds in iteration i in which Alice listens and gets 'meaningful' information.

Agreements. The protocol and analysis use the following abbreviated notation for agreement functions: Let $\pi \in \mathcal{P}$. We define:

$$\begin{aligned}
\texttt{Agree}_i^B(\pi) &= \max_{\substack{\Pi:\text{consist}^B(\Pi)=\pi \\ |\Pi|\le O(1/\varepsilon^6)}} \{\text{agree}(C_{\ell_i^B}(\Pi), W_i^B)\},\\
\texttt{UntagAgree}_i(\pi) &= \max_{\substack{\Pi:\text{consist}^A(\Pi)=\pi \\ |\Pi|\le O(1/\varepsilon^6)}} \{\text{agree}_{\overline{T_i}}(C_{\ell_i^A}(\Pi), W_i^A)\},\\
\texttt{TagAgree}_i(\pi) &= \max_{\substack{\Pi:\text{consist}^A(\Pi)=\pi \\ |\Pi|\le O(1/\varepsilon^6)}} \{\text{agree}_{T_i}(C_{\ell_i^A}(\Pi), W_i^A)\},\\
\texttt{Agree}_i^A(\pi) &= \texttt{UntagAgree}_i(\pi) + \texttt{TagAgree}_i(\pi).
\end{aligned}$$

We define $\texttt{Agree}^B_{\le i}(\pi) = \sum_{i' \in [i]} \texttt{Agree}^B_{i'}(\pi)$, and similarly also $\texttt{UntagAgree}_{\le i}(\pi)$, $\texttt{TagAgree}_{\le i}(\pi)$, $\texttt{Agree}^A_{\le i}(\pi)$, *etc.*. We define $\texttt{Agree}^A_i(\emptyset)$ and $\texttt{Agree}^B_i(\emptyset)$ by replacing π by $\emptyset$ in the terms for $\texttt{Agree}^A_i(\pi)$ and $\texttt{Agree}^B_i(\pi)$.

When we write '$\texttt{Agree}^A(\pi)$', '$\texttt{Agree}^B(\pi)$', and '$\texttt{TagAgree}(\pi)$' in the protocol, we mean $\texttt{Agree}^A_{\le i}(\pi)$, $\texttt{Agree}^B_{\le i}(\pi)$, and $\texttt{TagAgree}_{\le i}(\pi)$, where i is the current iteration of the protocol. When we write '$\texttt{Agree}^A(\emptyset)$' and '$\texttt{Agree}^B(\emptyset)$', we mean $\texttt{Agree}^A_i(\emptyset)$ and $\texttt{Agree}^B_i(\emptyset)$, where i is the current iteration of the protocol.

5.2 The Protocol Chunk

Let $r^A, r^B \in [0 : 10]$. The protocol $\texttt{Chunk}_{r^A:r^B}$ simulates one execution of the original protocol P when the inputs are x, y. When running $\texttt{Chunk}_{r^A:r^B}$, Alice transmits r^A messages for every r^B messages transmitted by Bob. When $r^B = 0$, Alice is the only one to transmit. Similarly, when $r^A = 0$, Bob is the only one to transmit.

When $r^A, r^B > 0$, the protocol $\texttt{Chunk}_{r^A:r^B}$ performs one execution of the protocol $P_{r^A:r^B}$ of Lemma 4.2 besides updating some global variables. When $r^A \cdot r^B = 0$, we ensure that $(r^A, r^B) \in \{(0,1),(1,0)\}$. In this case, the protocol $\texttt{Chunk}_{r^A:r^B}$ simply works on and updates the global variables.

We make statements like 'If Alice transmits in round j of $P_{r^A:r^B}$' with the understanding that if $r^A \cdot r^B = 0$ and $P_{r^A:r^B}$ is not defined, then, if $(r^A, r^B) = (0,1)$, Alice never transmits and if $(r^A, r^B) = (1,0)$, then Alice always transmits (vice versa for Bob). Furthermore, in case $r^A \cdot r^B = 0$, the functions $P^A_{r^A:r^B}$ are defined to always be $\perp$.

The protocol $\texttt{Chunk}_{r^A:r^B}$ is given in Algorithm 2 (Alice's side) and Algorithm 3 (Bob's side). We note that the protocols are not entirely symmetric. Also, note that the sets Π^A, Π^B maintained by Alice and Bob are of size at most $O(1/\varepsilon^6)$. Since M was chosen to be sufficiently larger that $O(1/\varepsilon^6)$ in Lemma 4.2, the parties can always encode Π^A and Π^B using the codes C_{ℓ^*} for all $\ell^* = \Theta(M)$ used in the protocols.

Algorithm 2 The protocol $\texttt{Chunk}^A_{r^A:r^B}(\text{tag})$

$\tau, W, T \leftarrow \epsilon$.
$\ell \leftarrow \frac{r^B}{r^A+r^B} \cdot \epsilon N$.
for $j \in [M]$ **do**
    If Alice transmits in round j of $P_{r^A:r^B}$: Let $a = P^A_{r^A:r^B}(x, \tau)$, $b = C_{\varepsilon N-\ell}(\Pi)_{j-|W|}$. Transmit (a, b). Set $\tau \leftarrow \tau \circ a$.
    If Bob transmits in round j of $P_{r^A:r^B}$: Receive $(\tilde{a}, \tilde{b}, \tilde{c})$. Set $\tau \leftarrow \tau \circ \tilde{a}$, $W \leftarrow W \circ \tilde{b}$, $T \leftarrow T \circ (\tilde{c} \wedge \text{tag})$.
end for
$\Pi \leftarrow \Pi \cup \text{listdec}^A_{r^A:r^B}(\tau)$.
$k \leftarrow k + \min\left\{1, \frac{\max\{1,r^A\}}{\max\{1,r^B\}}\right\} \cdot \texttt{Agree}^A(\emptyset)$.
$m \leftarrow m + \ell - \texttt{Agree}^A(\emptyset)$.

Algorithm 3 Bob's side of the protocol $\texttt{Chunk}^B_{r^A:r^B}(\text{tag})$

$\tau, W \leftarrow \epsilon$.
$\ell \leftarrow \frac{r^A}{r^A+r^B}\epsilon N$.
for $j \in [M]$ **do**
    If Bob transmits in round j of $P_{r^A:r^B}$: Let $a = P^B_{r^A:r^B}(y, \tau)$, $b = C_{\varepsilon N-\ell}(\Pi)_{j-|W|}$, $c = \text{tag} \wedge \mathsf{True}(j - |W| > \frac{2}{3}\Delta)$. Transmit (a, b, c). Set $\tau \leftarrow \tau \circ a$, $t \leftarrow t + \mathbb{1}(c = \texttt{true})$.
    If Alice transmits in round j of $P_{r^A:r^B}$: Receive $(\tilde{a}, \tilde{b})$. Set $\tau \leftarrow \tau \circ \tilde{a}$, $W \leftarrow w \circ \tilde{b}$.
end for
$\Pi \leftarrow \Pi \cup \text{listdec}^B_{r^A:r^B}(\tau) \cup \left(\bigcup_{\substack{\tilde{\Pi} \in \text{listdec}_\ell(W) \\ |\tilde{\Pi}| \le O(1/\varepsilon^2)}} \text{consist}^B(\tilde{\Pi}) \right)$.
$\Delta \leftarrow \min\left\{1, \frac{\max\{1,r^B\}}{\max\{1,r^A\}}\right\} \cdot \texttt{Agree}^B(\emptyset)$.
$k \leftarrow k + \Delta$.
$m \leftarrow m + \ell - \texttt{Agree}^B(\emptyset)$.
$\alpha \leftarrow m - \max_\pi\{\texttt{Agree}^B(\pi)\}$.

5.3 The Simulation Protocol

Our simulation protocol, with error resilience $\frac{7}{24}$, is described in Algorithm 4 (Alice's side) and Algorithm 5 (Bob's side).

Algorithm 4 Alice's side of the simulation protocol

$k, m \leftarrow 0$, $\Pi \leftarrow \emptyset$.

▷ stageNA-1 (non-adaptive):

Run $\texttt{Chunk}^A_{4:3}(\texttt{true})$ $14/\varepsilon$ times.

▷ stageNA-2 (non-adaptive):

Run $\texttt{Chunk}^A_{1:2}(\texttt{true})$ $6/\varepsilon$ times.

▷ stageA (adaptive):

if $\max_\pi\{\texttt{Agree}^A(\pi)\} \geq \texttt{AliceSure}(k)$ **then**

    Run $\texttt{Chunk}^A_{1:0}(\texttt{false})$ $4/\varepsilon$ times.

else

    $a \leftarrow \frac{1}{\epsilon N} \cdot \min\{2N, \max_\pi\{\texttt{TagAgree}(\pi)\} - 2N\}$.

    Run $\texttt{Chunk}^A_{1:0}(\texttt{false})$ a times.

    Run $\texttt{Chunk}^A_{0:1}(\texttt{false})$ $4/\varepsilon - a$ times.

end if

▷ Compute output:

if $\max_\pi\{\texttt{Agree}^A(\pi)\} \geq \texttt{AliceSure}(k)$ **then**

    Output $\arg\max_\pi\{\texttt{Agree}^A(\pi)\}$.

else

    Output $\arg\max_\pi \left\{\texttt{Agree}^A(\pi) - \frac{\texttt{TagAgree}(\pi)}{2} \cdot \mathbb{1}\left(\pi = \arg\max_{\pi'}\{\texttt{TagAgree}(\pi')\}\right)\right\}$.

end if

Algorithm 5 Bob's side of the simulation protocol

$k, m, \alpha, t \leftarrow 0$, $\Pi \leftarrow \emptyset$, $tStart, tEnd \leftarrow$ `false`.

▷ stageNA-1 (non-adaptive):

for $i \in [1 : 14/\varepsilon]$ **do**
  $tStart \leftarrow tStart \vee \mathsf{True}(i > \mathtt{StartTag}_1(k))$.
  Run $\mathtt{Chunk}^B_{4:3}(tStart)$.
end for

▷ stageNA-2 (non-adaptive):

for $i \in (14/\varepsilon : 20/\varepsilon]$ **do**
  $tStart \leftarrow tStart \vee \mathsf{True}(i > \mathtt{StartTag}_2(k))$.
  $tEnd \leftarrow tEnd \vee \mathsf{True}(t > 2\alpha + 15\varepsilon N)$.
  Run $\mathtt{Chunk}^B_{1:2}(tStart \wedge \neg tEnd)$.
end for

▷ stageA (adaptive):

for $i \in (20/\varepsilon : 24/\varepsilon]$ **do**
  if $\alpha > \mathtt{BobReceives}(k, i)$ **then**
    Run $\mathtt{Chunk}^B_{1:0}(\mathtt{false})$.
  else
    Run $\mathtt{Chunk}^B_{0:1}(\mathtt{false})$.
  end if
end for

▷ Compute output:

Output $\pi \leftarrow \arg\max_\pi \{\mathtt{Agree}^B(\pi)\}$.

6 Protocol Analysis

Recall that we denote by π^* the transcript of the original protocol P when it is run on inputs x, y and there are no corruptions. For all that follows, fix an execution of our simulation protocol (Algorithm 4 and Algorithm 5) on inputs x, y, where the adversary corrupts at most $\mathtt{budget} = 7N - 1000\varepsilon N$ rounds of communication. It is clear from the way our protocols are defined that they use an alphabet of constant size and their communication is only a constant factor more than the communication of the noiseless protocol P. Our goal is to prove that the output of both parties for this execution is π^* and Theorem 1.1 follows.

6.1 Definitions

Variables. Recall that we use the notation var^A_i to indicate the value of the variable var in Alice's protocol at the end of iteration i. The notation var^B_i is similar. When $i = 0$, we

mean the value at the beginning of the protocol. We omit the subscript i when $i = 24/\varepsilon$.

RR rounds. Recall that we define RR to be the set of all iterations i such that there is a round in iteration i where both Alice and Bob are listening. The way our protocol is defined, it holds that $\mathsf{RR} \cap [20/\epsilon] = \emptyset$. Furthermore, if $i \in \mathsf{RR}$, then Alice and Bob are both receiving in *all* rounds in iteration i.

In Lemma 6.15, we show that $\mathsf{RR} = \emptyset$.

Indices. The analysis uses the following iteration indices in $[24/\varepsilon]$:

$\boldsymbol{i^A}$**:** The first iteration i such that $\pi^* \in \Pi_i^A$. If no such i exists, set $i^A = 24/\varepsilon + 1$.

$\boldsymbol{i^B}$**:** The first iteration i such that $\pi^* \in \Pi_i^B$. If no such i exists, set $i^B = 24/\varepsilon + 1$.

$\boldsymbol{i_{\mathsf{RR}}}$**:** The smallest $i \in \mathsf{RR}$. If $\mathsf{RR} = \emptyset$, set $i_{\mathsf{RR}} = 24/\varepsilon + 1$.

$\boldsymbol{i_s}$**:** The last iteration $i \leq 20/\varepsilon$ such that $tStart_i = \texttt{false}$.

$\boldsymbol{i_e}$**:** The last iteration $i \leq 20/\varepsilon$ such that $tEnd_i = \texttt{false}$.

Errors. The analysis uses the following notation to count the number of corruptions inserted by the adversary:

$\boldsymbol{e_S^{A\to B}}$: The total number of corruptions from Alice to Bob in the iterations in the set $S \subseteq [24/\varepsilon]$. When S is singleton, say $S = \{i\}$, then we simply write $e_i^{A\to B}$. When $S = [i], (i : 24/\epsilon], [i : 24/\epsilon]$, we write $e_{\leq i}^{A\to B}, e_{>i}^{A\to B}, e_{\geq i}^{A\to B}$ respectively. When $S = [24/\varepsilon]$, we omit S altogether.

$\boldsymbol{e_S^{B\to A}}$: The total number of corruptions from Bob to Alice in the iterations in the set $S \subseteq [24/\varepsilon]$. We use the same abbreviations as above.

The analysis also uses the notation t_i^A as short for $\max_\pi\{\texttt{TagAgree}_{\leq i}^A(\pi)\}$.

6.2 Lemmas Concerning List-Decodable Codes

Claim 6.1. *For all $s > 0$, $i \in [24/\epsilon]$, sets $\Pi'_1, \ldots, \Pi'_s \subseteq \mathcal{P}$ that are all different and satisfy $|\Pi'_r| \leq O(1/\varepsilon^6)$ for all $r \in [s]$ and all $T' \in \{\texttt{true}, \texttt{false}\}^{\ell_i^A}$, it holds that*

$$\sum_{r\in[s]} \mathrm{agree}_{T'}(C_{\ell_i^A}(\Pi'_r), W_i^A) \leq \sum_{j\in[\ell_i^A]} \mathbb{1}(T'_j = \texttt{true}) + \binom{s}{2}\varepsilon \cdot \ell_i^A.$$

The assertion also holds with A replaced by B everywhere.

Proof. We have:

$$\begin{aligned}
\sum_{r\in[s]} \mathrm{agree}_T(C_{\ell_i^A}(\Pi'_r), W_i^A) &= \sum_{r\in[s]}\sum_{j\in[\ell_i^A]} \mathbb{1}(T'_j = \texttt{true}) \cdot \mathbb{1}(C_{\ell_i^A}(\Pi'_r)_j = W_{i,j}^A) \\
&= \sum_{j\in[\ell_i^A]}\sum_{r\in[s]} \mathbb{1}(T'_j = \texttt{true}) \cdot \mathbb{1}(C_{\ell_i^A}(\Pi'_r)_j = W_{i,j}^A) \\
&= \sum_{j\in[\ell_i^A]} \mathbb{1}(T'_j = \texttt{true}) \cdot \left(\sum_{r\in[s]} \mathbb{1}(C_{\ell_i^A}(\Pi'_r)_j = W_{i,j}^A)\right) \\
&\le \sum_{j\in[\ell_i^A]} \mathbb{1}(T'_j = \texttt{true}) \cdot \left(1 + \sum_{r<r'\in[s]} \mathbb{1}(C_{\ell_i^A}(\Pi'_r)_j = C_{\ell_i^A}(\Pi'_{r'})_j)\right) \\
&\le \sum_{j\in[\ell_i^A]} \mathbb{1}(T'_j = \texttt{true}) + \sum_{j\in[\ell_i^A]}\sum_{r<r'\in[s]} \mathbb{1}(C_{\ell_i^A}(\Pi'_r)_j = C_{\ell_i^A}(\Pi'_{r'})_j) \\
&\le \sum_{j\in[\ell_i^A]} \mathbb{1}(T'_j = \texttt{true}) + \sum_{r<r'\in[s]} \mathrm{agree}(C_{\ell_i^A}(\Pi'_r), C_{\ell_i^A}(\Pi'_{r'})) \\
&\le \sum_{j\in[\ell_i^A]} \mathbb{1}(T'_j = \texttt{true}) + \binom{s}{2}\varepsilon \cdot \ell_i^A,
\end{aligned}$$

by the first property of the code $C_{\ell_i^A}$ in Lemma 4.1. □

Claim 6.2. *Let $i < \max\{i^A, i^B\}$. It holds that:*

1. *If $i \le 14/\varepsilon$, then $\frac{3}{4}e_i^{A\to B} + e_i^{B\to A} \ge \frac{3\varepsilon N}{7}(1-\varepsilon)$.*
2. *If $i \in (14/\varepsilon : 20/\varepsilon]$, then $e_i^{A\to B} + \frac{1}{2}e_i^{B\to A} \ge \frac{\varepsilon N}{3}(1-\varepsilon)$.*

Proof. Since $i < \max\{i^A, i^B\}$, the interactive list decoding code was not decoded correctly till iteration i. Formally, we have that $\pi^* \notin \Pi_i^A \cap \Pi_i^B \implies \pi^* \notin \mathrm{listdec}^A_{r_i^A:r_i^B}(\tau_i^A) \cap \mathrm{listdec}^B_{r_i^A:r_i^B}(\tau_i^B)$. As τ_i^A and τ_i^B are the transcripts $P_{r_i^A:r_i^B}$ with the same corruptions, we have by Lemma 4.2,

$$\frac{e_i^{A\to B}}{\ell_i^B} + \frac{e_i^{B\to A}}{\ell_i^A} \ge 1-\varepsilon. \tag{1}$$

1. Since $i \le 14/\varepsilon$, it holds that $\ell_i^A = \frac{3}{7}\varepsilon N$ and $\ell_i^B = \frac{4}{7}\varepsilon N$, and the assertion follows from Equation 1.
2. Since $i \le (14/\varepsilon : 20/\varepsilon]$, it holds that $\ell_i^A = \frac{2}{3}\varepsilon N$ and $\ell_i^B = \frac{1}{3}\varepsilon N$, and the assertion follows from Equation 1.

□

Corollary 6.3. *We have* $\max\{i^A, i^B\} \le \frac{17}{\varepsilon} < \frac{20}{\epsilon} < i_{\mathsf{RR}}$.

Proof. We have $\frac{20}{\varepsilon} < i_{\mathsf{RR}}$ by definition. Suppose $\max\{i^A, i^B\} > \frac{17}{\varepsilon}$. Applying the first part of Claim 6.2 to the iterations $i \in [14/\varepsilon]$ and the second part of Claim 6.2 to the iterations $i \in (14/\varepsilon : 17/\varepsilon]$, we get

$$e_i^{A\to B} + e_i^{B\to A} > 7N(1-\varepsilon) > \texttt{budget},$$

a contradiction. □

Claim 6.4. *Let $i \in (i^A : i^B)$. We have:*

$$e_i^{A\to B} \geq \ell_i^B \cdot (1-\epsilon)\,.$$

Proof. In iteration i, for $i \in (i^A : i^B)$, Alice sends $C_{\ell_i^B}(\Pi_{i-1}^A)$. As $i > i^A$ and the set Π^A only grows, we have that $\pi^* \in \Pi_{i-1}^A \implies \text{consist}^B(\Pi_{i-1}^A) = \{\pi^*\}$ while Bob receives W_i^B. Now, if $\Pi_{i-1}^A \in \text{listdec}_{\ell_i^B}(W_i^B)$, then, $\Pi_i^B \ni \{\pi^*\} \implies i \geq i^B$, a contradiction. Therefore, $\Pi_{i-1}^A \notin \text{listdec}_{\ell_i^B}(W_i^B)$. However, due to part 2 of Lemma 4.1, this means that $\text{agree}(C_{\ell_i^B}(\Pi_{i-1}^A), W_i^B) < \epsilon\ell_i^B$ implying the result.

□

6.3 Properties of $\texttt{Agree}^A$ and $\texttt{Agree}^B$

We start with the following corollaries of Claim 6.1. The reason we have separate claims for $\texttt{Agree}^A$ and $\texttt{Agree}^B$ is that their definitions are not symmetric.

Corollary 6.5. *For all $s > 0$, $i \in [24/\epsilon]$, if $\pi_1, \ldots, \pi_s \in \text{consist}^A(\mathcal{P}) \cup \{\emptyset\}$ are all different, it holds that*

$$\sum_{r\in[s]} \texttt{Agree}_i^A(\pi_r) \leq \left(1 + 2\cdot\binom{s}{2}\varepsilon\right)\ell_i^A.$$

Proof. We have

$$\begin{aligned}
\sum_{r\in[s]} \texttt{Agree}_i^A(\pi_r) &= \sum_{r\in[s]} \texttt{TagAgree}_i(\pi_r) + \sum_{r\in[s]} \texttt{UntagAgree}_i(\pi_r) \\
&= \sum_{r\in[s]} \max_{\substack{\Pi'_r:\text{consist}^A(\Pi'_r)=\pi_r \\ |\Pi'_r|\leq O(1/\varepsilon^6)}} \{\text{agree}_{T_i}(C_{\ell_i^A}(\Pi'_r), W_i^A)\} \\
&\qquad + \sum_{r\in[s]} \max_{\substack{\Pi''_r:\text{consist}^A(\Pi''_r)=\pi_r \\ |\Pi''_r|\leq O(1/\varepsilon^6)}} \{\text{agree}_{\overline{T_i}}(C_{\ell_i^A}(\Pi''_r), W_i^A)\} \\
&= \sum_{r\in[s]} \text{agree}_{T_i}(C_{\ell_i^A}(\Pi'_r), W_i^A) + \sum_{r\in[s]} \text{agree}_{\overline{T_i}}(C_{\ell_i^A}(\Pi''_r), W_i^A).
\end{aligned}$$

where we use Π'_r to denote $\arg\max_{\substack{\Pi'_r:\text{consist}^A(\Pi'_r)=\pi_r \\ |\Pi'_r|\leq O(1/\varepsilon^6)}} \{\text{agree}_{T_i}(C_{\ell_i^A}(\Pi'_r), W_i^A)\}$ and similarly, Π''_r denotes $\arg\max_{\substack{\Pi''_r:\text{consist}^A(\Pi''_r)=\pi_r \\ |\Pi''_r|\leq O(1/\varepsilon^6)}} \{\text{agree}_{\overline{T_i}}(C_{\ell_i^A}(\Pi''_r), W_i^A)\}$. Observe that Π'_r, for $r \in [s]$ are all

different as the values of $\text{consist}^A(\Pi'_r) = \pi_r$ are all different. Similarly, the sets Π''_r are all different. Thus, Claim 6.1 says

$$\sum_{r\in[s]} \texttt{Agree}_i^A(\pi_r) = \sum_{r\in[s]} \text{agree}_{T_i}(C_{\ell_i^A}(\Pi'_r), W_i^A) + \sum_{r\in[s]} \text{agree}_{\overline{T_i}}(C_{\ell_i^A}(\Pi''_r), W_i^A)$$
$$\leq \left(1 + 2\cdot\binom{s}{2}\varepsilon\right)\ell_i^B.$$

□

Corollary 6.6. *For all $s > 0$, $i \in [24/\epsilon]$, if $\pi_1, \ldots, \pi_s \in \text{consist}^B(\mathcal{P}) \cup \{\emptyset\}$ are all different, it holds that*

$$\sum_{r\in[s]} \texttt{Agree}_i^B(\pi_r) \leq \left(1 + \binom{s}{2}\varepsilon\right)\ell_i^B.$$

Proof. Similar to the proof of Corollary 6.5 above. □

Claim 6.7. *Let $i \notin \mathsf{RR}$. If $i \leq i^A$, we have $e_i^{A\to B} \geq \ell_i^B - \texttt{Agree}_i^B(\emptyset)$.*
If $i > i^A$, we have $e_i^{A\to B} \geq \ell_i^B - \texttt{Agree}_i^B(\pi^*)$.
The assertion also holds with the roles of A and B switched.

Proof. The set Π_i^A only contains transcripts that are consistent with Alice's inputs, and the only transcript that is consistent with both Alice's and Bob's inputs is π^*. Consider an iteration $i \leq i^A$. In iteration i, Alice sends $C_{\ell_i^B}(\Pi_{i-1}^A)$ while Bob receives W_i^B. Furthermore, as $i - 1 < i^A$, we have $\pi^* \notin \Pi_{i-1}^A \implies \text{consist}^B(\Pi_{i-1}^A) = \emptyset$. We also have $|\Pi_{i-1}^A| \leq O(1/\varepsilon^6)$. As $i \notin \mathsf{RR}$,

$$e_i^{A\to B} = \ell_i^B - \text{agree}(C_{\ell_i^B}(\Pi_{i-1}^A), W_i^B) \geq \ell_i^B - \texttt{Agree}_i^B(\emptyset).$$

Similarly, for an iteration $i > i^A$. In iteration i, we have $\pi^* \notin \Pi_{i-1}^A \implies \text{consist}^B(\Pi_{i-1}^A) = \{\pi^*\}$. We also have $|\Pi_{i-1}^A| \leq O(1/\varepsilon^6)$. As $i \notin \mathsf{RR}$,

$$e_i^{A\to B} = \ell_i^B - \text{agree}(C_{\ell_i^B}(\Pi_{i-1}^A), W_i^B) \geq \ell_i^B - \texttt{Agree}_i^B(\pi^*).$$

□

Claim 6.8. *For $i < i^A$ and all π,*

$$e_i^{A\to B} + e_i^{B\to A} \geq (k_i^B - k_{i-1}^B) + \ell_i^B - \texttt{Agree}_i^B(\emptyset) - \varepsilon\ell_i^B.$$

This assertion also holds with the roles of A and B switched.

Proof. By Corollary 6.3, we have $i < \frac{20}{\epsilon} < i_{\mathsf{RR}}$. For $i \leq 14/\varepsilon$, we use the first part of Claim 6.2 to get

$$e_i^{A\to B} + e_i^{B\to A} \geq \frac{3}{4}\cdot\ell_i^B\cdot(1-\varepsilon) + \frac{1}{4}e_i^{A\to B}.$$

Next, we use the first part of Claim 6.7 to get,

$$e_i^{A\to B} + e_i^{B\to A} \geq \ell_i^B - \frac{1}{4}\texttt{Agree}_i^B(\emptyset) - \varepsilon\ell_i^B \geq \ell_i^B - \texttt{Agree}_i^B(\emptyset) + (k_i^B - k_{i-1}^B) - \varepsilon\ell_i^B.$$

For $i > 14/\varepsilon$, we use the second part of Claim 6.2 to get

$$e_i^{A\to B} + e_i^{B\to A} \geq \ell_i^B \cdot (1-\varepsilon) \geq \ell_i^B - \texttt{Agree}_i^B(\emptyset) + (k_i^B - k_{i-1}^B) - \varepsilon\ell_i^B.$$

□

Claim 6.9. *Let $i \notin \mathsf{RR}$. If $i > i^B$, we have the following*

1. $e_i^{B\to A} \geq (t_i^B - t_{i-1}^B) - \texttt{TagAgree}_i(\pi^*)$.
2. $e_i^{B\to A} \geq \ell_i^A - \texttt{Agree}_i^A(\pi^*) + \max\{\texttt{TagAgree}_i(\pi^*) - (t_i^B - t_{i-1}^B), 0\}$.

Proof. Consider an iteration $i > i^B$. In iteration i, Bob sends $C_{\ell_i^A}(\Pi_{i-1}^B)$ in ℓ_i^A messages. As $i > i^B$ and the set Π^B only grows, we have that $\pi^* \in \Pi_{i-1}^B \implies \text{consist}^A(\Pi_{i-1}^B) = \{\pi^*\}$. We also have $|\Pi_{i-1}^B| \leq O(1/\varepsilon^6)$. A total $t_i^B - t_{i-1}^B$ of these ℓ_i^A messages are tagged. Alice receives W_i^A in ℓ_i^A messages.

The first part follows by only considering the $t_i^B - t_{i-1}^B$ messages that were tagged by Bob. For these messages, we have,

$$e_i^{B\to A} \geq (t_i^B - t_{i-1}^B) - \texttt{TagAgree}_i(\pi^*).$$

Note that if $\texttt{TagAgree}_i(\pi^*) \leq t_i^B - t_{i-1}^B$, then the second part follows from Claim 6.7. It is therefore sufficient to show that $e_i^{B\to A} \geq \ell_i^A - (t_i^B - t_{i-1}^B) - \texttt{UntagAgree}_i(\pi^*)$. This follows by considering the $\ell_i^A - (t_i^B - t_{i-1}^B)$ messages by Bob that we not tagged. For these messages, we get:

$$e_i^{B\to A} \geq \ell_i^A - (t_i^B - t_{i-1}^B) - \texttt{UntagAgree}_i(\pi^*).$$

□

6.4 Errors From Alice to Bob

Lemma 6.10. *For all $i \in [24/\varepsilon]$, it holds that*

$$\alpha_{i-1} - \varepsilon\ell_i^B \leq \alpha_i \leq \alpha_{i-1} + \ell_i^B - \Delta_i.$$

Proof. Let $i \in [24/\varepsilon]$. In Algorithm 3, if $r_i^A = 0$, then $\alpha_i = \alpha_{i-1}$ and $\Delta_i = 0$. In this case, the assertion reduces to $\alpha_{i-1} - \varepsilon\ell_i^B \leq \alpha_{i-1} \leq \alpha_{i-1} + \ell_i^B$, which clearly holds. We thus assume that $r_i^A \neq 0$.

We first show the upper bound on α_i. Since $r_i^A \neq 0$, we have $\Delta_i = \min\left\{1, \frac{\max\{1, r_i^B\}}{r_i^A}\right\} \cdot \texttt{Agree}_i^B(\emptyset) \leq \texttt{Agree}_i^B(\emptyset)$. Therefore,

$$\begin{aligned}\alpha_i &= m_i^B - \max_\pi \{\texttt{Agree}_{\leq i}^B(\pi)\} \leq m_{i-1}^B + \ell_i^B - \texttt{Agree}_i^B(\emptyset) - \max_\pi \{\texttt{Agree}_{\leq i-1}^B(\pi)\} \\ &= \alpha_{i-1} + \ell_i^B - \texttt{Agree}_i^B(\emptyset) \leq \alpha_{i-1} + \ell_i^B - \Delta_i.\end{aligned}$$

We turn to show the lower bound on α_i: It holds that

$$\alpha_{i-1} = m_{i-1}^B - \max_\pi \{\texttt{Agree}_{\leq i-1}^B(\pi)\} = m_i^B - \ell_i^B + \texttt{Agree}_i^B(\emptyset) - \max_\pi \{\texttt{Agree}_{\leq i-1}^B(\pi)\}$$

$$\leq m_i^B + \varepsilon \ell_i^B - \max_{\pi} \{\mathtt{Agree}_i^B(\pi)\} - \max_{\pi} \{\mathtt{Agree}_{\leq i-1}^B(\pi)\} \qquad \text{(by Corollary 6.6)}$$
$$\leq m_i^B + \varepsilon \ell_i^B - \max_{\pi} \{\mathtt{Agree}_{\leq i}^B(\pi)\} = \alpha_i + \varepsilon \ell_i^B.$$

□

Lemma 6.11. *It holds that:*

1. $e_{<i_{\mathsf{RR}}}^{A\to B} \geq \alpha_{i_{\mathsf{RR}}-1}$.
2. $e_{<i_{\mathsf{RR}}}^{A\to B} + e_{\leq i^A}^{B\to A} \geq \alpha_{i_{\mathsf{RR}}-1} + k_{i_{\mathsf{RR}}-1}^B - 100\varepsilon N$.
3. *For all* $i \leq \min\{\max\{i^A, i^B\}, 14/\varepsilon\}$, *it holds that* $e_{\leq i}^{A\to B} \geq \frac{4\varepsilon N}{7} i - \frac{4}{3} \cdot k_i^B - 25\varepsilon N$.
4. *For all* $\pi \neq \pi^*$, *it holds that* $\mathtt{Agree}_{<i_{\mathsf{RR}}}^B(\pi) \leq \mathtt{budget} - k_{i_{\mathsf{RR}}-1}^B + 400\varepsilon N$.

Proof. Note that Corollary 6.3 implies $i^A < i^{\mathsf{RR}}$. We prove each part separately, and use the following equations in our proofs. Firstly, note that, for $i \in [24/\epsilon]$,

$$\ell_i^B = (m_i^B - m_{i-1}^B) + \mathtt{Agree}_i^B(\emptyset) \geq (m_i^B - m_{i-1}^B) + (k_i^B - k_{i-1}^B).$$

The foregoing equation implies:

$$\begin{aligned}
&\sum_{i\leq i^A} \ell_i^B - \mathtt{Agree}_i^B(\emptyset) + \sum_{i\in(i^A:i_{\mathsf{RR}})} \ell_i^B - \mathtt{Agree}_i^B(\pi^*) \\
&\quad\geq \sum_{i\leq i^A} m_i^B - m_{i-1}^B + \sum_{i\in(i^A:i_{\mathsf{RR}})} (m_i^B - m_{i-1}^B) + (k_i^B - k_{i-1}^B) - \mathtt{Agree}_i^B(\pi^*) \\
&\quad\geq m_{i_{\mathsf{RR}}-1}^B + (k_{i_{\mathsf{RR}}-1}^B - k_{i^A}^B) - \max_{\pi} \{\mathtt{Agree}_{<i_{\mathsf{RR}}}^B(\pi)\} \\
&\quad\geq \alpha_{i_{\mathsf{RR}}-1} + (k_{i_{\mathsf{RR}}-1}^B - k_{i^A}^B).
\end{aligned} \tag{2}$$

We now prove the claims in the lemma statement.

1. Adding Claim 6.7 for all $i < i_{\mathsf{RR}}$, we get

   $$e_{<i_{\mathsf{RR}}}^{A\to B} \geq \sum_{i\leq i^A} \ell_i^B - \mathtt{Agree}_i^B(\emptyset) + \sum_{i\in(i^A:i_{\mathsf{RR}})} \ell_i^B - \mathtt{Agree}_i^B(\pi^*) \geq \alpha_{i_{\mathsf{RR}}-1},$$

   by Equation 2.

2. Adding Claim 6.8 for $i < i^A$ and Claim 6.7 for all $i^A < i < i_{\mathsf{RR}}$, we get (using $k_{i^A} - k_{i^A-1}, \ell_i^B \leq \epsilon N$)

   $$\begin{aligned}
   e_{<i_{\mathsf{RR}}}^{A\to B} + e_{\leq i^A}^{B\to A} &\geq k_{i^A}^B - 100\epsilon N + \sum_{i\leq i^A} \ell_i^B - \mathtt{Agree}_i^B(\emptyset) + \sum_{i\in(i^A:i_{\mathsf{RR}})} \ell_i^B - \mathtt{Agree}_i^B(\pi^*) \\
   &\geq \alpha_{i_{\mathsf{RR}}-1} + k_{i_{\mathsf{RR}}-1}^B - 100\varepsilon N. \qquad \text{(Equation 2)}
   \end{aligned}$$

3. Fix $i \le \min\{\max\{i^A, i^B\}, 14/\varepsilon\}$. Adding Claim 6.7 for all $i' \le \min\{i, i^A\}$, we get

$$e^{A\to B}_{\le \min\{i,i^A\}} \ge \frac{4\varepsilon N}{7} \cdot \min\{i, i^A\} - \mathtt{Agree}^B_{\le \min\{i,i^A\}}(\emptyset) \ge \frac{4\varepsilon N}{7} \cdot \min\{i, i^A\} - \frac{4}{3} \cdot k^B_{\min\{i,i^A\}}.$$

Adding Claim 6.4 for all $\min\{i, i^A\} < i' < i$, we get

$$e^{A\to B}_{\le i} - e^{A\to B}_{\le \min\{i,i^A\}} \ge \frac{4\varepsilon N}{7} \cdot (1-\epsilon) \cdot \big(i - 1 - \min\{i, i^A\}\big) \ge \frac{4\varepsilon N}{7} \cdot \big(i - \min\{i, i^A\}\big) - 25\epsilon N.$$

Combining, we get the result.

4. Fix $\pi \ne \pi^*$. Adding Claim 6.8 for $i < i^A$ and Claim 6.7 for all $i^A < i < i_{\mathsf{RR}}$, we get

$$\begin{aligned}
\mathtt{budget} &\ge e^{A\to B}_{<i_{\mathsf{RR}}} + e^{B\to A}_{\le i^A} \\
&\ge k^B_{i^A} - 100\epsilon N + \sum_{i \le i^A} \ell^B_i - \mathtt{Agree}^B_i(\emptyset) + \sum_{i \in (i^A : i_{\mathsf{RR}})} \ell^B_i - \mathtt{Agree}^B_i(\pi^*) \\
&\ge k^B_{i^A} - 395\epsilon N + \sum_{i \le i^A} \mathtt{Agree}^B_i(\pi) + \sum_{i \in (i^A : i_{\mathsf{RR}})} \mathtt{Agree}^B_i(\emptyset) + \mathtt{Agree}^B_i(\pi) \\
&\qquad\qquad \text{(Corollary 6.6)} \\
&\ge k^B_{i^A} - 395\epsilon N + \mathtt{Agree}^B_{<i_{\mathsf{RR}}}(\pi) + \sum_{i \in (i^A : i_{\mathsf{RR}})} (k^B_i - k^B_{i-1}) \\
&\ge k^B_{i_{\mathsf{RR}}-1} - 400\varepsilon N + \mathtt{Agree}^B_{<i_{\mathsf{RR}}}(\pi).
\end{aligned}$$

The result then follows.

□

6.5 Errors from Bob to Alice

In the following lemma, we denote $\psi = \arg\max_\pi\{\mathtt{TagAgree}(\pi)\}$ and define

$$\mathsf{util}(\pi) = \mathtt{Agree}^A(\pi) - \frac{t^A}{2} \cdot \mathbb{1}(\pi = \psi).$$

Recall that $t^A = \max_\pi\{\mathtt{TagAgree}(\pi)\}$.

Lemma 6.12. *We have*

1. *If $i^B \le i_s$, then $e^{B\to A}_{>i_s} \ge \max\{t^B - t^A, 0\}$.*
2. *If $e^{B\to A}_{\le 20/\varepsilon} \le 3N$, then $\mathtt{Agree}^A_{\le 20/\varepsilon}(\pi^*) \ge \mathtt{AliceSure}(k^A_{20/\varepsilon})$.*
3. *For all $\pi \ne \pi^*$, it holds that,*

$$\mathtt{Agree}^A_{<i_{\mathsf{RR}}}(\pi) \le \mathtt{budget} - k^A_{i_{\mathsf{RR}}-1} + 700\varepsilon N < \mathtt{AliceSure}(k^A_{i_{\mathsf{RR}}-1}).$$

Furthermore, if $\mathsf{RR} = \emptyset$, we also have

1. *If $\psi = \pi^*$ and $i^B \le i_s$, then*

$$e^{B\to A} \ge \sum_{i\in[24/\varepsilon]} \ell_i^A - k^A - \max\{i^B - 14/\epsilon, 0\} \cdot \frac{\epsilon N}{3} + \max\{t^A - t^B, 0\} - \texttt{Agree}(\pi^*).$$

2. *If $\psi \ne \pi^*$, then for all $\pi' \ne \pi^*$, we have $e^{B\to A} \ge \mathsf{util}(\pi') - 200\epsilon N + t^A/2$.*

Proof. We prove each part separately.

1. Clearly $e^{B\to A}_{>i_s} \ge 0$. Adding Claim 6.9 for all $20/\varepsilon \ge i > i_s \ge i^B$, we get

$$e^{B\to A}_{>i_s} \ge \sum_{i_s<i\le 20/\varepsilon} ((t_i^B - t_{i-1}^B) - \texttt{TagAgree}_i^A(\pi^*)) \ge t^B - t^A,$$

   as Bob does not tag messages in iterations $i \in [i_s] \cup (20/\epsilon : 24/\epsilon]$.

2. We first show that $i^B \le 14/\varepsilon$. Suppose not. Then, Claim 6.2 says

$$\frac{3}{4} \cdot e^{A\to B}_{\le 14/\varepsilon} + e^{B\to A}_{\le 14/\varepsilon} \ge 6N(1-\epsilon) \implies e^{A\to B}_{\le 14/\varepsilon} + e^{B\to A}_{\le 14/\varepsilon} \ge \texttt{budget},$$

   due to our assumption that $e^{B\to A}_{\le 20/\varepsilon} \le 3N$. This is a contradiction, and therefore, we have $i^B \le 14/\varepsilon$. Adding Claim 6.7 for all $i \in [20/\varepsilon]$, we get

$$\begin{aligned} 3N \ge e^{B\to A}_{\le 20/\varepsilon} &\ge \sum_{i\le 20/\varepsilon} \ell_i^A - \sum_{i\le i^B} \texttt{Agree}_i^A(\emptyset) - \sum_{i^B<i\le 20/\epsilon} \texttt{Agree}_i^A(\pi^*) \\ &\ge 10N - k^A_{i^B} - \texttt{Agree}^A_{\le 20/\epsilon}(\pi^*). \end{aligned}$$

   where the second step uses $i^B \le 14/\varepsilon$. Rearranging gives the result.

3. By Corollary 6.3, we have $i^B < i^{\mathsf{RR}}$. Adding Claim 6.8 for $i < i^B$ and Claim 6.7 for all $i^B < i < i_{\mathsf{RR}}$, we get

$$\begin{aligned} \texttt{budget} &\ge e^{A\to B}_{\le i^B} + e^{B\to A}_{<i_{\mathsf{RR}}} \\ &\ge k^A_{i^B} - 100\epsilon N + \sum_{i\le i^B} \ell_i^A - \texttt{Agree}_i^A(\emptyset) + \sum_{i\in(i^B:i_{\mathsf{RR}})} \ell_i^A - \texttt{Agree}_i^A(\pi^*) \\ &\ge k^A_{i^B} - 695\epsilon N + \sum_{i\le i^B} \texttt{Agree}_i^A(\pi) + \sum_{i\in(i^B:i_{\mathsf{RR}})} \texttt{Agree}_i^A(\emptyset) + \texttt{Agree}_i^A(\pi) \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(Corollary 6.5)} \\ &\ge k^A_{i^B} - 695\epsilon N + \texttt{Agree}^A_{<i_{\mathsf{RR}}}(\pi) + \sum_{i\in(i^A:i_{\mathsf{RR}})} k_i^A - k_{i-1}^A \\ &\ge k^A_{i_{\mathsf{RR}}-1} - 700\varepsilon N + \texttt{Agree}^A_{<i_{\mathsf{RR}}}(\pi). \end{aligned}$$

   The result then follows.

For the furthermore part, we have

1. Adding Claim 6.7 for $i \le i^B$ and part two of Claim 6.9 for $i > i^B$, we get:

$$e^{B\to A} \ge \sum_{i\in[24/\varepsilon]} \ell_i^A - \sum_{i\le i^B} \texttt{Agree}_i^A(\emptyset) + \sum_{i>i^B} \left(\max\{\texttt{TagAgree}_i(\pi^*) - (t_i^B - t_{i-1}^B), 0\} - \texttt{Agree}_i^A(\pi^*)\right).$$

Using the fact that $0 \le \texttt{TagAgree}_i(\pi^*) \le \texttt{Agree}_i^A(\pi^*)$ for all i, we get

$$\begin{aligned} e^{B\to A} \ge \sum_{i\in[24/\varepsilon]} \ell_i^A - \sum_{i\le i^B} \texttt{Agree}_i^A(\emptyset) + \max\{\texttt{TagAgree}_{\le i^B}(\pi^*), 0\} \\ + \sum_{i>i^B} \max\{\texttt{TagAgree}_i(\pi^*) - (t_i^B - t_{i-1}^B), 0\} - \texttt{Agree}^A(\pi^*). \end{aligned}$$

We next use $i^B \le i_s \implies t_{i^B}^B = 0$ and the fact that $\sum_i \max\{c_i, 0\} \ge \max\{\sum_i c_i, 0\}$ to get

$$\begin{aligned} e^{B\to A} &\ge \sum_{i\in[24/\varepsilon]} \ell_i^A - \sum_{i\le i^B} \texttt{Agree}_i^A(\emptyset) + \max\{\texttt{TagAgree}(\pi^*) - t^B, 0\} - \texttt{Agree}^A(\pi^*) \\ &\ge \sum_{i\in[24/\varepsilon]} \ell_i^A - \sum_{i\le i^B} \texttt{Agree}_i^A(\emptyset) + \max\{t^A - t^B, 0\} - \texttt{Agree}^A(\pi^*). \end{aligned}$$

Next, observe that, for $i \le 14/\varepsilon$, we have $\texttt{Agree}_i^A(\emptyset) = k_i^A - k_{i-1}^A$. For $i \in (14/\varepsilon : i^B]$, we have $\frac{1}{2}\texttt{Agree}_i^A(\emptyset) = (k_i^A - k_{i-1}^A) \implies \texttt{Agree}_i^A(\emptyset) \le (k_i^A - k_{i-1}^A) + \ell_i^A/2 = (k_i^A - k_{i-1}^A) + \frac{\varepsilon N}{3}$ as $i^B \le 20/\varepsilon$ by Corollary 6.3. We get

$$e^{B\to A} \ge \sum_{i\in[24/\varepsilon]} \ell_i^A - k^A - \max\{i^B - 14/\epsilon, 0\} \cdot \frac{\epsilon N}{3} + \max\{t^A - t^B, 0\} - \texttt{Agree}^A(\pi^*).$$

2. If $\pi' = \psi$, we use Claim 6.7 to get

$$\begin{aligned} e^{B\to A} &\ge \sum_{i\le i^B} \ell_i^A - \texttt{Agree}_i^A(\emptyset) + \sum_{i>i^B} \ell_i^A - \texttt{Agree}_i^A(\pi^*) \\ &\ge \texttt{Agree}_i^A(\psi) - 100\epsilon N && \text{(Corollary 6.5)} \\ &\ge \mathsf{util}(\psi) + \frac{t^A}{2} - 100\epsilon N. \end{aligned}$$

If $\pi' \ne \psi$, we again Claim 6.7 to get

$$\begin{aligned} e^{B\to A} &\ge \sum_{i\le i^B} \ell_i^A - \texttt{Agree}_i^A(\emptyset) + \sum_{i>i^B} \ell_i^A - \texttt{Agree}_i^A(\pi^*) \\ &\ge \texttt{Agree}^A(\pi') + \texttt{Agree}^A(\psi) - 200\epsilon N && \text{(Corollary 6.5)} \\ &\ge \mathsf{util}(\pi') + \texttt{TagAgree}(\psi) - 200\epsilon N \ge \mathsf{util}(\pi') + t^A - 200\epsilon N. \end{aligned}$$

□

6.6 Lemmas Concerning Tags

In this section, we give useful bounds on the number of messages tagged by Bob. Recall that i_s (resp. i_e) is the last iteration $i \leq 20/\epsilon$ such that $tStart_i = \texttt{false}$ (resp. $tEnd_i = \texttt{false}$). We claim that $i_s < 20/\varepsilon$. Indeed, as if not, then $\frac{20}{\varepsilon} \leq \frac{11}{\varepsilon} + \frac{k^B_{20/\varepsilon-1}}{\varepsilon N} \implies k^B_{i_{\mathsf{RR}}-1} \geq k^B_{20/\varepsilon-1} \geq 9N$, contradicting part two of Lemma 6.11.

Lemma 6.13. *We have that*

1. $t^B \leq \max\{2\alpha, 2N\} + 50\varepsilon N$. *If* $i_e > 14/\varepsilon$, *we also have* $t^B \leq 2\alpha + 50\varepsilon N$.
2. $t^B \leq \frac{2\varepsilon N}{3}(20/\varepsilon - i_s)$.
3. *If* $\alpha_{20/\varepsilon} > 3N + \epsilon N - \frac{1}{3} \cdot k^B_{20/\epsilon}$, *then* $t^B_{20/\varepsilon} \geq 6N - 2k^B_{20/\varepsilon}/3$.

Proof. We prove each part separately.

1. If $i_e > 14/\varepsilon$, then since also $i_e \leq 20/\varepsilon$ (by the definition of i_e), it holds that $t^B = t^B_{i_e} \leq t^B_{i_e-1} + \epsilon N \leq 2\alpha_{i_e-1} + 16\epsilon N$. Now, by Lemma 6.10, $t^B \leq 2\alpha + 50\varepsilon N$.

   If $i_e = 14/\varepsilon$, then no messages are tagged past stageNA-1. The total number of messages transmitted by Bob during stageNA-1 is $(14/\varepsilon) \cdot \varepsilon N \cdot (3/7) = 6N$. However, since $\texttt{StartTag}_1 \geq \frac{28}{3\varepsilon}$, the first $(28/3\varepsilon) \cdot \varepsilon N \cdot (3/7) = 4N$ messages transmitted by Bob were not tagged. Conclude that at most $2N$ messages were tagged by Bob in this case and $t^B \leq 2N$.

2. Bob starts tagging after iteration i_s, and only tags the non-adaptive part, which consists of $20/\varepsilon$ executions of `Chunk`. Every execution of `Chunk` in the non-adaptive part consists of εN rounds, and Bob transmits in at most $2/3$ fraction of these rounds.

3. Observe that $t^B = t^B_{20/\epsilon}$ as Bob does not tag messages after iteration $20/\epsilon$. We divide this part into two cases, based on whether or not $i_s < 14/\epsilon$. We first consider the case $i_s < 14/\epsilon$.

   If $i_s < 14/\epsilon$, then, in the iterations $i \in (i_s : 14/\epsilon]$ Bob tags $\frac{3\epsilon N}{7} - \frac{2}{3} \cdot \Delta_{i-1}$ messages. Similarly, in the iterations $i \in (14/\epsilon : i_e]$, Bob tags $\frac{2\epsilon N}{3} - \frac{2}{3} \cdot \Delta_{i-1}$ messages. As Bob does not tag messages in iterations $i \notin (i_s : i_e]$, the total number of messages tagged by Bob is,

$$\begin{aligned} t^B &= \frac{3\epsilon N}{7} \cdot \left(\frac{14}{\epsilon} - i_s\right) + \frac{2\epsilon N}{3} \cdot \left(i_e - \frac{14}{\epsilon}\right) - \sum_{i \in (i_s : i_e]} \frac{2}{3} \cdot \Delta_{i-1} \\ &= \frac{3\epsilon N}{7} \cdot \left(\frac{14}{\epsilon} - i_s\right) + \frac{2\epsilon N}{3} \cdot \left(i_e - \frac{14}{\epsilon}\right) - \frac{2}{3} \cdot \left(k^B_{i_e-1} - k^B_{i_s-1}\right). \end{aligned} \tag{3}$$

   using the definition of $\Delta_{i-1} = k^B_{i-1} - k^B_{i-2}$. Next, we claim that $i_e = 20/\epsilon$. Suppose not, then, we have $t^B_{i_e} > 2\alpha_{i_e} + 15\epsilon N$, and, by definition of i_s that $i_s + 1 >$

$\texttt{StartTag}_1(k^B_{i_s}) \implies i_s > \frac{28}{3\varepsilon} + \frac{14k^B_{i_s}}{9\epsilon N} - 1$. We get:

$$\begin{aligned}
\alpha_{20/\epsilon} &\le \alpha_{i_e} + \sum_{i\in(i_e:20/\varepsilon]} \left(\ell^B_i - \Delta_i\right) && \text{(by Lemma 6.10)}\\
&\le \alpha_{i_e} + \frac{\epsilon N}{3}\left(\frac{20}{\epsilon} - i_e\right) - \sum_{i\in(i_e:20/\varepsilon]} (k^B_i - k^B_{i-1}) && \text{(Definition of } \Delta)\\
&\le \frac{1}{2}\cdot t^B_{i_e} + \frac{\epsilon N}{3}\left(\frac{20}{\epsilon} - i_e\right) - (k^B_{20/\epsilon} - k^B_{i_e})\\
&\le 3N - \frac{1}{3}\cdot k^B_{i_s} + \frac{3\epsilon N}{14} - \frac{1}{3}\cdot\left(k^B_{i_e-1} - k^B_{i_s-1}\right) - \frac{1}{3}\cdot(k^B_{20/\epsilon} - k^B_{i_e}) && \text{(Equation 3)}\\
&\le 3N + \epsilon N - \frac{1}{3}\cdot k^B_{20/\epsilon},
\end{aligned}$$

a contradiction. But, if $i_e = 20/\epsilon$, then using $i_s \le \texttt{StartTag}_1(k^B_{i_s-1}) \implies i_s \le \frac{28}{3\varepsilon} + \frac{14k^B_{i_s-1}}{9\epsilon N}$, we get using Equation 3 that

$$t^B \ge 6N - \frac{2}{3}\cdot k^B_{i_s-1} - \frac{2}{3}\cdot\left(k^B_{i_e-1} - k^B_{i_s-1}\right) \ge 6N - \frac{2}{3}\cdot k^B_{20/\epsilon},$$

as desired. In the other case where $i_s \ge 14/\epsilon$, we proceed using the same arguments. Instead of Equation 3, we now have

$$t^B = \frac{2\epsilon N}{3}\cdot(i_e - i_s) - \frac{2}{3}\cdot\left(k^B_{i_e-1} - k^B_{i_s-1}\right). \tag{4}$$

We again claim that $i_e = 20/\epsilon$. Suppose not, then, we have $t^B_{i_e} > 2\alpha_{i_e} + 15\epsilon N$, and, by definition of i_s that $i_s + 1 > \texttt{StartTag}_2(k^B_{i_s}) \implies i_s > \frac{11}{\varepsilon} + \frac{k^B_{i_s}}{\epsilon N} - 1$. Similar to the above, we get:

$$\begin{aligned}
\alpha_{20/\epsilon} &\le \frac{1}{2}\cdot t^B_{i_e} + \frac{\epsilon N}{3}\left(\frac{20}{\epsilon} - i_e\right) - (k^B_{20/\epsilon} - k^B_{i_e})\\
&\le 3N - \frac{1}{3}\cdot k^B_{i_s} + \frac{\epsilon N}{3} - \frac{1}{3}\cdot\left(k^B_{i_e-1} - k^B_{i_s-1}\right) - \frac{1}{3}\cdot(k^B_{20/\epsilon} - k^B_{i_e}) && \text{(Equation 4)}\\
&\le 3N + \epsilon N - \frac{1}{3}\cdot k^B_{20/\epsilon},
\end{aligned}$$

a contradiction. But, if $i_e = 20/\epsilon$, then using $i_s \le \texttt{StartTag}_2(k^B_{i_s-1}) \implies i_s \le \frac{11}{\varepsilon} + \frac{k^B_{i_s-1}}{\epsilon N}$, we get

$$t^B \ge 6N - \frac{2}{3}\cdot k^B_{i_s-1} - \frac{2}{3}\cdot\left(k^B_{i_e-1} - k^B_{i_s-1}\right) \ge 6N - \frac{2}{3}\cdot k^B_{20/\epsilon},$$

as desired.

□

Lemma 6.14. *If* $\max_\pi\{\texttt{Agree}^A_{\le 20/\varepsilon}(\pi)\} < \texttt{AliceSure}(k^A_{20/\varepsilon})$ *then* $\max\{i^A, i^B\} \le i_s$.

Proof. Suppose for contradiction that $\max_\pi\{\texttt{Agree}^A_{\le 20/\varepsilon}(\pi)\} < \texttt{AliceSure}(k^A_{20/\varepsilon})$ and $\max\{i^A, i^B\} > i_s$. Then, we have $\min\{14/\epsilon, i_s\} \le \min\{\max\{i^A, i^B\}, 14/\epsilon\}$ and the third part of Lemma 6.11 gives:

$$e^{A\to B}_{\le \min\{14/\varepsilon, i_s\}} \ge \frac{4\epsilon N}{7} \cdot \min\{14/\epsilon, i_s\} - \frac{4}{3}k^B_{\min\{14/\epsilon, i_s\}} - 25\varepsilon N. \tag{5}$$

We consider the following cases:

Case $14/\varepsilon \le i_s$: By adding Claim 6.2 for all $i \in [i_s]$,

$$\begin{aligned}
\left(\frac{4N}{3} + \frac{\varepsilon N}{3} i_s\right)(1-\epsilon) &= \left(\frac{3\varepsilon N}{7} \cdot \frac{14}{\varepsilon} + \frac{\varepsilon N}{3}\left(i_s - \frac{14}{\varepsilon}\right)\right)(1-\varepsilon) \\
&\le \frac{3}{4}e^{A\to B}_{\le 14/\varepsilon} + e^{B\to A}_{\le 14/\varepsilon} + e^{A\to B}_{(14/\varepsilon : i_s]} + \frac{1}{2}e^{B\to A}_{(14/\varepsilon : i_s]} \\
&\le e^{A\to B}_{\le i_s} + e^{B\to A}_{\le i_s} - \frac{1}{4}e^{A\to B}_{\le 14/\varepsilon} \\
&\le e^{A\to B}_{\le i_s} + e^{B\to A}_{\le i_s} - 2N + \frac{1}{3}k^B_{14/\varepsilon} + 10\varepsilon N. \qquad \text{(Equation 5)}
\end{aligned}$$

We next use the definition of i_s to get $i_s > \frac{11}{\epsilon} + \frac{k^B_{i_s}}{\epsilon N} - 1$. This allows us to continue as:

$$\begin{aligned}
e^{A\to B}_{\le i_s} + e^{B\to A}_{\le i_s} - 2N + \frac{1}{3}k^B_{14/\varepsilon} + 10\varepsilon N &\ge \left(\frac{4N}{3} + \frac{\varepsilon N}{3} i_s\right)(1-\epsilon) \\
&\ge 5N + \frac{1}{3} \cdot k^B_{i_s} - 50\epsilon N.
\end{aligned}$$

Rearranging gives $e^{A\to B}_{\le i_s} + e^{B\to A}_{\le i_s} \ge 7N - 60\varepsilon N \ge \texttt{budget}$, a contradiction.

Case $i_s < 14/\varepsilon$: In this case, we use the definition of i_s to get $i_s > \frac{28}{3\epsilon} + \frac{14k^B_{i_s}}{9\epsilon N} - 1$ to get from Equation 5 that

$$\begin{aligned}
e^{A\to B}_{\le i_s} &\ge \frac{4\varepsilon N}{7} i_s - \frac{4}{3}k^B_{i_s} - 25\varepsilon N \\
&\ge \frac{4\varepsilon N}{7} i_s + 8N - \frac{6\epsilon N}{7} - \frac{6\epsilon N}{7} i_s - 25\varepsilon N \\
&= 4N - 30\varepsilon N.
\end{aligned}$$

This means that $e^{B\to A}_{\le 20/\varepsilon} \le \texttt{budget} - e^{A\to B}_{\le i_s} < 3N.$ contradicting, by part two of Lemma 6.12, that $\max_\pi\{\texttt{Agree}^A_{\le 20/\varepsilon}(\pi)\} < \texttt{AliceSure}(k^A_{20/\varepsilon})$.

□

6.7 No RR Rounds

Lemma 6.15. $\mathsf{RR} = \emptyset$.

Proof. Assume for contradiction that $\mathsf{RR} \neq \emptyset$. Then $i_{\mathsf{RR}} > 20/\epsilon \in [24/\epsilon]$. Since Alice receives in iteration i_{RR}, we get

$$\max_{\pi}\{\texttt{Agree}^A_{\leq 20/\varepsilon}(\pi)\} < \texttt{AliceSure}(k^A_{20/\varepsilon}) \tag{6}$$

Equation 6 with part two of Lemma 6.12 implies

$$e^{B\to A} > 3N. \tag{7}$$

We also get that

$$i_{\mathsf{RR}} > \frac{1}{\varepsilon N}\cdot \min\left\{2N, t^A_{i_{\mathsf{RR}}-1} - 2N\right\} + 20/\varepsilon = \frac{1}{\varepsilon N}\cdot \min\left\{2N, t^A - 2N\right\} + 20/\varepsilon, \tag{8}$$

as $i_{\mathsf{RR}} > 20/\epsilon$ and Alice does not change t^A after iteration $20/\epsilon$. Since Bob receives in iteration i_{RR}, we get (using Equation 8)

$$\begin{aligned}\alpha_{i_{\mathsf{RR}}-1} &> 3N - \frac{1}{3}\cdot k^B_{i_{\mathsf{RR}}-1} + (i_{\mathsf{RR}} - 20/\epsilon)\,\epsilon N \\ &> 3N - \frac{1}{3}\cdot k^B_{i_{\mathsf{RR}}-1} + \min\left\{2N, t^A - 2N\right\}.\end{aligned} \tag{9}$$

Additionally, we get that Bob does not execute Line 55 in iteration $20/\varepsilon + 1$. Otherwise, if Bob transmits in iteration $20/\varepsilon + 1$, then k^B and α will not be updated and the condition in Line 52 will never be satisfied. This implies $\alpha_{20/\varepsilon} > 3N - \frac{1}{3}\cdot k^B_{20/\varepsilon} + \epsilon N$. By Lemma 6.13, it holds that

$$t^B \geq 6N - 2k^B_{20/\varepsilon}/3 \geq 6N - 2k^B_{i_{\mathsf{RR}}-1}/3. \tag{10}$$

Define $\delta = e^{A\to B}_{<i_{\mathsf{RR}}} - \alpha_{i_{\mathsf{RR}}-1} \geq 0$ by the first part of Lemma 6.11. By the second part of Lemma 6.11, we get $e^{B\to A}_{\leq i^A} \geq k^B_{i_{\mathsf{RR}}-1} - \delta - 100\epsilon N$. By Equation 7, we get that

$$\begin{aligned}e^{A\to B} + e^{B\to A} &\geq e^{A\to B}_{<i_{\mathsf{RR}}} + e^{B\to A}_{\leq i^A} + e^{B\to A}_{>i^A} \\ &\geq \alpha_{i_{\mathsf{RR}}-1} + \delta + \max\left\{3N, k^B_{i_{\mathsf{RR}}-1} - \delta - 100\varepsilon N + e^{B\to A}_{>i^A}\right\} \\ &\geq \alpha_{i_{\mathsf{RR}}-1} + \max\left\{3N, k^B_{i_{\mathsf{RR}}-1} + e^{B\to A}_{>i^A}\right\} - 100\varepsilon N. \qquad (\text{as } \delta \geq 0)\end{aligned}$$

Next, we use Equation 6 with Lemma 6.14 to claim that $\max\{i^A, i^B\} \leq i_s$ and $e^{B\to A}_{>i^A} \geq e^{B\to A}_{>i_s}$. Adding part one of Claim 6.9 for all $i_s < i \leq 20/\epsilon < i_{\mathsf{RR}}$, we get $e^{B\to A}_{>i_s} \geq \max\{t^B - t^A, 0\}$. Together with Equation 10, we get

$$\begin{aligned}e^{A\to B} + e^{B\to A} &\geq \alpha_{i_{\mathsf{RR}}-1} + \max\left\{3N, k^B_{i_{\mathsf{RR}}-1}, 6N + \frac{1}{3}\cdot k^B_{i_{\mathsf{RR}}-1} - t^A\right\} - 100\epsilon N \\ &\geq \alpha_{i_{\mathsf{RR}}-1} + \max\left\{2N + \frac{1}{3}\cdot k^B_{i_{\mathsf{RR}}-1}, 6N + \frac{1}{3}\cdot k^B_{i_{\mathsf{RR}}} - t^A\right\} - 100\epsilon N \\ &\qquad\qquad (\text{as } \max\{a,b\} \geq \tfrac{2}{3}a + \tfrac{1}{3}b) \\ &\geq \alpha_{i_{\mathsf{RR}}-1} + 4N + \frac{1}{3}\cdot k^B_{i_{\mathsf{RR}}-1} + \max\left\{-2N, 2N - t^A\right\} - 100\epsilon N\end{aligned}$$

$$\geq 7N + \min\{2N, t^A - 2N\} + \max\left\{-2N, 2N - t^A\right\} - 100\epsilon N \quad \text{(Equation 9)}$$
$$\geq 7N - 100\varepsilon N \geq \texttt{budget},$$

a contradiction. □

6.8 Bob's Output is Correct

Henceforth, we sometimes use Lemma 6.15 without stating it explicitly.

Lemma 6.16. *Bob outputs π^*.*

Proof. Let $j \in [0 : 4/\varepsilon]$ be the number of times Bob executes Line 53. We first claim that

$$\alpha \leq 3N - k^B/3 + (j+1)\varepsilon N. \tag{11}$$

For $j < 4/\varepsilon$, this directly follows from the fact the condition in Line 52 in iteration $j+1$ and, as α and k^B are not updated in this iteration, they will no longer get updated, and the condition will never be satisfied.

For $j = 4/\varepsilon$, we use part two of Lemma 6.11 to get

$$\texttt{budget} = 7N - 1000\varepsilon N \geq e^{A\to B} + e^{B\to A} \geq \alpha + k^B - 100\varepsilon N,$$

and Equation 11 follows.

Observe that in all iterations $i \in [24/\varepsilon]$, it holds that $\Delta_i \geq \frac{3}{4}\texttt{Agree}_i^B(\emptyset)$, and therefore $\sum_{i\in[24/\varepsilon]} \texttt{Agree}_i^B(\emptyset) \leq \frac{4}{3}\sum_{i\in[24/\varepsilon]} \Delta_i = \frac{4}{3}k^B$. Equation 11 implies

$$\begin{aligned}
3N - \frac{k^B}{3} + (j+1)\varepsilon N \geq \alpha &= m^B - \max_\pi\{\texttt{Agree}^B(\pi)\} \\
&= \sum_{i\in[24/\varepsilon]} \ell_i^B - \sum_{i\in[24/\varepsilon]} \texttt{Agree}_i^B(\emptyset) - \max_\pi\{\texttt{Agree}^B(\pi)\} \\
&\geq \sum_{i\in[24/\varepsilon]} \ell_i^B - \frac{4}{3}k^B - \max_\pi\{\texttt{Agree}^B(\pi)\} \\
&= (10N + j\varepsilon N) - \frac{4}{3}k^B - \max_\pi\{\texttt{Agree}^B(\pi)\}.
\end{aligned}$$

Rearranging gives $\max_\pi\{\texttt{Agree}^B(\pi)\} \geq 7N - k^B - \epsilon N$. In particular, Bob outputs a transcript π such that $\texttt{Agree}^B(\pi) \geq 7N - k^B - \epsilon N > \texttt{budget} - k^B + 400\varepsilon N$. This π must be π^*, otherwise we derive a contradiction to the last part of Lemma 6.11 (in combination with Lemma 6.15). □

6.9 Alice's Output is Correct

By part three of Lemma 6.12, we conclude that in case Alice outputs in Line 37, then Alice outputs π^*. Since $\max_\pi\{\texttt{Agree}^A_{\leq 20/\varepsilon}(\pi)\} \geq \texttt{AliceSure}(k^A_{20/\varepsilon})$ implies that Alice outputs in

Line 37, we henceforth assume that Alice outputs in Line 39 and $\max_\pi\{\texttt{Agree}^A_{\leq 20/\varepsilon}(\pi)\} < \texttt{AliceSure}(k^A_{20/\varepsilon})$. Due to Lemma 6.14, this means that

$$\max\{i^A, i^B\} \leq i_s. \tag{12}$$

Recall the definitions of ψ and util from subsection 6.5 above. Finally, note that, as t^A does not change after iteration $20/\epsilon$, we have that

$$\sum_{i\in[24/\epsilon]} \ell_i^A = 14N - \min\{2N, t^A - 2N\}. \tag{13}$$

Lemma 6.17. *If $\psi = \pi^*$, then Alice outputs π^*.*

Proof. By Equation 12 and part one of the "furthermore" part of Lemma 6.12, we get that

$$e^{B\to A} \geq \sum_{i\in[24/\epsilon]} \ell_i^A - k^A - \max\left\{i^B - 14/\varepsilon, 0\right\} \cdot \frac{\varepsilon N}{3} + \max\{t^A - t^B, 0\} - \texttt{Agree}^A(\pi^*). \tag{14}$$

We now split the proof into various cases and show that in all cases

$$e^{B\to A} \geq 14N - k^A - \texttt{Agree}^A(\pi^*) - e^{A\to B} - 50\varepsilon N.$$

This, together with $e^{A\to B} + e^{B\to A} \leq \texttt{budget}$ implies that $e^{B\to A} \geq 7N - k^A - \texttt{Agree}^A(\pi^*) + e^{B\to A}$, which rearranges to $\texttt{Agree}^A(\pi^*) \geq 7N - k^A \geq \texttt{AliceSure}(k^A)$, and we are done by part three of Lemma 6.12.

The analysis of the cases uses the following inequality obtained by combining the first part of Lemma 6.13 and the first part of Lemma 6.11

$$t^B \leq \max\{2\alpha, 2N\} + 50\varepsilon N \leq \max\{2e^{A\to B}, 2N\} + 50\varepsilon N. \tag{15}$$

Note that in the case $i_e > 14/\varepsilon$ we get the stronger inequality

$$t^B \leq 2\alpha + 50\varepsilon N \leq 2e^{A\to B} + 50\varepsilon N. \tag{16}$$

- $\boldsymbol{i^B \leq 14/\varepsilon}$ **and** $\boldsymbol{e^{A\to B} \geq N}$**:** Using Equation 15, $t^B \leq \max\{2e^{A\to B}, 2N\} + 50\varepsilon N = 2e^{A\to B} + 50\varepsilon N$. By plugging the last inequality in Equation 14 and using $\sum_{i\in[24/\epsilon]} \ell_i^A = 14N - \min\{2N, t^A - 2N\} \geq 14N - t^A/2$ (Equation 13),

$$\begin{aligned} e^{B\to A} &\geq 14N - t^A/2 - k^A - \texttt{Agree}^A(\pi^*) + \max\{t^A - 2e^{A\to B} - 50\varepsilon N, 0\} \\ &\geq 14N - k^A - \texttt{Agree}^A(\pi^*) - e^{A\to B} - 25\varepsilon N. \qquad (\text{as } \max\{a,b\} \geq (a+b)/2) \end{aligned}$$

- $\boldsymbol{i^B \leq 14/\varepsilon}$ **and** $\boldsymbol{e^{A\to B} < N}$**:** Using Equation 15, $t^B \leq \max\{2e^{A\to B}, 2N\} + 50\varepsilon N = 2N + 50\varepsilon N$. By plugging the last inequality in Equation 14 and using $\sum_{i\in[24/\epsilon]} \ell_i^A = 14N - \min\{2N, t^A - 2N\} \geq 16N - t^A$ (Equation 13),

$$\begin{aligned} e^{B\to A} &\geq 16N - t^A - k^A + \max\{t^A - 2N - 50\varepsilon N, 0\} - \texttt{Agree}^A(\pi^*) \\ &\geq 14N - k^A - \texttt{Agree}^A(\pi^*) - 50\varepsilon N. \end{aligned}$$

- $\boldsymbol{i^B > 14/\varepsilon}$: By Equation 12, $14/\varepsilon < i^B \leq i_s$. Also, as $\alpha_i \geq -14\epsilon N$ for all i (Lemma 6.10), we have $i_s < i_e \implies 14/\epsilon < i_e$. Using Equation 16, $t^B \leq 2e^{A\to B} + 50\varepsilon N \equiv a$. By the second part of Lemma 6.13 it holds that $t^B \leq \frac{2\varepsilon N}{3}(20/\varepsilon - i_s) \leq \frac{2\varepsilon N}{3}(20/\varepsilon - i^B) \equiv b$. Since we have seen that $t^B \leq a, b$ it is also the case that $t^B \leq (a+b)/2 = e^{A\to B} + 25\varepsilon N + \frac{\varepsilon N}{3}(20/\varepsilon - i^B)$. Using the last inequality in step (a) below and $\sum_{i\in[24/\epsilon]} \ell_i^A = 14N - \min\{2N, t^A - 2N\} \geq 16N - t^A$ in Equation 14 (Equation 13), we get

$$\begin{aligned} e^{B\to A} &\geq 16N - t^A - k^A - \left(i^B - 14/\varepsilon\right)\cdot\frac{\varepsilon N}{3} + \max\{t^A - t^B, 0\} - \texttt{Agree}^A(\pi^*) \\ &\geq 16N - k^A - \left(i^B - 14/\varepsilon\right)\cdot\frac{\varepsilon N}{3} - t^B - \texttt{Agree}^A(\pi^*) \\ &\overset{(a)}{\geq} 14N - k^A - e^{A\to B} - \texttt{Agree}^A(\pi^*) - 25\epsilon N. \end{aligned}$$

□

Lemma 6.18. *If $\psi \neq \pi^*$, then Alice outputs π^*.*

Proof. Recall our assumption that Alice outputs in Line 39. To show that Alice outputs π^*, we show that $\mathsf{util}(\pi^*) > \mathsf{util}(\pi')$, for all $\pi' \neq \pi^*$. By part two of the "furthermore" part of Lemma 6.12, it is sufficient to show that $\mathsf{util}(\pi^*) > e^{B\to A} + 200\varepsilon N - t^A/2$. Summing Claim 6.7 for $i > i^B$, we get

$$\mathsf{util}(\pi^*) \geq \sum_{i>i^B} l_i^A - e_{>i^B}^{B\to A} \geq 14N - t^A/2 - \sum_{i\leq i^B} l_i^A - e_{>i^B}^{B\to A}, \tag{17}$$

using $\sum_{i\in[24/\epsilon]} \ell_i^A = 14N - \min\{2N, t^A - 2N\} \geq 14N - t^A/2$ (Equation 13). In case $i^B \leq 14/\varepsilon$, we continue Equation 17 as:

$$\begin{aligned} \mathsf{util}(\pi^*) &\geq 14N - t^A/2 - \frac{3\varepsilon N}{7} i^B - e_{>i^B}^{B\to A} \\ &> 7N - t^A/2 - \frac{3\varepsilon N}{7} i^B + e_{\leq i^B}^{B\to A} + e_{\leq i^B}^{A\to B} + 500\varepsilon N \\ &\geq 7N - t^A/2 \geq e^{B\to A} + 200\varepsilon N - t^A/2. && \text{(Claim 6.2)} \end{aligned}$$

On the other hand, if $i^B > 14/\varepsilon$, we continue Equation 17 as:

$$\begin{aligned} \mathsf{util}(\pi^*) &\geq 8N - t^A/2 - \frac{2\varepsilon N}{3}\left(i^B - 14/\varepsilon\right) - e_{>i^B}^{B\to A} \\ &> N - t^A/2 - \frac{2\varepsilon N}{3}\left(i^B - 14/\varepsilon\right) + e_{\leq i^B}^{A\to B} + e_{\leq i^B}^{B\to A} + 500\varepsilon N \\ &\geq -t^A/2 - \frac{2\varepsilon N}{3}\left(i^B - 14/\varepsilon\right) + e_{\leq i^B}^{A\to B} + e_{\leq i^B}^{B\to A} + e_{>14\varepsilon}^{A\to B} + e_{>14/\varepsilon}^{B\to A} + 500\varepsilon N \\ &&& \text{(by Claim 6.2, } e_{\leq 14\varepsilon}^{A\to B} + e_{\leq 14/\varepsilon}^{B\to A} \geq 6N - 300\varepsilon N) \\ &\geq -t^A/2 - \frac{2\varepsilon N}{3}\left(i^B - 14/\varepsilon\right) + e_{(14/\varepsilon : i^B]}^{B\to A} + 2e_{(14/\varepsilon : i^B]}^{A\to B} + e^{B\to A} + 500\varepsilon N \end{aligned}$$

$$\geq e^{B\to A} + 200\varepsilon N - t^A/2. \qquad \text{(Claim 6.2)}$$

□

References

[AGS16] Shweta Agrawal, Ran Gelles, and Amit Sahai. Adaptive protocols for interactive communication. In *Information Theory (ISIT)*, pages 595–599. IEEE, 2016. 4

[BE17] Mark Braverman and Klim Efremenko. List and unique coding for interactive communication in the presence of adversarial noise. *SIAM Journal on Computing*, 46(1):388–428, 2017. 4, 14

[Ber64] Elwyn R. Berlekamp. *Block Coding with Noiseless Feedback*. PhD thesis, Massachusetts Institute of Technology (MIT), 1964. 4

[BGMO17] Mark Braverman, Ran Gelles, Jieming Mao, and Rafail Ostrovsky. Coding for interactive communication correcting insertions and deletions. *IEEE Transactions on Information Theory*, 63(10):6256–6270, 2017. 4

[BKN14] Zvika Brakerski, Yael Tauman Kalai, and Moni Naor. Fast interactive coding against adversarial noise. *Journal of the ACM (JACM)*, 61(6):35, 2014. 4

[BR11] Mark Braverman and Anup Rao. Towards coding for maximum errors in interactive communication. In *Symposium on Theory of computing (STOC)*, pages 159–166. ACM, 2011. 2, 3, 4

[Bra12] Mark Braverman. Towards deterministic tree code constructions. In *Innovations in Theoretical Computer Science (ITCS)*, pages 161–167. ACM, 2012. 4

[EGH16] Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Maximal noise in interactive communication over erasure channels and channels with feedback. *IEEE Transactions on Information Theory*, 62(8):4575–4588, 2016. 3, 4

[EKS18] Klim Efremenko, Gillat Kol, and Raghuvansh Saxena. Interactive coding over the noisy broadcast channel. In *Symposium on Theory of Computing (STOC)*, pages 507–520. ACM, 2018. 4

[FGOS15] Matthew Franklin, Ran Gelles, Rafail Ostrovsky, and Leonard J. Schulman. Optimal coding for streaming authentication and interactive communication. *IEEE Transactions on Information Theory*, 61(1):133–145, 2015. 4

[Gel17] Ran Gelles. Coding for interactive communication: A survey. *Foundations and Trends® in Theoretical Computer Science*, 13(1–2):1–157, 2017. 4

[GH14] Mohsen Ghaffari and Bernhard Haeupler. Optimal Error Rates for Interactive Coding II: Efficiency and List Decoding. In *Foundations of Computer Science (FOCS)*, FOCS, pages 394–403, 2014. 4

[GH17] Ran Gelles and Bernhard Haeupler. Capacity of interactive communication over erasure channels and channels with feedback. *SIAM Journal on Computing*, 46(4):1449–1472, 2017. 4

[GHK+18] Ran Gelles, Bernhard Haeupler, Gillat Kol, Noga Ron-Zewi, and Avi Wigderson. Explicit capacity approaching coding for interactive communication. *IEEE Transactions on Information Theory*, 64(10):6546–6560, 2018. 4

[GHS14] Mohsen Ghaffari, Bernhard Haeupler, and Madhu Sudan. Optimal error rates for interactive coding i: Adaptivity and other settings. In *Symposium on Theory of computing (STOC)*, pages 794–803, 2014. 2, 3, 4, 5, 8, 11

[GI03] Venkatesan Guruswami and Piotr Indyk. Linear time encodable and list decodable codes. In *Symposium on Theory of computing (STOC)*, pages 126–135. ACM, 2003. 13

[GMS11] Ran Gelles, Ankur Moitra, and Amit Sahai. Efficient and explicit coding for interactive communication. In *Foundations of Computer Science (FOCS)*, pages 768–777. IEEE, 2011. 4

[GMS14] Ran Gelles, Ankur Moitra, and Amit Sahai. Efficient coding for interactive communication. *IEEE Transactions on Information Theory*, 60(3):1899–1913, 2014. 4

[Gur06] Venkatesan Guruswami. Algorithmic results in list decoding. *Foundations and Trends in Theoretical Computer Science*, 2(2), 2006. 13

[Hae14] Bernhard Haeupler. Interactive channel capacity revisited. In *Foundations of Computer Science (FOCS)*, pages 226–235. IEEE, 2014. 4

[HKV15] Bernhard Haeupler, Pritish Kamath, and Ameya Velingker. Communication with partial noiseless feedback. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, 2015. 4

[HSV18] Bernhard Haeupler, Amirbehshad Shahrasbi, and Ellen Vitercik. Synchronization strings: Channel simulations and interactive coding for insertions and deletions. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 75:1–75:14, 2018. 4

[HV17] Bernhard Haeupler and Ameya Velingker. Bridging the capacity gap between interactive and one-way communication. In *Symposium on Discrete Algorithms (SODA)*, pages 2123–2142, 2017. 4

[KR13] Gillat Kol and Ran Raz. Interactive channel capacity. In *Symposium on Theory of computing (STOC)*, pages 715–724, 2013. 4

[Pan13] Denis Pankratov. On the power of feedback in interactive channels. Manuscript, 2013. 4

[Sch92] Leonard J Schulman. Communication on noisy channels: A coding theorem for computation. In *Foundations of Computer Science (FOCS)*, pages 724–733. IEEE, 1992. 4

[Sch93] Leonard J Schulman. Deterministic coding for interactive communication. In *Symposium on Theory of computing (STOC)*, pages 747–756. ACM, 1993. 4

[Sch96] Leonard J. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996. 2, 4

[SW17] Alexander A. Sherstov and Pei Wu. Optimal interactive coding for insertions, deletions, and substitutions. In *Foundations of Computer Science (FOCS)*, pages 240–251, 2017. 4

[WQC17] Gang Wang, Yanyuan Qin, and Chengjuan Chang. Communication with partial noisy feedback. In *IEEE Symposium on Computers and Communications (ISCC)*, pages 602–607, 2017. 4

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il