Summary - Data Verification and Reconciliation With Generalized Error-Control Codes

Bowen Song[1]

August 27, 2018

Abstract— **This report is a part of an independent study for set and string reconciliation problems in distributed systems. The paper [1] considers the problem of reconciling two multisets of data residing on separated hosts using the minimum amount of communication. The authors make connections to graph coloring problem and apply coding theory to the problem of data verification and reconciliation.**

I. Introduction

The paper considers a situation where two multisets of data from a finite universal set wish to reconcile their difference using the minimum amount of communication. The approach is to apply the general error-correcting codes as a single-round set reconciliation protocol and use error-detecting codes to verify data consistency. The main problem is to find error-control codes as functions that apply to data verification and reconciliation. The paper makes a connection to graph coloring problem that reversibly generates verification and reconciliation functions. The verification and reconciliation functions evaluate vector as a unique representation of multisets data, compress them into codewords, and transmit them for the receiving hosts to verify or reconcile data.

II. Algorithm Overview

The paper proposes an algorithm for a system setting of hosts Alice and Bob wishing to verify and reconcile their multisets of data. The algorithm can successfully verify and reconcile the multisets of data if their characteristic vectors v_A and v_B are within ϵ-vicinity of each other. In another word, the data in Bob has to be a distortion of data in Alice via an error function ϵ in a specific error set. The assumption is that both Alice and Bob hold the same set of ϵ's.

The vector of each multiset from universe U is a unique characteristic vector of length $|U|$. The components of a vector stand for occurrences of elements in a multiset from universe U in canonical order. If v_A and v_B are the same and represent elements from the same universe, the two multisets are considered equal. According to the same set of ϵ's, the algorithm partitions universe U into partitions of error classes to ensure each error function and a vector within the partition has a unique value.

Hosts can verify their vector of multisets' consistency by sharing codewords generated from verification functions from coloring error set graph. The vector of multisets and error functions generate the codewords. To recover the verification function from a codeword, the reconciling host needs to find an error from the error set that maps the codeword to its own vector of multisets. If such error exists, the algorithm believes that reconciliation would be possible. There is an identity function in an error set which verifies that the two compared data are the same.

For Bob to reconcile with Alice's data, Alice computes a reconciling function based on the second-order coloring of error set graph. The next few steps are the same as the verification procedure. Once an error function is found, Bob can move to the reconciliation part by inverting the error function and compute v_A based on its own vector.

III. Algorithm Performance Analysis

Codes	Communication Complexity
Verification	$O(max_{c\in\mathcal{C}}\|\epsilon(c)\|)$
Reconciliation	$O(max_{c\in\mathcal{C}}\|\epsilon^2(c)\|)$

c = codeword
$\mathcal{C}$ = Maximal Code
ϵ = error set

TABLE I

Performance Metric

Maximal code is a code that is not extendable without change the maximum distance between its codewords.

IV. Conclusion

The algorithm only needs one round of communication for reconciliation with the constraint that the data of the reconciling multisets are similar enough to be recovered by available error functions. Since the available error functions can be restricted to a fixed size for each universe of data, there is a maximum amount of error between two reconciling multisets that the algorithm can handle. Although most reconciling algorithm loses its communication efficiency after reconciling data are getting too different from each other, this shortcoming does not reduce the range of application of the algorithm.

References

[1] M. G. Karpovsky, L. B. Levitin, and A. Trachtenberg, "Data verification and reconciliation with generalized error-control codes," *IEEE Transactions on Information Theory*, vol. 49, no. 7, pp. 1788–1793, 2003.

[1] B. Song is with Department of Electrical and Computer Engineering, Boston University, Boston MA, sbowen@bu.edu