Trading Bit, Message, and Time Complexity of Distributed Algorithms

TIK Technical Report 339 with omitted proofs of the DISC'11 conference version

Johannes Schneider, Roger Wattenhofer

Computer Engineering and Networks Laboratory, ETH Zurich, 8092 Zurich, Switzerland

Abstract

We present tradeoffs between time complexity t, bit complexity b, and message complexity m. Two communication parties can exchange $\Theta(m \log(tb/m^2) + b)$ bits of information for $m < \sqrt{bt}$ and $\Theta(b)$ for $m \geq \sqrt{bt}$. This allows to derive lower bounds on the time complexity for distributed algorithms as we demonstrate for the MIS and the coloring problems. We reduce the bit-complexity of the state-of-the art $O(\Delta)$ coloring algorithm without changing its time and message complexity. We also give techniques for several problems that require a time increase of t^c (for an arbitrary constant c) to cut both bit and message complexity by $\Omega(\log t)$. This improves on the traditional time-coding technique which does not allow to cut message complexity.

1 Introduction

The efficiency of a distributed algorithm is assessed with at least one out of three classic distributed complexity measures: time complexity (number of rounds for synchronous algorithms), communication or bit complexity (total number of bits transmitted), and message complexity (total number of messages transmitted). Depending on the application, one or another measure might be more relevant. Generally speaking, time complexity has received most attention; but communication complexity (bandwidth constraints) or message complexity (accounting for message overhead) play a vital role as well. One cannot just ignore one of the measures, as there are tradeoffs: One may for instance sometimes cut down on time by exchanging larger messages. Alternatively, one may save messages and bits by communicating "silently". Two parties may for instance communicate for free by telephone by simply never picking up the phone, and instead letting the phone ring for a long time when transmitting a binary 1, and just a short time for a binary 0. A more sophisticated example for silent communication employs time-coding to communicate information through time. As illustration consider pulse-position modulation, as used in wireless and optical communication. A k-bit message can be dispersed over time by encoding the message with a single pulse in one of 2^k possible slots. Employing a single pulse within time t allows to communicate at most $\log t$ bits.[1] Reducing message complexity is harder in general, and in some cases impossible as there are dependencies between messages. In this paper, we identify mechanisms for symmetry breaking that cut both message and bit complexity by a factor of $\log t$, even though multiple messages cannot be combined into a single message through time-coding.

Although it is well-known that these dependencies exist, to the best of our knowledge the tradeoffs are not completely understood. A considerable amount of work deals with both message size and time complexity. These two measures give a (rough) bound on the bit complexity, e.g. time multiplied by message size as an upper bound. However, we show that for a given fixed bit complexity allowing many arbitrary small messages (i.e. consisting of 1 bit) compared to allowing only one large message might cause a drastic (up to exponential) gain in time. For some examples both the time and overall message complexity, i.e. the messages transmitted by all nodes, have been optimized, but the tradeoffs between all three have not been investigated to the best of our knowledge.

In the first part of the paper we answer questions like "If we can prolong an algorithm by a factor t in time and can increase the number of messages by a factor m, what is the effect on the bit complexity b?" We give a tight bound on the amount of information exchangeable between two nodes of $\Theta(m \log(tb/m^2) + b)$ bits for $m < \sqrt{bt}$ and $\Theta(b)$ for larger m. A bound on the (communicable) information together with a bound on the minimum required information that has to be exchanged to solve a problem yields bounds on the time-complexity. We derive such bounds for typical symmetry breaking problems, such as coloring and maximal independent sets. We show that for $t \in [2, n]$ any MIS and $O(\Delta)$ coloring algorithms using up to $c_0 \log n / \log t$ bits and messages for a constant c_0 require time t. In light of the state of the art upper bounds for unrestricted communication of $\mathrm{O}(\log n)$ using $O(\log n)$ bits for the MIS problem, and $O(\log^* n)$ for the $O(\Delta)$ coloring problem, our lower bound indicates that even a logarithmic factor of $\log t$ in the amount of transmittable bits can make more than an exponential difference in time.

In the second part, we identify two coding schemes, i.e. transformations, to gain a factor $\log t$ in bit as well as message complexity by a time increase of t^c that cannot be achieved with traditional time coding. We employ them to deterministic and randomized coloring and maximal independent set algorithms. Our techniques are applicable beyond these problems, e.g. for certain divide-and-conquer algorithms. We also improve the bit complexity for the fastest randomized $\mathrm{O}(\Delta + \log^{1+1/\log^* n} n)$ coloring for Δ being at least polylogarithmic, i.e. from $\mathrm{O}(\log \Delta \log n)$ to $\mathrm{O}(\log n \log \log n)$, while maintaining its time complexity.

2 Related Work

In [7] the notion of "bit rounds" was introduced in the context of a coloring algorithm, where a node must transmit either 0 or 1 in one bit round. This

[1] The amount of information that can be communicated follows directly from our bound.

bit round complexity is an interesting hybrid between time and bit complexity, particularly useful in systems where sending a single bit does not incorporate a significant protocol overhead. The paper [7] also states a lower bound of $\Omega(\log n)$ bit rounds to compute a coloring on a ring. In contrast, our bit-time complexity model is motivated by the fact that in practice time is frequently divided into slots, i.e. rounds, and nodes might not transmit at all in a slot or they might transmit more than a single bit. In some sense, the bit-time complexity model unifies algorithm complexity focusing on running time and communication complexity. For a survey on communication complexity see [11]. In a common set-up two (or more) parties A, B want to compute a function f that depends on values held by A and B and the goal is to derive bounds on the needed information. For example, [16] shows that for some network topologies (involving more than two nodes) reducing the number of allowed message exchanges by 1 can exponentially increase the time complexity. The amount of exchangeable information between two parties given a certain number of messages and time can be found in e.g. [17]. To the best of our knowledge, we are the first to extend the tradeoff to allow for a variable number of bits per message.

In [3] the communication complexity of breaking symmetry in rings and chains is investigated. In [4] the running time of MIS algorithms is investigated depending on the amount of advice (measured in bits) that nodes are given before the start of the algorithm. For a ring graph it is shown that $\Omega(n/\log^{(k)} n)$ bits of information are needed for any constant k to break the lower bound of $\Omega(\log^* n)$ [12]. At first, asking a global instance, knowing the exact topology of the graph, for advice seems to contradict the distributed approach. But the question regarding the amount of needed information to solve a task is interesting and valuable for enhancing the understanding of distributed algorithms. In particular, since some (aggregate) knowledge of the graph is often necessary for an efficient computation, e.g. the type of the graph.

There is a myriad of papers for different problems that consider two complexity measures. However, whereas many papers concentrate on time complexity and merely mention message size [14, 13, 8, 1, 2], others derive explicit tradeoffs [5, 15, 6].

In a paper by Métivier et al. [15] an algorithm for the MIS problem was stated running in time $O(\log n)$ with bit complexity $O(\log n)$ for general graphs. It improves on the bit complexity of the fastest algorithm [14]. Essentially, each node draws a random number in $[0, n]$ and is joined the MIS, if its number is the smallest. Our MIS algorithm trading time for bit/message complexity improves on [14] through a different technique. For the MIS problem arbitrary large messages do not allow for an arbitrary fast algorithm, i.e. in general graphs every algorithm requires at least $\Omega(\sqrt{\log n/\log\log n})$ or $\Omega(\log \Delta/\log\log \Delta)$ communication rounds for computing a MIS [9]. Interestingly, the opposite is true: An arbitrarily slow algorithm allows for constant message and bit complexity. The lower bound is achieved with a pseudo-symmetric graph, such that a node needs to get to know its neighborhood up to distance $\Omega(\sqrt{\log n/\log\log n})$ or $\Omega(\log \Delta/\log\log \Delta)$. For the coloring problem, [18] presented a technique, where nodes select multiple colors and keep any of them, if it is not also selected by a neighbor. The bit complexity for $\Delta + 1$ as well as $O(\Delta)$ algorithms is $O(\log \Delta \log n)$. The latter runs in time $O(\log^* n)$ for graphs with $\Delta \in \Omega(\log^{1+1/\log^* n} n)$. Recently, [19] used [18] to derive a $(1 - 1/O(\chi))(\Delta + 1)$ coloring, where χ denotes the chromatic number of the graph. Deterministic $\Delta + 1$ coloring algorithms [1, 8] are faster for graphs of sublogarithmic degrees, i.e. need $O(\Delta + \log^* n)$ time and might require a message exchange of size $O(\log n)$ in a communication round.

Apart from applications that rely only on pure time-coding, in [20] a data gathering protocol is implemented saving on energy and bandwidth by coding messages partly through time.

3 Model and Definitions

The communication network is modeled with a graph $G = (V, E)$. For each node, there exists a distinct communication channel (edge) to each neighbor. Initially, a node v only knows the number of neighbors $|N(v)|$ and has no other information about them. We use the message passing model, i.e. each node can exchange one distinct message with each of its neighbors in one synchronous communication round. Communication is error free. All nodes start executing the algorithm concurrently. The time complexity denotes the number of rounds until the last node terminates.

In a (vertex) coloring any two neighboring nodes u, v have a different color. A set $T \subseteq V$ is said to be independent in G if no two nodes $u, v \in T$ are neighbors. A set $S \subseteq V$ is a maximal independent set (MIS), if S is independent and there exists no independent superset $T \supset S$. In our algorithm a node remains *active* as long as it might still join the MIS, i.e. as long as it has no neighbor in the MIS and it is not in the MIS itself. *Inactive* nodes are removed from the graph G. For a node v its neighborhood $N^r(v)$ represents all active nodes within r hops of v (not including v itself). We use $N(v)$ for $N^1(v)$. The r hop neighborhood $N^r(v)$ including v is denoted by $N^r_+(v)$. The term "with high probability" abbreviated by w.h.p. denotes the number $1-1/n^c$ for an arbitrary constant c. The maximum degree is denoted by Δ and $\Delta_{N_+(v)}$ denotes the maximum degree of a node in $N_+(v)$, i.e. $\Delta_{N_+(v)} := \max_{u \in N_+(v)} d(u)$.

The bit complexity denotes the maximum sum of the number of bits transmitted over any edge during the execution of the algorithm, i.e. if an algorithm has time complexity t then the bit complexity is $\max_{e \in E} \sum_{r=0}^{t-1} b_e(r)$, where $b_e(r)$ denotes the number of bits transmitted over edge e in round r. Analogously, the message complexity denotes the maximum number of messages transmitted over any edge.

The time complexity of a distributed algorithm is traditionally defined as the number of communication rounds until the *last* node completes the algorithm. Somewhat inconsistently, message respectively bit complexity often measure the *total* of all exchanged messages respectively bits (of all nodes) or the expectation of message and bits exchanges of a single node during the execution. In this paper, analogous to the definition of time complexity, we consider the worst node only; in other words, the message respectively bit complexity is given by the number of messages or bits exchanged by the most loaded node. Both views have their validity, are commonly used and sometimes even coincide. The focus on a more local "maximum" measure is motivated by the observation that for distributed systems, an individual node might often form a bottleneck, and delay an algorithm, although overall the constraints on bandwidth, energy etc. are fulfilled. For example, if a single node in a battery powered sensor network must transmit much more often than other nodes, it will become non-operational much more quickly. This might have devastating effects on the network topology, e.g. disconnecting it and thereby preventing further data aggregation.

4 Tight Bounds on the Transmittable Information

In all (reasonable) distributed algorithms nodes must exchange a certain minimum of information with their neighbors. The amount of exchanged information is not only given by the total amount of bits contained in the messages, but also by the times, when the messages were sent and the size of the messages. In other words, the number of different (observable) behaviors of a node v by a neighbor u, i.e. the number of different ways v can communicate with u, determines the total amount of exchanged information between two nodes. We first bound the number of exchangeable information between two communication parties. By using a lower bound for the minimum needed amount of exchanged information for any kind of problem one therefore gets a lower bound on the time complexity of any algorithm depending on the bits and messages exchanged. We illustrate this general technique by deriving lower bounds for the MIS and coloring problem.

Theorem 1 *If a node executes t rounds using up to $m \leq t$ messages (at most one message per round) with a total of $b \geq m$ bits within messages (i.e. at least one bit per message), it can communicate in total $\Theta(m \log(tb/m^2) + b)$ bits for $m < \sqrt{bt}$ and $\Theta(b)$ bits for $m \geq \sqrt{bt}$.*

Proof. A node can decide not to transmit at all or it can transmit in $n_r \in [1, m]$ rounds $\{r_1, r_2, ..., r_{n_r}\}$ with $r_i \in [0, t-1]$ for $1 \leq i \leq n_r$. In each chosen round r_i the node transmits at least one bit. The total number of choices of rounds is given by $\binom{t}{n_r}$. Say a node wants to transmit n_{r_i} bits in round r_i then the sum of all bits transmitted n_t in all rounds must be at least n_r and at most b, i.e. $n_r \leq n_t := \sum_{i=1}^{n_r} n_{r_i} \leq b$. Thus the number of all possible sequences $(n_{r_1}, n_{r_2}, ..., n_{r_{n_r}})$ with $n_{r_i} \in [1, b - n_r + 1]$ is given by the composition of n_t into exactly n_r parts, i.e. the number of ways we can write n_t as a sum of n_r terms, i.e. $\binom{n_t-1}{n_r-1} \leq \binom{n_t}{n_r}$. Each of the at most n_t transmitted bits can either be 0 or 1, yielding 2^{n_t} combinations. Multiplying, these three terms and adding one for the case that a node does not transmit, i.e. $\binom{t}{n_r} \cdot \binom{n_t-1}{n_r-1} \cdot 2^{n_t} + 1$, gives a bound on the different behaviors of a node for a fixed n_r. Thus, overall the number of behaviors is upper bounded by:

$$1 + \sum_{n_r=1}^{m} \sum_{n_t=n_r}^{b} \binom{t}{n_r} \cdot \binom{n_t}{n_r} \cdot 2^{n_t} \leq 1 + mb \cdot \max_{1 \leq n_r \leq m, 0 \leq n_t \leq b} \binom{t}{n_r} \cdot \binom{n_t}{n_r} \cdot 2^{n_t}$$

$$\leq 1 + mb \cdot \max_{1 \leq n_r \leq m} \binom{t}{n_r} \cdot \binom{b}{n_r} \cdot 2^b$$

The last inequality follows due to $n_r \leq n_t \leq b$. We have $\binom{n}{k} \leq (ne/k)^k$, thus $\binom{b}{n_r} \leq (eb/n_r)^{n_r}$. Continuing the derivation we get:

$$\leq 1 + mb \cdot \max_{1 \leq n_r \leq m} (et/n_r)^{n_r} \cdot (eb/n_r)^{n_r} \cdot 2^b \leq 1 + mb \cdot 2^b \max_{1 \leq n_r \leq m} (e^2 bt/n_r^2)^{n_r}$$

Next we compute the maximum:

$$\frac{d}{dn_r}(e^2 bt/n_r^2)^{n_r} = (e^2 bt/n_r^2)^{n_r} \cdot (\ln(e^2 bt/n_r^2) - 2) = 0$$

$$\Leftrightarrow \ln(e^2 bt/n_r^2) - 2 = 0 \Leftrightarrow e^2 bt/n_r^2 = e^2 \Leftrightarrow n_r = \sqrt{bt}$$

For $m \geq \sqrt{bt}$ we get: $1 + mb \cdot 2^b \max_{1 \leq n_r \leq m}(e^2 bt/n_r^2)^{n_r} \leq (m+1) \cdot 2^b \cdot (e^2)^{\sqrt{bt}}$
For $m < \sqrt{bt}$: $1 + mb \cdot 2^b \max_{1 \leq n_r \leq m}(e^2 bt/n_r^2)^{n_r} \leq (m+1)(b+1) \cdot 2^b (e^2 bt/m^2)^m$

Taking the logarithm yields the amount of transmittable information being asymptotically equal to $O(m \log(tb/m^2) + b)$ for $m < \sqrt{bt}$, since $t \geq m$ and $b \geq m$ and $O(b + \sqrt{bt}) = O(b)$ for $b \geq m \geq \sqrt{bt}$.

With the same reasoning as before a lower bound can be computed. We use $\binom{n}{k} \geq (n/k)^k$ we have $\binom{b-1}{n_r-1} \geq ((b-1)/(n_r-1))^{n_r-1}$.

$$1 + \sum_{n_r=1}^{m} \sum_{n_t=n_r}^{b} \binom{t}{n_r} \cdot \binom{n_t - 1}{n_r - 1} \cdot 2^{n_t} \geq \max_{1 \leq n_r \leq m, 0 \leq n_t \leq b} \binom{t}{n_r} \cdot \binom{n_t - 1}{n_r - 1} \cdot 2^{n_t}$$

$$\geq \max_{1 \leq n_r \leq m} \binom{t}{n_r} \cdot \binom{b-1}{n_r - 1} \cdot 2^b \geq 2^b \max_{1 \leq n_r \leq m} (t/n_r)^{n_r} \cdot ((b-1)/(n_r - 1))^{n_r - 1}$$

$$\geq 2^b \max_{1 \leq n_r \leq m} ((b-1)t/((n_r-1)n_r))^{n_r-1} \geq 2^b \max_{1 \leq n_r \leq m} ((b-1)t/n_r^2)^{n_r-1}$$

Next we compute the maximum:

$$\frac{d}{dn_r}((b-1)t/n_r^2)^{n_r-1} = ((b-1)t/n_r^2)^{n_r-1} \cdot (\ln(((b-1)t/n_r^2)) - 2 + 1/n_r) = 0$$

$$\Leftrightarrow \ln((b-1)t/n_r^2) - 2 + 1/n_r = 0 \Leftrightarrow (b-1)t/n_r^2 = e^{2-1/n_r} \Leftrightarrow n_r = \sqrt{(b-1)t/e^{1+1/(2n_r)}}$$

For $m \geq \sqrt{(b-1)t/e^{1+1/(2n_r)}}$ we have: $2^b \max_{1 \leq n_r \leq m}((b-1)t/n_r^2)^{n_r-1} \geq 2^b(e^2)^{\sqrt{(b-1)t/e^2}-1}$
For $m < \sqrt{(b-1)t/e^{1+1/(2n_r)}}$: $2^b \max_{1 \leq n_r \leq m}((b-1)t/n_r^2)^{n_r-1} \geq 2^b(e^2(b-1)t/m^2)^{m-1}$

This, yields $\Omega(m \log(tb/m^2) + b)$ for $m < \sqrt{(b-1)t/e^{1+1/(2n_r)}}$, since $t \geq m$ and $b \geq m$ and $\Omega(b + \sqrt{bt}) \geq \Omega(b)$ for $m \geq \sqrt{(b-1)t/e^{1+1/(2n_r)}}$.

Overall, the bounds become $\Theta(m \log(tb/m^2) + b)$ for $m < \sqrt{bt}$ and $\Theta(b)$ otherwise.

Corollary 2 *The amount of information that k parties can exchange within t rounds, where each party can communicate with each other party directly and uses up to $m \leq t$ messages (at most one message per round) with a total of $b \geq m$ bits (i.e. at least one bit per message) is $\Theta(km \log(tb/m^2) + b)$ for $m < \sqrt{bt}$ and $\Theta(kb)$ bits for $m \geq \sqrt{bt}$.*

Proof. Compared to Theorem 1, where one node A only transmits data to another node B, the number of observable behaviors if k nodes are allowed to transmit is raised to the power of k, i.e. if one node can communicate in x different ways then the total number of observable behaviors becomes x^k. Taking the logarithm gives the amount of exchangeable information for k parties, i.e. $\log(x^k) = k \log x$, where $\log x$ is the amount of information a single node can transmit as stated in Theorem 1.

4.1 Lower Bound on the Time Complexity Depending on the Bit (and Message) Complexity

We first bound the amount of information that must be exchanged to solve the MIS and coloring problem. Then we give a lower bound on the time complexity depending on the bit complexity for any MIS and coloring algorithm where a message consists of at least one bit.

Theorem 3 *Any algorithm computing a MIS (in a randomized manner) in a constant degree graph, where each node can communicate less than $c_0 \log n$ bits for some constant c_0 fails (w.h.p.).*

The intuition of the so called "fooling set" argument proof is as follows: If a node cannot figure out what its neighbors are doing, i.e. it is unaware of the IDs of its neighbors, its chances to make a wrong choice are high.

Proof. Let us look at a graph being a disjoint union of cliques of size 2, i.e. every node has only one neighbor. A node can communicate in up to $2^{c_0 \log n} = n^{c_0}$ distinct ways. In the deterministic case, let $B_u \in [0, n^{c_0} - 1]$ be the behavior that a node u decides on given that it sent and received the same information throughout the execution of the algorithm. Clearly, before the first transmission no information exchange has occurred and each node u fixes some value B_u. Since there are only n^{c_0} distinct values for n nodes, there exists a behavior $B \in [0, n^{c_0} - 1]$, which is chosen by at least n^{1-2c_0} nodes given that they sent and received the same information.

Consider four arbitrary nodes $U = \{u, v, w, x\}$ that receive and transmit the same information. Consider the graph with $G' = (U, \{(u, v), (w, x)\})$ where u, v and also w, x are incident, $G'' = (U, \{(v, w), (u, x)\})$ and $G''' = (U, \{(u, w), (v, x)\})$. Note that u, v, w, x have no knowledge about the identity of their neighbors (They only know that their degrees are 1). Assume a deterministic algorithm correctly computes a MIS for G' and v is joined the MIS, then u is not joined the MIS in G' but also not in G''', since it cannot distinguish G' from G'''. Thus w must join the MIS to correctly compute a MIS for

G'''. Therefore, both v, w are joined the MIS S in G'' and thus S violates the independence condition of a MIS.

For the randomized case the argument is similar. Before the first transmission all nodes have received the same information. Since there are only n^{c_0} distinct behavior for n nodes at least a set S of nodes of cardinality $|S| \geq n^{1-c_0}$ will decide to transmit the same value B with probability at least $1/n^{2c_0}$ (given a node sent and received the same information). Now, assume we create a graph by iteratively removing two randomly chosen nodes $u, v \in S$ and adding an edge (u, v) between them until S is empty. For each node $v \in S$ must specify some probability to be joined the MIS. Assume the algorithm sets at least $|S|/2$ nodes to join with probability at least $1/2$. Given that at most $|S|/4$ nodes have been chosen from S to form pairs, the probability that two nodes u, v out of the remaining nodes joining the MIS with probability $1/2$, i.e. $\geq |S|/2 - |S|/4 = |S|/4$ nodes, are paired up is at least $1/16$ independently of which nodes have been paired up before. The probability that for a pair u, v behaving identically both nodes u, v join (and thus the computation of the MIS fails) is at least $1/4$. Since we have $|S|/2 = n^{1-2c_0}/2$ pairs, we expect a set $S' \subset S$ of at least $n^{1-6c_0}/2 \cdot 1/16 \cdot 1/4$ pairs to behave identically and join the MIS. Using a Chernoff bound for any constant $c_0 < 1/6$ at least $|S'| \geq n^{1-6c_0}/1024$ nodes behave identically with probability at least $1 - 1/n^c$ for an arbitrary constant c.

An analogous argument holds if less $|S|/2$ nodes are joined with probability more than $1/2$. In this case for some pairs u, v w.h.p. no node will join the MIS.

Theorem 4 *Any algorithm computing a MIS or coloring deterministically (or in a randomized manner) transmitting only* $b \leq c_1 \frac{\log n}{\log(t/\log n)}$ *bits per edge with* $t \in [2\log n, n^{c_2}]$ *for constants* c_1, c_2 *requires at least* t *time (w.h.p.). For* $t < 2\log n$ *and* $b \leq c_1 \log n$ *bits no algorithm can compute a MIS (w.h.p.).*[2]

Proof. If $m \geq \sqrt{tb}$ using the bound of $\Theta(b)$ of Theorem 1, a node can communicate at most $c_{thm}c_1 \log n$ bits for a constant c_{thm}. We have $c_{thm}c_1 \log n \leq c_0 \log n$ for a suitable constant c_1. Due to Theorem 3 at least $c_0 \log n$ bits are needed. For $t < 2\log n$ and $b \leq c_1 \log n$, we have $\sqrt{tb} \leq 2c_1 \log n$. If $m < \sqrt{tb}$, then the amount of transmittable information becomes (neglecting c_{thm} for now) $(m \log(tb/m^2) + b) \leq \sqrt{tb}\log 2 + c_1 \log n \leq 3c_1 \log n \leq c_0 \log n$ for a suitable constant c_1.

For $t \geq 2\log n$ and $m \leq b \leq \sqrt{tb}$ the amount of transmittable information becomes $O(m \log(tb/m^2) + b)$. We have $\max_{m \leq b} m \log(tb/m^2) \leq b \log(t/b)$. The maximum is attained for $m = b$. Using the assumption $b \leq c_1 \frac{\log n}{\log(t/\log n)}$, we get further: $b \log(t/b) \leq c_1 \frac{\log n}{\log(t/\log n)} \cdot \log(t \log(t/\log n)/\log n) = c_1 \frac{\log n}{\log(t/\log n)} \cdot (\log(t/\log n) + \log(\log(t/\log n))) = c_1 \log n \left(1 + \frac{\log(\log(t/\log n))}{\log(t/\log n)}\right) \leq 2c_1 \log n$ (since $t \leq n$). Thus, we have $m \log(tb/m^2) + b \leq 2c_1 \log n + b \leq 3c_1 \log n$.

[2] Note that the theorem does not follow directly from Theorem 3, since the number of bits that can be communicated using time-coding is generally larger than b, i.e. see Theorem 1.

Due to Theorem 3 at least $c_0 \log n$ bits required, thus for $3c_1c_{thm} < c_0$ at least time t is required. The lower bound for the MIS also implies a lower bound for $O(\Delta)$ coloring, since a MIS for constant degree graphs can be computed from a coloring in constant time, i.e. in round i nodes with color i are joined the MIS, if no neighbor is already in the MIS.

In a later section, we give an algorithm running in time $O(t \log n)$ using $O(\log n/ \log t)$ messages and bits. Thus, there exists an algorithm running in $O(\log n)$ time transmitting only $\log n/c$ messages containing one bit for any constant c. On the other hand, due to our lower bound any algorithm that transmits only one message containing $\log n/c$ bits for a sufficiently large constant c requires time n^{1/c_1} for some constant c_1 and is thus exponentially slower.

5 Algorithms Trading among Bit, Message, and Time Complexity

We look at various deterministic and randomized algorithms for the coloring and the maximal independent set problem as case studies. Before showing the tradeoffs we reduce the bit complexity of the algorithms without altering the time complexity. Then we show two mechanisms how prolonging an algorithm can be used to reduce the bit and – at the same time – the message complexity.

The first mechanism is straight forward and useful for randomized algorithms for symmetry breaking tasks, e.g. for MAC protocols where nodes try to acquire a certain resource. Assume a node tries to be distinct from its neighbors or unique among them. For example, for the coloring problem, it tries to choose a distinct color from its neighbors. For the MIS problem it tries to mark itself, and joins the MIS, if no neighbor is marked as well. Thus, if two neighbors get marked or pick the same color, we can call this a collision. We can reduce the probability of collisions by reducing the probability of a node to pick a color or get marked in a round. Thus, if a node only transmits if it has chosen a color or got marked, this causes less bits to be transmitted.

The second mechanism is beneficial for certain distributed algorithms that solve problems by iteratively solving subproblems and combining the solutions. Often the size (or number) of subproblems determines the number of iterations required, e.g. for divide and conquer algorithms. Assume that a distributed algorithm requires the same amount of communication to solve a subproblem independent of the size of the subproblem. In this case, by enlarging the size of the subproblem, the total number of iterations and thus the total amount of information to be transmitted can be reduced.

Apart from that there are also general mechanisms that work for any algorithm. Encoding information using the traditional *time coding* approach for k rounds works as follows: To transmit a value x we transmit x div k in round $x \mod k$.[3] Thus, in case $k \geq 2x - 1$ a single message of one bit is sufficient.

[3] The division operation x div k returns an integer value that states how often number k is contained in x.

Otherwise $\log x - \log k$ bits are needed. Our lower bound (Theorem 1) shows that a value of $\log x$ bits can be communicated by transmitting less than $\log x$ bits using more than one message and more than one communication round.

5.1 Coloring Algorithm

In the randomized coloring algorithms using the Multi-Trials technique [18] a node v picks a random number in $[0, \Delta]$ for each color not taken by one of its neighbors. Thus, given that a node can choose among $C(v)$ unused colors, the size of a message is $\log \Delta \cdot |C(v)|$. In [18] this is improved by letting a node pick one color out of every $\max_{u \in N(v)} 2d(u)$ colors. This results in bit complexity of $O(\log \Delta \log n)$ for $O(\Delta)$ and $\Delta + 1$ coloring. We use the improved algorithms as subroutines.

To lower the bit complexity while maintaining the same time complexity we let nodes get a color in two steps. First, a node picks an interval of colors. Second, it attempts to obtain an actual color from the chosen interval.

5.2 Randomized $O(\Delta + \log^{1+1/\log^* n} n)$ Coloring

We assume that initially each node v has $|C(v)| = (1 + 1/2^{\log^* n - 2})(\Delta_{N_+(v)} + \log^{1+1/\log^* n} n)$ colors available. Each node v considers disjoint intervals $([0, l-1], [l, 2l-1], ...)$ of colors, where each interval contains $l := (1 + 1/2^{\log^* n - 1}) \log^{1+1/\log^* n} n$ colors and the total number of intervals is given by $|C(v)|/l$. A node v first picks one of these intervals $I(v) \in \{0, 1, ..., |C(v)|/l\}$ of colors uniformly at random. From then on, it only considers a subgraph $G_{I(v)}$ of G, i.e. only neighbors $u \in N(v)$ that have picked the same interval $I(u) = I(v)$. All other neighbors operate on different intervals and have no influence on node v. Then, a coloring is computed in parallel for all subgraphs. That is to say, node v executes Algorithm *ConstDeltaColoring* [18] on $G_{I(v)}$ and tries to get a color or better said an index $ind_{I(v)}$ from $\{0, 1, ..., l-1\}$ in the interval $I(v)$. Its final color is given by the $ind_{I(v)}$ plus the color offset $I(v) \cdot l$ of the chosen interval $I(v)$.

Lemma 1. *Each node v has at most $\log^{1+1/\log^* n} n$ neighbors $u \in N(v)$ with $I(u) = I(v)$ w.h.p.*

Proof. Initially, each node picks independently uniformly at random one interval out of $(1 + 1/2^{\log^* n - 2})\Delta_{N_+(v)}/((1 + 1/2^{\log^* n - 1}) \log^{1+1/\log^* n} n) = c_1 \cdot \Delta_{N_+(v)}/\log^{1+1/\log^* n} n$ many with $c_1 = (2^{\log^* n - 1} + 2)/(2^{\log^* n - 1} + 1)$. Thus, a node v expects $E \leq \frac{\Delta_{N_+(v)}}{c_1 \cdot \Delta_{N_+(v)}/\log^{1+1/\log^* n} n} = \log^{1+1/\log^* n} n/c_1$ neighbors to have chosen the same interval. Using a Chernoff bound the probability that there are more than a factor $1 + c_1/2$ nodes beyond the expectation for a single interval is bounded by $1 - 2^{-c_1^2/8 \cdot E} = 1 - 2^{-c_1/8 \log^{1+1/\log^* n} n/c_1} \geq 1 - 1/n^{c_0}$ for an arbitrary constant c_0. Thus, w.h.p. the number of nodes in an interval is at most $(1 + c_1/2) \cdot \log^{1+1/\log^* n} n/c_1 \leq \log^{1+1/\log^* n} n$. The probability that this holds for all intervals can be bounded to be $1 - 1/n^{c_0-3}$ using Theorem 2 from [18].

Theorem 5 *The algorithm computes an $O(\Delta + \log^{1+1/\log^* n} n)$ coloring with bit complexity $O(\log n \log\log n)$ in time $O(\log^* n)$ w.h.p. (for sufficiently large n)*

Proof. The initial transmission of the interval requires at most $\log n$ bits, i.e. $\log \Delta - \log\log n$. Afterwards, when all nodes are split into subgraphs, the same analysis applies as for the *ConstDeltaColoring* Algorithm from [18] with $\Delta \leq \log^{1+1/\log^* n} n$, since each node only competes with at most $\log^{1+1/\log^* n} n - 1$ other nodes due to Lemma 1 and we have $(1 + 1/2^{\log^* n - 1}) \log^{1+1/\log^* n} n$ available colors. The colors are picked such that the chance of getting a chosen color is constant, i.e. a node u picks one color for every sequence of $2\Delta_{N_+(v)}$ available colors, where $\Delta_{N_+(v)}$ denotes the maximum size of an uncolored neighborhood of an uncolored node $v \in N(u)$ before the current communication round. Thus, each node v that picks a color has probability $1/2$ to actually get a color independent of the choices of its neighbors, since the number of chosen colors of all neighbors together is at most $\Delta_{N_+(v)}$, i.e. half the colors of all available colors $2\Delta_{N_+(v)}$ and node v makes its choice independent of the concurrent choices of its neighbors. Thus, after a node has picked and transmitted $O(\log n)$ colors with probability $1 - 1/2^{O(\log n)} = 1 - 1/n^c$ for an arbitrary constant c, a node has obtained a color. Since each color requires $\log\log n$ bits the total bit complexity is $O(\log n \log\log n)$. We can apply Corollary 14 [18] that gives a running time of $O(\log^* n)$ w.h.p.

5.3 Rand. $O(\Delta)$ Coloring in Time t^c using $O(\log n/\log t)$ Bits

One could use the previously described algorithm and traditional time coding to save on the bit complexity maintaining the same number of transmitted messages. For readability and to illustrate both concepts quantitatively we focus on the case $t^c \geq \log^{2+\epsilon} n$ (for an arbitrary small constant ϵ), where one can save on both: the message complexity by a factor of $\log t$ and the bit complexity by a factor of $\log\log n \log t$.[4] A node initially chooses an interval consisting of $(1 + 1/2^{\log^* n - 2}) \log^{1+1/\log^* n} n$ colors. Then the node iteratively transmits a single bit in a random round out of every $t_p = t^c/(c_1 \log n^{1+1/\log^* n})$ rounds. If it is the only one transmitting, it chooses a color, informs its neighbors about the obtained color and ends the algorithm.

Theorem 6 *The algorithm computes an $O(\Delta + \log^{1+1/\log^* n} n)$ coloring with bit complexity $O(\log n/\log t)$ in time $t^c + O(\log^* n)$ for any parameter c and t such that $t^c \geq \log^{2+\epsilon} n$ and $t \leq n$ for an arbitrary constant $\epsilon > 0$ w.h.p.*

Proof. The initial transmission of the interval requires less than $\log n$ bits, i.e. $\log \Delta - \log\log n$. We can use Theorem 1 with $b = m = O(\log n/\log t)$

[4] For small $t \leq 2\log n$ it is not possible to achieve bit complexity $c_1 \log n/\log t$ for a fixed constant c_1 due to the lower bound given in Theorem 4.

Algorithm FewBitsDeltaColoring, i.e. $(1+\epsilon)\Delta$ for $\epsilon > 1/2^{\log^* n-2}$ and parameter $t > \log^{2+\epsilon} n$

$s(v) := none$; $ind_{I(v)} := none$; $C(v) := \{0, 1, ..., (1+\epsilon)\log^{1+1/\log^* n} n - 1\}$
$I(v) :=$ random integer $r \in [0, (1+\epsilon)\Delta_{N_+(v)}/\log^{1+1/\log^* n} n + 1]$
Transmit $I(v)$ to all neighbors $u \in N(v)$ using time $t^c/2$ and $\log n/\log t$ bits and messages
$N_{I(v)}(v) := \{u \in N(v) | I(v) = I(u)\}$ {Only consider nodes in the same interval}
$i := 0$; $t_p := t^c/(c_1 \log^{1+1/\log^* n} n)$ {with constant c_1}
repeat
  if $i \mod t_p = 0$ **then** $t_s(v) :=$ Random odd number in $[0, t_p]$ **end if**
  if $t_s(v) = i$ **then**
    Transmit 1
    if nothing received **then**
      $ind_{I(v)} :=$ arbitrary available color
      Transmit $ind_{I(v)}$
    end if
  end if
  $N(v) := \{u | u \in N_{I(v)}(v) \land color_I(u) = none\}$
  $C(v) := C(v) \setminus \{ind_I(u) | u \in N(v)\}$
  $i := i + 1$
until $ind_{I(v)} \neq none$
$color(v) := ind_{I(v)} + I(v) \cdot (1+\epsilon)\log^{1+1/\log^* n} n$

messages m and bits b and at least $t^c/2 \geq (\log^{2+\epsilon} n)/2$ rounds. Since $\sqrt{bt} > m$ the amount of information that can be communicated is $\Theta(m\log(tb/m^2) + b) = O(\log n/\log t \log(t/(\log n/\log t)) + \log n/\log t) \geq O(\log n/\log t \log(t \log t/\log n)) \geq O(\log n/\log t(\log t + \log(\log t/\log n))) = O(\log n + \log n(\log\log t/\log t - \log\log n/\log t) = O(\log n)$ bits, since $\log\log n/\log t^c < 1/2$ because $t^c \geq \log^{2+\epsilon} n$.

Due to Lemma 1 each node v has at most $\Delta_0 := \log^{1+1/\log^* n} n$ neighbors competing for the $(1+1/2^{\log^* n-2})\log^{1+1/\log^* n} n$ colors of v's chosen interval. A node v transmits one bit for each interval of length t_p. Since nodes make their choices independently, the probability that node v is the only node transmitting is at least $1 - \Delta_0/t_p$, corresponding to the worst case that all neighbors transmit in different rounds. We have $\Delta_0/t_p = \Delta_0/(t^c/(c_2\log^{1+1/\log^* n} n) = \Delta_0 \cdot c_2 \log^{1+1/\log^* n} n/t^c \leq c_2 \log^{2+2/\log^* n} n/t^c$(due to Lemma 1) $\leq 1/t^{c \cdot 1/c_3}$ for some constant c_3 since $t^c \geq \log^{2+\epsilon} n$.Thus the chance $O(\log n/\log t) = c_4 \log n/\log t$ trials fail is $(1/t^{c/c_3})^{c_4 \log n/\log t} = 1/n^{c_1}$ for an arbitrary constant c_1 and a suitable constant c_4.

5.4 Deterministic $\Delta + 1$ Coloring

We adapt an algorithm [10] to turn a Δ^k coloring for any constant k into a $\Delta+1$ coloring in time $O(t^c \Delta \log \Delta)$ using $O(\log \Delta/\log t)$ messages of size $O(\log \Delta)$ and for an arbitrary parameter t and arbitrary constant c. The algorithm reduces the number of used colors in an iterative manner by splitting up the whole range of colors into sequences of colors of size at least $2t^c(\Delta+1)$. Consider all nodes S_I that have a color in some sequence I consisting of $2t^c(\Delta+1)$ colors, e.g. $I=\{0,1,...,2t^c(\Delta+1)-1\}$. To compress the range of used colors to $[0,\Delta]$, we can sequentially go through all $2t^c(\Delta+1)$ colors and let node $v \in S_I$ choose the smallest available color, i.e. in round i a node having the ith color in the interval I can pick a new color from I not taken by any of its neighbors. After going through all colors, we combine $2t^c$ intervals $\{I_0, I_1, ..., I_{2t^c-1}\}$ to get a new interval I' of the same size, i.e. $2t^c(\Delta+1)-1$. A node v with color i from I_j with $i \in [0,\Delta]$ gets color $c(v)=j\cdot(\Delta+1)+i$ in I'. Then we (recursively) apply the procedure again on all intervals I'.

Theorem 7 *The deterministic $\Delta+1$ coloring terminates in time $O(t^c\Delta\log\Delta)$ having bit complexity $O(\log^2\Delta/\log t)$ and message complexity $O(\log\Delta/\log t)$ for any parameter $1<t\leq\Delta$ and any constant c.*

Proof. We start from a correct Δ^k coloring for some constant k. A node gets to pick a color out of a sequence $(c_1, c_1+1, ..., c_1+2t^c(\Delta+1)-1)$ of $2t^c(\Delta+1)$ colors for $c_1 := c_0 2t^c(\Delta+1)$ and an arbitrary integer c_0. Thus it can always pick a color being at most $c_1+\Delta$ since it has at most Δ neighbors. After every combination of intervals requiring time $2t^c(\Delta+1)$, the number of colors is reduced by a factor of $2t^c$. We require at most $x=k\log\Delta/\log(2t^c)$ combinations since $(2t^c)^x=\Delta^k$. Therefore, the overall time complexity is $O(t^c\Delta\log\Delta)$. In each iteration a node has to transmit one color out of $2t^c(\Delta+1)$ many, i.e. a message of $\log(2t^c(\Delta+1)) = O(\log\Delta)$ bits (since $t^c\leq\Delta$) giving $O(\log^2\Delta/\log t)$ bit complexity.

5.5 MIS Algorithm

Our randomized Algorithm LowBitAndFast is a variant of algorithm [14]. It proceeds in an iterative manner. A node v marks itself with probability $1/d(v)$. In case two or more neighboring nodes are marked, the choice which of them is joined the MIS is based on their degrees, i.e. nodes with higher degree get higher priority. Since degrees change over time due to nodes becoming adjacent to nodes in the MIS, the degree has to be retransmitted whenever there is a conflict. Our algorithm improves Luby's algorithm by using the fact that the degree $d(u)$ of a neighboring node is not needed precisely, but an approximation $\tilde{d}(u)$ is sufficient. Originally, each node maintains a power of two approximation of the degrees of its neighbors, i.e. the approximation is simply the index of the highest order bit equal to 1. For example, for $d(v)$ having binary value 10110, it is 4. The initial approximate degree consists of $\log\log n$ bits. It is transmitted using (well known) time coding for $x=\log n$ rounds, i.e. to transmit a value k we transmit k div x in round $k \mod x$. When increasing the time complexity by a factor of t^{c_0} for an arbitrary constant c_0, a node marks itself with probability $1/(t^{c_0}\tilde{d}(v))$ for t^{c_0} rounds, where the approximation is only updated after the t^{c_0} rounds. Afterwards, a node only informs its neighbors if the degree changed by

Algorithm LOWBITANDFAST FOR ARBITRARY VALUE $t^{c_0} \geq 16$

For each node $v \in V$:
1: $\tilde{d}(v)$:= index of highest order bit of $2|N(v)|$ {2 approximation of $d(v)$}
2: Transmit $\tilde{d}(v)$ to all neighbors $u \in N(v)$ using time coding for $\log n$ rounds
3: **loop**
4:    **for** $i = 1..t^{c_0}$ **do**
5:       Choose a random bit $b(v)$, such that $b(v) = 1$ with probability $\frac{1}{4t^{c_0} \cdot \tilde{d}(v)}$
6:       Transmit $b(v)$ to all nodes $u \in N(v)$
         if $b(v) = 1 \wedge \nexists u \in N(v), b(u) = 1 \wedge \tilde{d}(u) \geq \tilde{d}(v)$ **then** Join MIS **end if**
7:    **end for**
8:    $k(v) := \max\{\lceil \log i \rceil | \text{integer } i, \frac{\tilde{d}(v)}{i} \geq d(v)\}$
9:    **if** $k(v) > c_0/2 \log t$ **then**
10:       Transmit $k(v)$ using time message coding for t^{c_0} rounds using c_2 messages of size 1 bit
11:       $\tilde{d}(v) := \tilde{d}(v) \text{ div } 2^{k(v)} + \tilde{d}(v) \mod 2^{k(v)}$
12:    **end if**
13:    **for all** received messages $k(u)$ **do**
14:       $\tilde{d}(u) := \tilde{d}(u) \text{ div } 2^{k(u)} + \tilde{d}(u) \mod 2^{k(u)}$
15:    **end for**
16: **end loop**

a factor of at least two. For updating the approximation we use time message coding for t^{c_0} rounds and a constant number of messages and bits. Whenever a node is joined the MIS or has a neighbor that is joined, it ends the algorithm and informs its neighbors.

The analysis is analogous to [14] and differs only in the constants.

Lemma 2. *A node v maintains an approximation $\tilde{d}(u)$ of the degrees $d(u)$ of all neighbors $u \in N(v)$, such that $d(u) \leq \tilde{d}(u) \leq 2 \cdot d(u)$, when node v begins to join the MIS for t^{c_0} rounds (i.e. enters the for loop).*

Proof. Since initially a node transmits a two approximation of its degree to all neighbors, the lemma holds at the beginning of the algorithm, i.e. before the first execution of the for loop (lines 4 to 7). Whenever for a node v, the approximation $\tilde{d}(v)$ stored by its neighbors (and itself) is larger than $i \cdot d(v)$ with integer $i > 2$ before an execution of the for loop (line 4 to 7), node v adapts $\tilde{d}(v)$, i.e. sets $\tilde{d}(v) := \tilde{d}(v) \text{ div } 2^{\lceil \log i \rceil}$, such that $d(v) \leq \tilde{d}(v) \leq 2 \cdot d(v)$ and transmits i to its neighbors $u \in N(v)$ that also adapt their approximation $\tilde{d}(v)$ for node v's degree.

Lemma 3 (Joining MIS). *A node v is joined the MIS in t^{c_0} rounds with probability $p \geq \frac{1}{8t^{c_0} d_0(v)}$, where $d_0(v)$ denotes the degree of node v before the execution of the first out of t^{c_0} rounds.*

Proof: Let M be the set of marked nodes, i.e. with their bits equal to 1. Let $H(v)$ be the set of neighbors of v with higher or equal degree approximation. Using independence of v and $H(v)$ we get

$$
\begin{aligned}
Pr\left[v \notin \text{MIS} | v \in M\right] &= Pr\left[\exists w \in H(v), w \in M | v \in M\right] \\
&= Pr\left[\exists w \in H(v), w \in M\right] \\
&\leq \sum_{w \in H(v)} Pr\left[w \in M\right] = \sum_{w \in H(v)} \frac{1}{4t^{c_0}\tilde{d}(w)} \\
&\leq \frac{d_0(v)}{4\tilde{d}(v)t^{c_0}} \leq \frac{1}{2t^{c_0}}
\end{aligned}
$$

- The first equality is correct because $H(v)$ and v are independently marking themselves.
- The second inequality is because $w \in H(v)$, if $\tilde{d}(v) \leq \tilde{d(w)}$
- The third inequality is because $w \in H(v)$, if $d_0(v) \leq 2\tilde{d(v)}$

Then

$$
Pr\left[v \in \text{MIS}\right] = Pr\left[v \in \text{MIS} | v \in M\right] \cdot Pr\left[v \in M\right] \geq (1 - \frac{1}{2t^{c_0}}) \cdot \frac{1}{4t^{c_0}\tilde{d}(v)} \geq \frac{1}{8t^{c_0}d_0(v)}
$$

□

Lemma 4 (Good Nodes). *A node v is called* good *if*

$$
\sum_{w \in N(v)} \frac{1}{4d(w)} \geq \frac{1}{12}.
$$

Otherwise we call v a bad *node. A good node will be removed with probability $p \geq c_1$ for some constant $c_1 > 0$ in t^{c_0} rounds.*

Proof: Let node v be good. If there is a neighbor $w \in N(v)$ with degree at most 8 we are done: With Lemma 3 the probability that node w is joined the MIS is at least $\frac{1}{64t^{c_0}}$ for a single round and thus $1 - (1 - \frac{1}{64t^{c_0}})^{t^{c_0}} \geq c_1$ for t^{c_0} rounds, and our good node will be removed.

So all we are worried about is that all neighbors have at least degree 9: For any neighbor w of v we have $\frac{1}{4d(w)} \leq \frac{1}{12}$. Since $\displaystyle\sum_{w \in N(v)} \frac{1}{4d(w)} \geq \frac{1}{12}$ there is a subset of neighbors $S \subseteq N(v)$ such that $\displaystyle\frac{1}{12} \leq \sum_{w \in S} \frac{1}{4d(w)} \leq \frac{1}{3}$

We can now bound the probability that node v will be removed. Let therefore R be the event of v being removed. Again, if a neighbor of v is joined the MIS, node v will be removed. We have

$$
\begin{aligned}
Pr\left[R\right] &\geq Pr\left[\exists u \in S, u \in \text{MIS}\right] \\
&\geq \sum_{u \in S} Pr\left[u \in \text{MIS}\right] - \sum_{u,w \in S; u \neq w} Pr\left[u \in \text{MIS} \textbf{ and } w \in \text{MIS}\right].
\end{aligned}
$$

For the last inequality we used the inclusion-exclusion principle truncated after the second order terms. Let M again be the set of marked nodes after step 1. Using $Pr\left[u \in M\right] \geq Pr\left[u \in \text{MIS}\right]$ we get

$$\begin{aligned}
Pr\left[R\right] &\geq \sum_{u \in S} Pr\left[u \in \text{MIS}\right] - \sum_{u,w \in S; u \neq w} Pr\left[u \in M \textbf{ and } w \in M\right] \\
&\geq \sum_{u \in S} Pr\left[u \in \text{MIS}\right] - \sum_{u \in S} \sum_{w \in S} Pr\left[u \in M\right] \cdot Pr\left[w \in M\right] \\
&\geq \sum_{u \in S} \frac{1}{8t^{c_0} d(u)} - \sum_{u \in S} \sum_{w \in S} \frac{1}{4t^{c_0} d(u)} \frac{1}{4t^{c_0} d(w)} \\
&\geq \sum_{u \in S} \frac{1}{4t^{c_0} d(u)} \left(\frac{1}{2} - \sum_{w \in S} \frac{1}{2t^{c_0} d(w)} \right) \geq \frac{1}{48t^{c_0}} \left(\frac{1}{2} - \frac{1}{6t^{c_0}} \right) = \frac{1}{c_2 t^{c_0}} . \text{(for some constant } c_2\text{)}
\end{aligned}$$

Thus, the probability that a node is not removed for t^{c_0} rounds becomes $(1 - \frac{1}{c_2 t^{c_0}})^{t^{c_0}} \leq c_3$ for some constant $c_3 < 1$. Therefore, the probability p that a good node is removed is a constant $c_1 := 1 - c_3 > 0$.

Lemma 5 (Good Edges). *An edge $e = (u, v)$ is called* bad *if both u and v are bad; else the edge is called* good. *The following holds: At any time at least half of the edges are good.*

Proof: For the proof we construct a directed auxiliary graph: Direct each edge towards the higher degree node (if both nodes have the same degree direct it towards the higher identifier). Now we need a little helper lemma before we can continue with the proof. The indegree is defined as the number of edges pointed toward a node. The outdegree is defined as the number of edges incident to a node that point away from it.

Lemma 6. *A bad node has outdegree at least twice its indegree.*

Proof: For the sake of contradiction, assume that a bad node v does not have outdegree at least twice its indegree. In other words, at least one third of the neighbor nodes (let us call them S) have degree at most $d(v)$. But then

$$\sum_{w \in N(v)} \frac{1}{4d(w)} \geq \sum_{w \in S} \frac{1}{4d(w)} \geq \sum_{w \in S} \frac{1}{4d(v)} \geq \frac{d(v)}{3} \frac{1}{4d(v)} = \frac{1}{12}$$

which means v is good, a contradiction. □

Continuing the proof of Lemma 5: According to Lemma 6 the number of edges directed into bad nodes is at most half the number of edges directed out of bad nodes. Thus, the number of edges directed into bad nodes is at most half the number of edges. Thus, at least half of the edges are directed into good nodes. Since these edges are not bad, they must be good.

Theorem 8 *Algorithm* LowBitAndFast *terminates in time* $O(t^{c_0} \log n)$ *w.h.p. having bit and message complexity* $O(\log n / \log t)$.

Proof: With Lemma 4 a good node (and therefore a good edge) will be deleted with constant probability within t^{c_0} rounds. Since at least half of the edges are good (Lemma 5) a constant number of edges will be deleted in each phase.

More formally: With Lemmas 4 and 5 we know that an edge will be removed with probability at least $c_1/2$. Let R be the number of edges to be removed. Using linearity of expectation we know that $E[R] \geq m \cdot (c_1/2)$, m being the total number of edges at the start of a phase. Now let $p := Pr[R \leq E[R]/2]$. Bounding the expectation yields

$$E[R] = \sum_r Pr[R = r] \cdot r \leq p \cdot E[R]/2 + (1 - p) \cdot m.$$

Solving for p we get

$$p \leq \frac{m - E[R]}{m - E[R]/2} < \frac{m - E[R]/2}{m} \leq 1 - c_1/2.$$

In other words, with probability at least $c_1/4$ for at least $m \cdot (c_1/4)$ edges are removed in a phase. We call a phase successful if this is the case. Since phases are independent, we can bound the probability that half of $c_2 \cdot \log n$ phases are successful using a Chernoff bound by $1 - e^{c_2/8 \log n}$. Thus, with probability $1 - 1/n^{c_2/8}$ there are at least $c_2/2 \log n$ successful phases. Since the number of edges $m \leq n^2$, the theorem follows for a sufficiently large constant c_2. □

Initially a node transmits its two approximation, which corresponds to an index in the binary representation of its degree. The degree can be encoded using $\log n$ bits and thus the two approximation requires at most $\log \log n$ bits.

Using time coding for $\log n$ rounds these bits are transmitted using a message of 1 bit. From then on the degree is only updated if a node's approximation differs from its correct value by more than a factor of 2 after t^{c_0} rounds. To perform an update, a node v transmits a value $k(v) = \log \max\{i | \text{integer } i, \frac{\tilde{d}(v)}{i} \geq |N(v)|\}$ using time coding for t^{c_0} rounds. Say a node transmits $r \leq O(\log n / \log t)$ factors $k_0(v), k_1(v), ..., k_{r-1}(v)$. We must have that $\prod_{i=0}^{r-1} 2^{k_i(v)} \leq d(v) \leq n$ or equivalently $\log \prod_{i=0}^{r-1} 2^{k_i(v)} \leq \log n = \sum_{i=0}^{r-1} k_i(v) \leq \log n$. Thus, the maximal amount of bits that has to be transmitted within messages when using time coding for t^{c_0} rounds for each message is given by $\max_{r, k_i(v)} \sum_{i=0}^{r-1} (\max \log k_i(v) - c_0 \log t, 1)$ given $\sum_{i=0}^{r-1} k_i(v) \leq \log n$. The maximum is $\mathrm{O}(\log n / \log t)$ for $k_i(v) = 1$ and $r = O(\log n / \log t)$ yielding an overall bit complexity of $\mathrm{O}(\log n / \log t)$.

References

1. L. Barenboim and M. Elkin. Distributed $(\delta + 1)$-coloring in linear (in δ) time. In *Symposium on Theory of Computing(STOC)*, 2009.
2. L. Barenboim and M. Elkin. Deterministic distributed vertex coloring in polylogarithmic time. In *Symp. on Principles of distributed computing(PODC)*, 2010.

3. Y. Dinitz, S. Moran, and S. Rajsbaum. Bit complexity of breaking and achieving symmetry in chains and rings. *J. ACM*, 2008.
4. P. Fraigniaud, C. Gavoille, D. Ilcinkas, and A. Pelc. Distributed computing with advice: information sensitivity of graph coloring. *Distributed Computing*, 2009.
5. P. Fraigniaud and G. Giakkoupis. On the bit communication complexity of randomized rumor spreading. In *SPAA*, 2010.
6. G. N. Frederickson and N. A. Lynch. Electing a leader in a synchronous ring. *J. ACM*, 34(1), 1987.
7. K. Kothapalli, C. Scheideler, M. Onus, and C. Schindelhauer. Distributed coloring in $O(\sqrt{\log n})$ bit rounds. In *International Parallel & Distributed Processing Symposium (IPDPS)*, 2006.
8. F. Kuhn. Weak Graph Coloring: Distributed Algorithms and Applications. In *Parallelism in Algorithms and Architectures (SPAA)*, 2009.
9. F. Kuhn, T. Moscibroda, and R. Wattenhofer. What Cannot Be Computed Locally! In *Symposium on Principles of Distributed Computing (PODC)*, 2005.
10. F. Kuhn and R. Wattenhofer. On the Complexity of Distributed Graph Coloring. In *Symp. on Principles of Distributed Computing (PODC)*, 2006.
11. E. Kushilevitz and N. Nisan. Communication complexity. In *Cambridge University Press*, 1997.
12. N. Linial. Locality in Distributed Graph Algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992.
13. Z. Lotker, B. Patt-Shamir, and S. Pettie. Improved distributed approximate matching. In *SPAA*, 2008.
14. M. Luby. A Simple Parallel Algorithm for the Maximal Independent Set Problem. *SIAM Journal on Computing*, 15:1036–1053, 1986.
15. Y. Métivier, J. M. Robson, S.-D. Nasser, and A. Zemmar. An optimal bit complexity randomised distributed mis algorithm. In *Colloquium on Structural Information and Communication Complexity (SIROCCO)*, 2009.
16. N. Nisan and A. Wigderson. Rounds in communication complexity revisited. *SIAM J. Comput.*, 22(1), 1993.
17. N. Santoro. *Design and Analysis of Distributed Algorithms.* Wiley-Interscience, 2006.
18. J. Schneider and R. Wattenhofer. A New Technique For Distributed Symmetry Breaking. In *Symp. on Principles of Distributed Computing(PODC)*, 2010.
19. J. Schneider and R. Wattenhofer. Distributed Coloring Depending on the Chromatic Number or the Neighborhood Growth. In *In Colloquium on Structural Information and Communication Complexity (SIROCCO)*, 2011.
20. J. Schneider and R. Wattenhofer. Poster Abstract: Message Position Modulation for Power Saving and Increased Bandwidth in Sensor Networks. In *10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2011.