Foundations and Trends® in
Theoretical Computer Science
Vol. 13, No. 1-2 (2017) 1–157

DOI: 10.1561/0400000079

now
the essence of knowledge

Coding for Interactive Communication: A Survey

Ran Gelles
Faculty of Engineering, Bar-Ilan University
ran.gelles@biu.ac.il

Contents

Abstract

Coding for interactive communication augments coding theory to the interactive setting: instead of communicating a message from a sender to a receiver, here the parties are involved in an interactive conversation.

Coding schemes allow the parties to complete their conversation despite noise added by the channel. Similar to the unidirectional case, good coding schemes can withstand a large amount of noise and succeed with high probability, while adding only a small amount of redundant information.

We aim at giving a comprehensive view on the foundations of coding for interactive communication. In particular, we review basic features of coding schemes in the interactive setting, and survey the main techniques used in designing such schemes. Furthermore, we survey recent developments in interactive coding schemes, and their applications to other related fields.

R. Gelles. *Coding for Interactive Communication: A Survey*. Foundations and Trends® in Theoretical Computer Science, vol. 13, no. 1-2, pp. 1–157, 2017.
DOI: 10.1561/0400000079.

1

Introduction

1.1 Motivation

Assume Alice and Bob play chess with each other. Since they live in different countries, they play over the phone—every evening Alice calls Bob and they communicate the next move. Now assume the phone line is noisy. While Bob declares his next move "`B2 to B4`", Alice hears "`D2 to D4`". Many days later, when Bob declares a victory, Alice rejects his claims: on her board Bob is not even close to being victorious.

This situation—where two parties *interact* with each other over a noisy communication—is the topic of this manuscript. As opposed to the standard error-correction setting in which one side has some information to convey to the other side, here both sides need to convey information to each other. One could let the parties simply use standard error-correcting codes to send all their information to the other side in a noise-robust way. Such a naïve approach would cause the conversation to be very long: the possibility of interacting is crucial for having efficient conversations. Consider, for instance, the preceding chess game. We can think of each player as having a fixed playing strategy that defines, for every position of the board, the next move that should be played. Compared to the (approximately) one hundred moves an average chess

game takes [21], a description of a player's complete strategy may be extremely long as it needs to describe its move for all the (more than) 10^{40} different board positions [54].

A second naïve approach would be to employ error-correcting codes independently to each round of the conversation. Such an approach would result in poor performance—it could tolerate only a very small amount of noise, namely, the noise it takes to corrupt a *single* message. The ideal solution is one that tolerates a large amount of noise (e.g., a coding that works even if a constant fraction of the messages are corrupted), and yet does not increase the communication by too much (e.g., it multiplies the communication by at most a constant).

1.2 The setting

In the standard interactive communication setting [83], two parties (Alice and Bob) compute a function $f(x, y)$ by holding a conversation. Alice is given the input x, Bob is given y, and they aim to compute $f(x, y)$ by exchanging as few bits as possible. Kushilevitz and Nisan's book [60] gives an excellent description of the communication complexity of computing functions within the interactive setting. In the setting of *coding for interactive communication*, the channel that connects the parties may be noisy (see Figure 1.1). The parties' goal is now to succeed, with high probability, in computing $f(x, y)$ despite the channel's noise, while sending as few bits as possible.

An interactive computation is performed via a *protocol* π, which is a pair of algorithms $\pi = (\pi_A, \pi_B)$ run by Alice and Bob, respectively. Each round, the protocol defines the next message to send, as a function of the party's input, the round number, and the symbols that party has received so far (the *transcript*). For example, in the first round Alice sends $\pi_A(x, 1, \emptyset) \in \Sigma$ and Bob sends $\pi_B(y, 1, \emptyset) \in \Sigma$, where Σ is the channel's alphabet. It is possible that only a single party speaks at each round—for example, Alice at odd rounds and Bob at even rounds. In this case we assume $\pi_A(x, i, \cdot) = \emptyset$ for even i's, and $\pi_B(x, i, \cdot) = \emptyset$ for odd i's.

After a fixed number n of rounds the protocol concludes and outputs a value. Alice's output is given by $\pi_A(x, n + 1, \mathsf{trans}_A)$, and Bob's, by

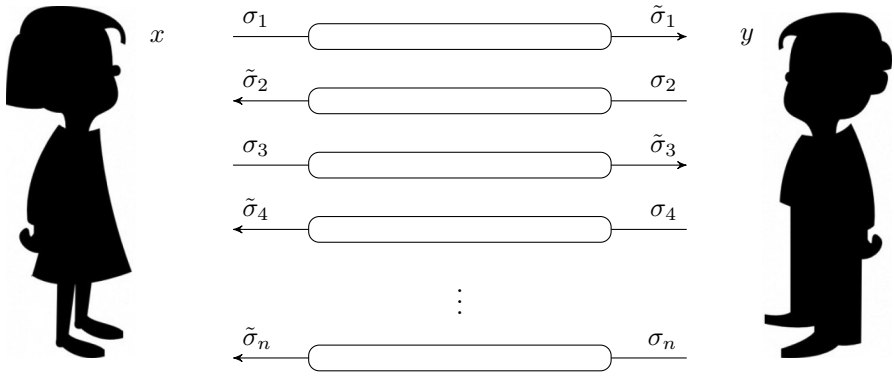

Figure 1.1: An alternating interactive protocol π of length $|\pi| = n$ rounds between Alice that holds the input x, and Bob that holds y, where the communication takes place over a noisy channel.

$\pi_B(x, n+1, \mathsf{trans}_B)$, where trans is the transcript seen by that party. Recall that due to the noisy channel, the parties may receive different symbols than the ones sent to them.

Let X, Y, Z be some finite sets. We say that π computes the function $f : X \times Y \to Z$ if for any pair of inputs $(x, y) \in X \times Y$, both parties output $f(x, y)$. We define the length of the protocol to be its round complexity and denote it by $|\pi| = n$. At each round, the parties send a symbol out of the channels' alphabet Σ. The communication complexity of a protocol, $\mathsf{CC}(\pi)$, is the number of bits communicated by both parties; specifically, assuming one symbol is sent at each round, we have $\mathsf{CC}(\pi) = n \log |\Sigma|$.[1]

When evaluating a noise-resilient protocol π for some function f, it is convenient to compare it to the noiseless protocol π_0 of the same function. In this manuscript, we will care about *coding schemes* that, given a noiseless protocol π_0 construct a resilient version π that outputs a valid transcript of π_0 (and, thus, computes f). Our noiseless protocol π_0 is always defined over a binary alphabet and takes $|\pi_0| = n_0$ rounds. Unless otherwise stated, Alice and Bob talk in π_0 in alternating rounds: Alice sends one bit in odd rounds and Bob sends one bit in even rounds, so that the total communication complexity of the noiseless protocol

[1] Throughout this manuscript, all logarithms are taken to base two.

is $\mathsf{CC}(\pi_0) = n_0$. Note that any protocol can be converted into a binary alternating form at the cost of increasing the communication by at most $2\log|\Sigma|$. Hence, the preceding format of the noiseless protocol π_0 can be considered without loss of generality as its communication complexity differs by a factor of $2\log|\Sigma|$ (this difference does affect the generality when considering communication-optimized coding schemes). Due to technical reasons, we will sometimes need π_0 to be defined for rounds greater than n_0. In this case we can assume that after round n_0, π_0 sends zeros indefinitely.

Remark 1.1. In the following we will use the Landau notations to describe how a coding scheme π behaves with respect to the noiseless protocol π_0. In particular, we write $O(), \Omega()$, and so on, to denote the asymptotic behavior of quantities as $n_0 \to \infty$.

1.3 Parameters that we care about

We can evaluate the performance of an interactive coding scheme according to several parameters.

- **Maximal Noise Rate:** The maximal noise rate that the resilient protocol can tolerate. Usually, the noise rate $\varepsilon \in [0, 1]$ is measured as the fraction of corrupted symbols out of all the symbols that were communicated during the protocol. We will be mostly interested in coding schemes that tolerate a *constant fraction* of noise (i.e., when $\varepsilon = O(1)$). We also consider the case where the noise is stochastic (i.e., where each symbol is corrupted independently with probability ε; see §1.4).

- **Code Rate:** The *rate* of the coding scheme π with respect to the noiseless π_0, defined by

$$r = \frac{\mathsf{CC}(\pi_0)}{\mathsf{CC}(\pi)}.$$

  The rate indicates how much redundancy was added in order to make the computation noise resilient. We will be mostly interested in resilient schemes that have a *constant rate* (positive rate), in

which the communication complexity of the resilient scheme is at most a constant times more than the complexity of the noiseless computation, $\mathsf{CC}(\pi) = O(\mathsf{CC}(\pi_0)) = O(n_0)$. If a scheme does not have a constant rate, that is, when $\lim_{n_0 \to \infty} r = 0$, we say the coding scheme has a *vanishing* rate.

- **Success Probability** The probability that both parties output the correct value. The probability is over the randomness of the protocol (if randomized) and the noise (if randomized). We aim to obtain coding schemes that succeed with exponentially high probability in the length of the noiseless protocol, $1 - 2^{-\Omega(n_0)}$.

- **Efficiency:** The computational efficiency of the protocol. We aim for protocols for which the next symbol can be computed in at most polynomial time in n_0 (assuming a black-box access to π_0, as the noiseless protocol by itself may be inefficient).

1.4 Parameters that make a difference

When defining the setting, several variables come into play. Many times, these seemingly meaningless tweaks make a substantial difference in the capabilities of coding schemes.

- **The Channel.** We assume a channel Ch over alphabet Σ is a causal function $\mathsf{Ch} : \Sigma \to \Sigma$, where each instantiation of the function may (implicitly) depend on all previous channel instantiations. The channel is characterized by the following parameters:

  - *Alphabet size.* While the channel is always assumed to have a fixed-size alphabet (which is independent of the function we compute), the specific size of the alphabet may affect the noise resilience of the coding scheme. It is common that the alphabet in use is determined as a function of the noise resilience, and as the resilience approaches the limit, the alphabet size increases. The most difficult setting is thus when the alphabet is set to be of size 2, that is, a binary channel.

– *Noise (type).* In the standard noisy channel, the noise may substitute an input symbol $\sigma \in \Sigma$ into any other symbol $\sigma' \in \Sigma$. A different type of noisy channel is the *erasure channel*, $\mathsf{Ch} : \Sigma \to \Sigma \cup \{\perp\}$, in which the input symbol is either delivered without any disturbance or turns into a special erasure mark $\perp \notin \Sigma$. In the more general *channel with insertions and deletions*, the channel $\mathsf{Ch} : (\Sigma \cup \emptyset) \to (\Sigma \cup \emptyset)$ is allowed to completely remove transmitted symbols (so that the receiver will not be aware a symbol was sent to it) or inject new symbols (so that a symbol arrives at the receiver without the sender sending it).

– *Noise (power).* The power of the noise can be classified into three main categories:

  (i) *adversarial noise*, where the adversarial channel is considered to be all powerful, and the only restriction on the noise is the total amount of corruptions the channel is allowed to make. As mentioned before, the corruption budget is usually given as a fraction of the total communication. That is, an adversarial noise rate of ε means that at most εn symbols can be corrupted.

  (ii) *computationally efficient noise*, where the adversarial channel is considered to be computationally limited, in addition to being restricted to corrupting an ε-fraction of the transmissions.

  (iii) *random noise*, where each symbol is disturbed with some fixed probability, independently of previous transmissions, that is, a memoryless channel. The prominent example is the *binary symmetric channel* with flipping probability $\varepsilon < 1/2$, denoted BSC_ε, where each bit goes through undisturbed with probability $1 - \varepsilon$ or gets flipped with probability ε, independently per transmission. Note that random noise is a special type of a computationally bounded noise (yet, there is no limit on the fraction of corrupted transmissions).

– *Feedback.* In the case of channels with feedback, we assume the sender instantly learns the symbol received at the other

side via a separate *noiseless* feedback channel. The feedback channel is not counted towards the communication complexity nor the corruption budget.

- **Order of Speaking.** The order of speaking, both in the noiseless protocol π_0 and in the simulation π, may have a great effect on the properties of the coding scheme—specifically, its rate and noise resilience. We distinguish the case of *fixed order of speaking* in which the party that sends a symbol at the i-th round is predetermined and independent of the inputs of the protocol and the observed noise, and the case of *adaptive order of speaking*, where each party independently determines whether to send a symbol at the next round according to its input and received transcript.

- **Shared Randomness.** Whether or not the parties begin the computation with a random string unknown to the adversarial channel may have an effect either on the maximal obtainable rate of the coding scheme or on its noise resilience. In a sense, having a shared randomness has a certain effect of converting adversarial noise into a random one [64]. Practically, the parties can use shared randomness to better detect corruptions, reducing bit flips into erasure marks (with high probability).

- **Number of Parties.** The above interactive setting can be augmented to include the multiparty case, where m parties $\{p_i\}$ hold a private input $\{x_i\}$, respectively, and wish to compute some function $f(x_1, \ldots, x_m)$ while communicating over a noisy network. The network's topology has an important effect on the coding scheme's properties.

1.5 Organization

We begin in **Section 2** by discussing coding for interactive communication in the presence of adversarial noise. We discuss the maximal noise that can be tolerated and show a scheme with an optimal resilience. To that end we discuss two main techniques for coding (tree codes and

rewind-if-error) that will be used throughout the manuscript. In **Section 3** we discuss (computationally) efficient constructions of coding schemes. We begin with the random noise setting and show several relaxations to tree codes that yield an efficient coding scheme. We then turn to the adversarial noise setting and show that optimal noise resilience can be achieved by an efficient coding scheme, using list-decoding techniques.

An advanced family of coding schemes adaptively change their structure (i.e., their length and the order in which the parties speak) according to the observed noise. In **Section 4** we discuss two models for adaptive protocols and show that a better noise resilience can be achieved in each of these more general models. The rate of coding schemes is discussed in **Section 5**. In particular, we show coding schemes in the random noise setting, whose rate approaches one as the noise probability approaches zero. We also discuss the maximal possible rate, that is, the capacity of interactive communication over memoryless noisy channels.

Section 6 explores other types of noisy channels. In particular, we survey coding schemes for channels that allow a noiseless feedback, erasure channels, channels with insertions and deletions, and quantum channels. In **Section 7** we extend the discussion to the multiparty case and discuss how to code distributed protocols performed over a noisy network, both in the random and the adversarial noise settings. When more than two parties perform a distributed computation, it is important to define whether messages pass in a synchronous or an asynchronous way. We survey coding schemes in both message-passing models and compare their properties. Applications of coding for interactive communications, and related topics that build on the techniques of interactive coding are presented in **Section 8**.

Finally, in **Appendix B** we provide several tables that summarize the coding schemes discussed in this manuscript and compare their properties.

References

[1] Shweta Agrawal, Ran Gelles, and Amit Sahai. Adaptive protocols for interactive communication. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 595–599, July 2016. See also longer version in arXiv:1312.4182 (cs.DS).

[2] Noga Alon, Mark Braverman, Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Reliable communication over highly connected noisy networks. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, PODC '16, pages 165–173, New York, NY, USA, 2016. ACM.

[3] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures & Algorithms*, 3(3):289–304, 1992.

[4] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., Hoboken, NJ, 3rd edition, 2008.

[5] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[6] Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K. Wootters. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Phys. Rev. Lett.*, 70:1895–1899, March 1993.

[7] Elwyn R. Berlekamp. *Block Coding with Noiseless Feedback*. PhD thesis, Massachusetts Institute of Technology, 1964.

[8] M. R. Best, A. E. Brouwer, F. Jessie MacWilliams, Andrew M. Odlyzko, and Niel J. A. Sloane. Bounds for binary codes of length less than 25. *Information Theory, IEEE Transactions on*, 24(1):81–93, January 1978.

[9] Allison Bishop and Yevgeniy Dodis. Interactive coding for interactive proofs. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography: 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II*, pages 352–366, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[10] Zvika Brakerski and Yael T. Kalai. Efficient interactive coding against adversarial noise. *Foundations of Computer Science (FOCS), IEEE Annual Symposium on*, pages 160–166, 2012.

[11] Zvika Brakerski, Yael T. Kalai, and Moni Naor. Fast interactive coding against adversarial noise. *J. ACM*, 61(6):35:1–35:30, December 2014.

[12] Zvika Brakerski and Moni Naor. Fast algorithms for interactive coding. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 443–456, 2013.

[13] Gilles Brassard, Ashwin Nayak, Alain Tapp, Dave Touchette, and Falk Unger. Noisy interactive quantum communication. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, FOCS, pages 296–305, Oct 2014.

[14] Mark Braverman. Towards deterministic tree code constructions. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 161–167. ACM, 2012.

[15] Mark Braverman and Klim Efremenko. List and unique coding for interactive communication in the presence of adversarial noise. In *Foundations of Computer Science (FOCS), IEEE 55th Annual Symposium on*, pages 236–245, 2014.

[16] Mark Braverman, Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Constant-rate coding for multiparty interactive communication is impossible. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 999–1010, New York, NY, USA, 2016. ACM.

[17] Mark Braverman, Ran Gelles, Jieming Mao, and Rafail Ostrovsky. Coding for Interactive Communication Correcting Insertions and Deletions. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 61:1–61:14, Dagstuhl, Germany, 2016.

[18] Mark Braverman and Anup Rao. Towards coding for maximum errors in interactive communication. In *STOC '11: Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, pages 159–166, 2011.

[19] Mark Braverman and Anup Rao. Toward coding for maximum errors in interactive communication. *Information Theory, IEEE Transactions on*, 60(11):7248–7255, November 2014.

[20] Keren Censor-Hillel, Ran Gelles, and Bernhard Haeupler. Making asynchronous distributed computations robust to noise. Preprint arXiv:1702.07403 (cs.DS), 2017.

[21] What is the average length of a game of chess?, 2013. Chess StackExchange. [Online:] `https://chess.stackexchange.com/questions/2506/what-is-the-average-length-of-a-game-of-chess`.

[22] Kai-Min Chung, Rafael Pass, and Sidharth Telang. Knowledge-preserving interactive coding. In *Proceedings of the 54th annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 449–458, 2013.

[23] Richard Cleve and Harry Buhrman. Substituting quantum entanglement for communication. *Phys. Rev. A*, 56:1201–1204, August 1997.

[24] Claude Crépeau and Joe Kilian. Achieving oblivious transfer using weakened security assumptions. In *Foundations of Computer Science, 29th Annual Symposium on*, pages 42–52, 1988.

[25] Ivan Damgård, Serge Fehr, Kirill Morozov, and Louis Salvail. Unfair noisy channels and oblivious transfer. In Moni Naor, editor, *Theory of Cryptography*, volume 2951 of *Lecture Notes in Computer Science*, pages 355–373. Springer Berlin Heidelberg, 2004.

[26] Marcel Kenji de Carli Silva, Nicholas J. A. Harvey, and Cristiane M. Sato. Sparse sums of positive semidefinite matrices. Preprint arXiv:1107.0088, 2011.

[27] Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Maximal noise in interactive communication over erasure channels and channels with feedback. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, ITCS '15, pages 11–20. ACM, 2015.

[28] 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, Aug 2016.

[29] Klim Efremenko, Gillat Kol, and Raghuvansh Saxena. Interactive coding over the noisy broadcast channel. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:93, 2017.

[30] Abbas El Gamal. Open problems presented at the 1984 workshop on specific problems in communication and computation sponsored by bell communication research. In Thomas M. Cover and B. Gopinath, editors, *Open problems in communication and computation*. Springer-Verlag New York, Inc., 1987.

[31] Peter Elias. List decoding for noisy channels. Technical Report 335, Research Laboratory of Electronics, Massachusetts Institute of Technology, 1957. Reprinted from the 1957 IRE WESCON convention record Part 2.

[32] Paul Erdős and László Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. *Infinite and finite sets*, 10:609–627, 1975.

[33] G. David Forney. Concatenated codes. Technical Report 440, Massachusetts Institute of Technology. Research Laboratory of Electronics, 1965.

[34] Matthew Franklin, Ran Gelles, Rafail Ostrovsky, and Leonard J. Schulman. Optimal coding for streaming authentication and interactive communication. In *Advances in Cryptology – CRYPTO 2013*, volume 8043 of *Lecture Notes in Computer Science*, pages 258–276. Springer Berlin Heidelberg, 2013.

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

[36] Robert G. Gallager. *Information Theory and Reliable Communication*, volume 2. Springer, 1968.

[37] Robert G. Gallager. Finding parity in a simple broadcast network. *Information Theory, IEEE Transactions on*, 34(2):176–180, Mar 1988.

[38] Ran Gelles and Bernhard Haeupler. Capacity of interactive communication over erasure channels and channels with feedback. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, pages 1296–1311, 2015.

[39] Ran Gelles, Bernhard Haeupler, Gillat Kol, Noga Ron-Zewi, and Avi Wigderson. Towards optimal deterministic coding for interactive communication. In *Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 1922–1936, Philadelphia, PA, USA, 2016. Society for Industrial and Applied Mathematics.

[40] Ran Gelles and Yael T. Kalai. Constant-rate interactive coding is impossible, even in constant-degree networks. In *Proceedings of the 8th Conference on Innovations in Theoretical Computer Science, ITCS '17*, 2017.

[41] Ran Gelles, Ankur Moitra, and Amit Sahai. Efficient and explicit coding for interactive communication. In *Foundations of Computer Science (FOCS), IEEE 52nd Annual Symposium on*, pages 768–777, 2011.

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

[43] Ran Gelles and Amit Sahai. Potent tree codes and their applications: Coding for interactive communication, revisited. Preprint arXiv:1104.0739 (cs.DS), 2011.

[44] Ran Gelles, Amit Sahai, and Akshay Wadia. Private interactive communication across an adversarial channel. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, ITCS '14, pages 135–144. ACM, 2014.

[45] Mohsen Ghaffari and Bernhard Haeupler. Optimal error rates for interactive coding II: Efficiency and list decoding. In *Foundations of Computer Science (FOCS), IEEE 55th Annual Symposium on*, pages 394–403, 2014.

[46] Mohsen Ghaffari, Bernhard Haeupler, and Madhu Sudan. Optimal error rates for interactive coding I: Adaptivity and other settings. In *STOC '14: Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 794–803, 2014.

[47] E. N. Gilbert, F. J. MacWilliams, and N. J. A. Sloane. Codes which detect deception. *Bell System Technical Journal*, 53(3):405–424, March 1974.

[48] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, pages 218–229. ACM, 1987.

[49] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[50] Navin Goyal, Guy Kindler, and Michael Saks. Lower bounds for the noisy broadcast problem. *SIAM Journal on Computing*, 37(6):1806–1841, 2008.

[51] Bernhard Haeupler. Interactive channel capacity revisited. In *Foundations of Computer Science (FOCS), IEEE 55th Annual Symposium on*, pages 226–235, 2014.

[52] Bernhard Haeupler and Ameya Velingker. Bridging the capacity gap between interactive and one-way communication. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '17, pages 2123–2142. Society for Industrial and Applied Mathematics, 2017.

[53] Richard W. Hamming. Error detecting and error correcting codes. *Bell System technical journal*, 29(2):147–160, April 1950.

[54] Eric Holcomb. So how many chess board positions are there? [online]: `www.nwchess.com/articles/misc/Chess_Board_Positions_article.pdf`.

[55] William M. Hoza and Leonard J. Schulman. The adversarial noise threshold for distributed protocols. In *Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 240–258, Philadelphia, PA, USA, 2016. Society for Industrial and Applied Mathematics.

[56] Abhishek Jain, Yael T. Kalai, and Allison B. Lewko. Interactive coding for multiparty protocols. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, ITCS '15, pages 1–10. ACM, 2015.

[57] Yael T. Kalai, Allison B. Lewko, and Anup Rao. Formulas resilient to short-circuit errors. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 490–499, 2012.

[58] Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM Journal on Discrete Mathematics*, 3(2):255–265, 1990.

[59] Gillat Kol and Ran Raz. Interactive channel capacity. In *STOC '13: Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 715–724, 2013.

[60] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[61] F.T. Leighton, Bruce M. Maggs, and Satish B. Rao. Packet routing and job-shop scheduling in O(congestion + dilation) steps. *Combinatorica*, 14(2):167–186, June 1994.

[62] Allison Lewko and Ellen Vitercik. Balancing communication for multiparty interactive coding. Preprint arXiv:1503.06381, 2015.

[63] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, June 1995.

[64] Richard Lipton. A new approach to information theory. In Patrice Enjalbert, Ernst Mayr, and Klaus Wagner, editors, *STACS '94*, volume 775 of *Lecture Notes in Computer Science*, pages 699–708. Springer, 1994.

[65] Ankur Moitra. Efficiently coding for interactive communication. *Electronic Colloquium on Computational Complexity (ECCC)*, 2011. TR11-042.

[66] Cristopher Moore and Leonard J. Schulman. Tree codes and a conjecture on exponential sums. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, ITCS '14, pages 145–154. ACM, 2014.

[67] Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM Journal on Computing*, 22(4):838–856, 1993.

[68] Rafail Ostrovsky, Yuval Rabani, and Leonard J. Schulman. Error-correcting codes for automatic control. *Information Theory, IEEE Transactions on*, 55(7):2931–2941, July 2009.

[69] Denis Pankratov. On the power of feedback in interactive channels. [Online:] `http://people.cs.uchicago.edu/~pankratov/papers/feedback.pdf`, 2013.

[70] Marcin Peczarski. An improvement of the tree code construction. *Information Processing Letters*, 99(3):92–95, August 2006.

[71] Pavel Pudlák. Linear tree codes and the problem of explicit constructions. *Linear Algebra and its Applications*, 490:124–144, 2016.

[72] Michael O. Rabin. How to exchange secrets with oblivious transfer. IACR Cryptology ePrint Archive, 2005. Originally published as: Technical Report TR-81, Aiken Computation Lab, Harvard University, 1981.

[73] Sridhar Rajagopalan and Leonard J. Schulman. A coding theorem for distributed computation. In *STOC '94: Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, pages 790–799. ACM, 1994.

[74] Leonard J. Schulman. Communication on noisy channels: A coding theorem for computation. *Foundations of Computer Science, Annual IEEE Symposium on*, pages 724–733, 1992.

[75] Leonard J. Schulman. Deterministic coding for interactive communication. In *STOC '93: Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing*, pages 747–756, 1993.

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

[77] Leonard J. Schulman. A postcript to "coding for interactive communication". [Online:] `http://www.cs.caltech.edu/~schulman/Papers/intercodingpostscript.txt`, 2003.

[78] Adi Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, October 1992.

[79] Claude E. Shannon. The zero error capacity of a noisy channel. *Information Theory, IRE Transactions on*, 2(3):8–19, September 1956.

[80] Claude E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001. Originally appeared in *Bell System Tech. J.* 27:379–423, 623–656, 1948.

[81] Alexander A. Sherstov and Pei Wu. Optimal interactive coding for insertions, deletions, and substitutions. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:79, 2017.

[82] John M. Wozencraft. List decoding. Technical report, Research Laboratory of Electronics, Massachusetts Institute of Technology. Quarterly Progress Report, no. 48, 1958.

[83] Andrew C.-C. Yao. Some complexity questions related to distributive computing(preliminary report). In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, STOC '79, pages 209–213, 1979.

[84] Andrew C.-C. Yao. Quantum circuit complexity. In *Foundations of Computer Science, 1993. Proceedings., 34th Annual Symposium on*, pages 352–361, 1993.