Editorial

Frédérique Oggier*

Division of Mathematical Sciences,
Nanyang Technological University,
21 Nanyang Link,
637371 Singapore
Email: frederique@ntu.edu.sg
*Corresponding guest editor

Alexander Barg

Department of Electrical and Computer Engineering,
University of Maryland,
College Park, MD 20742, USA
Email: abarg@umd.edu

Gérard Cohen

Telecom ParisTech,
Informatique et Reseaux,
46 rue Barrault,
Paris cedex 13, 75634, France
Email: gerard.cohen@telecom-paristech.fr

Classically, communication takes place over a noisy channel that connects a sender and a receiver. Coding is used to add redundancy to transmitted messages in order to protect them from the channel noise, so that the receiver be able to extract the message meant for him from the noisy signal. Coding techniques are also used to increase efficiency of storage systems, including recent applications of codes in distributed storage. In problems related to distributed storage, one usually assumes that data are stored across a network of nodes each of which is prone to failure. In this scenario, erasure coding is used to store data with added redundancy to ensure low overhead and efficient recovery of data stored on the failed drives.

The basic analogy between storage and special communication is clear: erasures in the communication channel correspond to node failures in the distributed storage system, and good codes for communication are capable of providing good protection against failures in storage systems. But there are also important differences, one of them being maintenance over time: communication is instantaneous, and once a corrupted message is received, it does not deteriorate anymore. This is not the case for a distributed storage system, where data need to be kept over months or years. Once some failures have happened, nothing prevents more failures to happen; on the contrary, they may trigger correlated failures. This justifies the need for a maintenance mechanism, where once a node failure is detected, the node content is recovered using redundant data stored at other functional nodes.

The system is capable of creating a copy of the failed node and maintaining the data integrity as well as the original encoding structure. In recent years, this view of distributed storage systems has motivated a line of research around the notion of data 'repairability', i.e., design of coding schemes that enable low-overhead storage with efficient maintenance of the system that supports protection against node failures.

This special issue contains recent research results on the design and analysis of codes for distributed storage. The papers naturally fall under two main trends related to codes with repairability, namely regenerating codes, which seek to minimise the amount of bandwidth used to repair the failed nodes; and locally repairable codes, which are designed to perform repair by contacting only a small number of live nodes.

The papers "Outer bounds on the storage-repair bandwidth trade-off of exact-repair regenerating codes" by Birenjith Sasidharan, N. Prakash, M. Nikhil Krishnan, Myna Vajha, Kaushik Senthoor and P. Vijay Kumar, and "Cooperative repair of multiple node failures in distributed storage systems", by Kenneth Shum and Junyu Chen, consider regenerating codes and the trade-off between storage and repair bandwidth with or without node collaboration.

The papers "Integrated interleaved codes as locally recoverable codes: properties and performance" by Mario Blaum and Steven Hetzler, and "Cyclic LRC codes, binary LRC codes, and upper bounds on the distance of cyclic codes" by Itzhak Tamo, Alexander Barg, Sreechakra Goparaju and Robert Calderbank look at the design of locally repairable codes emphasising small alphabet size, a feature particularly important in applications.

The paper "Updatable encryption in distributed storage networks using key-homomorphic pseudo random functions" by Jhordany Rodriguez Parra, Terence Chan and Siu-Wai Ho investigates the issue of securing storage systems which use codes to provide redundancy.

A combination of two papers on regenerating codes and two papers on locally repairable codes, together with one paper addressing security issue, provides a well-rounded overview of important research directions of codes for distributed storage. We thank the authors and reviewers for their contribution to this project as well as the Editor in Chief of IJICoT for suggesting to put together a special issue on codes for storage and supporting us at all the stages.