# Repairing Reed-Solomon Codes With Two Erasures

Hoang Dau*, Iwan Duursma†, Han Mao Kiah‡, and Olgica Milenkovic§
*†§University of Illinois at Urbana-Champaign, ‡Nanyang Technological University
Emails: {*hoangdau, †duursma, §milenkov}@illinois.edu, ‡hmkiah@ntu.edu.sg

***Abstract*—Despite their exceptional error-correcting properties, Reed-Solomon (RS) codes have been overlooked in distributed storage applications due to the common belief that they have poor repair bandwidth: A naive repair approach would require the whole file to be reconstructed in order to recover a single erased codeword symbol. In a recent work, Guruswami and Wootters (STOC'16) proposed a single-erasure repair method for RS codes that achieves the optimal repair bandwidth amongst all linear encoding schemes. We extend their trace collection technique to cope with two erasures.**

## I. Introduction

### A. Background

The *repair bandwidth* is an important performance metric of erasure codes in the context of distributed storage [1]. In such a system, for a chosen field $F$, a data vector in $F^k$ is mapped to a codeword vector in $F^n$, whose entries are stored at different storage nodes. When a node fails, the symbol stored at that node is erased (lost). A replacement node (RN) has to recover the content stored at the failed node by downloading information from the other nodes. The repair bandwidth is the total amount of information that the RN has to download in order to successfully complete the repair process.

Reed-Solomon (RS) codes [2], which have been extensively studied in theory [3] and widely used in practice, were believed to have prohibitively high repair bandwidth. In a naive repair scheme, recovering the content stored at a *single* failed node would require downloading the *whole* file, i.e., $k$ symbols over $F$. The poor performance in repairing failed nodes of RS codes motivated the introduction of repair-efficient codes such as regenerating codes [1] and locally repairable codes [4], [5], [6].

Guruswami and Wootters [7] recently proposed a bandwidth-optimal linear repair method based on RS codes. The key idea behind their method is to recover a single erased symbol by collecting a sufficiently large number of its (field) traces, each of which can be constructed from a number of traces of other symbols. As all traces belong to a subfield $B$ of $F$ and traces from the same symbol are related, the total repair bandwidth can be significantly reduced. The repair scheme obtained by Guruswami and Wootters [7], however, only applies to the case of one erasure, or in other words, one failed node.

### B. Our Contribution

We propose an extension of the Guruswami-Wootters repair scheme that can ensure recovery from two erasures. We provide two *distributed* schemes for Reed-Solomon codes, both of which use the same repair bandwidth *per erasure* as in the case of a single erasure. In these repair schemes, the two RNs first download repair data from all available nodes (Download Phase). They subsequently collaborate to exchange the data in order to complete the repair process at each node (Collaboration Phase). The first scheme has a *collaboration depth one*, that is, in the Collaboration Phase, the two RNs send out repair data to each other simultaneously in one round. This scheme works whenever the field extension degree $t$ is divisible by the characteristic of the field $F$. An example illustrating the first scheme is given in Fig. 1. The second scheme has a *collaboration depth two*, that is, in the Collaboration Phase, one RN receives the repair data from the other RN, completes its repair process, and then sends out its repair data to the other node. This scheme applies to all field extension degrees.

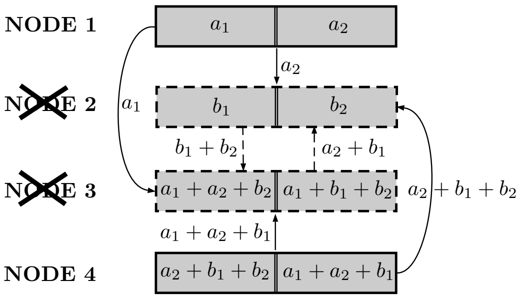

Fig. 1: A toy example illustrating the repair procedure for *two* failed nodes in a four-node storage system based on a $[4, 2]$ Reed-Solomon code over $\mathbb{F}_4$. The stored file is $\big((a_1, a_2), (b_1, b_2)\big) \in \mathbb{F}_4^2$, where $a_1$, $a_2$, $b_1$, and $b_2$ are bits in $\mathbb{F}_2$. Suppose that Node 2 and Node 3 fail simultaneously. In the Download Phase, each replacement node first downloads two bits (along the solid arrows) from the two available nodes, namely Node 1 and Node 4. In the Collaboration Phase, the replacement nodes communicate with each other to complete their own repair processes by exchanging two extra bits (along the dashed arrows), computed based on the previously downloaded bits.

### C. Organization

The paper is organized as follows. We first provide relevant definitions and introduce the terminology used throughout the paper. We then proceed to discuss the Guruswami-Wootters repair scheme for RS codes in the presence of a single erasure in Section II. Our main results – repair schemes for RS codes in the presence of two erasures – are presented in Section III. For a thorough literature review on the related works on *cooperative regenerating codes* and the motivation for repairing multiple erasures, the interested reader is referred to our companion paper [8].

## II. Repairing One Erasure in Reed-Solomon Codes

We start by introducing relevant definitions and the notation used in all subsequent derivations, and then proceed to review the approach proposed by Guruswami and Wootters [7] for repairing a single erasure/node failure in RS codes.

978-1-5090-4096-4/17/$31.00 ©2017 IEEE

### A. Definitions and Notations

Let $[n]$ denote the set $\{1,2,\ldots,n\}$. Let $B=\text{GF}(p^m)$ be the finite field of $p^m$ elements, for some prime $p$ and $m\geq 1$. Let $F=\text{GF}(p^{mt})$ be a field extension of $B$, where $t\geq 1$. We often refer to the elements of $F$ as *symbols* and the elements of $B$ as *sub-symbols*. We can also treat $F$ as a vector space of dimension $t$ over $B$, i.e. $F\cong B^t$, and hence each symbol in $F$ may be represented as a vector of length $t$ over $B$. A linear $[n,k]$ code $\mathcal{C}$ over $F$ is a subspace of $F^n$ of dimension $k$. Each element of a code is referred to as a codeword. The dual of a code $\mathcal{C}$, denoted $\mathcal{C}^{\perp}$, is the orthogonal complement of $\mathcal{C}$.

**Definition 1.** Let $F[x]$ denote the ring of polynomials over $F$. The Reed-Solomon code $\text{RS}(A,k)\subseteq F^n$ of dimension $k$ over a finite field $F$ with evaluation points $A=\{\alpha_1,\alpha_2,\ldots,\alpha_n\}\subseteq F$ is defined as:

$$\text{RS}(A,k)=\Big\{\big(f(\alpha_1),\ldots,f(\alpha_n)\big)\colon f\in F[x],\deg(f)<k\Big\}.$$

A *generalized* Reed-Solomon code, $\text{GRS}(A,k,\boldsymbol{\lambda})$, where $\boldsymbol{\lambda}=(\lambda_1,\ldots,\lambda_n)\in F^n$, is defined similarly to a Reed-Solomon code, except that the codeword corresponding to a polynomial $f$ is now defined as $\big(\lambda_1 f(\alpha_1),\ldots,\lambda_n f(\alpha_n)\big)$, $\lambda_i\neq 0$ for all $i\in[n]$. It is well known that the dual of an RS code $\text{RS}(A,k)$, for any $n\leq|F|$, is a generalized RS code $\text{GRS}(A,n-k,\boldsymbol{\lambda})$, for some multiplier vector $\boldsymbol{\lambda}$ (see [3, Chp. 10]). Whenever clear from the context, we use $f(x)$ to denote a polynomial of degree at most $k-1$, which corresponds to a codeword of the RS code $\mathcal{C}=\text{RS}(A,k)$, and $p(x)$ to denote a polynomial of degree at most $n-k-1$, which corresponds to a dual codeword in $\mathcal{C}^{\perp}$. Since $\sum_{\alpha\in A}p(\alpha)(\lambda_\alpha f(\alpha))=0$, we refer to such a polynomial $p(x)$ as a check polynomial for $\mathcal{C}$. Note that when $n=|F|$, we have $\lambda_\alpha=1$ for all $\alpha\in F$. In general, as recovering $f(\alpha)$ is equivalent to recovering $\lambda_\alpha f(\alpha)$, to simplify the notation, we often omit the factor $\lambda_\alpha$ in the equation above.

### B. The Guruswami-Wootters Repair Scheme for One Erasure

Suppose that the polynomial $f(x)\in F[x]$ corresponds to a codeword in the RS code $\mathcal{C}=\text{RS}(A,k)$ and that $f(\alpha^*)$ is the erased symbol, where $\alpha^*\in A$ is an evaluation point of the code.

Given that $F$ is a field extension of $B$ of degree $t$, i.e. $F=\text{GF}(p^{mt})$ and $B=\text{GF}(p^m)$, for some prime $p$, one may define the field trace of any symbol $\alpha\in F$ as $\mathsf{Tr}_{F/B}(\alpha)=\sum_{i=0}^{t-1}\alpha^{|B|^i}$, which is always a sub-symbol in $B$. We often omit the subscript $F/B$ for succinctness. The key points in the repair scheme proposed by Guruswami and Wootters [7] can be summarized as follows. Firstly, each symbol in $F$ can be recovered from its $t$ independent traces. More precisely, given a basis $u_1,u_2,\ldots,u_t$ of $F$ over $B$, any $\alpha\in F$ can be uniquely determined given the values of $\mathsf{Tr}(u_i\,\alpha)$ for $i\in[t]$, i.e. $\alpha=\sum_{i=1}^{t}\mathsf{Tr}(u_i\alpha)u_i^{\perp}$, where $\{u_i^{\perp}\}_{i=1}^t$ is the dual (trace-orthogonal) basis of $\{u_i\}_{i=1}^t$ (see, for instance [9, Ch. 2, Def. 2.30]). Secondly, when $n-k\geq|B|^{t-1}$, the trace function also provide checks that generate repair equations whose coefficients are linearly dependent over $B$, which keeps the repair cost low.

Note that the checks of $\mathcal{C}$ are precisely those polynomials $p(x)\in F[x]$ with $\deg(p)<n-k$. It turns out that for $n-k\geq|B|^{t-1}$, we can define checks that take part in the repair process via the trace function described above. For each $u\in F$ and $\alpha\in F$, we define the polynomial

$$p_{u,\alpha}(x)=\mathsf{Tr}\big(u(x-\alpha)\big)/(x-\alpha). \qquad (1)$$

By the definition of a trace function, the following lemma follows in a straightforward manner.

**Lemma 1** ([7]). *The polynomial $p_{u,\alpha}(x)$ defined in* (1) *satisfies the following properties.*
*(a)* $\deg(p_{u,\alpha})=|B|^{t-1}-1$*;* *(b)* $p_{u,\alpha}(\alpha)=u$*.*

By Lemma 1 (a), $\deg(p_{u,\alpha})=|B|^{t-1}-1<n-k$. Therefore, the polynomial $p_{u,\alpha}(x)$ corresponds to a codeword of $\mathcal{C}^{\perp}$ and is a check for $\mathcal{C}$. Now let $U=\{u_1,\ldots,u_t\}$ be a basis of $F$ over $B$, and set

$$p_i(x)\triangleq p_{u_i,\alpha^*}(x)=\mathsf{Tr}\big(u_i(x-\alpha^*)\big)/(x-\alpha^*),\quad i\in[t].$$

These $t$ polynomials correspond to $t$ codewords of $\mathcal{C}^{\perp}$. Therefore, we obtain $t$ equations of the form

$$p_i(\alpha^*)f(\alpha^*)=-\sum_{\alpha\in A\setminus\{\alpha^*\}}p_i(\alpha)f(\alpha),\quad i\in[t]. \qquad (2)$$

A key step in the Guruswami-Wootters repair scheme is to apply the trace function to both sides of (2) to obtain $t$ different *repair equations*

$$\mathsf{Tr}\big(p_i(\alpha^*)f(\alpha^*)\big)=-\sum_{\alpha\in A\setminus\{\alpha^*\}}\mathsf{Tr}\big(p_i(\alpha)f(\alpha)\big),\ i\in[t]. \qquad (3)$$

According to Lemma 1 (b), $p_i(\alpha^*)=u_i$, for $i=1,\ldots,t$. Moreover, by the linearity of the trace function, we can rewrite (3) as follows. For $i=1,\ldots,t$,

$$\mathsf{Tr}\big(u_if(\alpha^*)\big)=-\sum_{\alpha\in A\setminus\{\alpha^*\}}\mathsf{Tr}\big(u_i(\alpha-\alpha^*)\big)\times\mathsf{Tr}\Big(\frac{f(\alpha)}{\alpha-\alpha^*}\Big). \qquad (4)$$

The right-hand side sums of the equations (4) can be computed by downloading the repair trace $\mathsf{Tr}\Big(\frac{f(\alpha)}{\alpha-\alpha^*}\Big)$ from the node storing $f(\alpha)$, for each $\alpha\in A\setminus\{\alpha^*\}$. As a consequence, the $t$ independent traces $\mathsf{Tr}\big(u_if(\alpha^*)\big)$, $i=1,\ldots,t$, of $f(\alpha^*)$ can be determined by downloading one sub-symbol from each of the $n-1$ available nodes. The erased symbol $f(\alpha^*)$ can subsequently be recovered from its $t$ independent traces. By [10, Cor. 1], this scheme is bandwidth-optimal when $n=|F|$ and $k=n(1-1/|B|)$.

## III. Repairing Two Erasures in Reed-Solomon Codes

We consider the same setting as in Section II-B, i.e. $n-k\geq|B|^{t-1}$, where $B=\text{GF}(p^m)$ and $F=\text{GF}(p^{mt})$, and assume that $\mathcal{C}$ is an RS code $\text{RS}(A,k)$ over $F$. However, we now suppose that two codeword symbols, say $f(\alpha^*)$ and $f(\overline{\alpha})$, are erased. Two repair schemes are proposed, both of which use the same bandwidth per erasure as in the case of a single erasure in [7].

### A. General Idea

We first discuss the challenges associated with repairing two erased symbols and then proceed to describe our strategy for dealing with this repair scenario. A check $p(x)$ is said to *involve* a codeword symbol $f(\alpha)$ if $p(\alpha)\neq 0$. When only one symbol $f(\alpha^*)$ is erased, every check $p(x)$ that involves $f(\alpha^*)$ can be used to generate a repair equation as follows.

$$\mathsf{Tr}\big(p(\alpha^*)f(\alpha^*)\big)=-\sum_{\alpha\in A\setminus\{\alpha^*\}}\mathsf{Tr}\big(p(\alpha)f(\alpha)\big). \qquad (5)$$

However, when two symbols $f(\alpha^*)$ and $f(\overline{\alpha})$ are erased, in order to, say, recover $f(\alpha^*)$, we no longer have the freedom to use every possible check that involves $f(\alpha^*)$. Indeed, those checks that involve both $f(\alpha^*)$ and $f(\overline{\alpha})$ cannot be used in a straightforward manner for repair, because we cannot simply compute the right-hand side sum of (5) without retrieving some information from $f(\overline{\alpha})$.

The gist of our approach is to first generate those checks that only involve one codeword symbol, $f(\alpha^*)$ or $f(\overline{\alpha})$, but not both. We show that there exist $2(t-1)$ such checks, which are used in the Download Phase. Each RN uses $t-1$ checks and downloads the corresponding $n-2$ sub-symbols from each available node. Apart from the $t-1$ checks that involve $f(\alpha^*)$ but not $f(\overline{\alpha})$, and the $t-1$ checks that involve $f(\overline{\alpha})$ but not $f(\alpha^*)$, we also introduce *two* additional checks that involve both $f(\alpha^*)$ and $f(\overline{\alpha})$, which are useful in the Collaboration Phase. It is not immediately clear how these last two checks can be used at all. However, we prove that when the extension degree $t$ is divisible by the characteristic of the field $F$, each erased symbol can be recovered at each RN using the aforementioned $t$ checks, at the cost of downloading in total $n-1$ sub-symbols from $n-2$ surviving nodes and from the other RN. In the first repair scheme, the two RNs exchange their repair data simultaneously (parallel repair), while in the second scheme, one node *waits* to receive the data from the other node before sending out its own repair data (sequential repair). By allowing one node to wait in the Collaboration Phase, we obtain a repair scheme that works for every field extension degree.

To identify check equations that involve one codeword symbol $f(\alpha)$ but not the other symbol $f(\beta)$, we first introduce a special polynomial $Q_{\alpha,\beta}(z)$, defined as follows:

$$Q_{\alpha,\beta}(z) = \mathsf{Tr}\big(z(\beta-\alpha)\big), \quad \alpha \neq \beta. \tag{6}$$

Let $K_{\alpha,\beta}$ denote the root space of $Q_{\alpha,\beta}(z)$. Then

$$K_{\alpha,\beta} = \{z^* \in F \colon \mathsf{Tr}(z^*\alpha) = \mathsf{Tr}(z^*\beta)\}. \tag{7}$$

**Lemma 2.** *The following statements hold for every $\alpha$ and $\beta$ in $F$, $\alpha \neq \beta$.*

(a) *$K_{\alpha,\beta} \equiv K_{\beta,\alpha}$. In other words, the polynomial $Q_{\alpha,\beta}$ and the polynomial $Q_{\beta,\alpha}$ have the same root spaces.*

(b) $\dim_B(K_{\alpha,\beta}) = \dim_B(K_{\beta,\alpha}) = t-1$.

*Proof.* From (7), due to symmetry, $K_{\alpha,\beta} \equiv K_{\beta,\alpha}$. As the trace function is a linear mapping from $F$ to $B$, its kernel $K = \{\kappa \in F \colon \mathsf{Tr}(\kappa) = 0\}$ is a subspace of dimension $t-1$ over $B$ (see [9, Thm. 2.23]). Therefore, the root space of $Q_{\alpha,\beta}(z)$ is $K_{\alpha,\beta} = \frac{1}{\beta-\alpha}K$, which is also a subspace of dimension $t-1$ over $B$. ■

We then use a root $z^*$ of the polynomial $Q_{\alpha,\beta}(z)$ to define a check equation according to (1).

$$p_{z^*,\alpha}(x) = \mathsf{Tr}\big(z^*(x-\alpha)\big)/(x-\alpha).$$

The following properties of $p_{z^*,\alpha}(x)$ will be used in our subsequent proofs.

**Lemma 3.** *Suppose that $\alpha$ and $\beta$ are two distinct elements of $F$, and $z^*$ is a root of $Q_{\alpha,\beta}(z)$ or $Q_{\beta,\alpha}(z)$ in $F$, i.e. $z^* \in K_{\alpha,\beta}$. Then the following claim holds.*

(a) $p_{z^*,\alpha}(\beta) = 0$.

*Moreover, if the extension degree $t$ is divisible by char$(F)$ then*

(b) *$p_{u,\alpha}(\beta)$ is a root of $Q_{\alpha,\beta}(z)$ and $Q_{\beta,\alpha}(z)$, for every $u \in F$.*

*Proof.* Note that according to Lemma 2 (a), the root spaces of $Q_{\alpha,\beta}(z)$ and $Q_{\beta,\alpha}(z)$ are the same. The first claim is clear based on the definitions of $Q_{\alpha,\beta}(z)$ and $p_{z^*,\alpha}(x)$. For the second claim, it is sufficient to show that $p_{u,\alpha}(\beta)$ is a root of $Q_{\alpha,\beta}(z)$.

For simplicity, let $\Delta \triangleq \beta-\alpha$ and $b \triangleq \mathsf{Tr}\big(u(\beta-\alpha)\big) \in B$. By definition of $p_{u,\alpha}(x)$, we have $p_{u,\alpha}(\beta) = \mathsf{Tr}\big(u(\beta-\alpha)\big)/(\beta-\alpha) = b/\Delta$. By definition of $Q_{\alpha,\beta}(z)$, we also have $Q_{\alpha,\beta}(z) = \mathsf{Tr}\big(z(\beta-\alpha)\big) = \mathsf{Tr}(z\Delta)$. Therefore, $Q_{\alpha,\beta}\big(p_{u,\alpha}(\beta)\big) = \mathsf{Tr}\Big((b/\Delta)\Delta\Big) = \mathsf{Tr}(b) = 0$, because for $b \in B$, we always have $\mathsf{Tr}(b) = tb = 0$, whenever $t$ is divisible by the char$(F)$. Hence, $p_{u,\alpha}(\beta)$ is a root of $Q_{\alpha,\beta}(z)$. ■

The following lemma restates what is shown in Section II-B.

**Lemma 4.** *For $\alpha \neq \alpha^*$ and $u \in F$,*

$$\mathsf{Tr}\big(p_{u,\alpha^*}(\alpha)f(\alpha)\big) = \mathsf{Tr}\big(u(\alpha-\alpha^*)\big)\mathsf{Tr}\Big(\frac{f(\alpha)}{\alpha-\alpha^*}\Big). \tag{8}$$

*Hence, $\mathsf{Tr}\big(p_{u,\alpha^*}(\alpha)f(\alpha)\big)$ can be computed by downloading the repair trace $\mathsf{Tr}\Big(\frac{f(\alpha)}{\alpha-\alpha^*}\Big)$ from the node storing $f(\alpha)$.*

### B. A Depth-One Repair Scheme for Two Erasures

The scheme comprises of two phases, the Download Phase, where each RN contacts and downloads data from the other $n-2$ available nodes, and the Collaboration Phase, where the two RNs exchange the data, based on what they receive earlier in the Download Phase. The main task is to design the data to be exchanged during the two phases. This task can be completed via a selection of proper check polynomials to be used by each RN. We discuss the generation of these polynomials below.

Let $K_{\alpha^*,\overline{\alpha}}$ be the root space of the polynomial $Q_{\alpha^*,\overline{\alpha}}(z)$. By Lemma 2 (b), $\dim_B(K_{\alpha^*,\overline{\alpha}}) = t-1$. Let $U = \{u_1, u_2, \ldots, u_{t-1}\} \subseteq F$ and $V = \{v_1, v_2, \ldots, v_{t-1}\} \subseteq F$ be two arbitrary bases of $K_{\alpha^*,\overline{\alpha}}$ over $B$. We extend $U$ and $V$ to obtain the two bases $U' = \{u_1, \ldots, u_t\}$ and $V' = \{v_1, \ldots, v_t\}$ of $F$ over $B$, respectively. For $i \in [t]$, we set

$$p_i(x) \triangleq p_{u_i,\alpha^*}(x) = \mathsf{Tr}\big(u_i(x-\alpha^*)\big)/(x-\alpha^*), \tag{9}$$

$$q_i(x) \triangleq p_{v_i,\overline{\alpha}}(x) = \mathsf{Tr}\big(v_i(x-\overline{\alpha})\big)(x-\overline{\alpha}). \tag{10}$$

**Download Phase.** In this phase, each RN contacts $n-2$ available nodes to download repair data. To determine what to download, the RN for $f(\alpha^*)$ uses the first $t-1$ checks $p_1, \ldots, p_{t-1}$ to construct the following $t-1$ repair equations.

$$\mathsf{Tr}\big(p_i(\alpha^*)f(\alpha^*)\big) = -\sum_{\alpha \in A \setminus \{\alpha^*\}} \mathsf{Tr}\big(p_i(\alpha)f(\alpha)\big),\ i \in [t-1]. \tag{11}$$

Similarly, the RN for $f(\overline{\alpha})$ creates the following repair equations.

$$\mathsf{Tr}\big(q_i(\overline{\alpha})f(\overline{\alpha})\big) = -\sum_{\alpha \in A \setminus \{\overline{\alpha}\}} \mathsf{Tr}\big(q_i(\alpha)f(\alpha)\big),\ i \in [t-1]. \tag{12}$$

By Lemma 3 (a), we have $p_i(\overline{\alpha}) = 0$ and $q_i(\alpha^*) = 0$ for all $i = 1, \ldots, t-1$. Therefore, the right-hand sides of (11) and (12) do not involve $f(\alpha^*)$ and $f(\overline{\alpha})$. As a result, each RN can recover $t-1$ independent traces of the corresponding erased symbol by downloading $n-2$ sub-symbols (traces) from the available nodes. Corollary 1, which follows directly from Lemma 4, formally states this fact.

**Corollary 1.** *In the Download Phase, the replacement node for $f(\alpha^*)$ can recover $t-1$ independent traces, namely $\mathsf{Tr}\big(p_1(\alpha^*)f(\alpha^*)\big), \ldots, \mathsf{Tr}\big(p_{t-1}(\alpha^*)f(\alpha^*)\big)$, by downloading $n-2$ repair traces, i.e. $\mathsf{Tr}\Big(\frac{f(\alpha)}{\alpha-\alpha^*}\Big)$ from the available node storing $f(\alpha)$, for all $\alpha \in A \setminus \{\alpha^*, \overline{\alpha}\}$. A similar statement holds for the replacement node for $f(\overline{\alpha})$, where the checks are $q_i$ and the repair traces are $\mathsf{Tr}\Big(\frac{f(\alpha)}{\alpha-\overline{\alpha}}\Big)$.*

**Collaboration Phase.** As one more independent trace of each erased symbol is needed for a complete recovery, the two RNs create two additional repair equations for $f(\alpha^*)$, $f(\overline{\alpha})$, respectively.

$$\mathsf{Tr}\big(p_t(\alpha^*)f(\alpha^*)\big) + \mathsf{Tr}\big(p_t(\overline{\alpha})f(\overline{\alpha})\big) = -\sum_{\alpha \in A \setminus \{\alpha^*, \overline{\alpha}\}} \mathsf{Tr}\big(p_t(\alpha)f(\alpha)\big). \quad (13)$$

$$\mathsf{Tr}\big(q_t(\overline{\alpha})f(\overline{\alpha})\big) + \mathsf{Tr}\big(q_t(\alpha^*)f(\alpha^*)\big) = -\sum_{\alpha \in A \setminus \{\alpha^*, \overline{\alpha}\}} \mathsf{Tr}\big(q_t(\alpha)f(\alpha)\big). \quad (14)$$

It is clear that from the repair traces $\mathsf{Tr}\Big(\frac{f(\alpha)}{\alpha-\alpha^*}\Big)$, $\alpha \in A \setminus \{\alpha^*, \overline{\alpha}\}$, retrieved in the Download Phase, the RHS of (13) can be determined. However, to determine the desired trace $\mathsf{Tr}\big(p_t(\alpha^*)f(\alpha^*)\big)$, the RN for $f(\alpha^*)$ needs to know the missing trace $\mathsf{Tr}\big(p_t(\overline{\alpha})f(\overline{\alpha})\big)$, which would have been downloaded from the node storing $f(\overline{\alpha})$ if it had not failed. The following lemma states that for certain field extension degrees, this missing piece of information can be created by the RN for $f(\overline{\alpha})$ based on what it obtains in the Download Phase. It can then send this trace to the RN for $f(\alpha^*)$ to help complete the recovery of that symbol. A similar scenario also holds for $f(\overline{\alpha})$.

**Lemma 5.** *If the field expansion degree $t$ is divisible by the characteristic of the fields $F$ and $B$, then $p_t(\overline{\alpha})$ is dependent (over $B$) on the set $\{q_i(\overline{\alpha})\colon i \in [t-1]\}$. Also, in this case, $q_t(\alpha^*)$ is dependent (over $B$) on the set $\{p_i(\alpha^*)\colon i \in [t-1]\}$.*

*Proof.* Because of symmetry, it suffices to just prove the first statement of the lemma. By Lemma 1 (b), we have $q_i(\overline{\alpha}) = v_i$, for every $i \in [t-1]$. Therefore, $\{q_i(\overline{\alpha})\colon i \in [t-1]\} = \{v_1, \ldots, v_{t-1}\} = V$, which is a basis of the root space $K_{\alpha^*,\overline{\alpha}}$ of the polynomial $Q_{\alpha^*,\overline{\alpha}}(z)$. Therefore, in order to show that $p_t(\overline{\alpha})$ is dependent on $V$, it is sufficient to prove that $p_t(\overline{\alpha})$ is a root of $Q_{\alpha^*,\overline{\alpha}}(z)$. But this follows immediately from Lemma 3 (b), because $p_t(x)$ equals $p_{u_t,\alpha^*}(x)$ by its definition in Step 4. ■

From the linearity of the trace function, we arrive at the following corollary of Lemma 5.

**Corollary 2.** *If the field expansion degree $t$ is divisible by the characteristic of the fields $F$ and $B$, then the trace $\mathsf{Tr}\big(p_t(\overline{\alpha})f(\overline{\alpha})\big)$ can be written as a linear combination (over $B$) of the traces in $\Big\{\mathsf{Tr}\big(q_i(\overline{\alpha})f(\overline{\alpha})\big)\colon i \in [t-1]\Big\}$. Also, the trace $\mathsf{Tr}\big(q_t(\alpha^*)f(\alpha^*)\big)$ can be written as a linear combination (over $B$) of the traces in $\Big\{\mathsf{Tr}\big(p_i(\alpha^*)f(\alpha^*)\big)\colon i \in [t-1]\Big\}$. Moreover, the coefficients of these combinations do not depend on $f$.*

Note that by Lemma 4, the traces $\mathsf{Tr}\big(p_t(\overline{\alpha})f(\overline{\alpha})\big)$ and $\mathsf{Tr}\big(q_t(\alpha^*)f(\alpha^*)\big)$ can be determined based on the traces $\mathsf{Tr}\Big(\frac{f(\overline{\alpha})}{\overline{\alpha}-\alpha^*}\Big)$ and $\mathsf{Tr}\Big(\frac{f(\alpha^*)}{\alpha^*-\overline{\alpha}}\Big)$, respectively. Therefore, in the Collaboration Phase, the RNs can send their repair data to each other, which matches precisely what they would have sent if they had not failed. The graphical illustration of the two phases of this scheme is depicted in Fig. 2. We refer to this as a *depth-one* collaborative repair scheme because in the Collaboration Phase, two RNs exchange repair data in one round and do not have to wait for each other.

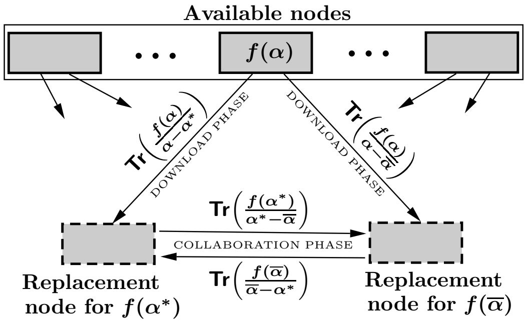
Available nodes
$f(\alpha)$
$\mathsf{Tr}\Big(\frac{f(\alpha)}{\alpha-\alpha^*}\Big)$
DOWNLOAD PHASE
DOWNLOAD PHASE
$\mathsf{Tr}\Big(\frac{f(\alpha)}{\alpha-\overline{\alpha}}\Big)$
$\mathsf{Tr}\Big(\frac{f(\alpha^*)}{\alpha^*-\overline{\alpha}}\Big)$
COLLABORATION PHASE
$\mathsf{Tr}\Big(\frac{f(\overline{\alpha})}{\overline{\alpha}-\alpha^*}\Big)$
Replacement node for $f(\alpha^*)$
Replacement node for $f(\overline{\alpha})$

Fig. 2: Illustration of the depth-one collaborative repair scheme of two erasures for Reed-Solomon codes.

**Lemma 6.** *In the Collaboration Phase, the replacement node for $f(\alpha^*)$ can recover the $t$-th trace $\mathsf{Tr}\big(p_t(\alpha^*)f(\alpha^*)\big)$, by downloading one repair trace $\mathsf{Tr}\Big(\frac{f(\overline{\alpha})}{\overline{\alpha}-\alpha^*}\Big)$ from the replacement node for $f(\overline{\alpha})$. Similarly, the replacement node for $f(\overline{\alpha})$ can recover the $t$-th trace $\mathsf{Tr}\Big(\frac{f(\alpha)}{\alpha-\overline{\alpha}}\Big)$ by downloading one repair trace $\mathsf{Tr}\Big(\frac{f(\alpha^*)}{\alpha^*-\overline{\alpha}}\Big)$ from the replacement node for $f(\alpha^*)$.*

**Theorem 1.** *The depth-one collaborative repair scheme can be used to repair any two erased symbols of a Reed-Solomon codes RS$(A, k)$ at a repair bandwidth of $n-1$ sub-symbols per symbol, given that $n-k \geq |B|^{t-1}$ and the characteristic of $F$ divides $t$.*

*Proof.* By Lemma 1 (b), $p_i(\alpha^*) = u_i$ and $q_i(\overline{\alpha}) = v_i$ for $i \in [t]$. Recall that the sets $U' = \{u_1, \ldots, u_t\}$ and $V' = \{v_1, \ldots, v_t\}$ are both linearly independent over $B$. Therefore, after the two phases, each RN obtains $t$ independent traces for each erased symbol, $\mathsf{Tr}\big(u_i f(\alpha^*)\big)$, for $f(\alpha^*)$, and $\mathsf{Tr}\big(v_i f(\overline{\alpha})\big)$, for $f(\overline{\alpha})$, for all $i \in [t]$. Thus, each erased symbol will have $t$ independent traces for its recovery. Each RN downloads $n-2$ sub-symbols in the Download Phase and one sub-symbol in the Collaboration Phase, according to Corollary 1 and Lemma 6, which sum up to a total repair bandwidth of $n-1$ sub-symbols. ■

**Remark 1.** In our repair scheme, each RN uses a bandwidth of $n-1$ sub-symbols, which is the same as the case of one erasure in [7]. In a naive scheme, one RN first downloads $kt$ sub-symbols from some $k$ available nodes, recover both erased symbols, and then sends the corresponding symbol to the other RN. Its total bandwidth used is $kt+t$, which is worse than ours if $\frac{k+1}{n-1} \geq \frac{2}{t}$, i.e. when $t$ is not small or the code has high rate.

**Example 1.** Let $q=2$, $t=2$, $n=4$, and $k=2$. Let $\mathbb{F}_4 = \{0, 1, \xi, \xi^2\}$, where $\xi^2+\xi+1=0$. Then $\{1, \xi\}$ is a basis of $F = \mathbb{F}_4$ over $B = \mathbb{F}_2$. Moreover, each element $\boldsymbol{a} \in \mathbb{F}_4$ can be represented by a pair of bits $(a_1, a_2)$ where $\boldsymbol{a} = a_1 + a_2\xi$. Suppose the stored file is $(\boldsymbol{a}, \boldsymbol{b}) \in \mathbb{F}_4^2$. To devise a systematic RS code, we associate with each file $(\boldsymbol{a}, \boldsymbol{b}) \in \mathbb{F}_4^2$ a polynomial $f(x) = f_{\boldsymbol{a},\boldsymbol{b}}(x) \triangleq \boldsymbol{a} + (\boldsymbol{b}-\boldsymbol{a})x$. We have

$$\begin{aligned}
f(0) &= a_1 + a_2\xi = \boldsymbol{a},\\
f(1) &= b_1 + b_2\xi = \boldsymbol{b},\\
f(\xi) &= (a_1 + a_2 + b_2) + (a_1 + b_1 + b_2)\xi,\\
f(\xi^2) &= (a_2 + b_1 + b_2) + (a_1 + a_2 + b_1)\xi.
\end{aligned}$$

The four codeword symbols $f(0)$, $f(1)$, $f(\xi)$, and $f(\xi^2)$ are stored at Node 1, Node 2, Node 3, and Node 4, respectively, as depicted in Fig. 1.

**Download Phase.** Set

$$Q_{1,\xi}(z) \triangleq \mathsf{Tr}\big(z(\xi-1)\big) = \xi z^2 + \xi^2 z.$$

We choose two bases $U = V = \{\xi\}$ of the root space of $Q_{1,\xi}(z)$. Set $p_1(x) = \mathsf{Tr}\big(\xi(x-1)\big)/(x-1) = \xi^2 x + 1$, and $q_1(x) = \mathsf{Tr}\big(\xi(x-\xi)\big)/(x-\xi) = \xi^2 x + \xi^2$. RN2 (RN for Node 2) downloads two bits from the two available nodes, namely $a_2 = \mathsf{Tr}\big(f(0)/(0-1)\big)$ from Node 1 and $a_2+b_1+b_2 = \mathsf{Tr}\big(f(\xi^2)/(\xi^2-1)\big)$ from Node 4. It then uses (11) to obtain the first trace $b_1+b_2 = \mathsf{Tr}(\xi f(1)) = 1\times a_2 + 1\times(a_2+b_1+b_2)$. Similarly, RN3 (RN for Node 3) also downloads $a_1 = \mathsf{Tr}\big(f(0)/(0-\xi)\big)$ from Node 1 and $a_1+a_2+b_1 = \mathsf{Tr}\big(f(\xi^2)/(\xi^2-\xi)\big)$ from Node 4. It then recovers $a_2+b_1 = \mathsf{Tr}(\xi f(\xi)) = 1\times a_1 + 1\times(a_1+a_2+b_1)$.

**Collaboration Phase.** RN2 sends $b_1+b_2$ over to the RN3, which, by Lemma 5, is the same as $\mathsf{Tr}\big(f(1)/(1-\xi)\big)$. Conversely, RN3 sends $a_2+b_1$ over to RN2, which is the same as $\mathsf{Tr}\big(f(\xi)/(\xi-1)\big)$. $U$ and $V$ are extended to the basis $\{\xi,\xi^2\}$ of $\mathbb{F}_4$ over $\mathbb{F}_2$. Set $p_2(x) = \mathsf{Tr}\big(\xi^2(x-1)\big)/(x-1) = \xi x + 1$, and $q_2(x) = \mathsf{Tr}\big(\xi^2(x-\xi)\big)/(x-\xi) = \xi x$. Now, RN2 has three repair traces to recover the second trace of $f(1)$ using $p_2$, i.e. $b_1 = \mathsf{Tr}\big(\xi^2 f(1)\big) = 1\times a_2 + 0\times(a_2+b_1+b_2) + 1\times(a_2+b_1)$. Based on the two traces $b_1+b_2$ and $b_1$, the erased symbol $\boldsymbol{b} = f(1)$ can be recovered. Similarly, RN3 can recover the second trace of $f(\xi)$ as $a_1+a_2+b_2 = \mathsf{Tr}\big(\xi^2 f(\xi)\big) = 0\times(a_1) + 1\times(b_1+b_2) + 1\times(a_1+a_2+b_1)$, and then can recover $f(\xi)$ completely.

### *C. A Depth-Two Repair Scheme for Two Erasures*

We modify the depth-one repair scheme developed in the previous subsection to obtain a depth-two scheme that works for *all* field extension degrees. We still generate the checks $p_1,\dots,p_t$ and $q_1,\dots,q_t$ as in the first scheme, given by (9) and (10), respectively. However, the RN for $f(\alpha^*)$, instead of $p_i$, uses the following checks

$$p_i^*(x) \triangleq \tau p_i(x), \qquad \text{for all } i = 1,\dots,t. \tag{15}$$

where $\tau \triangleq q_1(\overline{\alpha})/p_t(\overline{\alpha}) = v_1/p_t(\overline{\alpha})$ is a *nonzero* constant, which only depends on $\alpha^*$, $\overline{\alpha}$, $u_t$, and $v_1$, and not on $f$. Note that since $u_t \notin K_{\alpha^*,\overline{\alpha}}$, $p_t(\overline{\alpha}) \neq 0$. Hence, $\tau$ is well defined. Clearly, $\deg(p_i^*) = \deg(p_i) < n-k$ and hence, $p_i^*(x)$ serve as check polynomials of the code.

**Lemma 7.** *The check polynomials $p_i^*(x)$ defined as in* (15) *satisfy the following properties.*

(P1) $p_1^*(\overline{\alpha}) = p_t^*(\overline{\alpha}) = \cdots = p_{t-1}^*(\overline{\alpha}) = 0.$

(P2) $p_t^*(\overline{\alpha}) = q_1(\overline{\alpha}) = v_1 \neq 0.$

(P3) $\{p_1^*(\alpha^*),\dots,p_t^*(\alpha^*)\}$ *is a basis of $F$ over $B$.*

*Proof.* The first property (P1) holds because $p_i^*(\overline{\alpha}) = \tau p_i(\overline{\alpha})$ and $p_i(\overline{\alpha}) = 0$ for every $i = 1,\dots,t-1$ by Lemma 3 (a). Property (P2) is obvious. Property (P3) follows from the fact that $p_i^*(\alpha^*) = \tau p_i(\alpha^*)$, $\tau \neq 0$, and that $U' = \{u_1,\dots,u_t\} = \{p_1(\alpha^*),\dots,p_t(\alpha^*)\}$ is a basis of $F$ over $B$. ■

**Download Phase.** In this phase, the RN for $f(\alpha^*)$ uses the first $t-1$ checks $p_1^*,\dots,p_{t-1}^*$ to construct the $t-1$ repair equations. For $i = 1,\dots,t-1$,

$$\begin{aligned}\mathsf{Tr}\big(p_i^*(\alpha^*)f(\alpha^*)\big) &= -\sum_{\alpha\in A\setminus\{\alpha^*\}} \mathsf{Tr}\big(p_i^*(\alpha)f(\alpha)\big)\\ &= -\sum_{\alpha\in A\setminus\{\alpha^*\}} \mathsf{Tr}\big(u_i(\alpha-\alpha^*)\big)\mathsf{Tr}\Big(\frac{\tau f(\alpha)}{\alpha-\alpha^*}\Big). \end{aligned}\tag{16}$$

By Lemma 7 (a), the RHS of (16) does not involve $f(\overline{\alpha})$. Thus, the RN for $f(\alpha^*)$ can determine $t-1$ traces $\mathsf{Tr}\big(p_i^*(\alpha^*)f(\alpha^*)\big)$, $i \in [t-1]$, of $f(\alpha^*)$ by downloading $n-2$ sub-symbols $\mathsf{Tr}\Big(\frac{\tau f(\alpha)}{\alpha-\alpha^*}\Big)$ from the available nodes storing $f(\alpha)$, $\alpha \in A \setminus \{\alpha^*,\overline{\alpha}\}$. The RN for $f(\overline{\alpha})$ follows the same procedure as in the first scheme (Section III-B).

**Collaboration Phase.** The last repair equation for $f(\alpha^*)$ is

$$\begin{aligned}\mathsf{Tr}\big(p_t^*(\alpha^*)f(\alpha^*)\big) + \mathsf{Tr}\big(p_t^*(\overline{\alpha})f(\overline{\alpha})\big) &= -\sum_{\alpha\in A\setminus\{\alpha^*,\overline{\alpha}\}} \mathsf{Tr}\big(p_t^*(\alpha)f(\alpha)\big)\\ &= -\sum_{\alpha\in A\setminus\{\alpha^*,\overline{\alpha}\}} \mathsf{Tr}\big(u_t(\alpha-\alpha^*)\big)\mathsf{Tr}\Big(\frac{\tau f(\alpha)}{\alpha-\alpha^*}\Big). \end{aligned}\tag{17}$$

Clearly, the RN for $f(\alpha^*)$ can compute the RHS of (17) based on what it downloaded in the Download Phase. To obtain the last trace $\mathsf{Tr}\big(p_t^*(\alpha^*)f(\alpha^*)\big)$, it downloads $\mathsf{Tr}\big(p_t^*(\overline{\alpha})f(\overline{\alpha})\big)$ from the RN for $f(\overline{\alpha})$, which is possible because $\mathsf{Tr}\big(p_t^*(\overline{\alpha})f(\overline{\alpha})\big) = \mathsf{Tr}\big(q_1(\overline{\alpha})f(\overline{\alpha})\big)$, due to Lemma 7 (P2), which is already available at the RN for $f(\overline{\alpha})$. Then, due to Lemma 7 (P3), the RN for $f(\alpha^*)$ has $t$ independent traces of $f(\alpha^*)$ to recover this lost symbol. As $f(\alpha^*)$ has been recovered, the RN for $f(\overline{\alpha})$ downloads the repair trace $\mathsf{Tr}\big(f(\alpha^*)/(\alpha^*-\overline{\alpha})\big)$ from the RN for $f(\alpha^*)$ to compute the trace $\mathsf{Tr}\big(q_t(\overline{\alpha})f(\overline{\alpha})\big)$, and then can recover $f(\overline{\alpha})$ completely. Note that the RN for $f(\alpha^*)$ has to first receive the repair trace from the RN for $f(\overline{\alpha})$ before computing and sending out its repair trace for $f(\overline{\alpha})$. Theorem 2 summarizes the discussion.

**Theorem 2.** *The depth-two collaborative repair scheme can be used to repair any two erased symbols of a Reed-Solomon codes $RS(A,k)$ at a repair bandwidth of $n-1$ sub-symbols per symbol, for every field extension degree, given that $n-k \geq |B|^{t-1}$.*

## Acknowledgment

This work has been supported in part by the NSF grant CCF 1526875, the Center for Science of Information under the grant NSF 0939370, and the NSF grant CCF 1619189. The work for H. M. Kiah has been supported in part by a Singapore Ministry of Education (MOE) Tier 1 grant 2016-T1-001-156.

## References

[1] A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inform. Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.

[2] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 2, pp. 300–304, 1960.

[3] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam: North-Holland, 1977.

[4] F. Oggier and A. Datta, "Self-repairing homomorphic codes for distributed storage systems," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2011, pp. 1215–1223.

[5] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *IEEE Trans. Inform. Theory*, vol. 58, no. 11, pp. 6925–6934, 2012.

[6] D. Papailiopoulos and A. Dimakis, "Locally repairable codes," in *IEEE Int. Symp. Inform. Theory (ISIT)*, 2012, pp. 2771–2775.

[7] V. Guruswami and M. Wootters, "Repairing Reed-Solomon codes," in *Proc. Annu. Symp. Theory Comput. (STOC)*, 2016.

[8] H. Dau, I. Duursma, H. M. Kiah, and O. Milenkovic, "Repairing Reed-Solomon codes with multiple erasures," https://arxiv.org/abs/1612.01361.

[9] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications*. Cambridge University Press, 1986.

[10] H. Dau and O. Milenkovic, "Optimal repair schemes for some families of full-length Reed-Solomon codes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, accepted, https://arxiv.org/abs/1701.04120.