[6] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 471–480, July 1973.
[7] H. Witsenhausen and A. Wyner, "Interframe coder for video signals," U. S. Patent 4 191 970, 1980.
[8] A. E. Gamal and A. Orlitsky, "Interactive data compression," in *Proc. 25th Annu. Symp. Foundations of Computer Science*, 1984, pp. 100–108.
[9] H. Witsenhausen, "The zero-error side information problem and chromatic numbers," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 592–593, Sept. 1976.
[10] P. Gemmell, R. Lipton, R. Rubinfeld, M. Sudan, and A. Widgerson, "Self-testing/correcting for polynomials and for approximate functions," in *Proc. ACM Symp. Theory of Computing*, vol. 25, no. 23, 1991, pp. 32–42.
[11] M. Blum and S. Kannan, "Designing programs that check their work," *J. ACM*, vol. 42, no. 1, pp. 269–291, 1995.
[12] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1978.
[13] M. Naor, A. Orlitsky, and P. Shor, "Three results on interactive communication," *IEEE Trans. Information Theory*, vol. 39, pp. 1608–1615, Sept. 1993.
[14] R. McEliece, E. Rodemich, H. Rumsey, Jr., and L. Welch, "New upper bounds on the rate of a code via the Delsarte–Macwilliams inequalities," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 157–166, Mar. 1977.
[15] V. Levenshtein, "On the minimal redundancy of binary error-correcting codes," *Probl. Pered. Inform.*, vol. 10, pp. 26–42, 1974.
[16] E. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.

Data Verification and Reconciliation With Generalized Error-Control Codes

Mark G. Karpovsky, *Fellow, IEEE*, Lev B. Levitin, *Fellow, IEEE*, and Ari Trachtenberg, *Member, IEEE*

***Abstract*—We consider the problem of data reconciliation, which we model as two separate multisets of data that must be reconciled with minimum communication. Under this model, we show that the problem of reconciliation is equivalent to a variant of the graph coloring problem and provide consequent upper and lower bounds on the communication complexity of reconciliation. Further, we show by means of an explicit construction that the problem of reconciliation is, under certain general conditions, equivalent to the problem of finding error-correcting codes for a general class of errors. Under this equivalence, reconciling with little communication is linked to codes with large size, and *vice versa*. We show analogous results for the problem of multiset verification, in which we wish to determine whether two multisets are equal using minimum communication. As a result, a wide body of literature in coding theory may be applied to the problems of reconciliation and verification.**

***Index Terms*—Data reconciliation and verification, error-correcting codes, graph coloring.**

I. Introduction

The problem of reconciling data is inherent to applications that require consistency among distributed information, including diverse examples such as gossip protocols for distributing networked data [1], resource discovery [2], mobile data [3], [4], and sequences of symbols from a given alphabet, such as nucleotide sequences in DNA or amino acids sequences in proteins [5]. In each of these examples, the system needs to determine and, thereafter, reconcile the differences between data stored in physically separate locations.

From the perspectives of scalability and performance, it is important that reconciliations occur with minimum communication, measured both by the number of transmitted bits and by the number of rounds of communication. When data are represented by sets, as can be reasonably modeled for the examples cited above, this problem is known as the *set reconciliation* problem [6], [7]. The *data reconciliation* problem is a natural generalization in which data is represented by multisets rather than sets; the case where only a single message of communication is permitted is termed *one-way* reconciliation.

This correspondence examines the one-way data reconciliation problem within a generalized framework in which differences between multisets correspond to evaluations of arbitrary "error" functions. We show that this problem of reconciliation is equivalent to a variation of the problem of graph coloring: second-order coloring or distance-2 coloring. A second-order coloring of a graph assigns colors to vertices in such a way that any two nodes separated by a path of length at most two are colored differently. Applying well-known results from graph coloring, we then provide lower and upper bounds on the amount of information that must be sent between two hosts for this type of general reconciliation.

In many practical cases, it is not necessary to reconcile two multisets, but merely to determine whether they are in fact the same. This may be the case when testing of a remote device is performed by verification of its signature [8], [9]. Such a determination can often be made with substantially less communication than a full-scale reconciliation. In this context, we consider the problem of *one-way data verification*: verifying that two multisets are the same, subject to a given range of possible differences, by communication of a single message. Again we show that such verification is equivalent to graph coloring and error detection. We also provide both lower and upper bounds on the amount of information that must be exchanged for data verification.

The main contribution of this work is a constructive connection between generalized error-correcting codes and one-way data reconciliation on the one hand, and between generalized error-detecting codes and one-way data verification on the other. Protocols for reconciling multisets that differ by some transformation in a set $\mathcal{E}$ can generate (and be generated from) codes capable of correcting errors represented by $\mathcal{E}$. The communication complexity of such a reconciliation protocol is linked to the size of the corresponding error-correcting code, so that good data reconciliation schemes result from good codes, and *vice versa*. In particular, perfect codes result in optimum one-way data reconciliation protocols. Similar results are provided for verification and error detection.

The problem of reconciliation has been studied extensively from many different perspectives in the literature. We can broadly characterize these perspectives based on their model of the differences between two reconciling hosts.

One model involves synchronizing two discrete random variables with some known joint probability distribution using a minimum communication complexity. Witsenhausen [10] followed by Alon and Orlitsky [11] show a connection between such random variable reconciliation and graph coloring, giving results analogous to those of Sections III-A and IV-A. In addition, Orlitsky [12] showed how to use linear error-correcting codes for a specific instance of data reconciliation.

Manuscript received July 2, 2001; revised January 29, 2002. This work was supported in part by the National Science Foundation under Grants MIP–9630096 and CCR-0133521.

The authors are with the Department of Electrical and Computer Engineering, Reliable Computing Laboratory, Boston University, Boston, MA 02215 USA (trachten@bu.edu).

Communicated by S. Litsyn, Associate Editor for Coding Theory.
Digital Object Identifier 10.1109/TIT.2003.813498

Another model involves two hosts reconciling files (or strings) that differ by a bounded number of insertions, deletions, or modifications (collectively: "edits"). The problem of efficient reconciliation under these circumstances, also known as the edit-distance problem [13], has been extensively studied [14], [15] because of its connections to important fields such as file synchronization and pattern recognition. Levenshtein [16] pioneered work in this area by developing error-correcting codes capable of correcting precisely these types of errors. Recently, in [17], he also examined the problem of reconstructing a sequence from several copies distorted with these types of errors.

In our model, data is represented by *multisets*. The data is thus inherently unindexed, meaning that only the content of the individual data items, and not their relative position, matters. Unlike other models, we also assume that reconciliation is agnostic to the roles of participating hosts, as explained in Section II. Our results extend the bridge between coding theory and data reconciliation originally started in [12], providing statements of conditioned equivalence and corresponding bounds.

We begin in Section II with a brief formal introduction of the three problems connected in this correspondence. Section III addresses the general problem of data verification, proving connections between graph coloring, data verification, and error detection, and describing consequent communication bounds. Section IV provides analogous results for data reconciliation and error correction. Finally, in Section V, we describe several applications.

II. Background

A. Graph Coloring

Definition 1: A *proper coloring* of a graph G with set of vertices V and edges E is an assignment of colors to each vertex in such a way that the vertices of any edge $e \in E$ are colored differently.

A proper coloring using at most k colors will be called a k*-coloring* of the graph. The *chromatic number* of a graph, denoted $\gamma(G)$, is the minimum integer k for which there exists an k-coloring of G.

Definition 2: A *second-order coloring* of a graph is a proper coloring of a graph with the extra property that no two neighbors of any vertex have the same color.

A second-order coloring of G is also a proper coloring of the square of the graph, which is the graph G^2 obtained from G by additionally connecting with an edge each pair of vertices that are of distance two apart. The minimum number of colors needed to second-order color a graph is the *second order chromatic number* of the graph, denoted $\gamma_2(G)$.

B. Error Detection and Correction

Consider the module $\mathbb{Z}_q^n$ consisting of all n-dimensional vectors over the ring $\mathbb{Z}_q$. A q-ary code of length n is simply a subset of the elements of this module.

Definition 3: An *error set* for $\mathbb{Z}_q^n$ is a set $\mathcal{E} = \{e_0, e_1, e_2, \ldots e_{|\mathcal{E}|}\}$ whose elements are functions $e_i\colon \mathbb{Z}_q^n \longrightarrow \mathbb{Z}_q^n$, one of which is the identity function $e_0(x) = x$. If the functions $e_i \in E$ are all bijections and their inverses e_i^{-1} are also in $\mathcal{E}$, then we shall call this set *bijective*. If the functions commute with each other, so that

$$e_i(e_j(x)) = e_j(e_i(x)) \; \forall\, x \in \mathbb{Z}_q^n, \;\; \forall\, e_i, e_j \in \mathcal{E}$$

we shall call this set *commutative*.

We also generalize the concept of an error-ball around a vector with the following definition.

Definition 4: Given an error set $\mathcal{E}$ and a vector $x \in \mathbb{Z}_q^n$, the $\mathcal{E}$*-image* of x is defined to be $\mathcal{E}[x] = \{e(x) | e \in \mathcal{E}\}$. The $\mathcal{E}$*-vicinity* of x is defined as

$$\mathcal{E}(x) = \mathcal{E}[x] \cup \{z \in \mathbb{Z}_q^n | e(z) = x, \, e \in \mathcal{E}\}.$$

More generally, the $\mathcal{E}^k$*-vicinity* is defined to be

$$\mathcal{E}^k(x) = \bigcup_{y \in \mathcal{E}^{k-1}(x)} \mathcal{E}(y)$$

where $\mathcal{E}^1(x) = \mathcal{E}(x)$. The $\mathcal{E}^k$*-image* is defined likewise with brackets replacing parentheses.

Note that if $\mathcal{E}$ is bijective, then $\mathcal{E}^k[x] = \mathcal{E}^k(x)$ for all integers $k \geq 1$.

Definition 5: A code $C \in \mathbb{Z}_q^n$ *detects* the error set $\mathcal{E}$ if $c_i \notin \mathcal{E}[c_j]$ for all $c_i \neq c_j (c_i, c_j) \in \mathcal{C}$ and *corrects* $\mathcal{E}$ if $\mathcal{E}[c_i] \cap \mathcal{E}[c_j] = \emptyset$ for all $c_i \neq c_j (c_i, c_j) \in \mathcal{C}$, where $\emptyset$ denotes the empty set.

C. Set and Multiset Reconciliation and Verification

The traditional formalization of the set reconciliation problem is as follows [6], [7]: given a pair of hosts A and B, each with a set (S_A and S_B, respectively) of length-b bit-strings and no *a priori* knowledge of the other host's set, how can each host determine the mutual difference of the two sets with a minimal amount of communication.

In general, we may consider data represented as multisets whose elements are chosen (possibly with repetition) from a finite, universal set U. Every multiset M whose elements are taken from U may be associated uniquely with a *characteristic vector* $v(M)$ of length $n = |U|$ whose ith component is j if and only if the ith element of U occurs j times in M, for some canonical ordering of the elements of U. We shall generally assume that the multiplicity j of an element is at most $q-1$ so that $v(M) \in \mathbb{Z}_q^{|U|}$. We further limit ourselves to the case where only one of the two hosts needs to determine the multiset held by the other host, based on information transmitted in one message.

From the following definition, we see that one-way data reconciliation functions are precisely those functions that are injective over any given $\mathcal{E}$-vicinity.

Definition 6: The function $\sigma\colon \mathbb{Z}_q^n \longrightarrow \Sigma$ is a *one-way data reconciliation function* for an error set $\mathcal{E}$ if there exists a recovery function $R\colon (\Sigma \times \mathbb{Z}_q^n) \longrightarrow \mathbb{Z}_q^n$ reconciling multisets that differ by one of the functions in $\mathcal{E}$. More precisely, the recovery function must have the property that

$$\forall\, v_A, v_B \in \mathbb{Z}_q^n, \quad v_A \in \mathcal{E}(v_B) \implies R(\sigma(v_A), v_B) = v_A.$$

The *transmission size* of such a reconciliation function is the number of signals $|\Sigma|$ that need to be transmitted for reconciliation.

To perform a reconciliation with such a function, host A, which has the multiset M_A with characteristic vector $v_A = v(M_A)$, would send $\sigma(v_A)$ to host B. By computing $R(\sigma(v_A), v_B)$, host B would then determine the characteristic vector v_A and, consequently, the multiset M_A.

We are also interested in the problem of data verification, due to its connections to set reconciliation and a variety of independent applications such as off-line testing [8] and signature analysis [9]. In these cases, two hosts seek to confirm that they have the same multiset, subject to a known list $\mathcal{E}$ of possible differences.

Definition 7: A function $\alpha\colon \mathbb{Z}_q^n \longrightarrow \Phi$ is a *one-way data verification function* for an error set $\mathcal{E}$ if there exists a decision function $\mathcal{D}\colon (\Phi \times \mathbb{Z}_q^n) \longrightarrow \{0, 1\}$ with the property that

$$\forall\, v_A, v_B \in \mathbb{Z}_q^n,$$
$$v_A \in \mathcal{E}(v_B) \text{ and } \mathcal{D}(\alpha(v_A), v_B) = 1 \iff (v_A = v_B).$$

The *transmission size* of such a verification function is the number of signals $|\Phi|$ that need to be transmitted for verification.

D. Differences Between Reconciliation and Error Correction

There is a subtle, but important, difference between the definition of error correction and that of data reconciliation, especially in the face of a nonbijective set $\mathcal{E}$. Error correction presumes side information about communicating parties: one host has an uncorrupted codeword of the code, and the other host has a corrupted version; however, both hosts know who has the uncorrupted codeword and who has the corrupted version. In the reconciliation case, neither host is restricted to having a codeword, and, more importantly, no side information is available about the direction of the differences between the hosts.

To highlight this issue, consider the case of disseminating data in a peer-to-peer fashion along a branch of a multicast tree, where packet losses may occur in any communication. It is clear that any two hosts along this branch will have a subset relationship, meaning that one host will have a subset of the data held by the other. However, without global knowledge of the multicast tree, two arbitrary hosts would not initially know which has a subset of the other. In other words, reconciliation is assumed to be agnostic to the role (e.g., subset or superset) played by the hosts. A similar situation occurs when reconciling output from two circuits, where one has incurred a stuck at-fault causing several 0's to become 1's or *vice versa* (depending on the location and nature of the error). One cannot know, *a priori*, which circuit has failed or the direction of its errors.

III. Data Verification, Coloring, and Error Detection

The data verification problem can be reformulated formally as follows. Consider two hosts A and B with multisets M_A and M_B, respectively. The goal of verification is to determine whether $M_A = M_B$, subject to the sole *a priori* assumption that $v(M_B)$ is in the $\mathcal{E}$-vicinity of $v(M_A)$. The data verification problem is thus to determine the minimum amount of information A should send to B so that B can decide whether or not $M_A = M_B$.

A. Graph Coloring

Consider a natural graph structure [10] corresponding to an error set $\mathcal{E}$.

Definition 8: The *characteristic graph* of an error set $\mathcal{E}$ is the undirected graph $G_\mathcal{E} = (V, E)$ whose vertices are all characteristic vectors of multisets. Any two vertices $v_1, v_2 \in V$ are connected by an edge in this graph iff there exists a nonidentity error $e \in \mathcal{E}$ such that $e(v_1) = v_2$ or $v_1 = e(v_2)$.

Theorem 1: Any proper coloring of $G_\mathcal{E}$ generates a one-way data verification function α for the error set $\mathcal{E}$. Conversely, any verification function α yields a proper coloring of $G_\mathcal{E}$. The minimum transmission size required for any such verification is precisely the chromatic number $\gamma(G_\mathcal{E})$.

Theorem 1 follows from Definitions 1, 5, and 7 and the fact that each monochromatic set in a proper coloring may be selected as a *level set* (i.e., a set on which the function takes a constant value) for a corresponding one-way data verification function α.

Corollary 1: The minimum transmission size $T_V(\mathcal{E})$ for a one-way data verification function over an error set $\mathcal{E}$ satisfies the inequalities

$$T_V(\mathcal{E}) = \gamma(G_\mathcal{E}) \le \max_{v \in \mathbb{Z}_q^n} |\mathcal{E}(v)| \le 2|\mathcal{E}|. \qquad (1)$$

For bijective errors, $T_V(\mathcal{E}) \le |\mathcal{E}|$.

Proof: The left inequality follows directly from an application of Brook's theorem [18] to Theorem 1, since the degree of a vertex v in the characteristic graph of $\mathcal{E}$ is $|\mathcal{E}(v)| - 1$. The right inequality follows from the definition of $\mathcal{E}(v)$. □

Example 1: Consider the characteristic graph of an error set consisting of all odd-weight translation errors in $\mathbb{Z}_2^n$. Clearly, this graph can be two-colored, indicating that set verification can be done with the transmission of one bit. However, if we simply change the error set to consist of all even-weight translations, then $n - 1$ bits of transmission are needed.

Though a nonoptimal proper coloring satisfying the upper bound in (1) can be generated in linear time $O(|V|) = O(q^n)$, practical use of such techniques is severely limited by the fact that the size of the characteristic graph grows exponentially in the size n of the underlying universal set U. For bijective and commutative errors, a more practical approach is based on error-detecting codes, described in the next section.

B. Error Detection

The following theorem shows that monochromatic vertices in a proper coloring of $G_\mathcal{E}$ and level sets of a verification function each produce error-detecting codes for error set $\mathcal{E}$.

Theorem 2: Any one-way data verification function α for an error set $\mathcal{E}$ with transmission size τ generates a code in $\mathbb{Z}_q^n$ which detects $\mathcal{E}$ and has at least $\frac{q^n}{\tau}$ codewords. Moreover, for any $\mathcal{E}$, the level set of a one-way data verification function for $\mathcal{E}$ is a code $\mathbb{C} \subseteq \mathbb{Z}_q^n$ detecting $\mathcal{E}$.

Proof: Let $\mathbb{C}$ be the code corresponding to a level set of a given verification function α. Then, from Definition 7, it cannot be that $c_i \in \mathcal{E}(c_j)$ for codewords $c_i \neq c_j$. In other words, $\mathbb{C}$ must detect errors in $\mathcal{E}$. Since, by assumption, there are τ level sets for α, at least one of them must correspond to an error-detecting code of size $\ge \frac{q^n}{\tau}$. The second claim in the theorem follows from Definition 5, which implies that coloring codewords monochromatically in $G_\mathcal{E}$ will not violate a proper coloring of the graph. □

Putting together Theorem 2 and Corollary 1 with Brooks' theorem gives the following generalization of the Gilbert–Varshamov bound to our general class of errors; Corollary 4 in a subsequent section provides a similar generalization for error-correcting codes.

Corollary 2: For any error set $\mathcal{E}$ with characteristic graph $G_\mathcal{E}$ there exists an error-detecting code $\mathbb{C}$ with a number of codewords

$$|\mathbb{C}| \ge \frac{q^n}{\gamma(G_\mathcal{E})} \ge \frac{q^n}{\max_{v \in \mathbb{Z}_q^n} |\mathcal{E}(v)|} \ge \frac{q^n}{2|\mathcal{E}|}.$$

For bijective errors, $|C| \ge \frac{q^n}{|\mathcal{E}|}$.

The complement to Theorem 2, presented as Theorem 3, relies upon the notion of vector *orbits*, which partition the space $\mathbb{Z}_q^n$ into equivalence classes.

Definition 9: The *orbit* of an element $v \in \mathbb{Z}_q^n$ under $\mathcal{E}$ is the union of all $\mathcal{E}^k$-vicinities of v

$$O(v) = \bigcup_{k=1}^{q^n-1} \mathcal{E}^k(v). \qquad (2)$$

The following important property of orbits follows straightforwardly.

Lemma 1: If $e_1(v) = e_2(v)$ for $e_1, e_2 \in \mathcal{E}$, for any bijective, commutative error set $\mathcal{E}$, then the functions e_1 and e_2 are identical on the entire orbit $O(v)$, that is, $e_1(x) = e_2(x), \ \forall x \in O(v)$.

We now can state our main theorem for generating a verification function from an error-detecting code.

Theorem 3: Any maximal[1] code $\mathbb{C} \subseteq \mathbb{Z}_q^n$ that detects a bijective and commutative error set $\mathcal{E}$ also generates a one-way data verification function α with transmission size at most

$$\max_{c\in\mathbb{C}} |\mathcal{E}[c]| = \max_{c\in\mathbb{C}} |\mathcal{E}(c)|.$$

Proof: Because $\mathbb{C}$ is maximal and $\mathcal{E}$ is bijective, $\cup_{c\in\mathbb{C}}\mathcal{E}(c) = \mathbb{Z}_q^n$. Thus, there exists a partition D of $\mathbb{Z}_q^n$ into *domains* D_i such that $D_i \subseteq \mathcal{E}(c_i)$ for $c_i \in \mathbb{C}$. In addition, Lemma 1 implies that any given orbit O imposes a partitioning of $\mathcal{E}$ into equivalence classes of error functions acting on O. As such, we may designate a complete set of distinct error functions $\mathcal{E}_i$ around the orbit of a codeword c_i with the defining property that if $e_1, e_2 \in \mathcal{E}_i$, then $e_1(v) \neq e_2(v)$ for all $v \in O(c_i)$. It is assumed that both verifying hosts generate the same partition D and set of functions $\mathcal{E}_i$.

Computing the one-way data verification function $\alpha(v)$ then simply requires finding the domain D_i containing v and the error $e \in \mathcal{E}_i$ that maps the codeword of the domain $c_i \in \mathbb{C}$ onto v (i.e., $e(c_i) = v$ implies $\alpha(v) = e$). Different vectors in the same domain necessarily have different values of α. On the other hand, vectors in different domains either have different values of α or else need not be verified, as the following argument shows.

Suppose $\alpha(v_i) = \alpha(v_j)$ for vectors $v_i \in D_i$ and $v_j \in D_j$ with $i \neq j$. Then, the commutativity and bijectivity of $\mathcal{E}$ insures that v_i and v_j cannot differ by $\tilde{e} \in \mathcal{E}$ or else

$$\begin{aligned} e(v_i) = c_i \qquad e(v_j) &= c_j \qquad \tilde{e}(v_i) = v_j \\ e(\tilde{e}(v_i)) &= \tilde{e}(c_i) \\ e(v_j) &= \tilde{e}(c_i) \\ c_j &= \tilde{e}(c_i) \end{aligned} \tag{3}$$

with (3) contradicting the error-detecting capability of $\mathbb{C}$. Since there is no $e \in \mathcal{E}$ mapping $e(v_i) = v_j$, vectors v_i and v_j are beyond the verification requirements of α.

Thus, $\alpha(v)$ takes on a unique value for each $v \in \mathcal{E}_i$. However, Lemma 1 insures that $|\mathcal{E}_i| = |\mathcal{E}[c_i]|$, giving a maximum transmission size of $\max_{c\in\mathbb{C}} |\mathcal{E}[c]| = \max_{c\in\mathbb{C}} |\mathcal{E}(c)|$. □

Theorem 3 shows an equivalence of data verification and error detection for bijective and commutative error sets $\mathcal{E}$; the theorem does not hold for arbitrary error sets as shown by the following example.

Counterexample 1: Consider the space $\mathbb{Z}_3$ under two error functions: the identity e_0 and the incrementor $e_1(x) = x + 1 \pmod 3$. The code $\mathbb{C} = \{0\}$ is a maximal code detecting $\mathcal{E}$ with $\max_{c\in C} |\mathcal{E}[c]| = 2$. However, the characteristic graph of the error set has chromatic number 3, corresponding to a transmission size of $3 > \max_{c\in C} |\mathcal{E}[c]|$ for the best verification function.

IV. Data Reconciliation, Coloring, and Error Correction

A. Graph Coloring

Theorem 4: Any second-order coloring of $G_{\mathcal{E}}$ generates a one-way data reconciliation function σ for the error set $\mathcal{E}$. Conversely, any such reconciliation function σ yields a second-order coloring of $G_{\mathcal{E}}$. The minimum transmission size required for reconciliation is precisely the second-order chromatic number $\gamma_2(G_{\mathcal{E}})$.

The theorem follows from Definitions 2 and 6 and by an association of vertex colors to values of σ. We may apply Brooks' theorem to the square of the characteristic graph of $\mathcal{E}$ to get the following corollary.

Corollary 3: The minimum transmission size $T_R(\mathcal{E})$ for a one-way data reconciliation function over an error set $\mathcal{E}$ satisfies the inequalities

$$\max_{v\in\mathbb{Z}_q^n} |\mathcal{E}(v)| \leq T_R(\mathcal{E}) \leq \max_{v\in\mathbb{Z}_q^n} |\mathcal{E}^2(v)|. \tag{4}$$

It is interesting to note that for certain error sets, such as those that form a group under composition, the transmission size for reconciliation and verification meet (i.e., $T_V(\mathcal{E}) = \max_v |\mathcal{E}(v)| = T_R(\mathcal{E})$).

B. Error Correction

The following theorem is proved analogously to Theorem 2.

Theorem 5: Any one-way data reconciliation function σ for an error set $\mathcal{E}$ with transmission size τ generates a code in $\mathbb{Z}_q^n$ which corrects $\mathcal{E}$ and has at least $\frac{q^n}{\tau}$ codewords. Moreover, each monochromatic set of vertices in a second-order coloring of $G_{\mathcal{E}}$ is a code that corrects $\mathcal{E}$.

Applying Theorem 5 and Corollary 3 gives the following result, which reduces to the well-known Gilbert–Varshamov bound for Hamming errors.

Corollary 4: For any error set $\mathcal{E}$ with characteristic graph $G_{\mathcal{E}}$, there exists an error-correcting code $\mathbb{C}$ with number of codewords

$$\frac{q^n}{\max\limits_{v\in\mathbb{Z}_q^n}|\mathcal{E}^2(v)|} \leq \frac{q^n}{\gamma_2(G_{\mathcal{E}})} \leq |\mathbb{C}| \leq \frac{q^n}{\max\limits_{v\in\mathbb{Z}_q^n}|\mathcal{E}(v)|}. \tag{5}$$

By Theorem 2, any code that *detects* error set $\mathcal{E}$ is a monochromatic set of vertices in a proper coloring of $G_{\mathcal{E}}$. In contrast with this, not every code that *corrects* $\mathcal{E}$ is a monochromatic set in a second-order coloring of $G_{\mathcal{E}}$, as demonstrated by the following counterexample.

Counterexample 2: Consider $\mathbb{Z}_2^3$ under the error set $\mathcal{E} = \{e_0, e_1, e_2\}$, where e_0 is the identity function and e_1 and e_2 are given by the table shown at the bottom of the page.

It is easy to see that code $\mathbb{C} = \{000, 011, 110\}$ corrects the error set $\mathcal{E}$. However, $\mathbb{C}$ cannot be a monochromatic set in any second-order coloring of the characteristic graph $G_{\mathcal{E}}$ because codewords 000 and 011 are of distance 2 from each other.

As with verification, an error-correcting code for a bijective, commutative error set generates a reconciliation function.

$x =$	000	001	010	011	100	101	110	111
$e_1(x) =$	000	010	100	001	000	110	100	011
$e_2(x) =$	000	000	011	010	101	001	101	110

[1] A code is maximal if codewords cannot be added to it without affecting its error-detecting/correcting capability.

Theorem 6: Any maximal code $\mathbb{C} \subseteq \mathbb{Z}_q^n$ that corrects a bijective and commutative error set $\mathcal{E}$ generates a one-way data reconciliation function σ with transmission size at most $\max_{c \in \mathbb{C}} |\mathcal{E}^2(c)|$.

The proof follows similarly to that for Theorem 3, but makes additional use of the fact that the set $\mathcal{E}_v^2$ of distinct functions $e \in \mathcal{E}^2$ acting on the orbit $O(v)$ has cardinality equal to the number of points in $\mathcal{E}^2(v)$ (i.e., $|\mathcal{E}_v^2| = |\mathcal{E}^2(v)|$). We may thus formulate the following protocol for reconciliation using any code $\mathbb{C}$ correcting $\mathcal{E}$.

Protocol 1: (**Data reconciliation using a code**) All hosts interested in reconciling must agree upon fixed methods for generating

- a code $\mathbb{C}$ correcting $\mathcal{E}$,
- a partitioning D of $\mathbb{Z}_q^n$ into domains $D_i \subseteq \mathcal{E}^2(c_i)$, and
- a complete set of distinct errors $\mathcal{E}_i \subseteq \mathcal{E}$ acting upon each orbit $O(c_i)$, $c_i \in \mathbb{C}$.

Host B with characteristic vector v_B can then reconcile with host A with vector v_A as follows.

2) A sends to B the unique value $\sigma(v_A) = e_A$ such that $e_A(c_A) = v_A$ for $c_A \in \mathbb{C}$ and $e_A \in \mathcal{E}_A^2$.

3) B finds the unique $e \in \mathcal{E}_B$ with the property that $e_A^{-1}(v_B) = e(c)$ for $c \in \mathbb{C}$. If such an e exists, then Theorem 6 assures that $c = c_A$; otherwise, v_A and v_B are irreconcilable.

4) B determines A's characteristic vector as $v_A = e^{-1}(v_B)$.

In the case of Hamming errors (i.e., $\mathcal{E}$ consists of functions $e_i(x) = x + v$ for v of Hamming weight at most some fixed t), one can understand Theorem 6 in terms of the covering radius of a code [19], defined to be the minimum value ρ for which balls of radius ρ around codewords will completely cover $\mathbb{Z}_q^n$.

Corollary 5: For code $\mathbb{C}$ of length n, covering radius ρ, and minimum distance $2t+1$, there corresponds a one-way data reconciliation function for at most t Hamming errors with transmission size

$$T_R(\mathcal{E}) \le \sum_{i=0}^{\rho} \binom{n}{i} (q-1)^i. \tag{6}$$

Note that for Hamming errors, a maximal code has $t \le \rho \le 2t$ [19], so that (6) provides a stronger bound than that of Corollary 3.

A consequence of Corollary 5 is that perfect codes produce optimal one-way data reconciliation schemes. For example, the length 23 binary, triple-error-correcting Golay code [20] can be used to produce an optimal scheme for reconciling subsets of $\mathbb{Z}_{23}$.

V. Examples and Applications

A. Group Errors

Our first examples are of error sets that form a group under composition, in which case several of the bounds in the correspondence meet.

Example 2: (Cyclic shift) Consider the set of errors corresponding to left or right cyclic shifts of up to $t \le n$ positions of a vector

$$\begin{aligned}\mathcal{E}_{cycle} = \{e_i(\langle x_0 x_2 x_3 \cdots x_{n-1}\rangle) \\ = \langle x_i x_{1+i} x_{2+i} \cdots x_{n-1+i}\rangle | -t \le i \le t\}\end{aligned}$$

where indexes of x are taken $\bmod n$. When $t \ge \lfloor \frac{n}{2} \rfloor$, the set of shifts of a given vector forms a clique in the characteristic graph $G_{\mathcal{E}_{cycle}}$ so that the minimum transmission size is

$$T_V(\mathcal{E}_{cycle}) = \gamma(G_{\mathcal{E}_{cycle}}) = n = T_R(\mathcal{E}_{cycle})$$

meaning that the upper bound of Corollary 1 and the lower bound of Corollary 3 are attained. In the alternate case, where $t < \lfloor \frac{n}{2} \rfloor$, the transmission size is

$$\begin{aligned}T_V(\mathcal{E}_{cycle}) = \gamma(G_{\mathcal{E}_{cycle}}) = t+1 \\ < \max_x |\mathcal{E}_{cycle}(x)| = 2t+1 = T_R(\mathcal{E}_{cycle}).\end{aligned}$$

Example 3: (Permutations) Consider the error set of permutation errors, in which vectors of length n may be changed by any permutation of their bits. This error set, though noncommutative, affords a simple reconciliation scheme derived from a corresponding error-correcting code $\mathbb{C}_\Pi$. The codewords of $\mathbb{C}_\Pi$ are vectors whose rightmost bits are 1 and remaining bits 0, so that $|C_\Pi| = n+1$. Reconciling two vectors v_A and v_B merely involves transmitting the permutations π_A and π_B that transform these vectors to respective codewords c_A, $c_B \in \mathbb{C}_\Pi$. Having received π_B, host A can simply compute $\pi_B^{-1}(c_A)$ to determine v_B if the two vectors differ by a permutation error.

Since the characteristic graph for permutation errors consists of disjoint cliques (one for each codeword c_i), the amount of communication required for both verification and reconciliation of permutations of length n vectors is equal to the size of the largest clique

$$T_V(\mathcal{E}_{\text{perm}}) = T_R(\mathcal{E}_{\text{perm}}) = \binom{n}{\lceil \frac{n}{2} \rceil}.$$

B. Reconciliation for Hamming Errors

Consider the traditional formulation of the set reconciliation problem, as stated in Section II-C. This form of set reconciliation has many applications in networking, including gossip protocols [1], [21], [22], resource discovery [2], [6], and synchronization [3], [4].

In the context of this work, we may view classical set reconciliation as a one-way data reconciliation problem under Hamming errors of weight at most t. Corollary 3 shows that the minimum number of bits needed for such reconciliation when $t = n = 2^b$ is $\log_2(\max_x |\mathcal{E}(x)|) = 2^b$. In other words, the optimal way to surely reconcile two sets in the face of arbitrary additive errors is to transmit the entire set. On the other hand, if a t is known *a priori* then the size of the corresponding error set is

$$\sum_{i=0}^{t} \binom{2^b}{i} \sim 2^{bt}, \qquad \text{for } \frac{t}{2^b} \longrightarrow 0. \tag{7}$$

The upper bound transmission size of bt bits is essentially achieved in [7] using polynomial interpolation, and can be alternatively achieved by using reconciliation schemes based on BCH (Bose–Chaudhuri–Hocquenghem) codes.

C. Page Errors

The model of single page errors assumes that errors occur only in the same region. Consider, for the sake of an example, that two hosts each have subsets of $\mathbb{Z}_{16}$, with page regions defined every four elements; thus, the first page contains elements $\{1, 2, 3, 4\}$, the next page contains $\{5, 6, 7, 8\}$, etc. The error set $\mathcal{E}_{\text{page}}$ for this model contains all functions that corrupt a single page. For example, a corruption of the first page by a toggling the presence of set elements 2 and 4 is given by

$$e(x) = x \oplus 0101\ 0000\ 0000\ 0000$$

where x is the characteristic vector of the uncorrupted set. Since $|\mathcal{E}(x)| = 61$ for any x, Corollary 1 implies that one-way set verification requires at most 6 bits of communication. One-way set reconciliation, on the other hand, requires

$$6 \le \log T_R(\mathcal{E}_{page}) \le \log(|\mathcal{E}^2(X)|) = \log(1411) < 11$$

bits of communication. Using extended Reed–Solomon codes of length 4 over $\mathbb{Z}_2^4$, we have $\log T_V(\mathcal{E}_{\rm page}) \leq 4$ bits and $\log T_R(\mathcal{E}_{\rm page}) \leq 8$ bits of communication. In the general case, where each page contains 2^i elements, each encoded by a b-bit vector, and any t pages can be corrupted, the following asymptotic result holds provided that $t/2^{b-i} \longrightarrow 0$ and $t/2^i \longrightarrow 0$:

$$\log T_R(\mathcal{E}_{\rm page}) \leq \log \left[\sum_{j=0}^{2t} (2^i - 1)^j \binom{2^{b-i}}{j} \right] \sim 2bt. \tag{8}$$

D. Client–Server Reconciliation

Consider the case of a client host maintaining a subset of data on a serving host. The error set for verification and reconciliation is the set of unidirectional errors $\mathcal{E}_U$, that is, $e(x) = y$ for $e \in \mathcal{E}_U$ if and only if the set $v^{-1}(x) \subseteq v^{-1}(y)$, where v^{-1} is the mapping transforming characteristic vectors to sets. This error set is commutative, but not bijective.

Consider the characteristic graph of this error set over binary vectors of length n and for up to t unidirectional errors. The vector of all zeros is clearly contained in a clique of size t consisting of vectors starting with up to t ones. Applying Corollary 1, we see that verification requires a transmission size of at least $T_V(\mathcal{E}_U) \geq t+1$ signals. In fact, this lower bound is easily achievable since the server can send to the client the number of entries it has, $\bmod t$, from which the client can verify equality.

For reconciliation, we note that the vectors in the ball of errors around the all-zero vector must each be colored differently, giving $T_R(\mathcal{E}_U) \geq \sum_{i=0}^{t} \binom{n}{i}$. Note that this is identical to the lower bound (7) for reconciling under classical errors. Thus, though it is much easier to verify unidirectional errors than classical errors, it is just as hard to reconcile under either error set.

VI. Conclusion

In this work, we have studied the problems of verification and reconciliation of remote data with a minimum amount of communication. We have demonstrated connections between one-way data reconciliation/verification, error-control codes, and graph coloring over a general error set. In particular, we have described in Section IV how to transform an arbitrary code that corrects a general class of commutative and bijective errors into an algorithm for data reconciliation, and *vice versa*; similarly, in Section III, we have shown how such an error-detecting code can be used to perform data verification. The quality of the derived error-correcting/detecting codes is dependent on the quality of the chosen data reconciliation/verification schemes, and *vice versa*, with the particular example that perfect codes generate optimal schemes.

Finally, we have presented a number of examples throughout the work and in Section V, thereby demonstrating the applicability of this work to such diverse areas as testing, file synchronization, and client–server network updates.

Acknowledgment

The authors wish to thank Alon Orlitsky, Krishnamurthy Viswanathan, and Junan Zhang for their interest and thoughtful discussions, and the anonymous referees and Associate Editor for insightful comments. Ari Trachtenberg would also like to thank Yaron Minsky for stimulating discussions and Alexander Vardy for encouragement.

References

[1] R. van Renesse, Y. Minsky, and M. Hayden, “A gossip-style failure detection service,” in *Middleware ’98: IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing*, N. Davies, K. Raymond, and J. Seitz, Eds. New York: Springer-Verlag, 1998, pp. 55–70.

[2] M. Harchol-Balter, T. Leighton, and D. Lewin, “Resource discovery in distributed networks,” in *Proc. 18th Annu. ACM-SIGACT/SIGOPS Symp. Principles of Distributed Computing*, Atlanta, GA, May 1999.

[3] A. Trachtenberg and D. Starobinski, “Toward global synchronization,” presented at the Large Scale Networks Workshop, Mar. 2001. [Online]. Available: http://ana.lcs.mit.edu/sollins/LSN-Workshop/papers/.

[4] A. Trachtenberg, D. Starobinski, and S. Agarwal, “Fast PDA synchronization using characteristic polynomial interpolation,” in *Proc. INFOCOM*, June 2002.

[5] R. Durbin, S. Eddy, A. Krogh, and G. Mitchéson, *Biological Sequence Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[6] Y. Minsky, A. Trachtenberg, and R. Zippel, “Set reconciliation with nearly optimal communication complexity,” Cornell Univ., Ithaca, NY, Tech. Reps. TR1999-1778, TR2000-1796,TR2000-1813, 2000.

[7] ——, “Set reconciliation with nearly optimal communication complexity,” in *Proc. IEEE Int. Symp. Information Theory*, Washington, DC, June 2001, p. 232. Also, accepted for publication in *IEEE Trans. Inform. Theory*.

[8] D. Siewiorek and R. Swarz, *Reliable Computer Systems: Design and Evaluation*. Bedford, MA: Digital Press, 1992.

[9] M. G. Karpovsky and P. Nagvajara, “Design of self-diagnostic boards by signature analysis,” *IEEE Trans. Ind. Electron.*, vol. 36, pp. 241–246, May 1989.

[10] H. Witsenhausen, “The zero-error side information problem and chromatic numbers,” *IEEE Trans. Inform. Theory*, vol. IT-22, p. 592, Sept. 1976.

[11] N. Alon and A. Orlitsky, “Source coding and graphs entropies,” *IEEE Trans. Inform. Theory*, vol. 42, pp. 1329–1339, Sept. 1996.

[12] A. Orlitsky, “Interactive communication of balanced distributions and correlated files,” *SIAM J. Discr. Math.*, vol. 6, no. 4, pp. 548–564, Nov. 1993.

[13] ——, “Interactive communication: Balanced distributions, correlated files, and average-case complexity,” in *Proc. 32nd Annu. Symp. Foundations of Computer Science*, 1991, pp. 228–238.

[14] G. Cormode, M. Paterson, S. Sahinhalp, and U. Vishkin, “Communication complexity of document exchange,” presented at the ACM-SIAM Symposium on Discrete Algorithms, Jan. 2000.

[15] T. Schwarz, R. Bowdidge, and W. Burkhard, “Low cost comparisons of file copies,” in *Proc. Int. Conf. Distributed Computing Systems*, 1990, pp. 196–202.

[16] V. Levenshtein, “Binary codes capable of correcting spurious insertions and deletions of ones,” *Probl. Inform. Transm.*, vol. 1, no. 1, pp. 8–17, 1965.

[17] ——, “Efficient reconstruction of sequences,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 2–22, Jan. 2001.

[18] R. Brooks, “On coloring the nodes of a network,” in *Proc. Cambridge Phil. Soc.*, vol. 37, 1941, pp. 194–197.

[19] G. Cohen, I. Honkala, S. Litsyn, and A. Lobstein, *Covering Codes*. Amsterdam, The Netherlands: Elsevier, 1997.

[20] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*. New York: North-Holland, 1977.

[21] K. Guo, M. Hayden, R. v. Renesse, W. Vogels, and K. P. Birman, “GSGC: An efficient gossip-style garbage collection scheme for scalable reliable multicast,” Cornell Univ., Tech. Rep., Dec. 1997.

[22] M. Hayden and K. Birman, “Probabilistic broadcast,” Cornell Univ., Tech. Rep., 1996.