Communication Complexity: Lecture 1

Nathan Harms

January 9, 2026

1 Introduction

Communication is inherent in any computation. When performing any calculation, information must flow from one place to another: across wires in a circuit, between machines in a data center, across internet connections, even between nature and scientists trying to learn about it. As a consequence, understanding the amount of communication required to solve problems is fundamental to understanding computation, and has become central to theoretical computer science.

The study of communication complexity was initiated by Yao in 1979 [Yao79].

Definition 1.1 (Communication Problem)**.** A *communication problem* is a function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ that takes two inputs.

Let's begin with a very informal definition of a communication protocol. We will formalize it later.

Definition 1.2 (Communication Protocol (Informal))**.** For a communication problem $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, two players Alice and Bob agree in advance on a communication protocol, which proceeds as follows. Alice receives an input $x \in \mathcal{X}$, Bob receives an input $y \in \mathcal{Y}$, and they cooperate to compute $f(x, y)$ by taking turns sending messages (binary strings) to each other according to the protocol.
The cost of the protocol on inputs (x, y) is the total number of bits communicated between Alice and Bob to compute $f(x, y)$. The cost of the protocol is

$$\max_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \text{cost of protocol on inputs } (x, y).$$

The cost of a *problem* f is denoted $\mathsf{D}(f)$, and it is the minimum cost of any protocol computing f.

Here is a problem that has very low communication cost:

Example 1.3 (Parity)**.** Consider the function $\mathsf{parity} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ defined as

$$\mathsf{parity}(x, y) := \begin{cases} 0 & \text{if number of 1s in } xy \text{ is even} \\ 1 & \text{if number of 1s in } xy \text{ is odd} \end{cases}$$

Here is a protocol that solves this problem. Alice receives x, Bob receives y. Alice sends $|x| \mod 2$ (the number of 1s in x mod 2) and Bob sends $|y| \mod 2$ back. They output $|x| + |y| \mod 2$. The cost of the protocol is 2.

Here is a problem with very high communication cost:

Example 1.4 (Equality)**.** The function $\textsc{Equality}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is defined as

$$\mathsf{Equality}(x,y) \coloneqq \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$$

Here is a protocol that solves this problem. Alice receives x, Bob receives y. Alice sends x itself. Bob sends back 1 if $x = y$ and 0 otherwise. The cost of this protocol is $n + 1$.

Later, we will prove that this protocol for $\textsc{Equality}$ is optimal.

Theorem 1.5. *The cost of* $\textsc{Equality}$ *on n-bit inputs is* $n + 1$.

2 Selected Applications of Communication Complexity

Before making the above definitions formal, let's jump to some examples of communication complexity, to get a taste for the variety of areas where it is useful.

2.1 Time and Space Tradeoffs for Turing Machines

Our first application can be understood using only our informal definition. In this theorem, we consider multi-tape Turing machines, with a read-only input tape, any fixed number of work tapes, and a binary alphabet. The space complexity $S(n)$ of a Turing machine with k work tapes is

$$\max_{x \in \{0,1\}^n} \sum_{i=1}^{k} \ell_i(x)$$

where $\ell_i(x)$ is the largest index on the i^{th} work tape that the i^{th} tape head reaches during the execution on x. The time complexity $T(n)$ is the maximum number of time steps executed by the machine on any input $x \in \{0,1\}^n$ (or ∞ if there is $x \in \{0,1\}^n$ where it never halts).

Theorem 2.1 ([KN96])**.** *Let $L \coloneqq \{ww^R | w \in \{0,1\}^*\}$ be the language of palindromes (where w^R is w in reverse order). Let M be any multi-tape Turing Machine deciding L in time $T(n)$ and space $S(n)$ on n-bit inputs. Then*

$$T(4n) \cdot S(4n) = \Omega(n^2).$$

Proof sketch. Let M be any Turing machine deciding L. We will design a communication protocol for the $\textsc{Equality}$ problem. For any n, Alice and Bob perform the following protocol. Given $x, y \in \{0,1\}^n$:

1. Alice and Bob each simulate M on input $x0^{2n}y^R$, where y^R is y in reverse order, as follows. The input and work tape heads both start at position 0. Initialize the current player to Alice. At each time t:

   (a) If the current state of M is a halting state, halt and output the decision.

(b) If the current player is Alice, the input tape head is guaranteed to be in the prefix $x0^{2n}$, which Alice knows. Alice simulates step t of M. If the input tape head stays in $x0^{2n}$ then continue without communication. Otherwise the tape head moves to the suffix y^R; in this case, Alice sends the contents of the work tape to Bob, together with the new state of M, and Bob becomes the current player.

(c) If the current player is Bob, the input tape head is guaranteed to be in the suffix $0^{2n}y^R$, which Bob knows. Bob simulates step t of M. If the tape head stays in the suffix $0^{2n}y^R$ then continue without communication. Otherwise the tape head moves to the prefix x; in this case, Bob sends the contents of the work tape to Alice, together with the new state of M, and Alice becomes the current player.

The protocol computes EQUALITY because $x = y$ if and only if $x0^{2n}y^R \in L$. The cost of the protocol can be calculated as follows:

1. Each message sent by Alice or Bob contains the contents of the work tape, which is at most $S(4n)$ bits, together with the current state of M, which is $O(1)$ bits.

2. The number of messages sent in the protocol is at most $T(4n)/2n$, because the input tape head must cross the center 0^{2n} between each round of communication, which takes time at least $2n$.

Therefore

$$\mathsf{D}(\textsc{Equality}) = O\left(S(4n) \cdot T(4n)/n\right).$$

From Theorem 1.5, we know $\mathsf{D}(\textsc{Equality}) = \Omega(n)$, which concludes the proof. ■

2.2 Circuits

Theorem 2.2. *The smallest depth of a circuit computing any function $f\colon \{0,1\}^n \to \{0,1\}$ is equal to the cost of the optimal communication protocol of the* Karchmer-Wigderson Game *defined by f.*

And many more...

2.3 Learning Theory

Theorem 2.3 ([PS86])**.** *A concept class $\mathcal{H}$ can be represented as $\Theta(d)$-dimensional halfspaces if and only if its matrix representation has unbounded-error randomized communication cost* $\log(d) + O(1)$.

Theorem 2.4 ([FX14])**.** *The sample complexity of PAC learning with pure differential privacy is equivalent to one-way public-coin randomized communication cost.*

And many more...

2.4 Property Testing

Theorem 2.5 ([BBM12]). *Testing whether an unknown function* $f\colon \{0,1\}^n \to \{0,1\}$ *is either the parity function on some subset of* k *bits, or is* $1/2$*-far from all such functions, requires* $\Omega(k)$ *queries of the form* $x \mapsto f(x)$*.*

2.5 Graph Theory

Theorem 2.6 ([EHZ24]). *For all* $N \in \mathbb{N}$*, there is a graph on* $\mathrm{poly}(N)$ *vertices which contains all* N*-vertex subgraphs of the hypercube as subgraphs.*

And many more...

3 Formal Definition and Communication Matrices

The formal definition of a communication protocol is annoyingly tedious:

Definition 3.1 (Deterministic Communication Complexity)**.** Let $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ be a function on any (finite) domains $\mathcal{X}, \mathcal{Y}$. A *communication protocol* Π for f consists of:

1. A binary tree T with inner nodes $V(T)$ and leaves $L(T)$;
2. A function player: $V(T) \to \{A, B\}$ (which indicates whether it is Alice's or Bob's turn to speak at each node);
3. A set of functions $m_{v,p}\colon \mathcal{X} \cup \mathcal{Y} \to \{0,1\}$ defined for each inner node $v \in V(T)$ and player $p \in \{A, B\}$;
4. A set of labels $\ell : L(T) \to \{0,1\}$ which indicate the output of the protocol at each leaf.

On input $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the protocol proceeds as follows. The current node variable v is initiated at the root of T. In each round:

- If v has reached a leaf, output $\ell(v)$ and terminate the protocol.
- If $\mathrm{player}(v) = A$ (i.e. it is Alice's turn to speak), then v moves to its left child if $m_{v,A}(x) = 0$ and to the right child if $m_{v,A}(x) = 0$.
- If it is Bob's turn to speak, then do the same, with y as the input instead.

We write $\Pi(x, y)$ for the output of the protocol on input x, y, and the output is required to satisfy $\Pi(x, y) = f(x, y)$ for all x, y.
The *cost* of a protocol Π is the depth of the tree T. The cost $\mathsf{D}(f)$ of a function is the minimum cost of any protocol computing f.

Communication problems are equivalent to matrices, and indeed we will usually think of them as matrices.

Definition 3.2 (Communication Matrix)**.** Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets, and let $f\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a communication problem. Then the *communication matrix* of f is the matrix $M_f \in \mathbb{R}^{\mathcal{X} \times \mathcal{Y}}$ with rows indexed by $\mathcal{X}$ and columns indexed by $\mathcal{Y}$, defined by

$$\forall (x, y) \in \mathcal{X} \times \mathcal{Y}\colon \qquad M_f(x, y) := f(x, y).$$

Example 3.3 (Parity)**.** Recall the Parity problem $\mathrm{par}(x, y) := (|x| + |y| \mod 2)$. The communication matrix looks like a checkerboard.

Example 3.4. Recall the Equality problem $\textsc{Eq}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ where $\textsc{Eq}(x, y) = 1$ iff $x = y$. The communication matrix of Equality is the identity matrix.

It can be helpful to think of every boolean-valued communication problem $f\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ as the adjacency relation on some bipartite graph G with vertices $\mathcal{X} \cup \mathcal{Y}$, as follows. Let G be the bipartite graph with edge set

$$\{(x, y) \mid f(x, y) = 1\}.$$

Therefore $f(x, y) = \textsc{Adj}(x, y)$ where $\textsc{Adj}(x, y)$ is the adjacency function for the graph G. Note that the communication matrix M_f is the adjacency matrix of G.

4 Open Problems: Computing Communication Complexity

Open Problem 1. *Are there constants $c \geq 1, L_1, L_2$ such that there exists a polynomial time algorithm for the following problem? Given a matrix $M \in \{0,1\}^{N \times N}$, output a number k such that $L_1 \cdot k \leq \mathsf{D}(M) \leq L_2 \cdot k^c$.*

The most famous conjecture in communication complexity is the *log-rank conjecture*, which we will look at next time. This conjecture implies:

Conjecture 1 (Consequence of Log-Rank Conjecture)**.** *The answer to the above problem is YES.*

There is some recent progress on this problem: Hirahara, Ilango, & Loff [HIL25] showed last year that it is NP-hard to compute $\mathsf{D}(M)$. But their result leaves open the possibility that there is a polynomial time algorithm for computing $\mathsf{D}(M)$ up to additive $+1$ error! However, “we” do not believe there is one; indeed, we expect it to be NP-hard to approximate communication complexity up to any constant factor:

Open Problem 2. *Show that there is no polynomial-time algorithm which approximates $\mathsf{D}(M)$ up to a constant-factor, unless $\mathsf{P} = \mathsf{NP}$.*

5 Monochromatic Rectangles and Partition Number

A key concept in communication complexity is *monochromatic rectangles* (see e.g. the cover of the textbook).

Definition 5.1 (Rectangles)**.** A *combinatorial rectangle* is a set $R = X \times Y$ where X and Y are each sets, where the cartesian product $\times$ is defined by

$$X \times Y := \{(x, y) \mid x \in X, y \in Y.$$

Let M be a matrix with rows indexed by $[m]$ and columns indexed by $[n]$, and let $R = X \times Y$ be a rectangle with $X \subseteq [m]$ and $Y \subseteq [n]$. Let b be any value. Then R is *b-monochromatic* for M if

$$\forall (i, j) \in X \times Y : \qquad M(i, j) = b.$$

We say a rectangle R is *monochromatic* for M if there exists b such that R is b-monochromatic.

Let's see how monochromatic rectangles relate to communication protocols.

5.1 Properties of Protocols

Let $M \in \mathbb{R}^{\mathcal{X} \times \mathcal{Y}}$ be any matrix, let Π be any communication protocol computing M, and let T be its communication tree.

Observation 5.2. *For each input (x, y), there is a unique root-to-leaf path $p(x, y)$ that Π takes through T.*

We say input (x, y) *reaches* node v of T if v is in the path $p(x, y)$. For every node v of Π, define the set

$$R_v := \{(x, y) \in \mathcal{X} \times \mathcal{Y} \mid \Pi \text{ on } (x, y) \text{ reaches } v\}.$$

We may observe the following properties of protocols:

Proposition 5.3 (Properties of Protocols)**.** *Let M be a matrix with rows indexed by $[m]$ and columns indexed by $[n]$, and let Π be a communication protocol computing M. For each node v of Π:*

1. *R_v is a combinatorial rectangle;*
2. *If v is an inner node, let w_1 and w_2 be its children. R_v is a rectangle $R_v = X \times Y$. Then exactly one of these two statements holds:*
   (a) *There is $X' \subset X$ such that $R_{w_1} = X' \times Y$ and $R_{w_2} = (X \setminus X') \times Y$; or*
   (b) *There is $Y' \subset Y$ such that $R_{w_1} = X \times Y'$ and $R_{w_2} = X \times (Y \setminus Y')$.*
3. *Let $w \neq v$ be another node. Then*
   (a) *If w is an ancestor, then $R_v \subset R_w$.*
   (b) *If w is a descendent, then $R_w \subset R_v$.*
   (c) *Otherwise, $R_v \cap R_w = \varnothing$.*
4. *If v is a leaf node, then R_v is a monochromatic rectangle.*

Proof sketch. Properties 1 and 2 hold by induction on the depth of node v (base case is the root node). Property 3 follows from property 2. Property 4 holds by property 1 together with the fact that Π computes M. ■

Theorem 5.4 (Protocol Balancing)**.** *Let M be any matrix and let Π be a protocol computing M. Suppose Π has ℓ leaves. Then there exists a protocol Π' computing M with cost $\leq 2\log_{3/2}(\ell)$.*

Proof. We did not have time for this in class, so it will be an exercise in the homework. ■

5.2 Partition Number

Definition 5.5. Let $M \in \{0,1\}^{m\times n}$. A *monochromatic rectangle partition* of M is a set $\mathcal{P} = \{R_1, \dots, R_k\}$ of disjoint rectangles R_i such that each rectangle is monochromatic in M. The *partition number* $\chi(M)$ is the minimum k such that there exists a monochromatic rectangle partition $\mathcal{P}$ of size $|\mathcal{P}| = k$.

Theorem 5.6. *For any $M \in \{0,1\}^{m\times n}$,*

$$\log(\chi(M)) \leq \mathsf{D}(M) \leq O(\log^2(\chi(M))).$$

Proof. For the lower bound, observe that the leaves of any protocol Π computing M correspond to a monochromatic partition. Since there are at most 2^d leaves in a protocol of depth d, we have

$$\chi(M) \leq 2^{\mathsf{D}(M)}.$$

Let us now prove the upper bound. We prove the claim by induction on the partition number $k = \chi(M)$. The base case is when $k = 1$, i.e. , M is monochromatic, and $\mathsf{D}(M) \leq 1$.

Now let $k = \chi(M) > 1$, and let $\mathcal{P} = \{R_1, \dots, R_k\}$ be a monochromatic partition of size k. Write $R_i = X_i \times Y_i$ for each $i \in [k]$.

Alice and Bob are given inputs (x, y). Call a rectangle R_i *x-good* if at most $3k/4$ rectangles R_j satisfy $X_j \cap X_i \neq \varnothing$, and *$x$-bad* otherwise. Similarly, call R_i *y-good* if at most $3k/4$ rectangles R_j satisfy $Y_j \cap Y_i \neq \varnothing$, and *$y$-bad* otherwise.

Claim 5.7. *For all inputs x, y, the unique rectangle $R_i \in \mathcal{P}$ that contains x, y is either x-good or y-good.*

Proof of claim. Suppose R_i is both x-bad and y-bad. Then $3k/4$ rectangles $R_j = X_j \times Y_j$ satisfy $X_j \cap X_i \neq \varnothing$, and $3k/4$ rectangles $R_j = X_j \times Y_j$ satisfy $Y_j \cap Y_i \neq \varnothing$. Then there exists $R_j \neq R_i$ such that both $X_j \cap X_i \neq \varnothing$ and $Y_j \cap Y_i \neq \varnothing$. But then R_j and R_i are not disjoint, a contradiction. ■

As a consequence of this claim, there must exist a rectangle $R_i \in \mathcal{P}$ containing (x, y) that is either x-good or y-good. Alice and Bob may therefore communicate as follows:

1. Alice sends one bit to Bob to indicate whether there is an x-good rectangle $R_i = X_i \times Y_i$ where $x \in X_i$. If so, she uses $\lceil \log k \rceil$ bits to identify an arbitrary x-good rectangle. Both players recurse on rows X_i of M.

2. If there is no x-good rectangle, then Bob uses $\lceil \log k \rceil$ bits to identify an arbitrary y-good rectangle $R_i = X_i \times Y_i$. Both players recurse on columns Y_i of M.

Call the new matrix in the recursion M'. In each case, M' has a monochromatic partition of size at most $3k/4$, because we may partition M' using only the rectangles in $\mathcal{P}$ that intersect the chosen rows or columns.

If the protocol runs for r rounds, then the remaining matrix has partition number at most $\left(\frac{3}{4}\right)^r \geq 1$, so it must terminate in at most $r = \log_{4/3}(\chi(M))$ rounds. In each round, at most $O(\log(\chi(M)))$ bits are transmitted between players. ∎

Recall Theorem 1.5 from earlier, which said that the communication cost of EQUALITY is $n+1$. We may now prove this:

Proof of Theorem 1.5. Let M be the communication matrix for $\textsc{Eq}_n$. We will show that $\chi_1(M) = 2^n$ (and $\chi_0(M) > 0$), so that

$$\mathsf{D}(\textsc{Eq}_n) \geq \log(\chi(M)) > \log(\chi_1(M)) = n.$$

Suppose for the sake of contradiction that $k := \chi_1(M) < 2^n$, so there are k disjoint 1-chromatic rectangles $R_1, \ldots, R_k$ that contain all 2^n diagonal entries of M. By the pigeonhole principle, there are two distinct diagonal entries (i,i) and (j,j) of M contained in a single 1-chromatic rectangle R_ℓ. Then R_ℓ also contains (i,j). But $M(i,j) = 0$ since $i \neq j$, which contradicts the fact that R_ℓ is 1-chromatic. ∎

References

[BBM12] Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. *Computational Complexity*, 21(2):311–358, 2012.

[EHZ24] Louis Esperet, Nathaniel Harms, and Viktor Zamaraev. Optimal adjacency labels for subgraphs of cartesian products. *SIAM Journal on Discrete Mathematics*, 38(3):2181–2193, 2024.

[FX14] Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. In *Conference on Learning Theory*, pages 1000–1019. PMLR, 2014.

[HIL25] Shuichi Hirahara, Rahul Ilango, and Bruno Loff. Communication complexity is NP-hard. *arXiv:2507.10426*, 2025.

[KN96] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1996.

[PS86] Ramamohan Paturi and Janos Simon. Probabilistic communication complexity. *Journal of Computer and System Sciences*, 33(1):106–123, 1986.

[Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *STOC*, 1979.