Two-party function computation on the reconciled data

Ivo Kubjas, Vitaly Skachek

Institute of Computer Science
University of Tartu
Tartu, Estonia

February 22, 2017

Problem idea

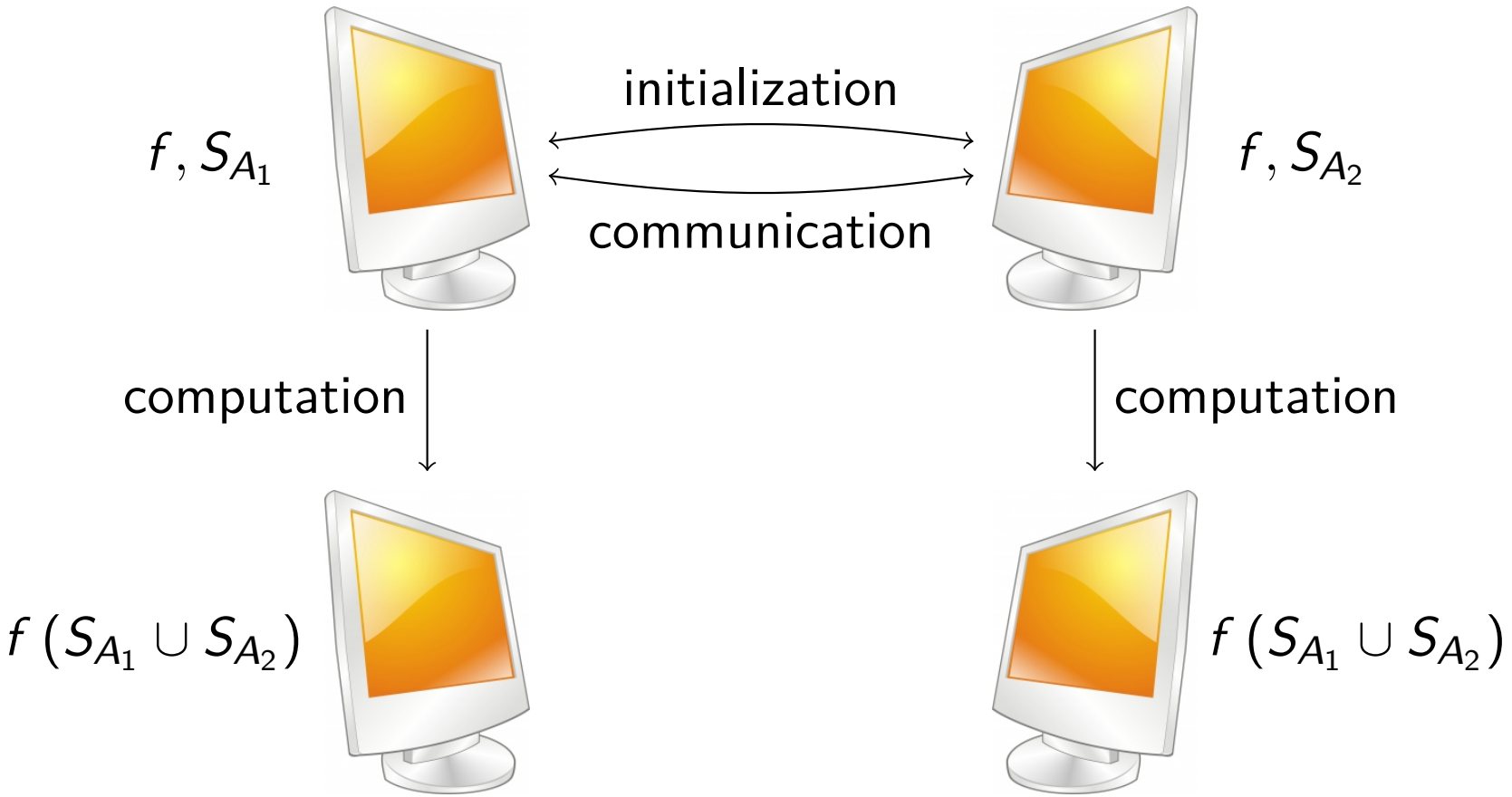


Figure: Cooperative function computation

Example 1.

There is a network with two wireless temperature measuring sensors. The sensors perform the measurements at fixed intervals. Occasionally, either of the sensors is turned off for maintenance.

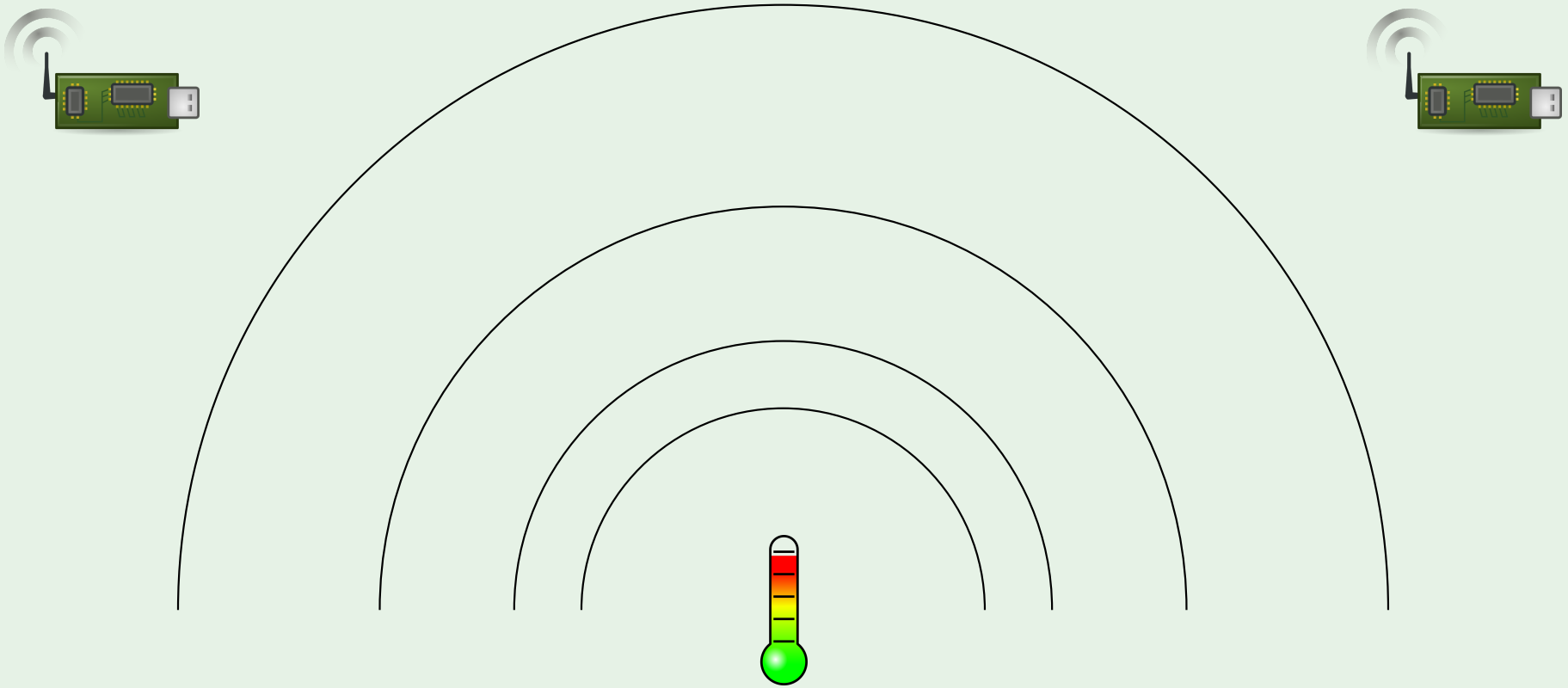


Figure: Sensor network

Example (cont.)

Example 2.

The goal is to obtain the average temperature during the time frame, omitting multiple measurements by both sensors.
We denote each measurement as (t, τ), where t is the time of the measurement and τ is the temperature. The sensors compute cooperatively the function

$$f(S) = \frac{\sum_{(t,\tau)\in S} \tau}{|S|},$$

where S_{A_1} are the measurements of the first sensor, S_{A_2} are the measurements of the second sensor and $S = S_{A_1} \cup S_{A_2}$.

1. Introduction
   - Motivating example

2. Deterministic communication complexity
   - Yao's model
   - Protocol trees
   - f-monochromatic rectangles
   - Fooling set method
   - Matrix rank method

3. Cooperative function computation
   - Set reconciliation
   - Problem statement
   - Bounds for sum

Yao's deterministic model

Let $M = \{0, \ldots, m\}$ and $N = \{0, \ldots, n\}$, and $f : M \times N \to \{0, 1\}$. Parties A and B have $i \in M$ and $j \in N$, respectively. They want to determine the value of $f(i, j)$ by alternately sending a single bit at a time.

Example 3.

Function $f(i, j) = (i + j) \pmod 2$ requires one bit of communication. Party A sends parity bit a of i to B. B determines $f(i, j) = (a + j) \pmod 2$.

Example 4.

Equality function $\mathrm{EQ} : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}$,

$$\mathrm{EQ}(i, j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

requires n bits of communication to determine f.

Yao's deterministic model (cont.)

Definition 5.

A protocol Π consists of Boolean functions

$$\{h_k(i; u_1, \ldots, u_{k-1}), l_k(j; v_1, \ldots, v_k) : k = 1, \ldots\},$$

where $a_k = h_k(i; b_1, \ldots, b_{k-1})$ computes the k-th bit sent by A and $b_k = l_k(j; a_1, \ldots, a_k)$ computes the k-th bit sent by B.

- The protocol terminates if either A or B has determined $f(i, j)$.
- There always exists a terminating protocol.

Yao's deterministic model (cont.)

Definition 6.

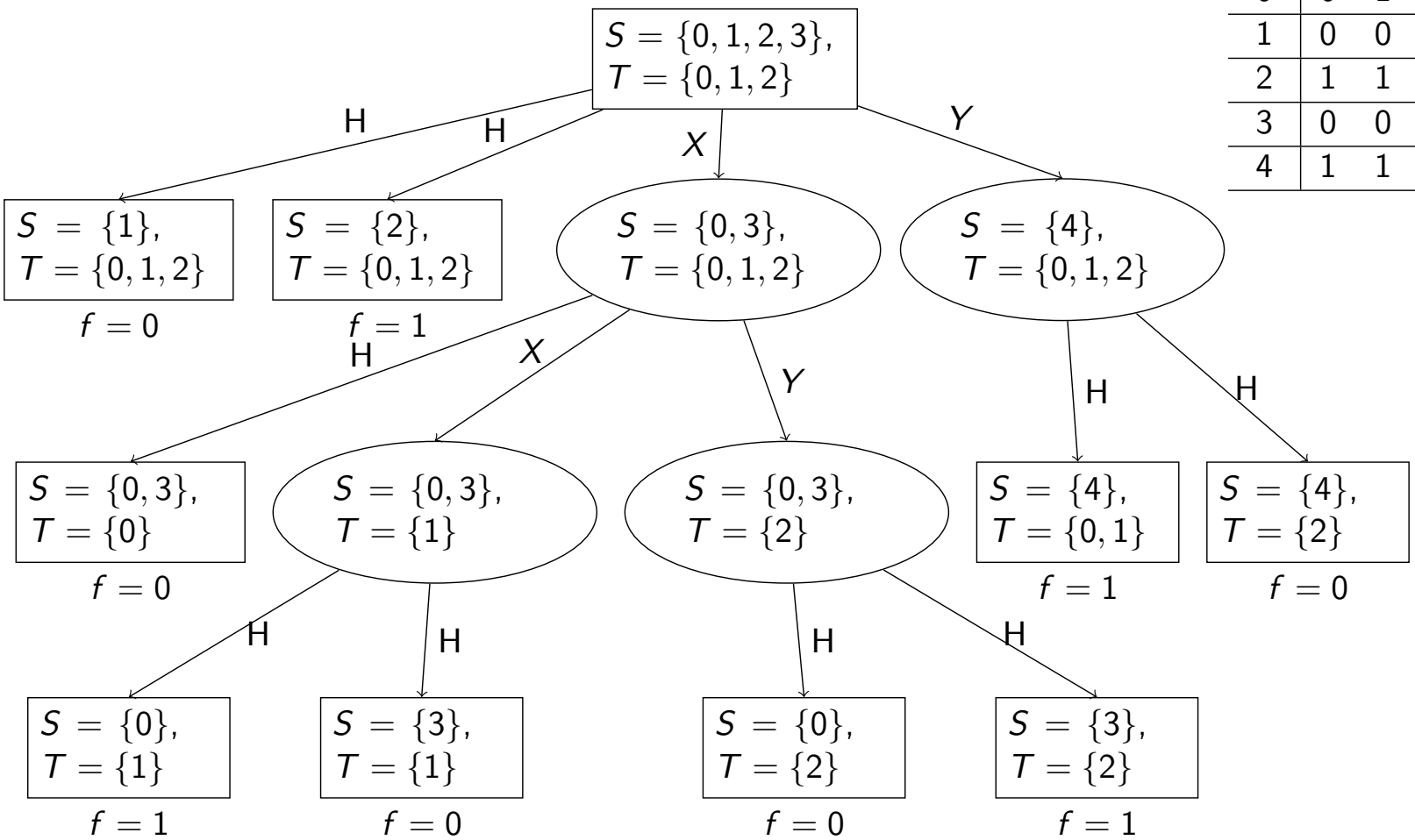
Cost of the protocol Π is the maximum number of transmitted bits for any $i \in N$ and $j \in M$.

Definition 7.

Deterministic communication complexity of function f is the minimum cost over all protocols Π that compute f. We denote communication complexity of f with $D(f)$.

Yao's deterministic model (alternative representation)

i/j	0	1	2
0	0	1	0
1	0	0	0
2	1	1	1
3	0	0	1
4	1	1	0



Yao's deterministic model (alternative representation II)

- Leafs $\{\ell\}$ partition the input set $M \times N$ into disjoint partitions $\{R_\ell\}$.
- A set $R = U \times V$ with $U \subset M$ and $V \subset N$ is a *(combinatorial) rectangle*.
- If f is fixed on R, then R is called f-monochromatic rectangle.

Example 8.

	000	001	010	011	100	101	110	111
000	0	1	1	0	1	0	0	0
001	1	0	0	0	0	0	0	1
010	1	0	0	0	1	0	0	0
011	0	0	1	0	0	0	0	1
100	1	0	0	0	1	0	0	1
101	1	1	1	0	0	0	1	1
110	0	0	0	0	1	0	0	0
111	0	1	1	0	1	1	0	1

f-monochromatic rectangles

Lemma 9.

Any protocol Π for a function f induces a partition of $M \times N$ into f-monochromatic rectangles. The number of rectangles is the number of leaves of Π.

Corollary 10.

If any partition of $M \times N$ into f-monochromatic rectangles requires at least t rectangles, then $D(f) \geq \log_2 t$.

Bounds on communication complexity (fooling set)

1. Construct a matrix $M_f = (f(i,j))$.
2. Pick all pairs (i,j) which belong to different partitions. Denote such (fooling) set S.

Lemma 11.

$R \subset M \times N$ is a rectangle if and only if

$$(x_1, y_1) \in R \wedge (x_2, y_2) \in R \Rightarrow (x_1, y_2) \in R.$$

3. From Corollary 10, $D(f) \geq \log_2 |S|$.
4. More generally, if μ is a probability distribution on $M \times N$, and if any f-monochromatic rectangle R has measure $\mu(R) \leq \delta$, then $D(f) \geq \log_2 1/\delta$.

Fooling set method

Example 12.

	000	001	010	011	100	101	110	111
000	0	1	1	0	1	0	0	0
001	1	0	0	0	0	0	0	1
010	1	0	0	0	1	0	0	0
011	0	0	1	0	0	0	0	1
100	1	0	0	0	1	0	0	1
101	1	1	1	0	0	0	1	1
110	0	0	0	0	1	0	0	0
111	0	1	1	0	1	1	0	1

Fooling set method

Example 13 (Example 4 continued).

Claimed that the communication complexity for equality function EQ is n bits. The corresponding matrix is:

$$\begin{pmatrix} & 0 & 1 & \dots & 2^n \\ 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 2^n & 0 & 0 & \dots & 1 \end{pmatrix}$$

The number of f-monochromatic rectangles is at least 2^n. Thus, the communication complexity $D(\mathrm{EQ}) \geq \log_2(2^n) = n$.

Bounds on communication complexity (matrix rank)

Let a $|M| \times |N|$-dimensional matrix $M_f = (f(i,j))$ be a matrix describing function f.

Definition 14.

$\mathrm{rank}(f)$ denotes the linear rank of the matrix M_f over the reals.

Lemma 15.

For any function $f : M \times N \to \{0,1\}$, $D(f) \geq \log_2 \mathrm{rank}(f)$.

1. Introduction
   - Motivating example

2. Deterministic communication complexity
   - Yao's model
   - Protocol trees
   - f-monochromatic rectangles
   - Fooling set method
   - Matrix rank method

3. Cooperative function computation
   - Set reconciliation
   - Problem statement
   - Bounds for sum

Set reconciliation problem

- Party A has set $S_A \subset \mathbb{F}^n$ and party B has set $S_B \subset \mathbb{F}^n$.
- Goal is to find $S_A \cup S_B$ with smallest communication complexity.

Lemma 16.

Let $S_A, S_B \subset X$ and denote $m = S_A \triangle S_B$. Then every set reconciliation algorithm has communication complexity at least mn.

- Bound achieving but computationally inefficient protocol by Minsky, Trachtenberg and Zippel in 2003.
- Asymptotically bound achieving and computationally efficient protocol by Goodrich and Mitzenmacher in 2011.

Cooperative function computation

- Party A has $S_A \subset \mathbb{F}^n$ and B has $S_B \subset \mathbb{F}^n$.
- Parties A and B want to cooperatively compute the function $\phi : \mathcal{P}(\mathbb{F}^n) \to V$.
- Generalization of set reconciliation: $\phi(S) = S$.
- Denoting $f(S_A, S_B) = \phi(S_A \cup S_B)$, then can easily apply methods for finding lower bounds.
- Naive approach: reconcile sets and then compute the value of the function.

Naive approach is not optimal

Example 17.

Assume that A and B are interested in computing $f(S_A, S_B) = \max\{S_A \cup S_B\}$, where all entries in $S_A \cup S_B$ are viewed as non-negative integer numbers in their binary representation. The following protocol requires only $2n$-bit communication.

1. The users A and B compute $x_A = \max\{S_A\}$ and $x_B = \max\{S_B\}$, respectively.
2. The users A and B exchange the values of x_A and x_B.
3. Each user computes $\max\{x_A, x_B\}$.

Similar protocols for other idempotent functions (minimum, logical OR, logical AND).

Lower bound for sum using fooling set

- Consider $\phi(S) = \sum_{x \in S} x$ over integers.

Theorem 18.

The number of bits communicated between A and B in any deterministic protocol Π that computes the sum function $\sum$ is at least $D(\sum) \geq 2^n + n - 1$.

	$\emptyset$	1	2	3	1, 2	1, 3	2, 3	1, 2, 3
1, 2, 3	6	6	6	6	6	6	6	6
2, 3	5	6	5	5	6	6	5	6
1, 3	4	4	6	4	6	4	6	6
1, 2	3	3	3	6	3	6	6	6
3	3	4	5	3	6	4	5	6
2	2	3	2	5	3	6	5	6
1	1	1	3	4	3	4	6	6
$\emptyset$	0	1	2	3	3	4	5	6

- Main diagonal 2^{2^n-1}
- Side diagonal 2^{2^n-2}
- Number of side diagonals $2^n - 1$
- At least 2^{2^n+n-2} rectangles
- Similarly to Lemma 10, communication complexity is $2^n + n - 1$.

Other bounds for sum

- Upper bound $2^n + 2n - 2$:
  - $2^n - 1$ bits to represent the set
  - $2n - 1$ bits to represent the sum
- Reducing set disjointness to sum:
  - Set disjointness function:

$$\mathrm{DISJ}(S_A, S_B) = \begin{cases} 1 & \text{if } S_A \cap S_B = \varnothing \\ 0 & \text{otherwise} \end{cases} .$$

  - Determine if sum on union equals sum on parties' sets.
  - Reduction overhead $2n$ bits.
  - Lower bound of $2^n + 1$ in literature for deterministic protocols.
  - Asymptotically tight bound $\Theta(2^n)$ for randomized protocols.

Upper bound using hash functions

- Reduction to randomized protocol using hash functions.
- Let $H \triangleq \mathbb{F}^k$ and $\mathcal{H} = \{h\}$ be a family of all hash functions $h : \mathbb{F}^n \to H$, such that

$$\forall K \in H, \forall h \in \mathcal{H} \ : \ |\{x \ : \ h(x) = K\}| = 2^{n-k} \ .$$

- Assume that $\mathcal{H}$ is ordered and can choose $h_0, h_1, \ldots \in \mathcal{H}$.

Upper bound using hash functions (cont.)

1: **procedure** PROTOCOL
2: **for** $i = 0$; true; $i = i + 1$ **do**
3: B sends the set $K_i = \{h_i(x) : x \in S_B\}$ to A
4: A creates empty set L_i
5: **for** $x \in S_A$ **do**
6: **if** $h_i(x) \notin K_i$ **then**
7: A adds x to L_i
8: **end if**
9: **end for**
10: **if** $|L_i| = d_A$ **then**
11: **break**
12: **end if**
13: **end for**
14: A sends $s = \sum_{x \in L_i} x$ to B
15: B computes $s' = s + \sum_{x \in S_B} x$
16: B sends s' to A
17: **end procedure**

Upper bound using hash functions (cont.)

- Number of bits sent at different steps:
  $t_0 = km_B, t_1 = 2n-1, t_2 = 2n-1$.
- Measure the probability of success (no collision) and failure:

$$p_a = \Pr[\text{accept}] = \Pr[|L_i| = d_A] = \left(1 - \frac{2^{n-k}-1}{2^n-1}\right)^{d_A}$$

$$p_n = \Pr[\text{not accept}] = 1 - p_a.$$

- Expected number of transmitted bits is
  $E[T_\infty] = km_B \left(1 - \frac{2^{n-k}-1}{2^n-1}\right)^{-d_A} + 4n - 2$.
- Taking $k = \log_2(\frac{d_A}{c})$, we minimize $E[T_\infty]$.
- Replacing $k = \log_2(\frac{d_A}{c})$, then $D(\sum) = O(m_B \cdot \log d_A + n)$.

Bounds for sum (overview)

Communication Complexity	Protocol Type	Comments
$\Theta(d \cdot n)$	Deterministic	Reconciliation first, difference size is d
$\geq 2^n + n - 1$	Deterministic	
$\leq 2^n + 2n - 2$	Deterministic	
$\geq 2^n - 2n + 1$	Deterministic	Reduction to set disjointness
$\Theta(2^n)$	Randomized	Reduction to set disjointness
$O(\kappa) + 4n$	Shared randomness	Reduction to finding the intersection, set sizes are κ
$O(\kappa) + 4n + O(\log n)$	Private randomness	Reduction to finding the intersection, set sizes are κ
$O(\kappa \cdot \log d_A + n)$	"Las Vegas" type	Set sizes are κ, $d_A = \lvert S_A \setminus S_B \rvert$