Self-Stabilizing Clock Synchronization with 1-bit Messages

Paul Bastide* George Giakkoupis† Hayk Saribekyan‡

Abstract

We study the fundamental problem of distributed clock synchronization in a basic probabilistic communication setting. We consider a synchronous fully-connected network of n agents, where each agent has a local clock, that is, a counter increasing by one modulo T in each round. The clocks have arbitrary values initially, and they must all indicate the same time eventually. We assume a pull communication model, where in every round each agent receives an ℓ-bit message from a random agent. We devise several fast synchronization algorithms that use small messages and are self-stabilizing, that is, the complete initial state of each agent (not just its clock value) can be arbitrary.

We first provide a surprising algorithm for synchronizing a binary clock ($T = 2$) using 1-bit messages ($\ell = 1$). This is a variant of the voter model and converges in $O(\log n)$ rounds w.h.p., unlike the voter model which needs polynomial time. Next we present an elegant extension of our algorithm that synchronizes a modulo $T = 4$ clock, with $\ell = 1$, in $O(\log n)$ rounds. Using these two algorithms, we refine an algorithm of Boczkowski et al. (SODA'17), that synchronizes a modulo T clock in polylogarithmic time (in n and T). The original algorithm uses $\ell = 3$ bit messages, and each agent receives messages from two agents per round. Our algorithm reduces the message size to $\ell = 2$, and the number of messages received to one per round, without increasing the running time. Finally, we present two algorithms that simulate our last algorithm achieving $\ell < 2$, without hurting the asymptotic running time. The first algorithm uses a message space of size 3, i.e., $\ell = \log_2(3)$. The second requires a rough upper bound on $\log n$, and uses just 1-bit messages. More generally, our constructions can simulate any self-stabilizing algorithm that requires a shared clock, without increasing the message size and by only increasing the running time by a constant factor and a polylogarithmic term.

1 Introduction

We study the following clock synchronization problem. We have a synchronous fully-connected system of n processors, which we call agents. Each agent is equipped with a local clock, that is, a counter increasing by one modulo T in each round, where T is an integer common across all agents. The initial state of each agent, including the value of its clock, can be arbitrary, and the clocks of all agents must agree eventually. We assume a probabilistic pull communication model, where in each round each agent receives a message from an agent sampled uniformly at random. The message is just a function of the state of the sampled agent at the beginning of the round. In this setting, we investigate synchronization algorithms that are simple and efficient, converging in a small (polylogarithmic in n and T) number of rounds and using small (constant-size) messages.

Clock synchronization is a fundamental distributed task, both in engineered and natural systems. In engineered systems, clock synchronization is an essential building block, as most algorithms require that processors have a common notion of time (e.g., so that they all start an execution at the same point in time). In natural systems, spontaneous synchronization of clock oscillators is ubiquitous, e.g., in populations of synchronously flashing fireflies, electrically synchronous pacemaker cells, and groups of women whose menstrual cycles become mutually synchronized [36].

In the model we consider, we assume agents take steps in synchronous rounds, but do not have a consistent numbering of the rounds. (E.g., agents regularly receive a common pulse.) Clock synchronization in this setting is often referred to as *digital clock synchronization*, or *synchronous counting* [13, 19, 33, 34]. The focus of most previous work on this problem has been to achieve resilience against *Byzantine* agents, whose behaviour can be arbitrary (and malicious), along with *self-stabilization*, which guarantees convergence from any configuration of the agents' states (and thus resilience to transient failures). However, achieving resilience to Byzantine agents is known to incur significant communication overhead, even for the simpler problem of Byzantine agreement [21].

We focus, instead, on settings where agents are unlikely to demonstrate malicious behaviour, and aim at achieving self-stabilization. Moreover, we assume agents of limited computational and communication

*Inria, ENS Rennes, Rennes, France

†Inria, Univ Rennes, CNRS, IRISA, Rennes, France. Partially supported by ANR Projects PAMELA (ANR-16-CE23-0016-01) and DESCARTES (ANR-16-CE40-0023).

‡University of Cambridge, UK. Supported by Gates Cambridge programme.

power, such as mobile sensor networks, or insect populations. In such settings it makes sense to explore solutions that are simple, and use small space and messages. For that, we adopt the popular gossip-based model of communication where each agent interacts with a single random agent in each step [5, 30, 31]. This model is attractive for its simplicity, and inherent robustness to various kinds of faults.

Boczkowski, Korman, and Natale [15, 16] were the first to study clock synchronization in a setting (almost) identical to ours. A key result of their paper is an elegant recursive construction for reducing the message size of a general family of algorithms. Combining this construction with a known stabilizing consensus algorithm [18], they provided a self-stabilizing synchronization algorithm for a modulo T clock with running time $\tilde{O}(\log n \log T)$, using messages of 3 bits. Their algorithm requires that each agent receives messages from two random agents in each round, instead of one. The authors posed as an open problem "whether the message size can be reduced to 2 bits or even to 1 bit, while keeping the running time poly-logarithmic." It was also left open whether the requirement of receiving messages from two agents in each round can be lifted.

Our Contribution. We answer both the above questions in the affirmative. We first consider the simplest instance of the problem, namely, synchronizing a *binary* clock ($T = 2$). A natural approach, and one used by Boczkowski et al. [16], is to exploit the similarity between clock synchronization and consensus, as the former is just an agreement problem on a counter. Indeed, there are several well-studied self-stabilizing consensus protocol, such as 3-median, 2-choices, and 3-majority [12, 18, 28], which can be trivially used to synchronize a binary clock in a self-stabilizing manner, in logarithmic rounds w.h.p.[1] However, all currently known such protocols require that each agent receives the clock values of two other agents per round.

It is plausible that one can drop this requirement, by having each agent use the message from the previous round together with the current round's. This approach, however, is not exactly equivalent to the original consensus protocol, thus requires a new analysis. (E.g., even if agents only update their state every other round, they need to do that simultaneously to achieve equivalence, but a synchronized binary clock is needed for that.) More importantly, the approach is unnecessarily complicated, requiring extra space in addition to the bit counter.

We present a surprisingly simple protocol for the problem: The state of an agent is just one bit, the actual clock. Whenever an agent u with clock 1 samples an agent v with clock 0, u changes its clock to 0. Also at the end of each round every agent increments its clock mod 2, i.e., flips its bit value. The next statement gives the properties of the protocol.

THEOREM 1.1. *There is a self-stabilizing algorithm for synchronizing a binary clock, which uses 1-bit messages and 2 states, and converges in $O(\log n)$ rounds w.h.p.*

Our algorithm has a superficial similarity to the well-known voter model [35], where each agent has a binary state, and in each round each agent copies the state of a random agent. It is not hard to see that the dynamics of our algorithm is equivalent to that of a slightly modified voter model, where agents in state 1 update their state in odd rounds, and agents in state 0 in even rounds (this alternation is realized in our algorithm by flipping the agent states at the end of each round). This seemingly small modification has a dramatic affect on the convergence time, as the standard voter model converges in expected $\Theta(n)$ rounds [2].

Unlike the voter model, and similarly to the stabilising consensus algorithms with two messages, mentioned earlier, the process underlying our algorithm has a single fixed point (which, however, is not $n/2$), and a slight deviation from that point creates a bias away from the point, which increases the closer we move to 0 or n. Our analysis follows a similar line as that of [18].

Next we focus on extending our algorithm to synchronize a mod T clock, for a small integer $T \geq 3$, as that can be used to directly improve the algorithm of [16]. We devise a simple algorithm for synchronizing a mod 4 clock: The state of each agent consists of a 2-bit string $b_1 b_0$, that is the binary representation of the mod 4 clock. The message of an agent is just its most significant bit, b_1. Whenever an agent u with $b_1^u = 1$ samples an agent v with $b_1^v = 0$, u flips both its clock bits, i.e., $b_1^u \leftarrow 1 - b_1^u$ and $b_0^u \leftarrow 1 - b_0^u$. Also at the end of each round every agent increments its clock mod 4. The next statement gives the properties of the protocol.

THEOREM 1.2. *There is a self-stabilizing algorithm for synchronizing a modulo 4 clock, which uses 1-bit messages and 4 states, and converges in $O(\log n)$ rounds w.h.p.*

Though not immediately obvious, there is a simple connection between the new algorithm and our first algorithm, which reduces the analysis of the former to the latter. If we consider only every other round of

[1] *With high probability (w.h.p.)* means with probability at least $1-O(n^{-c})$, for a constant $c > 0$ that can be made arbitrarily large at the cost of the other constants involved (e.g., the constant factor in the logarithmic number of rounds, in the case above).

an execution of the mod 4 algorithm, and look just at the most significant bit of the agents, the observed process is distributed identically to an execution of the mod 2 algorithm. It follows from Theorem 1.1 that after $2 \cdot O(\log n)$ rounds, all agents agree on the most significant clock bit w.h.p. Applying this argument twice (in parallel) starting from rounds 0 and 1, yields the desired logarithmic convergence time.

The two algorithms we have presented above are space- and message-optimal. They are also extremely simple, and thus, it is plausible that they are relevant to some biological or other natural processes, although we do not currently have any results in that direction.

We use the two algorithms as building blocks to implement a (more sophisticated) synchronization algorithm for a general modulo T clock. The high level approach is the same as in Boczkowski et al. [16]. We use a recursive construction, with $\log^* T$ layers. The first layer consists of two bits, and each subsequent layer consists of (roughly) the maximum number of bits that can be indexed using the bits of the previous layer. The bits in all layers taken together constitute the mod T clock, with the bits in the first layer being the least significant ones. Each message consists of two bits, one for synchronizing the mod 4 clock of the first layer (using our second algorithm above), and one bit with the value of the bit indexed by the first non-zero layer. This bit is then updated using our binary clock synchronization algorithm (instead of a consensus algorithm as done in [16]). The precise way that the binary clock synchronization algorithm is used is slightly subtle, as it takes into account the frequency with which each bit in the clock increases, and the frequency in which it is updated. A detailed description of the construction is given in Section 6. The next statement summarizes the main properties of the protocol.

THEOREM 1.3. *There is a self-stabilizing algorithm for synchronizing a modulo T clock, for any T that is a power of 2, which uses 2-bit messages and T states, and converges in $\tilde{O}(\log n \log T)$ rounds w.h.p.*[2]

We also address the case where T is not a power of 2, but we discuss that later, in Remark 1.1, as it relies on some additional results.

Next we provide two general constructions that, when applied to the algorithm above, further reduce the message size. First, we introduce some terminology. The *τ-clocked model* is an extension of the (standard) model we have considered so far, equipped with a shared modulo τ clock, i.e., all agents have a consistent numbering of the rounds mod τ; other than that, the agents' initial state can be arbitrary, as in the standard model. An algorithm S *simulates* (in the standard model) an algorithm A for the τ-clocked model, with *delay* d and *slowdown* s, if, roughly speaking, after at most d steps, S achieves the shared clock abstraction, and from then on all agents execute (in sync) a round of A once in every s rounds of S. In all statements below, the slowdown is constant and is a power of 2.

The algorithm of Theorem 1.3, which uses 2-bit messages, can be directly converted into an algorithm for the 2-clocked model with 1-bit messages: each message of the original algorithm is split into two 1-bit messages, sent in an odd and the next even round, while each agent updates its state only in even rounds. This approach works because our analysis does not require that the two bits are received from the same agent, a property termed *bitwise-independence* in [16].

Our first construction achieves the following result.

THEOREM 1.4. *Any protocol A for the 2-clocked model using 1-bit messages and σ states, can be simulated in the standard model using a message space of size 3 and 4σ states, with delay $O(\log n)$ w.h.p. and constant slowdown.*

The algorithm for simulating A in Theorem 1.4 is a simple adaptation of our modulo 4 clock synchronization algorithm. Agents simulate A when the most significant bit of their mod 4 clock is $b_1 = 1$ and use bit b_0 as the shared mod 2 clock value. When $b_1 = 0$ the message the agent sends is 0, as in the standard mod 4 clock protocol, and when $b_1 = 1$ it sends the corresponding message of the simulated algorithm A *increased by one* (i.e., the message is either 1 or 2). Thus an agent with $b_1 = 1$ which receives message $\mu \in \{0, 1, 2\}$, updates its mod 4 clock if $\mu = 0$, and updates its local state of the simulated algorithm A if $\mu \neq 0$. The operation of the mod 4 clock synchronization algorithm is not affected by these changes, and once all clocks are in sync, agents correctly simulate A, twice every 4 rounds.

Next we show how to reduce the message size to a single bit. At a high level the protocol is divided into two phases, one for synchronizing a mod 4 clock, whose role is similar to the clock's in the previous construction, and a second phase which assumes already synchronized clocks, and is when the actual simulation takes place. An agent stays in the first phase for a logarithmic number of rounds before switching to the second, and from the second phase it moves back to the first if it has an indication that some agent is out of sync. Unlike all our previous results, this requires that agents know a rough upper bound on $\log n$, which is hardcoded into

[2] The tilde notation hides factors that are at most polynomial in $\log\log T$ and $\log\log n$.

the algorithm.[3] A detailed description of the algorithm and explanation of its various subtle details is given in Section 8.

Theorem 1.5. *If each agent knows a linear upper bound on* $\log n$*, then any protocol A for the 2-clocked model using 1-bit messages and σ states, can be simulated in the standard model using 1-bit messages and* $\Theta(\sigma \log n)$ *states, with delay* $O(\log n)$ *w.h.p. and constant slowdown.*

Applying Theorem 1.5 to the algorithm of Theorem 1.3, which, as we pointed out, can be transformed to an equivalent algorithm for the 2-clocked model with 1-bit messages (by the bitwise-independence property), we obtain the following.

Corollary 1.1. *There is a self-stabilizing algorithm for synchronizing a modulo T clock, for any T that is a power of 2, which uses 1-bit messages and* $\Theta(T \log n)$ *states, and converges in* $\tilde{O}(\log n \log T)$ *rounds w.h.p.*[4]

Using the construction of Theorem 1.5 and the algorithm of Corollary 1.1, we can efficiently simulate any algorithm A for the T-clocked model: In the 2-clocked model, one can simulate A by running the mod T clock synchronization algorithm in odd rounds, and A in even rounds using that clock instead of a shared clock. The resulting algorithm can then be simulated in the standard model.

Corollary 1.2. *Any protocol A for the T-clocked model, where T is a power of 2, that uses 1-bit messages and σ states, can be simulated in the standard model using 1-bit messages and* $\Theta(\sigma T \log^2 n)$ *states, with delay* $\tilde{O}(\log n \log T)$ *w.h.p. and constant slowdown.*

In a similar way, we can use Theorem 1.5 (or Theorem 1.4) and Corollary 1.1 to simulate any k-bit message protocol A for the T-clocked model, by a k-bit message protocol in the standard model.

So far we have assumed that T is a power of 2. We can easily extend Corollaries 1.1 and 1.2 to the case in which T is not a power of 2 as follows. Boczkowski et al. [16] provided a simple and clever way to implement a mod T clock synchronization algorithm in the τ-clocked model, where $\tau = \Theta(\log n \log T)$ is a power of 2. The algorithm uses T states and 1-bit messages, and converges in $O(\log n \log T)$ rounds w.h.p. Simulating this algorithm using Corollary 1.2, we obtain an extension of Corollary 1.1 to arbitrary T. Combining then Theorem 1.5 with the resulting algorithm, we can obtain a similar extension of Corollary 1.2. The next remark gives the precise changes we need make to the statements of Corollaries 1.1 and 1.2.

Remark 1.1. *For any integer $T \geq 2$ that is not a power of 2, Corollary 1.1 still holds if we increase the number of states by factor* $\tau = \Theta(\log n \log T)$*, from* $\Theta(T \log n)$ *to* $\Theta(T \log^2 n \log T)$*. Similarly, if T is not a power of 2, Corollary 1.2 still holds if we increase the number of states from* $\Theta(\sigma T \log^2 n)$ *to* $\Theta(\sigma T \log^3 n \log T)$*.*

Corollary 1.1 (together with Remark 1.1) settles the open problem posed by Boczkowski et al. [16], establishing that fast clock synchronization is possible even when just single bit messages are used, and even when each agent receives a single message per round, from a random agent.

Boczkowski et al. [16] showed the following *message reduction theorem*. Let $\mathcal{A}(\eta, \ell)$ denote the class of all algorithms in a variant of our model, where each agent receives η messages per round (from η random agents), and ℓ is the message size. The theorem says that any algorithm in $\mathcal{A}(\eta, \ell)$ with the *bitwise-independence property* (i.e., each bit of each message can be received from a different random agent), can be simulated in $\mathcal{A}(3, 2)$. Using the construction of Theorem 1.5 and the algorithm of Corollary 1.1, and applying a similar reasoning as for Corollary 1.2, we can directly strengthen the message reduction theorem to allow simulation in the $\mathcal{A}(1, 1)$ model, with the same overhead as in the original theorem.

Due to space limitations, the analysis of the binary clock synchronization algorithm of Theorem 1.1, and the formal description and analysis of the simulator of Theorem 1.4 are omitted from this text, and can be found in the full version of the paper [10].

2 Related Work

The problem of self-stabilizing clock synchronization in fully-connected synchronous systems, has been studied extensively in the Byzantine failure model, under the names *digital clock synchronization* and *synchronous counting* [13, 19, 20, 22, 33, 34]. The goal has been to achieve resilience to the optimal 1/3 fraction of Byzantine agents, while at the same time minimizing the number of rounds and the message size. This requires significantly more communication than in our model, and typically all-to-all communication is employed in each round. Without Byzantine faults, the same prob-

[3] Having some knowledge of n is a common assumption in the literature, e.g., the final synchronization algorithm of [16] also uses an upper bound on $\log n$.

[4] No slowdown is involved, because simulating a modulo T clock with slowdown s gives a modulo sT clock. And from a modulo sT clock, by taking the mod T of the clock value, one trivially obtains a modulo T clock.

lem has been studied as the *self-stabilizing unison problem* [9,17,29]. The unison problem assumes an arbitrary underlying graph, unlike the fully-connected setting we consider in our work, and every process can read the state of all its neighbors in each round.

Clock synchronization is a fundamental task in the model of *population protocols* [5, 8]. This is an asynchronous model, where a random pair of agents interact in each step, by observing each other's state before updating their state. The synchronization task here is to implement a *phase clock* abstraction [7], which allows agents to collectively count time in phases of $\Theta(n \log n)$ interactions, with bounded skew. Efficient phase clock algorithms have been proposed in [3,27], and self-stabilizing algorithms based on oscillation dynamics were proposed in [23,32]. Very recently, a variant of the standard population protocol model was proposed [4], where interacting agents can only observe a function of the other agent's state, similarly to our model. Among other results, they proposed a phase clock implementation using 1-bit messages.

The *beeping* model [1,24] is another model in which communication is severely restricted. There is typically an arbitrary underlying graph, and in each round each agent can either send a 'beep' to all its neighbors, or stay silent. Optimal clock synchronization algorithms have been proposed recently [26], assuming arbitrary activation times, which is a weaker property than self-stabilization.

The stochastic process underlying our binary clock algorithm, behaves similarly to consensus dynamics that have been studied recently under the names of *stabilizing consensus*, *majority consensus* (if there are two opinions), or *plurality consensus* (for $k > 2$ opinions). Typically, in these algorithms, each agent observes the opinions of two (or more) random agents in each round and updates its opinion as a function of the observed opinions and its own, e.g., adopting the median value, or the majority [11, 14, 18, 28]. An interesting variant, where each agent observes the opinion of only one other agent per round, called the *undecided state dynamics*, was analysed in [6, 12]. A majority algorithm that uses push communication, and is optimal for a failure model where 1-bit messages are flipped independently with constant probability, was proposed in [25].

In [16], a self-stabilizing majority protocol is described, which converges in $\tilde{O}(\log n)$ rounds w.h.p., uses 3 bits-messages, and each agent receives two messages per round (from random agents). Our improvement to the message reduction theorem of [16], directly implies that the message size of the above protocol can be reduced to 1-bit, and the number of messages received to one, without hurting the convergence time.

3 Model

We assume a synchronous system of n agents that execute a protocol in a sequence of parallel rounds. The set of all agents is denoted N. In each round t, every agent u can sample one agent v uniformly at random, and receive information about v's state. Precisely, u receives a *message* $m(s_v)$, where s_v is v's state prior to round t, and m is a function specified by the protocol, whose domain is the agent state space. We call m the protocol's *message function*. After receiving $m(s_v)$, u updates its state according to the protocol, completing the round. We assume the initial state of each agent (before the first round) is arbitrary.

We refer to the model described above as the *standard model*. We also consider an extension of this model, equipped with a *shared* modulo τ clock. Precisely, at the beginning of each round, the current value of the shared clock is broadcast to each agent. The initial value of the shared clock (at round 0) can be an arbitrary value in $\{0, \ldots, \tau - 1\}$. We will refer to this model as the *τ-clocked model*.

For a protocol A in the standard model, we write $A(s, msg)$ to denote the state of an agent after a round of the protocol, when the agent's state prior to the round is s, and the message it receives is msg. Similarly, if A is a protocol in the τ-clocked model, we define $A(s, k, msg)$ as above, except that we additionally assume that the shared clock value in the round is k.

A protocol B *simulates* protocol A if for a random execution of B (from any initial configuration), there is some round d such that after d, all agents execute in sync a rounds of A in every b rounds of B periodically. (For a formal definition of a simulation see [10].) We call B a *simulator* of A. The efficiency of the simulator is measured in terms of d and the ratio $b : a$, which we call *delay* and *slowdown*, respectively. We will use simulators to simulate in the standard model a protocol for the τ-clocked model, or to simulate a protocol in the same model but using shorter messages.

For a binary string s of length k, we write $\text{val}(s)$ to denote the value of the string interpreted as a binary number, and write $\text{incr}(s)$ to denote the string with value $\text{val}(s) + 1 \bmod 2^k$.

4 Binary Clock

In Algorithm 1, we present a simple binary clock synchronization protocol. The state of each agent is a single bit $b \in \{0, 1\}$, which is the clock value of the agent. In each round, first, every agent u whose clock is 1 samples a random agent v, and if v's clock is 0 then u sets its own clock to 0 (thus the message function is just $m(b) = b$). Then, every agent increases its clock modulo 2. The algorithm satisfies Theorem 1.1, i.e., uses 1-

Algorithm 1: Binary clock synchronization protocol.

state: $b \in \{0,1\}$ `// binary clock value`

foreach *round* t **do**
  if $b = 1$ **then**
    `// copy the clock of a random agent`
    sample an agent v u.a.r., and let b' be v's state prior to round t
    $b \leftarrow b'$
  $b \leftarrow 1 - b$ `// increment clock mod 2`

bit messages and 2 states, and converges in $O(\log n)$ rounds w.h.p. The proof of the convergence time bound is similar to the analysis in [18], and can be found in [10].

5 Modulo 4 Clock

Next we present a mod 4 clock synchronization protocol. The pseudocode is given in Algorithm 2. The state of each agent consists of two bits b_1, b_0. These correspond to the bits in the binary representation of the modulo 4 clock, with b_1 being the most significant bit. Thus the clock value is $2b_1 + b_0$. In each round, first, every agent u with $b_1 = 1$ samples a random agent v, and if v's most significant bit is 0, then u flips both its bits (thus the message function is $m(b_1, b_0) = b_1$). After that, every agent increases its clock modulo 4. The algorithm satisfies the state space and message size requirements of Theorem 1.2. Below we prove the logarithmic bound on the convergence time.

THEOREM 5.1. *Starting from any initial configuration, Algorithm 2 synchronizes all modulo 4 clocks in $O(\log n)$ rounds w.h.p.*

Proof. Let N^t_{ij} denote the set of agents whose clock bits are $b_1 = i$ and $b_0 = j$ right after the first t rounds. The state transitions in round $t+1$ are then as follows:

Algorithm 2: Modulo 4 clock synchronization protocol.

state: $b_1, b_0 \in \{0,1\}$ `// clock value:` $2b_1 + b_0$

foreach *round* t **do**
  if $b_1 = 1$ **then**
    sample an agent v u.a.r., and let b'_1 be v's most significant state bit prior to round t
    if $b'_1 = 0$ **then**
      `// flip both bits`
      $b_1 \leftarrow 1 - b_1$; $b_0 \leftarrow 1 - b_0$
  $b_1 b_0 \leftarrow \mathrm{incr}(b_1 b_0)$ `// increment clock mod 4`

1. If $u \in N^t_{00}$, then $u \in N^{t+1}_{01}$.
2. If $u \in N^t_{01}$, then $u \in N^{t+1}_{10}$.
3. If $u \in N^t_{10}$, then $u \in N^{t+1}_{10}$ if u samples an agent from $N^t_{00} \cup N^t_{01}$, and $u \in N^{t+1}_{11}$ otherwise.
4. If $u \in N^t_{11}$, then $u \in N^{t+1}_{01}$ if u samples an agent from $N^t_{00} \cup N^t_{01}$, and $u \in N^{t+1}_{00}$ otherwise.

Let Z_{t+1} be the set of agents $u \in N^t_{10} \cup N^t_{11}$ that sample an agent from $N^t_{00} \cup N^t_{01}$ in round $t+1$. It follows

$$\begin{aligned} N^{t+1}_{01} \cup N^{t+1}_{10} &= N^t_{00} \cup N^t_{01} \cup Z_{t+1} \\ N^{t+1}_{10} \cup N^{t+1}_{11} &= N^t_{01} \cup N^t_{10}. \end{aligned} \tag{5.1}$$

CLAIM 5.1. *Let $y_t = |N^t_{10} \cup N^t_{11}|/n$. W.h.p., $y_t \in \{0,1\}$ for all $t \geq c \log n$, for some constant c.*

Proof. From (5.1),

$$N^{t+2}_{10} \cup N^{t+2}_{11} = N^{t+1}_{01} \cup N^{t+1}_{10} = N^t_{00} \cup N^t_{01} \cup Z_{t+1}.$$

Let $z_{t+1} = |Z_{t+1}|/n$. The above equation then implies

$$y_{t+2} = (1 - y_t) + z_{t+1}.$$

It follows that for any $y, y' \in \{0, 1/n, 2/n, \ldots, n/n\}$,

$$\begin{aligned} \Pr[y_{t+2} = 1 - y + y' \mid y_t = y] &= \Pr[z_{t+1} = y' \mid y_t = y] \\ &= \Pr[B(ny, 1-y) = ny'], \end{aligned}$$

since $|Z_{t+1}| \sim B(ny, 1-y)$ given $y_t = y$. We can also easily show that

$$\Pr[x_{t+1} = 1 - y + y' \mid x_t = y] = \Pr[B(ny, 1-y) = ny'],$$

where x_t is the fraction of agents whose binary clock is 1 after executing the first t rounds of Algorithm 1. Thus,

$$\Pr[y_{t+2} = z \mid y_t = y] = \Pr[x_{t+1} = z \mid x_t = y],$$

for any z. Since the sequences $(x_t)_{t \geq 0}$ and $(y_{2t})_{t \geq 0}$ are Markov chains, the equation above implies they have the same distribution, if they start from the same initial state. From Theorem 1.1, we have that w.h.p. $x_t = \{0,1\}$ for all $t \geq c' \log n$, for some constant c', and any initial configuration. It follows $y_{2t} = \{0,1\}$ for all $t \geq c' \log n$, w.h.p., for any initial configuration. Moreover by taking as initial configuration the one reached after the first round, we obtain also that $y_{2t+1} = \{0,1\}$ for all $t \geq c' \log n$, w.h.p. Therefore, by a union bound, w.h.p., $y_t = \{0,1\}$ for all $t \geq 2c' \log n + 1$. ∎

CLAIM 5.2. *If $y_t \in \{0,1\}$ and $y_{t+1} \in \{0,1\}$, then $|N^t_{ij}| = n$ for some pair i, j.*

Proof. We have $|N^t_{10} \cup N^t_{11}| = ny_t$, and from (5.1),

$$|N^t_{01} \cup N^t_{10}| = |N^{t+1}_{10} \cup N^{t+1}_{11}| = ny_{t+1}.$$

If $y_t = y_{t+1} = 0$, then the above equations imply that $|N^t_{10}| = |N^t_{11}| = |N^t_{01}| = 0$, thus $|N^t_{00}| = n$. If $y_t = y_{t+1} = 1$, then $|N^t_{00} \cup N^t_{01}| = n(1 - y_t) = 0$ and $|N^t_{11} \cup N^t_{00}| = n(1 - y_{t-1}) = 0$, thus $|N^t_{10}| = n$. Similarly, if $y_t = 0, y_{t+1} = 1$ we obtain $|N^t_{01}| = n$, and if $y_t = 1, y_{t+1} = 0$ then $|N^t_{11}| = n$. ◻

From Claims 5.1 and 5.2, there is a $t = \Theta(\log n)$ such that $|N^t_{ij}| = n$ w.h.p. A simple inductive argument shows that for all $t' > t$, no agent with $b_1 = 0$ is sampled in round t', and all clocks have value $2i+j+(t'-t) \bmod 4$ after the round. The theorem then follows. ◻

6 Modulo T Clock

We present a protocol that uses 2-bit messages to synchronize a mod T clock, where T is a power of 2.

6.1 Protocol Description. Let $T = 2^k$, and suppose that $k \geq 3$, as Sections 4 and 5 already give 1-bit message protocols for $k \in \{1, 2\}$. The state of each agent $u \in N$ simply consists of exactly k bits necessary to store the clock value. We denoted the clock by a bit-string $C = b_{k-1} \dots b_0$ of length k, where b_0 is the least significant bit. We represent C as a concatenation of its substrings $Q_h, Q_{h-1}, \dots, Q_1$, which we call *sub-clocks*. The lengths of the sub-clocks are given using a sequence of functions $(\rho_i(l))_{i \geq 0}$ defined on integers $l \geq 2$ as follows. Let $\rho(l) = \lceil \log_2(l+1) \rceil$. Then $\rho_i(l)$ is the iterative application of ρ on l, i times, i.e.,

$$\rho_i(l) = \begin{cases} l, & \text{if } i = 0, \\ \rho(\rho_{i-1}(l)), & \text{if } i \geq 1. \end{cases} \tag{6.2}$$

For any $l \geq 2$, we define $\nu(l) = \min\{i \geq 0 \colon \rho_i(l) = 2\}$, which is the number of iterations until the sequence reaches its fixed point $\rho(2) = 2$. Next we define

$$\lambda = \min \left\{ l \geq 2 \colon \sum_{i=0}^{\nu(l)} \rho_i(l) \geq k \right\}. \tag{6.3}$$

The number of sub-clocks is then $h = \nu(\lambda) + 1$. For $i \in \{1, \dots, h-1\}$, the length of the sub-clock Q_i is $l_i = \rho_{h-i}(\lambda)$, and the sub-clock Q_h contains the remaining $l_h = k - \sum_{i=1}^{h-1} l_i$ bits. For convenience, we define $s_0 = 0$, and $s_i = s_{i-1} + l_i$ for $1 \leq i \leq h$. Thus, for the ith sub-clock we have $Q_i = b_{s_i - 1} \dots b_{s_{i-1}}$. We will write $\oplus$ to denote the standard XOR operator, i.e., for $x, y \in \{0, 1\}$, $x \oplus y = 1$ if and only if $x \neq y$. Using this notation, the pseudocode of the mod T clock synchronization protocol is given in Algorithm 3.

Algorithm 3: Modulo $T = 2^k$ clock synchronization protocol.

---

state: clock $C = b_{k-1} \dots b_0$, also represented by the sub-clocks $Q_h, \dots, Q_1$, where $Q_i = b_{s_i - 1} \dots b_{s_{i-1}}$, for $1 \leq i \leq h$

msg function: $m(C)$ is the 2-bit string $b_1 b_{\pi(C)}$ if $\pi(C) \neq \bot$, and $b_1 0$ otherwise

1 **foreach** *round* t **do**
2   sample an agent v u.a.r., and let $\mu_1 \mu_2$ be the 2-bit message received from v
  `// sync. mod 4 clock` $Q_1 = b_1 b_0$ `using` μ_1
3   **if** $\mu_1 = 0$ **and** $b_1 = 1$ **then**
4     $b_1 \leftarrow 1 - b_1$; $b_0 \leftarrow 1 - b_0$
5   $p \leftarrow \pi(C)$ `// index of bit to sync.`
6   **if** $p \neq \bot$ **then**
    `//` i `is the same as in Line 14`
7   let i be such that $p \in [s_i, s_{i+1} - 1]$
    `// condition for updating` b_p
8   **if** $(s_i = p$ **and** $b_p = 1)$ **or**
9      $(s_i < p$ **and** $b_p \oplus b_{s_i} = 1$ **and**
10       $b_{p'} = 0$ *for some* $s_i \leq p' < p)$ **then**
11       $b_p \leftarrow \mu_2$
12   $C \leftarrow \mathrm{incr}(C)$

13 **function** $\pi(C)$
  `//` i `: index of first non-zero sub-clock`
14   $i \leftarrow \min\{i \geq 1 \colon \mathrm{val}(Q_i) \neq 0\} \cup \{+\infty\}$
15   **if** $i < h$ **and** $\mathrm{val}\,(Q_i) \leq l_{i+1}$ **then**
16   **return** $s_i + \mathrm{val}\,(Q_i) - 1$
17   **else return** $\bot$

We will write C^u to refer to the clock of agent u, and $C^{u,t}$ to refer to the clock's value right after round t. The same notation is used for variables Q_i and b_j.

The algorithm synchronizes the sub-clocks Q^u_i in the increasing order of i. The first sub-clock, Q^u_1, is a modulo 4 clock, which uses the first bit μ_1 of the message for synchronization, in Lines 3 to 4. The synchronization of this clock happens exactly as in Algorithm 2. The second bit μ_2 of the message is used for synchronizing the rest of the bits of the clocks. For each round, agent u uses $\pi(C^u)$ to determine the bit to synchronize in that round. Let i be the index of the first non-zero sub-clock ($i = +\infty$ if all sub-clocks are zero). If $i < h$ and $q = \mathrm{val}(Q^u_i) \leq l_{i+1}$, then we synchronize the qth least significant bit of sub-clock Q^u_{i+1} (Line 16 returns its index in C^u). By the construction of the sub-clocks, $2^{l_i} - 1 \geq l_{i+1}$, thus, $\mathrm{val}(Q^u_i)$ suffices to index all the bits of Q^u_{i+1}. If the condition in Line 15 fails, i.e., $i \geq h$ or $\mathrm{val}(Q^u_i) > l_{i+1}$, no bit synchronization takes place in that round.

Now suppose $p = \pi(C^u) \geq 0$ at some round. Under the conditions in Lines 8 to 10, the bit b_p^u adopts the received value μ_2. Notice that if the bits at indices $0, \dots, p$ are synchronized, then b_p^u does not change as a result. Additionally, if for some $p' < p$, the bit at index p' is not the same in all clocks, then in the analysis we ignore the update of b_p^u. In other words, we analyse the synchronization of the bits one by one, starting from the least significant bit.

Therefore, we consider the case in which the bits up to index $p-1$ are synchronized at rounds $t \geq t_0$. In this case, $\pi(C^{u,t}) = p$ implies $\pi(C^{v,t}) = p$ for all agents v, and thus, Line 11 synchronizes bits at index p of all clocks.

We argue that the synchronization is completed in at most $\tilde{O}(\log T \cdot \log n)$ rounds. Our first observation is that the period of the clock Q_i^u (i.e., the number of rounds between two consecutive resets of Q_i^u) is 2^{s_i}. This implies that if $\pi(C^{u,t}) = p$, then $\pi(C^{u,t'}) = p$ for $t' = t + 2^{s_i}$. It is not hard to compute that $2^{s_i} = \tilde{O}(\log T)$, since $i < h$.

Now observe that in the case in which $p = s_i$ in some round t, the condition on Line 8 is equivalent to that of Line 2 of Algorithm 1. Additionally, in the next 2^{s_i} rounds, as a result of clock increments on Line 12, the bit b_p^u will be flipped exactly once. It follows that the bits at index p of all clocks, considered only in rounds t when $\pi(C^{u,t-1}) = p$, emulate the modulo 2 clock of Algorithm 1. By Theorem 1.1 and the observation that $\pi(C^u) = p$ every $\tilde{O}(\log T)$ rounds, we conclude the clock bits of index p are synchronized in $\tilde{O}(\log T \cdot \log n)$ rounds, w.h.p.

If $s_i < p$ in round t, the condition of the update on Lines 9 and 10 is more subtle. We argue that the bit $c^{u,t-1} = b_p^{u,t-1} \oplus b_{s_i}^{u,t-1}$, when considered at *almost* all rounds t with $\pi(C^{u,t-1}) = p$, emulates the binary clock from Algorithm 1. Precisely, we exclude the rounds t with $\pi(C^{u,t-1}) = p$ for which

$$b_{s_i}^{t-1} = \cdots = b_{p-1}^{t-1} = 1. \tag{6.4}$$

We observe that when (6.4) does not hold, the bit $b_{s_i}^{u,t-1}$ flips exactly once in the next 2^{s_i} rounds due to the increments on Line 12, whereas b_p^u does not flip, and so bit c^u flips exactly once, emulating Line 5 of Algorithm 1. On the other hand, if (6.4) holds, then in the next 2^{s_i} rounds, *both* $b_{s_i}^u$ and b_p^u flip once due to Line 12, and so the bit c^u we consider will not flip. Therefore, the condition on Lines 9 and 10, which prevents updating b_p^u in the last case, ensures that bit c^u correctly emulates an execution of Algorithm 1. Again by Theorem 1.1, we conclude that the bits of the clocks at index p are synchronized in $\tilde{O}(\log T \cdot \log n)$ rounds, w.h.p. This implies that by considering the $k = \log T$ bits one by one (starting from the least significant one), we can prove that the clocks are synchronized in $\tilde{O}(\log^2 T \cdot \log n)$ rounds.

We can improve upon this sequential argument, by observing that for some $z = s_{h-1} + \log\log n + \Theta(1)$, the bits at indices $z+1, \dots, k-1$ become synchronized *simultaneously* in $\tilde{O}(\log T \cdot \log n)$ rounds after the bits at indices $0, \dots, z$ are synchronized. Since $z = O(\log\log T + \log\log n)$, almost all k bits become synchronized in parallel. Thus, by analysing the synchronization time of just the $(z+1)$ least significant bits, in a sequential order, we reduce the bound on the total synchronization time to $\tilde{O}(\log T \cdot \log n)$.

6.2 Properties. The protocol uses 2-bit messages and $T = 2^k$ states, as promised by Theorem 1.3. Next we provide a detailed convergence analysis.

Theorem 6.1. *For any $T \geq 8$ that is a power of 2 and any initial configuration, Algorithm 3 synchronizes all mod T clocks in $O(\log T \log n \cdot (\log\log T)^3 \log\log(nT))$ rounds, w.h.p.*

Remark 6.1. *Algorithm 3 satisfies the bitwise-independence property, as defined in [16]. This is because the two bits $\mu_1\mu_2$ that an agent receives in one round serve different purposes: one for synchronizing Q_1, and the other for synchronizing the rest of the sub-clocks. In particular, the proof of Theorem 6.1 remains the same if an agent receives the two bits of its message from two randomly selected agents.*

6.3 Notation. For any round $t \geq 0$, let $r_t \leq k$ be the largest integer such that the r_t least significant bits of all agents' clocks are the same immediately after round t, i.e., r_t equals

$$\max\{r \in \mathbb{N} : \; b_j^{u,t} = b_j^{v,t} \text{ for all } u, v \in N \text{ and } 0 \leq j < r\}.$$

For an arbitrary agent u, let $R_t = b_{r_t-1}^{u,t} \dots b_0^{u,t}$, which is the longest common suffix of the binary representations of all clocks.

Let $\mathcal{E}_T$ denote an execution of Algorithm 3, which also describes the uniformly random choices that the agents make in each round. We will also consider an execution $\mathcal{E}_2$ of Algorithm 1 which synchronizes a binary clock β. For an agent u and $j \geq 0$, let $\beta^{u,j}$ be the value of the binary clock of u immediately after round j of $\mathcal{E}_2$. (Unlike in Algorithm 1, we use β instead of b in order to avoid confusion with the mod 4 clock in Algorithm 3).

6.4 Analysis.

Lemma 6.1. *The mod 4 clocks $Q_1^u = b_1^u b_0^u$ are synchronized after at most $O(\log n)$ rounds, w.h.p., and stay synchronized thereafter.*

Proof. From the definition of π on Lines 13 to 17, it follows that for any agent u, $\pi(C^u) \geq 2$ or $\pi(C^u) = \perp$. In particular, if the condition in Line 15 holds then the expression in Line 16 is at least s_i, as $\text{val}(Q_i) \neq 0$, and $s_i \geq s_1 = l_1 = 2$. It follows that Lines 5 to 11 never affect the two least significant bits of clock C^u. The remaining lines of Algorithm 3 are identical to the mod 4 clock presented in Algorithm 2. Therefore the lemma is proved by Theorem 5.1. ∎

LEMMA 6.2. *For any $t \geq 0$, if $r_t \geq 2$ then (a) $r_{t+1} \geq r_t$; (b) $\text{val}(R_{t+1}) \equiv \text{val}(R_t) + 1 \pmod{2^{r_t}}$.*

Proof. Since $r_t \geq 2$, the 2-bit sub-clock Q_1 is synchronized, thus from Lemma 6.1 it follows that $r_{t+1} \geq 2$. Therefore, we just need to prove that the updates on Lines 5 to 11 keep the r_t least significant bits synchronized.

For an agent u, let $p_u = \pi(C^{u,t})$ as on Line 5 in round $t+1$. Fix u such that $p_u \neq \perp$, if such u exists, and let $i \geq 1$ be such that $p_u \in [s_i, s_{i+1} - 1]$ as in Line 7. If $p_u < r_t$, then the sub-clocks $Q_1, \ldots, Q_i$ are synchronized in all agents, and, in particular, for any agent v, $\text{val}(Q^u_{i'}) = \text{val}(Q^v_{i'}) = 0$ for $1 \leq i' < i$ and $\text{val}(Q^u_i) = \text{val}(Q^v_i) \neq 0$. This implies that $p_v = \pi(C^{v,t}) = p_u < r_t$. Therefore, the second bit received by all agents in round $t+1$ is the same and is equal to $\mu_2 = b^{u,t}_{p_u}$. This implies that after executing Lines 7 to 11, the r_t least significant bits of the clocks remain synchronized. Finally, Line 12 is a simple incrementing operation which preserves the above property and implies that $r_{t+1} \geq r_t$ in this case.

If $p_u \geq r_t$, then either $p_v = \perp$ or $p_v \geq r_t$ for all agents v (otherwise, the argument above with respect to v gives a contradiction). In this case (and in the remaining case when $p_u = \perp$ for all u), p_u and p_v may not be equal for some two agents u and v, but the Lines 5 to 11 do not modify the r_t least significant bits of the C^u and C^v. Once again, this implies that $r_{t+1} \geq r_t$, completing the proof of (a).

Finally, (b) holds due to Line 12 and the fact that in round $t+1$ Lines 3 to 11 do not change any of the first r_t bits of any agent clock. ∎

LEMMA 6.3. *If $2 \leq r_{t_0} < k$ for some $t_0 \geq 0$, then there is a round $t = t_0 + O(2^{s_{h-1}} \cdot \log n)$ such that $r_t \geq r_{t_0} + 1$, w.h.p.*

Proof. For conciseness we set $p = r_{t_0}$. We need to bound the number of rounds until the bit at index p of the clocks is synchronized. Let i be such that $p \in [s_i, s_{i+1} - 1]$. This implies that for any two agents u, v, if $t \geq t_0$ and $1 \leq i' \leq i$, then $Q^{u,t}_{i'} = Q^{v,t}_{i'}$.

Fix an agent u and consider a round $t \geq t_0$.By the definition of function π, $p = \pi(C^{u,t})$ if and only if for $1 \leq i' < i$, $\text{val}(Q^{u,t}_{i'}) = 0$ and $\text{val}(Q^{u,t}_i) = p - s_i + 1$, or equivalently, if

$$\text{val}(C^{u,t}) \equiv (p - s_i + 1) \cdot 2^{s_{i-1}} \pmod{2^{s_i}}.$$

By Lemma 6.2, $r_t \geq r_{t_0} \geq s_i$, so the s_i least significant bits of $C^{u,t}$ and R_t are identical, i.e.,

$$\text{val}(C^{u,t}) \equiv \text{val}(R_t) \equiv \text{val}(R_{t_0}) + t - t_0 \pmod{2^{s_i}}.$$

Combining the last two equations above, we get that $p = \pi(C^{u,t})$ if and only if $t = t_j$ for some $j \geq 1$, where $t_1 = \min\{t \geq t_0 : \pi(C^{u,t}) = p\} < t_0 + 2^{s_i}$, and $t_{j+1} = t_j + 2^{s_i}$ for $j \geq 1$. Moreover, for any other agent v, if $t \geq t_0$, then it is also the case that $p = \pi(C^{v,t})$ if and only if $t = t_j$ for some $j \geq 1$. In other words, the values $(t_j)_{j \geq 1}$ are universal among agents. We consider two cases.

Case $s_i = p$. In this case b_p is the least significant bit of Q_{i+1}. Consider an execution $\mathcal{E}_2$ of the binary clock protocol, and couple executions $\mathcal{E}_T$ and $\mathcal{E}_2$ as follows: If u samples v in round t_j of $\mathcal{E}_T$, then u samples v in round j of $\mathcal{E}_2$. We set the initial clocks of the binary clock protocol $\beta^{u,0} = b^{u,t_1-1}_p$, and prove by induction that for any $j \geq 0$, $\beta^{u,j} = b^{u,t_{j+1}-1}_p$.

The base case of $j = 0$ holds by construction. For $j \geq 1$, suppose $\beta^{u,j-1} = b^{u,t_j-1}_p$, and that in round t_j of $\mathcal{E}_T$ agent u receives a message $\mu_1\mu_2$ from agent v. The condition on Line 8 of Algorithm 3 is satisfied in round t_j if $b^{u,t_j-1}_p = 1$, or equivalently by the inductive hypothesis, if $\beta^{u,j-1} = 1$ as in Algorithm 1. By the inductive hypothesis again, $\mu_2 = b^{v,t_j-1}_p = \beta^{v,j-1} = \beta'$, where β' is the bit received by u in round j of $\mathcal{E}_2$. This implies that on Line 11 of Algorithm 3 and on Line 4 of Algorithm 1, the same bit value is assigned by both operations. We have that $\text{val}(Q^{u,t_j-1}_i) = 1$ because $s_i = p$, hence, the increment of C^u in Algorithm 3 does not change the bit at index p of C^u in round t_j. In Algorithm 1 however, the clock β^u changes in the corresponding round j. Therefore, $b^{u,t_j}_p = 1 - \beta^{u,j}$. Since $t_{j+1} - t_j = 2^{s_i} = 2^p$, in exactly one of the rounds in $\{t_j+1, \ldots, t_{j+1}-1\}$, the bit at index p of C^u will flip (due to the increments), and thus, $b^{u,t_{j+1}-1}_p = 1 - b^{u,t_j}_p = \beta^{u,j}$, which completes our inductive proof.

Finally, let $j_s \geq 0$ be the first round when the binary clock β is synchronized in $\mathcal{E}_2$. This implies that for $t \geq t_{j_s+1}$, the bit b_p is also synchronized among agents in $\mathcal{E}_T$, i.e., $r_{t+1} \geq r_{t_0} + 1$. From Theorem 1.1, $j_s = O(\log n)$, w.h.p. It follows $t_{j_s+1} - t_0 \leq (j_s+1) \cdot 2^{s_i} = O(2^{s_{h-1}} \cdot \log n)$ w.h.p.

Case $s_i < p$. In this case, we also use a coupling with a mod 2 clock, however in a more subtle way. Among the rounds $t \in \{t_j\}_{j \geq 1}$, i.e., when $\pi(C^{u,t}) = p$, consider the ones where the condition in Lines 9 and 10 is met. Formally, let

(6.5)
$$\mathcal{T}_u = \left\{ t_j \colon j \geq 1,\, \exists p' \in [s_i, p-1] \text{ s.t. } b_{p'}^{u,t_j-1} = 0 \right\}.$$

For any $j \geq 1$, the bits of the clocks at indices up to $p-1$ are synchronized, thus, $\mathcal{T}_v = \mathcal{T}_u$ for any agent v. We therefore simply refer to the sets $\mathcal{T}_u$ as $\mathcal{T}$. Denote by τ_j the jth smallest element of $\mathcal{T}$, and note that if $t_j \notin \mathcal{T}$, then $t_{j+1} \in \mathcal{T}$, implying that, $\tau_j \leq t_{2j}$. For any $t \geq 0$, define $c^{u,t} = b_p^{u,t} \oplus b_{s_i}^{u,t}$. (The bit c^u can be thought of as an implicit variable of Algorithm 3.) We prove that for $t \in \mathcal{T}$, the bit $c^{u,t-1}$ emulates the binary clock from Section 4. To formalize that, consider an execution $\mathcal{E}_2$ of Algorithm 1, and couple this execution to execution $\mathcal{E}_T$ restricted to rounds in $\mathcal{T}$ (similarly to the previous case). The binary clocks β^u are initialized to $\beta^{u,0} = c^{u,\tau_1-1}$ in $\mathcal{E}_2$. We prove, by induction, that for any $j \geq 0$, $\beta^{u,j} = c^{u,\tau_{j+1}-1}$.

Once again the base case $j = 0$ holds by construction, so for $j \geq 1$, we assume that $\beta^{u,j-1} = c^{u,\tau_j-1}$. Let v be the agent from which u receives message $\mu_1\mu_2$ in round τ_j. Then, $\mu_2 = b_p^{v,\tau_j-1}$. Since $\tau_j \in \mathcal{T}$, the condition on Lines 9 and 10 of Algorithm 3 is satisfied in round τ_j if $c^{u,\tau_j-1} = 0$, or equivalently, if $\beta^{u,j-1} = 0$ due to the inductive hypothesis. Therefore, Line 11 of Algorithm 3 is executed in round τ_j of $\mathcal{E}_T$ if and only if Line 4 of Algorithm 1 is executed in round j of the coupled $\mathcal{E}_2$. Following the argument from the previous case, we have that after round τ_j of $\mathcal{E}_T$, $c^{u,\tau_j} = 1 - \beta^{u,j}$. As before, it remains to show that before the next coupled round τ_{j+1} the implicit bit c^u is incremented exactly once, i.e.,

$$c^{u,\tau_{j+1}-1} = 1 - c^{u,\tau_j}. \tag{6.6}$$

First suppose that $\tau_j + 2^{s_i} \in \mathcal{T}$, i.e., the first coupled round after τ_j is $\tau_{j+1} = \tau_j + 2^{s_i}$. Since $\tau_j, \tau_{j+1} \in \mathcal{T}$, in each of C^{u,τ_j-1} and $C^{u,\tau_{j+1}-1}$ there is a bit equal to 0 at least on one of the indices in $\{s_i, \ldots, p-1\}$. This implies that the bit at index p does not change in rounds $\tau_j+1, \ldots, \tau_{j+1}-1$ (which could only happen due to the increments on Line 12). On the other hand, the bit at index s_i changes exactly once in those rounds. Thus,

$$b_p^{u,\tau_{j+1}-1} = b_p^{u,\tau_j} \quad \text{and} \quad b_{s_i}^{u,\tau_{j+1}-1} = 1 - b_{s_i}^{u,\tau_j},$$

which implies (6.6). If $\tau_j + 2^{s_i} \notin \mathcal{T}$, then the next coupled round is $\tau_{j+1} = \tau_j + 2 \cdot 2^{s_i} \in \mathcal{T}$. In this case, bit p flips once due to the increments, while bit s_i flips twice (and does not change), because $\tau_{j+1} - \tau_j = 2 \cdot 2^{s_i}$. Thus, (6.6) holds in this case too.

Note that if the binary clock β is synchronized in round j_s of $\mathcal{E}_2$, then so are the bits $c^{u,t}$ for $t \geq t_{2(j_s+1)} \geq \tau_{j_s+1}$. This in turn implies that the bits $b_p^{u,t} = c^{u,t} \oplus b_{s_i}^{u,t}$ are synchronized since $s_i \leq r_t$. By the fact that $t_{2(j_s+1)} - t_0 \leq 2(j_s+2) \cdot 2^{s_i}$ and Theorem 1.1, we complete the proof. ∎

Lemma 6.4. *There is a constant $\eta > 0$, such that, if $r_{t_0} \geq s_{h-1} + \log\log n + \eta$, for some $t_0 \geq 0$, then, there is a round $t = t_0 + O(2^{s_{h-1}} \cdot \log n)$, such that the clocks of all agents are synchronized after round t (i.e., $r_t = k$), w.h.p.*

Proof. Consider an index $p \in [r_{t_0}, k-1]$. We analyse the number of rounds before the bits at index p of the clocks are the same. The analysis is similar to the analysis of the case $p > s_i = s_{h-1}$ in Lemma 6.3; so we use the same notation as there. Unlike in Lemma 6.3, we do not have the assumption that all bits at indices $0, \ldots, p-1$ are synchronized. This implies that the sets $\mathcal{T}_u$, as defined in (6.5), are not identical, which was used to prove (6.6). We circumvent this problem by considering a set of rounds $\mathcal{T}'$, which is a subset of $\mathcal{T}_u$ for all agents u. Moreover, $\mathcal{T}'$ contains sufficiently many consecutive rounds from the sequence $(t_j)_{j\geq 1}$ to synchronize bit p.

Let $z = s_{h-1} + \lceil \log\log n \rceil + \eta \leq r_{t_0}$, where constant $\eta \in \mathbb{N}$ will be defined later. Denote by τ the first round after t_0, such that after round τ the bits at indices $0, \ldots, z-1$ are all 0, i.e.,

$$\tau = \min\{t \geq t_0 \colon \operatorname{val}(R_t) \equiv 0 \pmod{2^z}\}.$$

Note that $\tau - t_0 \leq 2^z$. Let

$$\mathcal{T}' = \{t_j \colon j \geq 1 \text{ and } \tau < t_j < \tau + 2^z - 2^{s_{h-1}}\}.$$

By Lemma 6.2, for any $t \in \mathcal{T}'$,

$$\operatorname{val}(R_{t-1}) \equiv \operatorname{val}(R_\tau) + t - 1 - \tau \equiv t - 1 - \tau \pmod{2^z}.$$

And since $t - 1 - \tau < 2^z - 2^{s_{h-1}}$, there is some index $p' \in [s_{h-1}, z-1]$ such that the bit at index p' of $C^{u,t}$ is 0 for all agents u, as $r_t \geq z$. Therefore, $\mathcal{T}' \subset \mathcal{T}_u$ for all u. Similarly to (6.6), we obtain that if τ'_j is the jth smallest element of $\mathcal{T}'$ and $j < |\mathcal{T}'|$, then for any u,

$$c^{u,\tau'_{j+1}-1} = 1 - c^{u,\tau'_j}.$$

Unlike in Lemma 6.3, where $\mathcal{T}$ is infinite, we have to argue that $\mathcal{T}'$ contains sufficiently many elements for bit p to become synchronized. By construction,

$$|\mathcal{T}'| \geq 2^{z - s_{h-1}} - 2 \geq 2^\eta \cdot \log n - 2.$$

By Theorem 1.1, there exists a constant η such that $|\mathcal{T}'|$ rounds are sufficient for a binary clock to synchronize w.h.p. This implies that for $\tau_{\max} = \max(\mathcal{T}')$, the bits at index p of the clocks are the same immediately after round $\tau_{\max}$, w.h.p. Since $\tau_{\max}$ is independent of p, this statement holds for all $p \in \{z+1, \ldots, k-1\}$, thus all clocks are synchronized at round $\tau_{\max}$ w.h.p., by a union bound. Since $\tau_{\max} \leq \tau + 2^z \leq t_0 + 2 \cdot 2^z = t_0 + O(2^{s_{h-1}} \cdot \log n)$, the proof is complete. ∎

Claim 6.1. $s_{h-1} \leq \log k + 3 \log \log k + O(1)$.[5]

Proof. We show by induction that for any $i \in \{1, \ldots, h-1\}$, $s_i \leq 3l_i$. The base case is trivial since $s_1 = l_1$, so suppose $s_{i-1} \leq 3l_{i-1}$ for some $i \geq 2$. Then,

$$\begin{aligned} s_i = l_i + s_{i-1} \leq l_i + 3l_{i-1} &= l_i + 3\rho(l_i) \\ &= l_i + 3\lceil \log(l_i + 1) \rceil \leq 3l_i, \end{aligned}$$

where the last inequality holds since $l_i \geq 3$. Therefore,

$$s_{h-1} = l_{h-1} + s_{h-2} \leq l_{h-1} + 3l_{h-2} = \rho(\lambda) + 3\rho_2(\lambda),$$

where λ is defined in (6.3), from which it also follows that $\lambda \leq k$. Since ρ is a non-decreasing function and $\rho(l) \leq \log(l+1) + 1 \leq \log l + 2$ for any $l \geq 2$, for some constant $c > 0$,

$$s_{h-1} \leq \rho(k) + 3\rho_2(k) \leq \log k + 3 \log \log k + c.$$

This completes the proof. ∎

Proof of Theorem 6.1. Let η be the constant guaranteed by Lemma 6.4, and let $z = s_{h-1} + \lceil \log \log n + \eta \rceil$. For $r \in \{2, \ldots, z\}$, let $\tau_r = \min\{t : r_t = r\}$ be the first round when the r least significant bits of the clocks C are synchronized. We have that $\tau_2 = O(\log n)$ and, for $r > 2$, $\tau_r - \tau_{r-1} = O(2^{s_{h-1}} \cdot \log n)$, w.h.p., by Lemmas 6.1 and 6.3, respectively. By a union bound, therefore, $\tau_z = O(z \cdot 2^{s_{h-1}} \cdot \log n)$. By Lemma 6.4 and another application of a union bound, we have that the clocks are synchronized in $O((z+1) \cdot 2^{s_{h-1}} \cdot \log n)$ rounds, w.h.p. By Claim 6.1, $z = O(\log \log(nT))$ and $2^{s_{h-1}} = O(\log T \cdot (\log \log T)^3)$, and the proof is complete by substituting these values. ∎

7 Simulator with Message Space of Size 3

In Algorithm 4, we present a simple protocol for simulating, in the standard model, any protocol A for the 2-clocked model that uses 1-bit messages. It is based on the modulo 4 clock synchronization protocol from Algorithm 2. Each agent stores the two bits b_1, b_0 of the modulo 4 clock, plus the complete state s of the simulated algorithm A. The message function m of the simulator equals 0 if $b_1 = 0$, and $m_A(s) + 1 \in \{1, 2\}$ if $b_1 = 1$, where m_A is the binary message function of A. Thus a single ternary digit suffices to encode each message of the simulator. The protocol is the same as Algorithm 2, except that, when the received message is $\mu \neq 0$, then a round of A is executed, using $\mu - 1$ as the message and b_0 as the shared binary clock value.

[5] The coefficient 3 can be made arbitrarily close to 1, by using a more refined argument. This implies that the factor $(\log \log T)^3$ in the bound of Theorem 6.1 can be improved to almost $\log \log T$.

Algorithm 4: Simulation in the standard model with message space size 3, of any protocol A for the 2-clocked model with 1-bit messages.

state: $b_1, b_0 \in \{0, 1\}$; s: state of A
msg function: $m(b_1, b_0, s) := b_1 \cdot (1 + m_A(s))$, where m_A is A's msg function

foreach *round* t **do**
  if $b_1 = 1$ **then**
    sample an agent v u.a.r., and let μ be the message received from v
    if $\mu = 0$ **then**
      $b_1 \leftarrow 1 - b_1$; $b_0 \leftarrow 1 - b_0$
    else
      `// execute a round of` A
      $s \leftarrow A(s, b_0, \mu - 1)$
  $b_1 b_0 \leftarrow \mathrm{incr}(b_1 b_0)$

The simulator above satisfies the state space and message space conditions of Theorem 1.4. Next we show that the delay is logarithmic and specify the slowdown.

Theorem 7.1. *The simulator in Algorithm 4 has delay $O(\log n)$ w.h.p., and slowdown $4:2$.*

Proof. By comparing Algorithm 4 with Algorithm 2, we observe that the algorithm for updating variables b_1, b_0 is identical in the two protocols. Note, in particular, that in Algorithm 4, $\mu = 0$ if and only if $b'_1 = 0$ for the most significant bit of the sampled agent v. It follows from Theorem 5.1, that the two bit clock $b_1 b_0$ is synchronized across all agents after $O(\log n)$ rounds, w.h.p. From that point on, all agents u execute a round of A in each round t at which $b_1 = 1$, using $\mu - 1$ as the message received in the simulated round and b_0 as the shared clock value. Thus the slowdown is $4:2$. ∎

8 Simulator with 1-bit Messages

We present a simulator that uses 1-bit messages, and simulates in the standard model any protocol A for the 2-clocked model with 1-bit messages.

8.1 Protocol Description.

The pseudocode of the simulator is given in Algorithm 5. We assume that every agent has a linear upper bound on $\log n$. It is not necessary that the bound is the same for all agents, but to simplify exposition we assume it is. Each agent

Algorithm 5: Simulation in the standard model with 1-bit messages, of any protocol A for the 2-clocked model with 1-bit messages.

state:
 // two 2-bit clocks, b_1b_0 and c_1c_0
 $b_1, b_0, c_1, c_0 \in \{0, 1\}$
 // phase and level counters
 $\phi \in \{0, 1\}$; $\ell \in \{0, \dots, \ell^*\}$, where $\ell^* = \Theta(\ln n)$
 s: state of A

msg function:
 $m := (b_1 + \phi b_0 c_1 + \phi(1-b_0)c_1 \cdot m_A(s) > 0)\,?\,1 : 0$,
 where m_A is A's binary msg function

1 **foreach** *round* t **do**
2 sample an agent v u.a.r., and let μ be the message received from v
3 **if** $\phi = 0$ **then**
4 **if** $b_1 = 1$ **and** $\mu = 0$ **then**
5 **if** $\ell < \ell^*/2$ **then**
 // modify clock b_1b_0
6 $b_1 \leftarrow 1 - b_1$; $b_0 \leftarrow 1 - b_0$
7 $\ell \leftarrow 0$ // reset level
8 **else if** $\ell < \ell^*$ **then**
9 $\ell \leftarrow \ell + 1$ // increase level
10 **else**
 // move to phase 1 & reset c_1c_0
11 $\phi \leftarrow 1$; $c_1c_0 \leftarrow 00$
12 **else** // $\phi = 1$ in this case
13 **if** $b_1 = 1$ **and** $\mu = 0$ **then**
 // move back to middle of phase 0
14 $\phi \leftarrow 0$; $\ell \leftarrow \ell^*/2$
15 **else if** $b_1 = b_0 = 1$ **then**
 // increment clock c_1c_0
16 $c_1c_0 \leftarrow \mathrm{incr}(c_1c_0)$
17 **else if** $b_1 = 0$ **and** $b_0 = 1$ **then**
 // modify clock c_1c_0; 0 and 1 are switched in update condition
18 **if** $c_1 = 0$ **and** $\mu = 1$ **then**
19 $c_1 \leftarrow 1 - c_1$; $c_0 \leftarrow 1 - c_0$
20 **else if** $b_1 = b_0 = 0$ **and** $c_1 = 1$ **then**
 // execute a round of A
21 $s \leftarrow A(s, c_0, \mu)$
22 $b_1b_0 \leftarrow \mathrm{incr}(b_1b_0)$ // increment clock b_1b_0

u stores two modulo 4 clocks, b_1b_0 and c_1c_0, and at any point, u is in one of two *phases*, $\phi \in \{0, 1\}$. Both clocks follow the protocol in Algorithm 2. Clock b_1b_0 is incremented in each round, in both phases, whereas clock c_1c_0 is incremented only when the agent is in phase 1, and only every 4 rounds, whenever $b_1b_0 = 11$. Thus, in phase 1, the two clocks constitute a modulo 16 clock, $c_1c_0b_1b_0$. While in phase 1, an agent also executes two rounds of protocol A every 16 rounds: an even round of A is executed when $c_1c_0b_1b_0 = 1000$, and an odd round when $c_1c_0b_1b_0 = 1100$.

The transition of agent u from phase 0 to phase 1 is controlled by the agent's level ℓ. The level is used only when the agent is in phase 0, and in each round, ℓ is either reset to 0 or increased by one. Precisely, if conditions $b_1 = 1$ and $\mu = 0$ hold (which, as we will see, means that the clocks b_1b_0 of u and the v are not in sync), then ℓ is reset; otherwise it increases by one. After ℓ reaches the maximum value $\ell^* = \Theta(\log n)$, the agent moves to phase 1. On the other hand, when an agent is in phase 1 and the conditions $b_1 = 1$ and $\mu = 0$ hold, then the agent returns to phase 0. For technical reasons discussed later, the level is not reset in that case, but is set to $\ell^*/2$.

In phase $\phi = 0$, agent u just runs the synchronization protocol for the modulo 4 clock b_1b_0, updating also the level as described above. In particular, the value of u's message is $m = b_1$, similarly to Algorithm 2. When conditions $b_1 = 1$ and $\mu = 0$ hold, then u flips its bits b_1, b_0 in Line 6, as in Algorithm 2, but only if $\ell < \ell^*/2$. If $\ell \geq \ell^*/2$ then the bits are not flipped (again for technical reasons discussed later).

In phase $\phi = 1$, the message of u is $m = 1$ if $b_1 = 1$. Similarly to phase $\phi = 0$, if conditions $b_1 = 1$ and $\mu = 0$ hold, it means that u's clock b_1b_0 is not in sync with v's. Then u moves back to phase 0 setting its level to $\ell^*/2$ as mentioned above (but does not flip its bits b_1, b_0). When $b_1b_0 = 11$ and $\mu \neq 0$, agent u increments clock c_1c_0. When $b_1 = 0$ then u's message is $m = 0$ if $b_0 \cdot c_1 + (1 - b_0) \cdot c_1 \cdot m_A(s) = 0$ and 1 otherwise. In particular, when $b_1b_0 = 01$, then $m = c_1$, and u updates its clock c_1c_0 in that round. For technical reasons discussed later, clock c_1c_0 uses the symmetric update rule of that in Algorithm 2, i.e., the clock's bits are flipped when $c_1 = 0$ and $\mu = 1$ (compare Lines 4 and 18). When $b_1b_0 = 00$ and $c_1 = 1$, then $m = m_A(s)$, and u executes a round of A using c_0 as the shared clock value.

We now provide some informal explanation of why the protocol works, and justify some subtle design choices. First, it is not hard to see that, once all clocks b_1b_0 are synchronized, they stay in sync forever and agents never reset their level. This follows from the observation that if $b_1 = 1$ then $m = 1$, thus Lines 6, 7 and 14 are never executed again. Hence, after clock b_1b_0 is synchronized, all agents reach phase 1 within a logarithmic number of rounds. Then clock c_1c_0 is synchronized in logarithmic additional rounds, as all agents execute the modulo 4 synchronization protocol in sync (clock c_1c_0 is updated when $b_1b_0 = 01$, and incremented when $b_1b_0 = 11$). Once both clocks are synchronized, all agents execute algorithm A in sync, twice every 16 rounds.

It suffices thus to focus on the synchronization of clock $b_1 b_0$. In the idealized case where all agents start at level 0 of phase 0, the clocks become synchronized before any agent reaches level $\ell^*/2$ (assuming ℓ^* is large enough), as agents execute just the synchronization protocol of Algorithm 2. However, this is not the case in general, when agents start from arbitrary states. The main source of complication is that an agent u's message in phase 1 can be $m = 1$ even if $b_1 = 0$, whereas in Algorithm 2 it is always $m = b_1$. This can result in "missed update opportunities," where b_1 is 1 for u and 0 for v, but u does not flip its bits b_1, b_0.

To circumvent this issue, we switch the roles of 0 and 1 in the condition for modifying the modulo 4 clock $c_1 c_0$. This guarantees that as long as u stays in phase 1, we have $m = 1$ "sufficiently often." More concretely, in at least twice every four cycles of clock $b_1 b_0$, we have $c_1 = 0$ when $b_1 = 0$, and thus $m = 0$.[6]

We use the fact that for any agent in phase 1 we have $m = 0$ sufficiently often, to show that the following property holds w.h.p. If S is the set of agents that do not reach level 0 in the first $\Theta(\log n) < \ell^*/4$ rounds, then either $S = \emptyset$, or there is a set $S' \supseteq S$, containing all but an ϵ-fraction of agents, such that all agents $u \in S'$ have the same clock value.

The case of $S = \emptyset$ is similar to the idealized case where all agents start at level 0 of phase 0, mentioned earlier, thus all clocks get synchronized quickly. If $S \neq \emptyset$, then we show that the small minority of the clocks which show a different value quickly converges to the majority value. To simplify this technical argument, we give an "edge" to the agents $u \in S$, by not modifying their clock when they move from phase 1 back to phase 0, and when they reset their level to zero from a level $\ell \geq \ell^*/2$.

8.2 Properties. The simulator satisfies the message size and state space conditions of Theorem 1.5, i.e., uses 1-bit messages and increases the number of states by at most a logarithmic factor. Next we specify the slowdown, and establish a logarithmic upper bound on the delay.

THEOREM 8.1. *The simulator in Algorithm 5 has delay $O(\log n)$ w.h.p., and slowdown $16 : 2$.*

8.3 Notation. For any agent $u \in N$, integer $t \geq 0$, and state variable σ (e.g., b_1, ℓ, or ϕ) we write $\sigma^{u,t}$ to denote the value of variable σ in u's state right after the first t rounds. By $m^{u,t}$ we denote the value of the message function applied to the state of u right after the first $t \geq 0$ rounds. By $\mu^{u,t}$ we denote the message that u receives in round $t \geq 1$, i.e., $\mu^{u,t} = m^{v,t-1}$, if u samples agent v in round t.

We assume $\ell^* = \Theta(\log n)$ is a multiple of 8.

For any $t \geq 0$, $i, j \in \{0, 1\}$, and $0 \leq k \leq \ell^*$, we define the following sets of agents,

$$\begin{aligned} \Phi_i^t &= \{u \colon \phi^{u,t} = i\}, & L_k^t &= \{u \colon \ell^{u,t} \leq k\} \cap \Phi_0^t, \\ B_{ij}^t &= \{u \colon b_1^{u,t} = i, b_0^{u,t} = j\}, & B_i^t &= B_{i0}^t \cup B_{i1}^t, \\ C_{ij}^t &= \{u \colon c_1^{u,t} = i, c_0^{u,t} = j\}, & C_i^t &= C_{i0}^t \cup C_{i1}^t, \\ \hat{B}_0^t &= B_{01}^t \cup B_{10}^t, & \hat{B}_1^t &= B_{00}^t \cup B_{11}^t. \end{aligned}$$

Also, for $t \geq 1$,

$$U^t = \{u \colon b_1^{u,t-1} = 1, \mu^{u,t} = 0\}.$$

Note, if $u \in \Phi_0^{t-1} \cap U^t$ then $u \in L_0^t$, and if $u \in \Phi_1^{t-1} \cap U^t$ then $u \in L_{\ell^*/2}^t \setminus L_{\ell^*/2-1}^t$. Finally, for $0 \leq t_1 \leq t_2$, let

$$Z_{t_1,t_2} = \bigcup_{t_1 \leq t \leq t_2} L_0^t,$$

and note that $Z_{t_1,t_2} = L_{t_2-t_1}^{t_2}$ if $t_2 - t_1 < \ell^*/2$, and $Z_{t_1,t_2} \subseteq L_{t_2-t_1}^{t_2}$ if $t_2 - t_1 \leq \ell^*$.

8.4 Analysis. All lemmas below hold for any given round $t \geq 0$, and any fixed value for the configuration $\mathcal{C}_t$ of the agents' states after that round. The first lemma says that once all clocks $b_1 b_0$ get synchronized, they stay in sync forever.

LEMMA 8.1. *If $B_{ij}^t = N$ then $B_{\mathrm{incr}(ij)}^{t+1} = N$.*

Proof. Suppose, for contradiction, that $B_{ij}^t = N$, and $u \notin B_{\mathrm{incr}(ij)}^{t+1}$ for some u. Consider round $t + 1$, and let v be the agent that u samples in that round. Since u increments $b_1 b_0$ in Line 22, but $u \notin B_{\mathrm{incr}(ij)}^{t+1}$, it follows that u executes Line 6 (which is the only other line where the clock is modified). Thus, the conditions in Line 4 hold, i.e., $b_1^{u,t} = 1$ and $\mu^{u,t+1} = 0$. Since $u, v \in B_{ij}^t$ and $b_1^{u,t} = 1$, it follows $b_1^{v,t} = 1$, and thus $m^{v,t} = 1$. Then $\mu^{u,t+1} = m^{v,t} = 1$, which contradicts $\mu^{u,t+1} = 0$. ∎

Next we give a simple lower bound on the number of agents $u \in B$, for a given set $B \subseteq B_1^t$, that receive message 1 in round $t + 1$.

LEMMA 8.2. *If $B \subseteq B_1^t$, $|B| = k_1$, and $|B_1^t| = k_2$, then for any $0 \leq \delta \leq 1$,*

$$\Pr\left[|B \setminus U^{t+1}| < (1 - \delta) k_1 k_2 / n\right] < e^{-\delta^2 k_1 k_2 / (2n)}.$$

[6] The reason is that the possible state transitions for clock $c_1 c_0$ (after an update and increment) are: $00 \to \{00, 01\}$, $01 \to \{10, 11\}$, $10 \to 11$, and $11 \to 00$, thus from any state, $c_1 = 0$ after at most two transitions.

Proof. Since $m^{u,t} = 1$ for any $u \in B_1^t$, $|B \setminus U^{t+1}|$ is lower bounded by the number of agents $u \in B$ that sample an agent from B_1^t in round $t+1$. Given $|B| = k_1$ and $|B_1^t| = k_2$, the expected number of those agents is $k_1 \cdot (k_2/n)$. The claim then follows by a standard Chernoff bound. ◻

Roughly speaking, the next lemma implies that for any interval of 12 rounds and any agent u, either $m^{u,t'} = 0$ for at least two consecutive rounds t' in that interval, or $u \in U^{t'}$ for some t' in the interval.[7]

LEMMA 8.3.

(a) If $u \in B_{00}^t \cap (\Phi_0^t \cup C_0^t)$, then $m^{u,t} = m^{u,t+1} = 0$.

(b) For any u, if $\tau_1 = \min\{t' \geq t \colon u \in B_{00}^{t'} \cap (\Phi_0^{t'} \cup C_0^{t'})\}$, and $\tau_2 = \min\{t' > t \colon u \in U^{t'}\}$, then $\min\{\tau_1, \tau_2\} \leq t + 11$.

Proof. We show (a) first. Equation $m^{u,t} = 0$ follows from the definition of the message function, and assumption $u \in B_{00}^t \cap (\Phi_0^t \cup C_0^t)$. We now consider $m^{u,t+1}$. Since $u \in B_{00}^t$ we have $u \in B_{01}^{t+1}$, as $b_1^{u,t} = 0$ and thus Line 6 is not executed in round $t+1$. Then, from the definition of the message function, $m^{u,t+1} = 0$ if $u \in \Phi_0^{t+1}$, and $m^{u,t+1} = c_1^{u,t+1}$ if $u \in \Phi_1^{t+1}$. Thus to prove $m^{u,t+1} = 0$ it suffice to show that if $u \in \Phi_1^{t+1}$ then $c_1^{u,t+1} = 0$.

Suppose $u \in \Phi_1^{t+1}$. If $u \in \Phi_0^t$, i.e., u moved from phase 0 to phase 1 in round $t+1$, then it must have executed Line 11 in round $t+1$, thus $c_1^{u,t+1} = 0$, as desired. So, suppose that $u \in \Phi_1^t$. Then, from the assumption that $u \in B_{00}^t \cap (\Phi_0^t \cup C_0^t)$, it follows that $u \in C_0^t$. Since u does not change clock c_1c_0 in round $t+1$, as $u \in B_{00}^t$, it follows $c_1^{u,t+1} = c_1^{u,t} = 0$. This completes the proof of (a).

Next we prove (b). Suppose $u \in B_{ij}^t$. Suppose also that for all $t < t' \leq t + 11$, $u \notin U_t'$ (otherwise (b) holds). This implies that u does not execute Line 6 in any round $t' \in \{t+1, \ldots, t+11\}$. Then, due to Line 22, u's clock b_1b_0 is incremented by exactly one in each of these rounds.

Let $t_0 = \min\{t' \geq t \colon u \in B_{00}^{t'}\}$. From the last observation above, it follows that $t \leq t_0 \leq t+3$. If also $u \in \Phi_0^{t_0}$, i.e, $u \in B_{00}^{t_0} \cap \Phi_0^{t_0}$, then $\tau_1 \leq t_0 \leq t+3$, which implies (b).

Suppose now that $u \in \Phi_1^{t_0}$. Then $u \in \Phi_1^{t'}$ for al $t_0 < t' \leq t+11$, as u does not execute Line 14 at any of those rounds t'. Let $t_1 = t_0 + 4$ and $t_2 = t_0 + 8 \leq t + 11$. Then $u \in B_{00}^{t'}$ for all $t' \in \{t_0, t_1, t_2\}$. Thus, to prove (b), it suffices to show that $u \in C_0^{t'} = 0$, for some $t' \in \{t_0, t_1, t_2\}$.

Suppose that $c_1^{u,t_0} = 1$, otherwise the claim above holds. If $u \in C_{11}^{t_0}$ then $u \in C_{00}^{t_1}$, because u does not execute Line 18 in round $t_0 + 2$ as $c_1^{u,t_0+1} = c_1^{u,t_0} = 1$; thus $c_1^{u,t_1} = 0$. Similarly, if $u \in C_{10}^{t_0}$ then $u \in C_{11}^{t_1}$, and also $u \in C_{00}^{t_2}$; thus $c_1^{u,t_2} = 0$. This completes the proof of the claim that $c_1^{u,t'} = 0$, for some $t' \in \{t_0, t_1, t_2\}$, and the proof of (b). ◻

We now use Lemmas 8.2 and 8.3 to show that if $u \in B_{ij}^t$ and $|B_{ij}^t| \leq (1-\epsilon) \cdot n$, then with at least a constant probability, $u \in U^{t'}$ for some of the next 13 rounds $t' > t$.

LEMMA 8.4. *For any constant $0 < \epsilon_1 < 1$ there is a constant $0 < \epsilon_2 < 1$ such that, if $u \in B_{ij}^t$ and $|B_{ij}^t| \leq (1-\epsilon_1) \cdot n$, then $\Pr\left[u \in \bigcup_{t < t' \leq t+13} U^{t'}\right] \geq \epsilon_2$.*

Proof. Since $|B_{ij}^t| \leq n - \epsilon_1 n$, it follows that $|B_{i'j'}^t| \geq \epsilon_1 n/3$, for some pair $(i', j') \neq (i, j)$. For each $r \in \{0, 1, \ldots\}$, let $A_r = B_{i'j'}^t \setminus \bigcup_{t < t' \leq t+r} U^{t'}$. Note that if $v \in A_r$, then v does not execute Line 6 in any of the rounds $t+1$ up to $t+r$, thus v' clock b_1b_0 is incremented by exactly one in each of those rounds. Note also that if $A_r \subseteq B_0^{t+r}$, i.e., $b_1 = 0$ for the agents in A_r after round $t+r$, then $A_{r+1} = A_r$; while if $A_r \subseteq B_1^{t+r}$, then Lemma 8.2 implies that for any $0 < \delta < 1$,

$$\Pr\left[|A_{r+1}| < (1-\delta)a_r^2/n \;\middle|\; |A_r| = a_r\right] < e^{-\delta^2 a_r^2/(2n)}.$$

By applying this iteratively, and using a union bound, we obtain

$$\Pr\left[|A_r| < (1-\delta)^{2^r-1} a_0^{2^r}/n^{2^r-1} \;\middle|\; |A_0| = a_0\right] \leq r \cdot e^{-\delta^2((1-\delta)^{2^r-2} a_0^{2^r}/n^{2^r-2})/(2n)}.$$

Substituting $r = 11$, $\delta = 1/2$, and $a_0 = |B_{i'j'}^t| \geq \epsilon_1 n/3$, we obtain

$$\Pr\left[|A_{11}| < \epsilon_3 n\right] = e^{-\Omega(n)}, \tag{8.7}$$

for some constant $\epsilon_3 > 0$.

For any $v \in A_{11}$, Lemma 8.3(b) implies that there is some round $t_v \in \{t, \ldots, t+11\}$ such that $v \in B_{00}^{t_v} \cap (\Phi_0^{t_v} \cup C_0^{t_v})$. Precisely, $t_v \in \{t_0, t_1, t_2\}$, where $t_0 = t + \min\{k \geq 0 \colon \mathrm{val}(i'j') + k \equiv 0 \pmod 4\} \leq t+3$, $t_1 = t_0 + 4$, and $t_2 = t_0 + 8$. It follows that there is a round $t^* \in \{t_0, t_1, t_2\} \subseteq \{t, \ldots, t+11\}$, and a set $A^* \subseteq A_{11} \subseteq A_{t^*-t}$ with $|A^*| \geq |A_{11}|/3$, such that $v \in B_{00}^{t^*} \cap (\Phi_0^{t^*} \cup C_0^{t^*})$ for all $v \in A^*$. Combining that with result (8.7), we obtain that the following event, $\mathcal{E}$, occurs with probability $1 - e^{-\Omega(n)}$: There is some $t^* \in \{t, \ldots, t+11\}$ and a set $A^* \subseteq A_{t^*-t}$ such that $|A^*| \geq \epsilon_3 n/3$ and $v \in B_{00}^{t^*} \cap (\Phi_0^{t^*} \cup C_0^{t^*})$ for all $v \in A^*$.

[7] The proof of this lemma relies on the fact that the update condition for modifying clock c_1c_2, in Line 18, is the symmetric of that in Line 4, with the roles of 0 and 1 swapped.

Let $u \in B^t_{ij}$. Suppose $\mathcal{E}$ occurs, and fix t^* and A^*. If $u \notin \bigcup_{t<t'\leq t^*} U^{t'}$, then u's clock b_1b_0 is not in sync with the clocks of the agents $v \in A^*$ after round t^*, thus $u \notin B^{t^*}_{00}$. Also, from Lemma 8.3(a), for any $v \in A^*$, $m^{v,t^*} = m^{v,t^*+1} = 0$. We have two cases: If $u \in B^{t^*}_1$, then the probability that u samples some agent from A^* in round t^*+1, and thus $u \in U^{t^*+1}$, is $|A^*|/n \geq \epsilon_3/3$. If $u \notin B^{t^*}_1$, then $u \in B^{t^*}_{01}$ and $u \in B^{t^*+1}_{10}$, thus the probability u samples some agent from A^* in round t^*+2, implying $u \in U^{t^*+2}$, is also $|A^*|/n \geq \epsilon_3/3$. Therefore, in both cases, $u \in U^{t^*+1} \cup U^{t^*+2}$ with probability at least $\epsilon_3/3$.

It follows that, with probability at least $\Pr[\mathcal{E}] \cdot \epsilon_3/3 \geq \epsilon_2 > 0$, $u \in \bigcup_{t<t'\leq t+13} U^{t'}$. ∎

We use Lemma 8.4 to prove Lemmas 8.5 and 8.6 below. Roughly speaking, the two lemmas say that, w.h.p., all agents that do not reach level 0 during an interval of $\Theta(\log n)$ rounds have synchronized b_1b_0 clocks with each other, and also with a $(1-\epsilon)$-fraction of all agents. Recall that $Z_{t_1,t_2} = \bigcup_{t_1\leq t\leq t_2} L^t_0$.

Lemma 8.5. *There are constants $\alpha, \lambda > 0$ such that if $\ell^* \geq \lambda \ln n$ then $\Pr[R_t > t + a\ln n] = O(1/n)$, where $R_t = \min\{t' \geq t \colon N \setminus Z_{t,t'} \subseteq B^{t'}_{ij}$, for some $i,j\}$. Also, if $N \setminus Z_{t,t'} \subseteq B^{t'}_{ij}$ then $N \setminus Z_{t,t'+1} \subseteq B^{t'+1}_{\mathrm{incr}(ij)}$.*

Proof. Let u_1, u_2 be any pair of agents such that $(b^{u_1,t}_1, b^{u_1,t}_0) \neq (b^{u_2,t}_1, b^{u_2,t}_0)$. We partition all rounds $t' > t$ into intervals of length 13, and for each interval define a 0-1 random variable X_k as follows. For every $k \geq 0$, X_k is the indicator random variable of the event

$$\left\{(b^{u_1,t+13k}_1, b^{u_1,t+13k}_0) = (b^{u_2,t+13k}_1, b^{u_2,t+13k}_0)\right\} \cup \left\{\{u_1,u_2\} \cap \bigcup_{t+13k<t'\leq t+13(k+1)} U^{t'} \neq \emptyset\right\}.$$

If $(b^{u_1,t+13k}_1, b^{u_1,t+13k}_0) \neq (b^{u_2,t+13k}_1, b^{u_2,t+13k}_0)$ then at least one $u \in \{u_1,u_2\}$ satisfies the condition of Lemma 8.4 for $\epsilon_1 = 1/2$ in round $t+13k$. Thus, if $(b^{u_1,t+13k}_1, b^{u_1,t+13k}_0) \neq (b^{u_2,t+13k}_1, b^{u_2,t+13k}_0)$,

$$\Pr\left[\{u_1,u_2\} \cap \bigcup_{t+13k<t'\leq t+13(k+1)} U^{t'} \;\middle|\; \mathcal{C}_{t+13k}\right] \geq \epsilon_2,$$

where ϵ_2 is the constant provided from Lemma 8.4 for $\epsilon_1 = 1/2$, and $\mathcal{C}_{t'}$ is the configuration after round t'. It follows $\Pr[X_k = 1 \mid \mathcal{C}_{t+13k}] \geq \epsilon_2$. Therefore, for any $k \geq 0$,

$$\Pr[X_k = 1 \mid X_1, \dots, X_{k-1}] \geq \epsilon_2.$$

By applying a standard Chernoff bound to $X = \sum_{0\leq k<\kappa} X_k$, where $\kappa = \lceil 12\epsilon_2^{-1}\ln n\rceil$, we obtain

$$\Pr[X < \epsilon_2\kappa/4] < e^{-(3/4)^2\epsilon_2\kappa/2} < n^{-3}. \tag{8.8}$$

Set $t^* = t + 13\kappa$ and $\ell^* \geq 2\cdot 13\kappa$.

We argue that if $X \geq 3$ then $\{u_1,u_2\} \cap Z_{t,t^*} \neq \emptyset$: Suppose, for contradiction, that $X \geq 3$ and $u_1,u_2 \notin Z_{t,t^*}$. Then $(b^{u_1,t+13k}_1, b^{u_1,t+13k}_0) \neq (b^{u_2,t+13k}_1, b^{u_2,t+13k}_0)$ for all $0 \leq k < \kappa$, because the inequality holds for $k = 0$, and the b_1b_0 clock of each u_1,u_2 is incremented by exactly one in each round $t' \in \{t+1,\dots,t^*\}$, by the assumption that $u_1,u_2 \notin Z_{t,t^*}$ which implies the agents do not execute Line 7 and thus neither Line 6. Then, from $X \geq 3$, it follows that some $u \in \{u_1,u_2\}$ belongs to two sets U^{t_1} and U^{t_2}, where $t < t_1 < t_2 \leq t^*$. This means that in round t_1, u executes either Lines 5 to 7, or Line 14. Since u does not execute Line 7 (otherwise $u \in L^{t_1}_0$), it must execute Line 14, thus u's phase is 0 and its level is $\ell^*/2$ after round t_1. Then, since $t_2 - t_1 < t^* - t \leq \ell^*/2$, it follows that in round t_2 the level of u is less than ℓ^*, thus u executes Lines 5 to 7, and in particular, Line 7, which is a contradiction.

Combining the above result with (8.8), we obtain that for any pair u_1,u_2 of agents for which $(b^{u_1,t}_1, b^{u_1,t}_0) \neq (b^{u_2,t}_1, b^{u_2,t}_0)$, we have $\{u_1,u_2\} \cap Z_{t,t^*} \neq \emptyset$, with probability at least $1 - n^{-3}$. By a union bound, the statement is true for all pairs u_1,u_2 simultaneously with probability at least $1 - n^{-1}$. By contrapositive, with probability at least $1 - n^{-1}$, for every pair $u_1,u_2 \notin Z_{t,t^*}$ we have $(b^{u_1,t}_1, b^{u_1,t}_0) = (b^{u_2,t}_1, b^{u_2,t}_0)$, and thus $(b^{u_1,t^*}_1, b^{u_1,t^*}_0) = (b^{u_2,t^*}_1, b^{u_2,t^*}_0)$, because as we argued above, $u_1,u_2 \notin Z_{t,t^*}$ implies the two agents increment their b_1b_0 clock by one in each round $t' \in \{t+1,\dots,t^*\}$. It follows $\Pr[R_t \leq t^*] \geq 1 - n^{-1}$, which proves the first part of the lemma.

For the second part, suppose $N \setminus Z_{t,t'} \subseteq B^{t'}_{ij}$ and $u \in N \setminus Z_{t,t'+1}$. Then $u \notin L^{t'+1}_0$. Also $u \in B^{t'}_{ij}$, because the fact $Z_{t,t'} \subseteq Z_{t,t'+1}$ implies $N \setminus Z_{t,t'+1} \subseteq N \setminus Z_{t,t'} \subseteq B^{t'}_{ij}$. Since $u \notin L^{t'+1}_0$, u does not execute Line 7 in round $t'+1$, thus neither Line 6. From that and $u \in B^{t'}_{ij}$, it follows $u \in B^{t'+1}_{\mathrm{incr}(ij)}$. ∎

Lemma 8.6. *For any constant $0 < \epsilon < 1$, there are constants $\alpha, \lambda > 0$ such that for any $i,j \in \{0,1\}$ and $\ell^* \geq \lambda\ln n$, $\Pr[R'_t > t + \alpha\ln n] = O(1/n)$, where $R'_t = \min\{t' \geq t \colon B^t_{ij} \subseteq Z_{t,t'}$ or $(B^t_{ij} \setminus Z_{t,t'} \subseteq B^{t'}_{ij}$ and $|B^{t'}_{ij}| \geq (1-\epsilon)\cdot n)\}$.*

Proof. The proof is similar to that of Lemma 8.5. Let $u \in B^t_{ij}$. We partition all rounds $t' > t$ into intervals of length 16, and for each interval define a 0-1 random variable Y_k as follows. For every $k \geq 0$, Y_k is the indicator random variable of the event

$$\{u \notin B_{ij}^{t+16k}\} \cup \{|B_{ij}^{t+16k}| \geq (1-\epsilon)\cdot n\} \cup \left\{u \in \bigcup_{t+16k<t'\leq t+16(k+1)} U^{t'}\right\}.$$

From Lemma 8.4, if $u \in B_{ij}^{t+16k}$ and $|B_{ij}^{t+16k}| \leq (1-\epsilon)\cdot n$, then

$$\Pr\left[u \in \bigcup_{t+16k<t'\leq t+16(k+1)} U^{t'} \;\middle|\; \mathcal{C}_{t+16k}\right] \geq \epsilon_2,$$

where ϵ_2 is the constant provided from Lemma 8.4 for $\epsilon_1 = \epsilon$. It follows $\Pr[Y_k = 1 \mid \mathcal{C}_{t+16k}] \geq \epsilon_2$. Therefore, for any $k \geq 0$,

$$\Pr[Y_k = 1 \mid Y_1, \ldots, Y_{k-1}] \geq \epsilon_2.$$

Applying a standard Chernoff bound to $Y = \sum_{0\leq k<\kappa} Y_k$, where $\kappa = \lceil 8\epsilon_2^{-1}\ln n\rceil$, we obtain

$$\Pr[Y < \epsilon_2\kappa/4] < e^{-(3/4)^2\epsilon_2\kappa/2} < n^{-2}. \tag{8.9}$$

Set $t^* = t + 16\kappa$ and $\ell^* \geq 2\cdot 16\kappa$.

Next we argue that if $Y \geq 2$ then (i) $u \in Z_{t,t^*}$, or (ii) there is some $k_u \in \{0,\ldots,\kappa-1\}$ for which $|B_{ij}^{t+16k_u}| \geq (1-\epsilon)\cdot n$. Suppose, for contradiction, that $Y \geq 2$ and neither (i) nor (ii) holds. Then $u \in B_{ij}^{t+16k}$ for all $0 \leq k < \kappa$, because $u \in B_{ij}^t$, and u's b_1b_0 clock is incremented by exactly one in each round $t' \in \{t+1,\ldots,t^*\}$, since the assumption that (i) does not hold implies u does not execute Line 7 and thus neither Line 6. Combing that with the assumption that (ii) does not hold, we obtain that $Y \geq 2$ implies that u belongs to two sets U^{t_1} and U^{t_2}, where $t < t_1 < t_2 \leq t^*$. Then by the same argument as that used in the proof of Lemma 8.5, we conclude that u executes Line 7 in step t_1 or step t_2, contradicting (i).

From the above result and (8.9), we obtain that for any $u \in B_{ij}^t$, at least one of (i) and (ii) holds, with probability at least $1-n^{-2}$. By a union bound, the statement is true for all $u \in B_{ij}^t$ simultaneously, with probability at least $1-n^{-1}$. It follows that, with probability at least $1-n^{-1}$, either (i) holds for all $u \in B_{ij}^t$, or (ii) holds for at least one u. In the case where (i) holds for all $u \in B_{ij}^t$, we have $R_t' \leq t^*$ as $B_{ij}^t \subseteq Z_{t,t^*}$. In the case where (ii) holds for some u, we have $R_t' \leq t_u$ because $|B_{ij}^{t_u}| \geq (1-\epsilon)\cdot n$, and $B_{ij}^t \setminus Z_{t,t_u} \subseteq B_{ij}^{t_u}$ since $t_u \equiv t \pmod 4$. Thus in both cases, we have $R_t' \leq t^* = t + O(\ln n)$. ∎

Using Lemmas 8.5 and 8.6, we argue later that it suffices to consider just two cases: (I) $N = L_{\ell^*/4}^t$, and (II) $N \setminus L_{\ell^*/4}^t \subseteq B_{00}^t$ and $|B_{00}^t| \geq (1-\epsilon)\cdot n$. For case (I) the analysis is reduced to that of the modulo 4 clock of Algorithm 2, as shown in Lemma 8.7. Case (II) is analyzed in Lemmas 8.8 to 8.11.

Lemma 8.7. *There is a constant $\lambda > 0$ such that if $\ell^* \geq \lambda\ln n$ and $L_{\ell^*/4}^t = N$, then $\Pr[T > t + \ell^*/4] = O(1/n)$, where $T = \min\{t' \colon B_{ij}^{t'} = N,$ for some $i,j\}$.*

Proof. Let $\tau = O(\log n)$ be an upper bound obtain from Theorem 5.1, on the number of rounds until Algorithm 2 synchronizes the modulo 4 clocks of all n agents with probability at least $1-n^{-1}$. Let $\ell^* \geq 4\tau$, and suppose that $L_{\ell^*/4}^t = N$. Observe that in each round $t' \in \{t+1,\ldots,t+\tau\}$, each agent u may execute only Lines 2 to 7 and Line 22, since the level of each u increases by at most 1 in each round, thus it is at most $\ell^*/4 + (t'-t) - 1 \leq \ell^*/4 + \tau - 1 < \ell^*/2$ before each round t'. By comparing the above lines of Algorithm 5 with Algorithm 2, we observe that the code for updating clock b_1b_0 is identical in the two protocols. Note, in particular, that $\mu = 0$ if and only if $b_1 = 0$ for the agent v sampled in Algorithm 5. It follows that by round $t + \tau \leq t + \ell^*/4$ of Algorithm 5, all clocks b_1b_0 are synchronized with probability at least $1-n^{-1}$. ∎

The next two results, Lemma 8.8 and Lemma 8.9, are used to prove Lemma 8.10. Roughly speaking, Lemma 8.10 shows that, when case (II) above applies, after a constant number of rounds (multiple of 4), $|B_1^t|$ decreases by at least a constant factor in expectation, while at the same time it remains small w.h.p.

Lemma 8.8. *For any $0 < \epsilon_1 < 1$ and $0 \leq \epsilon_2 \leq 1-\epsilon_1$, if $|B_0^t| \geq (1-\epsilon_1)\cdot n$, $B_1^t \subseteq L_{\ell^*/2-1}^t$, and $|B_0^t \cap (\Phi_0^t \cup C_0^t)| \geq \epsilon_2 n$, then*

(a) $\mathbf{E}\left[|B_1^{t+4}|\right] \leq |B_1^t|\cdot(1+\epsilon_1\epsilon_2-\epsilon_2^2)$, *and*

(b) $\Pr\left[|B_1^{t+4}| > |B_1^t|\cdot(1+\epsilon_1\epsilon_2-\epsilon_2^2)+\epsilon_3 n\right] = e^{-\Omega(n)}$, *for any constant $\epsilon_3 > 0$.*

Proof. Similarly to the analysis of Algorithm 2 in the proof of Theorem 5.1, we can argue that for any $r \geq 0$,[8]

$$\begin{aligned} B_0^{r+1} &= \hat{B}_1^r, & \hat{B}_0^{r+1} &= B_0^r \cup \left(U^{r+1}\cap L_{\ell^*/2-1}^r\right) \\ B_1^{r+1} &= \hat{B}_0^r, & \hat{B}_1^{r+1} &= B_1^r \setminus \left(U^{r+1}\cap L_{\ell^*/2-1}^r\right), \end{aligned} \tag{8.10}$$

where $U^{r+1}\cap L_{\ell^*/2-1}^r$ is the set of agents that modify their clock b_1b_0 in round $r+1$ (by executing Line 6). It follows

$$B_0^{r+2} = B_1^r \setminus \left(U^{r+1}\cap L_{\ell^*/2-1}^r\right), \tag{8.11}$$

$$B_1^{r+2} = B_0^r \cup \left(U^{r+1}\cap L_{\ell^*/2-1}^r\right). \tag{8.12}$$

[8] Recall that $\hat{B}_0^r = B_{01}^r \cup B_{10}^r$ and $\hat{B}_1^r = B_{00}^r \cup B_{11}^r$.

We use these formulas to upper bound first $|B_0^{t+2}|$, and then $|B_1^{t+4}|$.

Since $B_1^t \subseteq L_{\ell^*/2-1}^t$, we have $U^{t+1} \cap L_{\ell^*/2-1}^t = U^{t+1}$. Also, the set U^{t+1} contains (at least) all agents $u \in B_1^t$ that sample some agent $v \in B_0^t \cap (\Phi_0^t \cup C_0^t)$ in round $t+1$, since $m^{v,t} = 0$. Hence,

$$\mathbf{E}[|U^{t+1}|] \geq |B_1^t| \cdot |B_0^t \cap (\Phi_0^t \cup C_0^t)|/n. \tag{8.13}$$

From (8.11), $B_0^{t+2} = B_1^t \setminus U^{t+1}$, and thus

$$\mathbf{E}[|B_0^{t+2}|] \leq |B_1^t| - |B_1^t| \cdot |B_0^t \cap (\Phi_0^t \cup C_0^t)|/n. \tag{8.14}$$

From (8.12), we have $B_1^{t+4} = B_0^{t+2} \cup (U^{t+3} \cap L_{\ell^*/2-1}^{t+2})$. The set $U^{t+3} \cap L_{\ell^*/2-1}^{t+2}$ is a subset of the agents $u \in B_1^{t+2} \cap L_{\ell^*/2-1}^{t+2}$ that sample some agent from B_0^{t+2} in round $t+3$. From (8.12), $B_1^{t+2} \subseteq B_0^t \cup U^{t+1}$. Also, $L_{\ell^*/2-1}^{t+2} \cap (B_0^t \cap \Phi_1^t) = \emptyset$: For any $u \in B_0^t \cap \Phi_1^t$, we have $u \notin U^{t+1}$ since $u \in B_0^t$, thus $u \in \Phi_1^{t+1}$; and for the next round, $t+2$, we have that if $u \notin U^{t+2}$ then $u \in \Phi_1^{t+2}$, while if $u \in U^{t+2}$ then $u \in \Phi_0^{t+2}$ and $\ell^{u,t+2} = \ell^*/2 > \ell^*/2 - 1$; therefore $u \notin L_{\ell^*/2-1}^{t+2}$.

Combining the above we obtain $B_1^{t+2} \cap L_{\ell^*/2-1}^{t+2} \subseteq (B_0^t \cap \Phi_0^t) \cup U^{t+1}$, and $U^{t+3} \cap L_{\ell^*/2-1}^{t+2} \subseteq Z$, where Z is the set of $u \in (B_0^t \cap \Phi_0^t) \cup U^{t+1}$ that sample an agent from B_0^{t+2} in round $t+3$. Then,[9]

$$\mathbf{E}[|Z| \mid |B_0^{t+2}|] = (|B_0^t \cap \Phi_0^t| + |U^{t+1}|) \cdot |B_0^{t+2}|/n. \tag{8.15}$$

It follows that $B_1^{t+4} \subseteq B_0^{t+2} \cup Z$, and

$$\begin{aligned}\mathbf{E}[&|B_1^{t+4}| \mid |B_0^{t+2}|] \\ &\leq |B_0^{t+2}| + (|B_0^t \cap \Phi_0^t| + |U^{t+1}|) \cdot |B_0^{t+2}|/n.\end{aligned}$$

The unconditional expectation of $|B_1^{t+4}|$ is then

$$\begin{aligned}\mathbf{E}[|B_1^{t+4}|] &\leq \mathbf{E}[|B_0^{t+2}|] + |B_0^t \cap \Phi_0^t| \cdot \mathbf{E}[|B_0^{t+2}|]/n \\ &\quad + \mathbf{E}[|U^{t+1}| \cdot |B_0^{t+2}|]/n \\ &\leq \mathbf{E}[|B_0^{t+2}|] + |B_0^t \cap \Phi_0^t| \cdot \mathbf{E}[|B_0^{t+2}|]/n \\ &\quad + |B_1^t|^2/n - |B_1^t| \cdot \mathbf{E}[|B_0^{t+2}|]/n,\end{aligned}$$

where in the second inequality we used $|U^{t+1}| \cdot |B_0^{t+2}| = (|B_1^t| - |B_0^{t+2}|) \cdot (|B_1^t| - |U^{t+1}|) \leq (|B_1^t| - |B_0^{t+2}|) \cdot |B_1^t|$. Substituting (8.14) above, yields

$$\begin{aligned}\mathbf{E}[|B_1^{t+4}|] &\leq |B_1^t| \cdot (1 - |B_0^t \cap (\Phi_0^t \cup C_0^t)|/n) \\ &\qquad \cdot (1 + |B_0^t \cap \Phi_0^t|/n - |B_1^t|/n) + |B_1^t|^2/n \\ &\leq |B_1^t| \cdot \big(1 - |B_0^t \cap (\Phi_0^t \cup C_0^t)|^2/n^2 \\ &\qquad + |B_0^t \cap (\Phi_0^t \cup C_0^t)| \cdot |B_1^t|/n^2\big) \\ &\leq |B_1^t| \cdot (1 - \epsilon_2^2 + \epsilon_1\epsilon_2),\end{aligned}$$

[9] Note $|U^{t+1}| = |B_1^t| - |B_0^{t+2}|$ thus $|U^{t+1}|$ is fixed given $|B_0^{t+2}|$.

since $|B_0^t \cap (\Phi_0^t \cup C_0^t)| \geq \epsilon_2 n$ and $|B_1^t| \leq \epsilon_1 n$. This completes the proof of (a).

Next we show (b). Since $|U_{t+1}|$ is a sum of independent 0-1 random variables, a Chernoff bound gives $\Pr[|U_{t+1}| < \mathbf{E}[|U_{t+1}|] - \epsilon_3 n/4] = e^{-\Omega(n)}$. Using the lower bound on $\mathbf{E}[|U_{t+1}|]$ from (8.13), gives

$$\Pr[|U_{t+1}| < |B_1^t| \cdot |B_0^t \cap (\Phi_0^t \cup C_0^t)|/n - \epsilon_3 n/4] = e^{-\Omega(n)}.$$

Fix U_{t+1} such that $|U_{t+1}| \geq |B_1^t| \cdot |B_0^t \cap (\Phi_0^t \cup C_0^t)|/n - \epsilon_3 n/4$. Given the configuration $\mathcal{C}_{t+2}$ after round $t+2$, $|Z|$ is also a sum of 0-1 i.r.v., thus $\Pr[|Z| > \mathbf{E}[|Z|] + \epsilon_3 n/4 \mid \mathcal{C}_{t+2}] = e^{-\Omega(n)}$, and by (8.15),

$$\begin{aligned}\Pr[|Z| > (|B_0^t \cap \Phi_0^t| + |U^{t+1}|) \cdot |B_0^{t+2}|/n + \epsilon_3 n/4 \mid \mathcal{C}_{t+2}] \\ = e^{-\Omega(n)}.\end{aligned}$$

Fix Z such that $|Z| \leq (|B_0^t \cap \Phi_0^t| + |U^{t+1}|) \cdot |B_0^{t+2}|/n + \epsilon_3 n/4$. Using that $|B_0^{t+2}| = |B_1^t| - |U^{t+1}|$ and $|B_1^{t+4}| \leq |B_0^{t+2}| + |Z|$, and using also the bounds on $|U^{t+1}|, |Z|$ we fixed above, we obtain

$$\begin{aligned}|B_1^{t+4}| \leq (|B_1^t| - |U^{t+1}|) \cdot (1 + |B_0^t \cap \Phi_0^t|/n + |U^{t+1}|/n) \\ + \epsilon_3 n/4.\end{aligned}$$

The right side is maximized by using the lower bound of $|U^{t+1}|$. That, and some calculations give

$$\begin{aligned}|B_1^{t+4}| &\leq |B_1^t| \cdot \big(1 - |B_0^t \cap (\Phi_0^t \cup C_0^t)|^2/n^2 \\ &\qquad + |B_1^t| \cdot |B_0^t \cap (\Phi_0^t \cup C_0^t)|/n^2\big) + \epsilon_3 n \\ &\leq |B_1^t| \cdot (1 - \epsilon_2^2 + \epsilon_1\epsilon_2) + \epsilon_3 n.\end{aligned}$$

Part (b) then follows, as the bounds we used for $|U^{t+1}|, |Z|$ hold with probability $1 - e^{-\Omega(n)}$. ∎

Lemma 8.9. *For any round t, there is a round $\rho = t+4k$, where $k \in \{0, \ldots, 3\}$, such that $|B_0^\rho \cap (\Phi_0^\rho \cup C_0^\rho)| \geq |B_0^t|/4$.*

Proof. For any $k \geq 0$, let $t_k = t + 4k$. To prove the lemma, it suffices to show that for any $u \in B_0^t$, there is a round $t_u \in \{t_0, t_1, t_2, t_3\}$ such that

$$u \in B_0^{t_u} \cap (\Phi_0^{t_u} \cup C_0^{t_u}), \tag{8.16}$$

because then $|B_0^\rho \cap (\Phi_0^\rho \cup C_0^\rho)| \geq |B_0^t|/4$ for at least one $\rho \in \{t_0, t_1, t_2, t_3\}$.

If $u \in B_0^t \cap \Phi_0^t$ or $u \in B_0^t \cap \Phi_1^t \cap C_0^t$, then (8.16) holds trivially, for $t_u = t_0 = t$. The remaining case is when $u \in B_0^t \cap \Phi_1^t \cap C_1^t$. In this case, from Lemma 8.3(b), there is some $t'_u \in \{t+1, \ldots, t+11\}$ such that (i) $u \in B_{00}^{t'_u} \cap (\Phi_0^{t'_u} \cup C_0^{t'_u})$, or (ii) $u \in U^{t'_u}$. Next we consider the smallest such round t'_u.

First, suppose that (i) holds and (ii) does not hold. Then, the clock b_1b_0 of u is incremented by exactly one in each round $t' \in \{t+1,\ldots,t'_u\}$, because $u \notin U^{t'}$, and the same is true for $t' = t'_u + 1$, because $u \in B_{00}^{t'_u}$. It follows that if $u \in B_{00}^t$ then $t'_u \in \{t+4, t+8\} = \{t_1, t_2\}$, and (8.16) holds for $t_u = t'_u$. While if $u \in B_{01}^t$ then $t'_u \in \{t+3, t+7, t+11\}$, thus (8.16) holds for $t_u = t'_u + 1 \in \{t_1, t_2, t_3\}$.

Suppose now that (ii) holds, i.e., $u \in U^{t'_u}$. As before the clock b_1b_0 of u is incremented by exactly one in each round $t' \in \{t+1,\ldots,t'_u - 1\}$, since $u \notin U^{t'}$. In round t'_u, u moves to phase 0 and level $\ell^*/2$, and its clock b_1b_0 is again incremented by one.

If $u \in B_{00}^t$, it follows that $t'_u \in \{t+3, t+4, t+7, t+8, t+11\}$. If $t'_u \in \{t+4, t+8\}$ then $u \in B_{00}^{t'_u} \cap \Phi_0^{t'_u}$, thus (8.16) holds for $t_u = t'_u \in \{t_1, t_2\}$. If $t'_u \in \{t+3, t+7, t+11\}$ then $u \in B_{11}^{t'_u} \cap L_{\ell^*/2}^{t'_u}$. Thus, in round $t'_u + 1$, regardless of whether or not $u \in U^{t'_u+1}$, we have $u \in B_{00}^{t'_u+1} \cap \Phi_0^{t'_u+1}$. Therefore, in this case (8.16) holds for $t_u = t'_u + 1 \in \{t_1, t_2, t_3\}$.

If $u \in B_{01}^t$, we have $t'_u \in \{t+2, t+3, t+6, t+7, t+10, t+11\}$, and we can similarly argue that if $t'_u \in \{t+3, t+7, t+11\}$ then (8.16) holds for $t_u = t'_u + 1$, while if $t'_u \in \{t+2, t+6, t+10\}$ then (8.16) holds for $t_u = t'_u + 2$. ◻

Lemma 8.10. *There are constants $\varepsilon_1, \varepsilon_2 > 0$, such that if $|B_0^t| \geq (1-\varepsilon_1)\cdot n$, $B_1^t \subseteq L_{\ell^*/2-16}^t$, and $\rho \in \{t, t+4, t+8, t+12\}$ is the smallest round predicted by Lemma 8.9, then*

(a) $\mathbf{E}\left[|B_1^{\rho+4}|\right] \leq (1-\varepsilon_2)\cdot|B_1^t|$, *and*

(b) $\Pr\left[|B_0^{\rho+4}| < (1-\varepsilon_1)\cdot n\right] = e^{-\Omega(n)}$.

Proof. If $B_1^t = \emptyset$ then $\rho = 0$ and $B_1^{t+4} = \emptyset$, by (8.10), thus the lemma holds. Next we assume $B_1^t \neq \emptyset$.

For any $k \geq 0$, let $t_k = t + 4k$. Then, ρ is a random variable that takes values in $\{t_0, t_1, t_2, t_3\}$. We use the following trick which allows us to argue about the fixed round t_3 instead of round ρ. If $\rho = t_k \neq t_3$, then we replace rounds $t_k + 1$ up to t_3 by "dummy" rounds in which agents do nothing (in particular, they do not change their state). Assuming this modification, it suffices to show (a) and (b) using t_3 in place of ρ, which is what we do in the rest of the proof.

For any $k \geq 0$, given configuration $\mathcal{C}_{t_k}$, if $|B_0^{t_k}| \geq (1-\epsilon_1)\cdot n$ and $B_1^{t_k} \subseteq L_{\ell^*/2-1}^t$, then Lemma 8.8 applied for any ϵ_2 and any constant $\epsilon_3 > 0$ yields

(8.17)
$$\mathbf{E}\left[|B_1^{t_{k+1}}| \;\middle|\; \mathcal{C}_{t_k}\right] \leq |B_1^{t_k}|\cdot(1+\epsilon_1^2/4),$$
$$\Pr\left[|B_1^{t_{k+1}}| > |B_1^{t_k}|\cdot(1+\epsilon_1^2/4)+\epsilon_3 n \;\middle|\; \mathcal{C}_{t_k}\right] = e^{-\Omega(n)},$$

because $1 + \epsilon_1\epsilon_2 - \epsilon_2^2 \leq 1 + \epsilon_1^2/4$, for any $\epsilon_2 \geq 0$. Equations (8.17) are also trivially true if rounds $t_k + 1$ up to t_{k+1} are dummy rounds.

We fix $\varepsilon_1 = 0.1$. We also define shorthand notation $x_k = |B_1^{t_k}|$. Then $x_0 \leq \varepsilon_1 n = 0.1n$. By applying (8.17) for all $k \in \{0,1,2,3\}$, using a small enough constant $\epsilon_3 > 0$, we obtain

$$\mathbf{E}[x_1] \leq x_0\cdot(1+(0.1)^2/4),$$
$$\Pr[x_1 > 0.101n] = e^{-\Omega(n)},$$
$$\mathbf{E}[x_2 \mid \mathcal{C}_{t_1}, x_1 \leq 0.101n] \leq x_1\cdot(1+(0.101)^2/4),$$
$$\Pr[x_2 > 0.102n \mid \mathcal{C}_{t_1}, x_1 \leq 0.101n] = e^{-\Omega(n)},$$
$$\mathbf{E}[x_3 \mid \mathcal{C}_{t_2}, x_2 \leq 0.102n] \leq x_2\cdot(1+(0.102)^2/4),$$
$$\Pr[x_3 > 0.1025n \mid \mathcal{C}_{t_2}, x_2 \leq 0.102n] = e^{-\Omega(n)}.$$

By the union bound,

$$\Pr[x_2 > 0.102n] \leq 2\cdot e^{-\Omega(n)},$$
$$\Pr[x_3 > 0.1025n] \leq 3\cdot e^{-\Omega(n)}.$$

Also

$$\begin{aligned}\mathbf{E}[x_3] &\leq n\cdot\Pr[x_2 > 0.102n] + \mathbf{E}[x_2]\cdot(1+(0.102)^2/4)\\ &\leq 2ne^{-\Omega(n)} + \big(n\cdot\Pr[x_1 > 0.101n]\\ &\qquad + \mathbf{E}[x_1]\cdot(1+(0.101)^2/4)\big)\cdot(1+(0.102)^2/4)\\ &\leq 4ne^{-\Omega(n)} + x_0\cdot(1+(0.1)^2/4)\\ &\qquad\cdot(1+(0.101)^2/4)\cdot(1+(0.102)^2/4)\\ &\leq 4ne^{-\Omega(n)} + x_0\cdot 1.008.\end{aligned}$$

Using the above bounds for x_3, we can bound x_4 by applying Lemma 8.8 once more. Form Lemma 8.9, we have $|B_0^{t_3} \cap (\Phi_0^{t_3} \cup C_0^{t_3})| \geq |B_0^t|/4 \geq (1-\varepsilon_1)\cdot n/4 = 0.225n$. Then, from Lemma 8.8(a),

$$\begin{aligned}&\mathbf{E}[x_4 \mid \mathcal{C}_{t_3}, x_3 \leq 0.1025n]\\ &\quad\leq x_3\cdot(1+0.1025\cdot 0.225-(0.225)^2) \leq x_3\cdot 0.973,\end{aligned}$$

and

$$\begin{aligned}\mathbf{E}[x_4] &\leq n\cdot\Pr[x_3 > 0.1025n] + \mathbf{E}[x_3]\cdot 0.973\\ &\leq 7n\cdot e^{-\Omega(n)} + x_0\cdot 1.008\cdot 0.973\\ &\leq 7n\cdot e^{-\Omega(n)} + x_0\cdot 0.981\\ &\leq 0.99x_0,\end{aligned}$$

where the last inequality holds for all large enough n, because the assumption $B_1^t \neq \emptyset$ implies $x_0 \geq 1$. This completes the proof of (a). From Lemma 8.8(b),

$$\begin{aligned}&\Pr[x_4 > 0.1025n\cdot 0.973 + \epsilon'_3 n \mid \mathcal{C}_{t_3}, x_3 \leq 0.1025n]\\ &\qquad = e^{-\Omega(n)}.\end{aligned}$$

Since $0.103n \cdot 0.973 < 0.1n = \varepsilon_1 n$, combining the above inequality and $\Pr[x_3 > 0.1025n] \le 3 \cdot e^{-\Omega(n)}$, we obtain $\Pr[x_4 > \varepsilon_1 n] \le 4 \cdot e^{-\Omega(n)}$, which implies (b). ◻

We use Lemma 8.10 to show the following counterpart of Lemma 8.7.

Lemma 8.11. *There are constants $\varepsilon, \lambda > 0$ such that if $\ell^* \ge \lambda \ln n$, $N \setminus L^t_{\ell^*/4} \subseteq B^t_{00}$, and $|B^t_{00}| \ge (1-\varepsilon)\cdot n$, then $\Pr[T > t + \ell^*/4] = O(1/n)$, where $T = \min\{t' \colon B^{t'}_{ij} = N$, for some $i, j\}$.*

Proof. For $d \in \{0,1\}$, let $T_d = \min\{t' \ge t \colon t' \equiv t + d \pmod 4,\ B^{t'}_0 = N\}$. We will show

$$\Pr[T_d > t + \ell^*/4] = O(1/n). \tag{8.18}$$

Using that, we can argue similarly to the analysis of Algorithm 2 that $\Pr[T > t + \ell^*/4] = O(1/n)$: From (8.10), it follows that for all $t' \ge T_d$ with $t' \equiv T_d \pmod 4$, $B^{t'}_0 = \hat{B}^{t'+1}_0 = B^{t'+2}_1 = \hat{B}^{t'+3}_1 = N$. From that, and fact $T_1 \equiv T_0 + 1 \pmod 4$, we obtain that if $T_0 < T_1$, then $\hat{B}^{T_1}_0 = B^{T_1-1}_0 = N = B^{T_1}_0$, which implies $B^{T_1}_{01} = N$. While if $T_0 > T_1$, then $\hat{B}^{T_0}_1 = B^{T_0-3}_0 = N = B^{T_0}_0$, which implies $B^{T_0}_{00} = N$. Hence, in both cases $T \le \max\{T_0, T_1\}$. The claim then follows from (8.18), using a union bound.

It remains to prove (8.18). We consider T_0 first. We partition all rounds $t' > t$ into intervals of variable lengths, with the set of possible lengths being $\{4, 8, 12, 16\}$. The kth such interval is the set of rounds $\{s_{k-1}+1, \ldots, s_k\}$, where $s_0 = t$, $s_k = \rho_{s_{k-1}} + 4$ for $k \ge 1$, and $\rho_{t'}$ denotes the smallest round ρ predicted by Lemma 8.9 for a given round t', i.e., $t' \le \rho_{t'} \equiv t' \pmod 4$, and $|B^{\rho_{t'}}_0 \cap (\Phi^{\rho_{t'}}_0 \cup C^{\rho_{t'}}_0)| \ge |B^{t'}_0|/4$.

Set $\varepsilon = \varepsilon_1$ and $\ell^* \ge 64 \cdot \lceil \varepsilon_2^{-1} \ln(\varepsilon_1 n^2) \rceil$, where $\varepsilon_1, \varepsilon_2$ are the constants of Lemma 8.10.

For any $k \ge 1$, let X_k be a non-negative integer random variable, where $X_k = |B^{s_k}_1|$ if $|B^{s_{k'}}_1| \le \varepsilon n$ for all $0 \le k' < k$, and $X_k = 0$ otherwise. For any $1 \le k \le \ell^*/(4 \cdot 16)$, we have $s_{k-1} \le t + \ell^*/4 - 16$, and since $B^t_1 \subseteq N \setminus B^t_{00} \subseteq L^t_{\ell^*/4}$, it follows $B^{s_{k-1}}_1 \subseteq L^{s_{k-1}}_{\ell^*/2-16}$. Then, for any k in the above range, Lemma 8.10(a) implies

$$\mathbf{E}[X_k] \le (1-\varepsilon_2)\cdot\mathbf{E}[X_{k-1}] \le (1-\varepsilon_2)^k \cdot |B^t_1| \le (1-\varepsilon_2)^k \varepsilon_1 n.$$

For $\kappa = \lceil \varepsilon_2^{-1} \ln(\varepsilon_1 n^2) \rceil \le \ell^*/(4 \cdot 16)$, the above gives $\mathbf{E}[X_\kappa] \le 1/n$, and by Markov's inequality

$$\Pr[X_\kappa \ne 0] = \Pr[X_\kappa \ge 1] \le \mathbf{E}[X_\kappa]/1 \le 1/n.$$

Moreover, from Lemma 8.10(b), it follows

$$\Pr\left[|B^{s_k}_1| \le \varepsilon n, \text{ for all } 0 \le k < \kappa\right] \ge 1 - \kappa \cdot e^{-\Omega(n)}.$$

Combining the last two results above, yields $\Pr[|B^{s_\kappa}_1| \ne 0] = O(1/n)$. Since $s_\kappa \le t + \ell^*/4$, this implies (8.18) for $d = 0$.

The proof of (8.18) for $d = 1$ is the same except that we replace t by $t+1$, and observe that, from the lemma's assumptions $|B^t_{00}| \ge (1+\varepsilon)\cdot n$ and $N \setminus L^t_{\ell^*/4} \subseteq B^t_{00}$, it follows $|B^{t+1}_0| \ge (1+\varepsilon)\cdot n$ and $B^{t+1}_1 \subseteq N \setminus B^t_{00} \subseteq L^{t+1}_{\ell^*/4+1}$. The last two inequalities allow us to apply Lemma 8.10 in the same way we did for the case of $d = 0$ above. ◻

Proof of Theorem 8.1. First we upper bound $T = \min\{t \colon B^t_{ij} = N$, for some $i, j\}$, i.e., the first round when all clocks $b_1 b_0$ are synchronized.

From Lemma 8.5, there exists some constant λ_1 such that if $\ell^* \ge \lambda_1 \ln n$, then the round $T_1 = \min\{t \colon N \setminus Z_{0,t} \subseteq B^t_{00}\}$ satisfies $\Pr[T_1 \le \ell^*/8] = 1 - O(1/n)$. Precisely, the fist part of Lemma 8.5 gives $\Pr[T'_1 \le \ell^*/8 - 3] = 1 - O(1/n)$, where $T'_1 = \min\{t \colon N \setminus Z_{0,t} \subseteq B^t_{ij}$, for some $i, j\}$, and the second part of Lemma 8.5 gives $T_1 \le T'_1 + 3$.

Fix now round T_1, and suppose that $T_1 \le \ell^*/8$. Let $T_2 = \min\{t \ge T_1 \colon B^{T_1}_{00} \subseteq Z_{T_1,t}\}$, and $T_3 = \min\{t \ge T_1 \colon B^{T_1}_{00} \setminus Z_{T_1,t} \subseteq B^t_{00},\ B^t_{00} \ge (1-\varepsilon)\cdot n\}$, where ε is the constant of Lemma 8.11. From Lemma 8.6, there is a constant λ_2 such that if $\ell^* \ge \lambda_2 \ln n$ then $\Pr[\min\{T_2, T_3\} \le T_1 + \ell^*/8] = 1 - O(1/n)$.

Fix round $T_4 = \min\{T_2, T_3\}$, and suppose that $T_4 \le T_1 + \ell^*/8$. Then $T_4 \le \ell^*/4$, since we have assumed $T_1 \le \ell^*/8$. Moreover, from the definition of T_1 and $T_1 \le T_4 \le \ell^*/4$, we have

$$N \setminus B^{T_1}_{00} \subseteq Z_{0,T_1} \subseteq Z_{0,T_4} = L^{T_4}_{T_4} \subseteq L^{T_4}_{\ell^*/4}. \tag{8.19}$$

Next we consider the two cases $T_4 = T_2$ and $T_4 = T_3$ separately.

First, suppose $T_4 = T_2$. Then, $B^{T_1}_{00} \subseteq Z_{T_1,T_4} \subseteq L^{T_4}_{\ell^*/8}$. From that and (8.19), it follows $N = L^{T_4}_{\ell^*/4}$. Then, from Lemma 8.7, there is a constant $\lambda_3 > 0$ such that if $\ell^* \ge \lambda_3 \ln n$, then $\Pr[T \le T_4 + \ell^*/4] = 1 - O(1/n)$.

Suppose now that $T_4 \ne T_2$, thus $T_4 = T_3$. Then $B^{T_1}_{00} \setminus Z_{T_1,T_4} \subseteq B^{T_4}_{00}$ and $|B^{T_4}_{00}| \ge (1-\varepsilon)\cdot n$. The first equation implies $B^{T_1}_{00} \setminus L^{T_4}_{\ell^*/4} \subseteq B^{T_4}_{00}$. Also, from (8.19), we have $N \setminus L^{T_4}_{\ell^*/4} \subseteq B^{T_1}_{00}$. It follows $N \setminus L^{T_4}_{\ell^*/4} \subseteq B^{T_1}_{00} \setminus L^{T_4}_{\ell^*/4} \subseteq B^{T_4}_{00}$. Since also $|B^{T_4}_{00}| \ge (1-\varepsilon)\cdot n$, Lemma 8.11 implies there is a constant $\lambda_4 > 0$ such that if $\ell^* \ge \lambda_4 \ln n$, then $\Pr[T \le T_4 + \ell^*/4] = 1 - O(1/n)$.

Therefore, in both cases above, we have $\Pr[T \le T_4 + \ell^*/4] = 1 - O(1/n)$, which implies that $\Pr[T \le \ell^*/2] = 1 - O(1/n)$, since $T_4 \le \ell^*/4$.

Recall that the above result holds conditionally on event $\{T_1 \le \ell^*/8\} \cap \{T_4 \le T_1 + \ell^*/8\}$, and assuming ℓ^* is large enough. If $\ell^* \ge \lambda \ln n$, where

$\lambda = \max\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$, then all conditions for ℓ^* are met simultaneously, and $\Pr[\{T_1 \le \ell^*/8\} \cap \{T_4 \le T_1 + \ell^*/8\}] = 1 - 2 \cdot O(1/n)$. It follows that the unconditional probability that $T \le \ell^*/2$ is $1 - 3 \cdot O(1/n)$.

We can amplify the above probability to $1 - O(1/n^k)$, by repeating the argument k times. Moreover, from Lemma 8.1, once clocks $b_1 b_0$ get synchronized for the first time, they remain synchronized. This completes the proof that clocks $b_1 b_0$ become synchronized in $O(\log n)$ rounds w.h.p.

The rest of the proof is straightforward. As explained in Section 8.1, all agents reach level 1 by round $T + \ell^* + 1$. Then clocks $c_1 c_0$ are synchronized in $O(\log n)$ additional rounds w.h.p., as all agents execute the modulo 4 synchronization protocol in sync: clock $c_1 c_0$ is updated when $b_1 b_0 = 01$, and incremented when $b_1 b_0 = 11$. Once both clocks are synchronized across all agents, the agents execute A in sync, twice every 16 rounds: an even round of A is executed when $c_1 c_0 b_1 b_0 = 1000$, and an odd round when $c_1 c_0 b_1 b_0 = 1100$. ∎

APPENDIX

A Formal Definition of Simulations

An *execution* E_A of some protocol A is a sequence $C_0, I_1, C_1, I_2, C_2 \ldots$, where C_t, for $t \ge 0$, is the *configuration* (i.e., the vector of states of all agents) after the first t rounds, and I_t, for $t \ge 1$, is the *communication pattern* in round t, describing which agent v is sampled by each u.

For any integers $k \ge 0$ and $\ell \ge 1$, and any set $S \subseteq \{0, 1, \ldots, \ell - 1\}$, we denote by $J(k, \ell, S)$ an infinite subsequence of $(k, k+1, k+2, \ldots)$, where $i \in J(k, \ell, S)$ if and only if $i = k$, or $i > k$ and $(i - k) \bmod \ell \in S$.

A protocol B *simulates* protocol A if there is an integer $\ell \ge 1$, a set $S \subseteq \{0, 1, \ldots, \ell - 1\}$, and a function F from the state space of an agent in B to the state space of an agent in A,[10] such that the following is true: For a random execution $C_0, I_1, C_1, I_2, C_2 \ldots$ of B with an arbitrary initial configuration C_0, there is some d (which is a random variable that depends on C_0 and the communication patterns up to d), such that $F(C_{j_0}), I_{j_1}, F(C_{j_1}), I_{j_2}, F(C_{j_2}), \ldots$ is an execution of A, where $(j_0, j_1, \ldots) = J(d, \ell, S)$. We call B a *simulator* of A, and refer to d and the ratio $\ell : |S|$ as the *delay* and *slowdown* of the simulation, respectively.

B Analysis of Binary Clock

We prove the following statement.

[10] We will write $F(C_B)$ for a configuration C_B of B to denote the configuration C_A of A obtained by applying function F to the state of each agent in C_B.

THEOREM B.1. *Starting from any initial configuration, Algorithm 1 synchronizes all binary clocks in $O(\log n)$ rounds w.h.p.*

B.1 Preliminaries. For $t \ge 0$, let X_t denote the set of agents whose clock is 1 immediately after the first t rounds, and let $x_t = |X_t|/n$ denote the fraction of those agents. Recall that N is the set of all agents. By $B(m, p)$ we denote a binomial random variable with parameters m and p.

LEMMA B.1. *For any $x, x' \in \{0, 1/n, 2/n, \ldots, n/n\}$,*

$$\Pr[x_{t+1} = 1 - x + x' \mid x_t = x] = \Pr[B(nx, 1 - x) = nx']. \tag{B.1}$$

and

$$\mathbf{E}[x_{t+1} \mid x_t = x] = 1 - x^2. \tag{B.2}$$

Proof. The state transitions in round $t+1$ are as follows: (1) if $u \in N \setminus X_t$ then $u \in X_{t+1}$; and (2) if $u \in X_t$ then $u \in X_{t+1}$ if u samples an agents from $N \setminus X_t$ in round $t+1$, and $u \in N \setminus X_{t+1}$ otherwise. It follows that $X_{t+1} = (N \setminus X_t) \cup Z$, where Z is the set of agents $u \in X_t$ that sample some agent from $N \setminus X_t$ in round $t+1$. Therefore, $|X_{t+1}| = n - |X_t| + |Z|$, and given $|X_t| = nx$, we have $|Z| \sim B(nx, 1 - x)$. This implies (B.1). Also,

$$\begin{aligned}
\mathbf{E}[|X_{t+1}| \mid |X_t| = nx] &= n - nx + \mathbf{E}[|Z| \mid |X_t| = nx] \\
&= n - nx + \mathbf{E}[B(nx, 1 - x)] \\
&= n - nx + nx(1 - x) \\
&= n(1 - x^2).
\end{aligned}$$

This implies (B.2). ∎

By β we denote the unique non negative fixed point of the recurrence $F_{t+1} = 1 - F_t^2$, i.e., the non negative root of equation $\beta = 1 - \beta^2$. It is easy to compute that

$$\beta = \frac{\sqrt{5}}{2} - \frac{1}{2} \approx 0.618.$$

It follows from (B.2) that $\mathbf{E}[x_{t+1} \mid x_t = \beta] = \beta$, and also $\mathbf{E}[x_{t+1} \mid x_t = x] > \beta$ if $x < \beta$, and $\mathbf{E}[x_{t+1} \mid x_t = x] < \beta$ if $x > \beta$.

The following tail bounds are obtained using standard Chernoff bounds.

LEMMA B.2. *For any $\delta > 0$,*

$$\Pr[x_{t+1} > 1 - x^2 + \delta x(1 - x) \mid x_t = x] < e^{-\delta^2 nx(1-x)/(2+\delta)}, \tag{B.3}$$

and

$$\Pr[x_{t+1} < 1 - x^2 - \delta x(1 - x) \mid x_t = x] < e^{-\delta^2 nx(1-x)/2}. \tag{B.4}$$

Proof. Suppose that $x_t = x$. For (B.3), we have

$$\begin{aligned}&\Pr[x_{t+1} > 1 - x^2 + \delta x(1-x)]\\&= \Pr[x_{t+1} > 1 - x + (1+\delta)x(1-x)]\\&= \Pr[B(nx, 1-x) > (1+\delta)nx(1-x)], \text{ by (B.1)}\\&< e^{-\delta^2 nx(1-x)/(2+\delta)}.\end{aligned}$$

where in the last line we applied a standard Chernoff upper bound. The proof of (B.4) is similar, and uses the Chernoff lower bound

$$\Pr[B(nx, 1-x) < (1-\delta)nx(1-x)] < e^{-\delta^2 nx(1-x)/2}. \qquad \square$$

The next simple bound is useful when x_t is small.

Lemma B.3. $\Pr[x_{t+1} = 1 \mid x_t = x] \geq 1 - nx^2$.

Proof. In order to have $x_{t+1} = 1$, every agent $u \in X_t$ must sample an agent from set $N \setminus X_t$ in round $t+1$. When $x_t = x$, the probability that a given $u \in X_t$ samples some agent from $N \setminus X_t$ is $1 - x$, thus a union bound over all the $|X_t| = nx$ agents $u \in X_t$ proves the claim. $\square$

The following bound on the binomial distribution is obtained using Stirling's approximation.

Lemma B.4. *For the binomial random variable $B(m,p)$ with $0 < p \leq 1/2$, and any $0 < k \leq 2mp$,*

$$\Pr[B(m,p) = k] \geq \frac{\sqrt{2\pi}}{e^2} \cdot \sqrt{\frac{m}{k(m-k)}} \cdot e^{-\frac{m(mp-k)^2}{k(m-k)}}.$$

Proof. We use Stirling's approximation formula,

$$\sqrt{2\pi n}(n/e)^n < n! < e\sqrt{n}(n/e)^n.$$

Let $\lambda = mp - k$.

$$\begin{aligned}&\Pr[B(m,p) = k]\\&= \binom{m}{k} p^k (1-p)^{m-k}\\&= \frac{m!}{k!(m-k)!} \cdot p^k (1-p)^{m-k}\\&\geq \frac{\sqrt{2\pi m}(m/e)^m}{e^2\sqrt{k(m-k)}(k/e)^k((m-k)/e)^{m-k}} \cdot p^k(1-p)^{m-k}\\&= \frac{\sqrt{2\pi m}}{e^2\sqrt{k(m-k)}} \cdot \frac{(mp)^k}{k^k} \cdot \frac{(m-mp)^{m-k}}{(m-k)^{m-k}}\\&= \frac{\sqrt{2\pi m}}{e^2\sqrt{k(m-k)}} \cdot \left(\frac{k+\lambda}{k}\right)^k \cdot \left(\frac{m-k-\lambda}{m-k}\right)^{m-k}\\&\geq \frac{\sqrt{2\pi m}}{e^2\sqrt{k(m-k)}} \cdot e^{\lambda - \lambda^2/k} \cdot e^{-\lambda - \lambda^2/(m-k)}\\&= \frac{\sqrt{2\pi m}}{e^2\sqrt{k(m-k)}} \cdot e^{-m\lambda^2/(k(m-k))},\end{aligned}$$

where the second-last line was obtained using the fact $1 + x \geq e^{x - x^2}$, for $x \geq -1/2$. Condition $x \geq -1/2$ holds in our case, as $\lambda/k \geq 0$, and $-\lambda/(m-k) = -(mp-k)/(m-k) \geq -(mp)/m = -p \geq -1/2$. $\square$

Using Lemma B.4, we show the following anti-concentration result. Recall, $\mathbf{E}[x_{t+1} \mid x_t = x] = 1 - x^2$, from (B.2).

Lemma B.5. *For any $1/2 \leq x \leq 2/3$ and $4n^{-1/2} \leq \delta \leq (1/18) \cdot n^{1/2}$,*

$$\Pr[x_{t+1} \leq 1 - x^2 - \delta n^{-1/2} \mid x_t = x] \geq 0.62 \cdot \delta e^{-48\delta^2}.$$

Proof. Suppose that $x_t = x$. We have

$$\begin{aligned}&\Pr[x_{t+1} \leq 1 - x^2 - \delta n^{-1/2}]\\&\qquad = \Pr[x_{t+1} \leq 1 - x + x(1-x) - \delta x^{-1/2}]\\&\qquad = \Pr[B(nx, 1-x) \leq nx(1-x) - \delta x^{1/2}],\end{aligned}$$

by (B.1). Let $m = nx$ and $p = 1 - x$. Since $1/2 \leq x \leq 2/3$, it follows that $n/2 \leq m \leq 2n/3$, $1/3 \leq p \leq 1/2$, and $2n/9 \leq mp \leq n/4$ (we implicitly use these bounds below). Let also $\lambda = \delta n^{1/2}$, thus $4 \leq \lambda \leq n/18 \leq mp/4$. Then,

$$\begin{aligned}&\Pr[x_{t+1} \leq 1 - x^2 - \delta n^{-1/2}]\\&= \Pr[B(m,p) \leq mp - \lambda]\\&= \sum_{0 \leq k \leq mp-\lambda} \Pr[B(m,p) = k]\\&\geq \sum_{0 \leq k \leq mp-\lambda} \frac{\sqrt{2\pi}}{e^2} \cdot \sqrt{\frac{m}{k(m-k)}} \cdot e^{-\frac{m\cdot(k-mp)^2}{k\cdot(m-k)}}\\&\geq \sum_{mp-2\lambda \leq k \leq mp-\lambda} \frac{\sqrt{2\pi}}{e^2} \cdot \sqrt{\frac{m}{m^2/4}} \cdot e^{-\frac{m\cdot(2\lambda)^2}{(mp-2\lambda)(m-(mp-2\lambda))}}\\&\geq \lfloor\lambda\rfloor \cdot \frac{\sqrt{2\pi}}{e^2} \cdot \frac{2}{\sqrt{m}} \cdot e^{-\frac{m\cdot 4\lambda^2}{(mp/2)\cdot(m-mp/2)}}\\&\geq (3\lambda/4) \cdot \frac{\sqrt{2\pi}}{e^2} \cdot \frac{2}{\sqrt{2n/3}} \cdot e^{-\frac{m\cdot 4\lambda^2}{(n/9)\cdot(m-m/4)}}\\&> 0.62 \cdot \delta \cdot e^{-48\delta^2},\end{aligned}$$

where in the first inequality above was obtained using Lemma B.4. $\square$

B.2 Main Lemmas. Recall that $\beta^2 + \beta = 1$ and $\beta \approx 0.618$. The first lemma below upper bounds x_{t+2} when $x_t < \beta$, and the second lemma upper bounds x_{t+1} when $x_t > \beta$.

LEMMA B.6. *There is a constant $c > 0$, such that for any $0 < x < \beta$,*

$$\Pr[x_{t+2} \le x - \beta x(\beta - x) \mid x_t = x] \ge 1 - 2e^{-cnx(\beta-x)^2}.$$

Proof. Let $z = \beta - x$, and $\delta = z(2\beta - 1)/4$. We bound x_{t+1} using (B.4):

$$\begin{aligned} \Pr[x_{t+1} &< 1 - x^2 - \delta x(1-x) \mid x_t = x] \\ &< e^{-\delta^2 nx(1-x)/2} < e^{-c_1 nxz^2}, \end{aligned} \tag{B.5}$$

where $c_1 = (\delta/z)^2(1-\beta)/2$. Let $y_0 = 1 - x^2 - \delta x(1-x)$. We use (B.3) to bound x_{t+2} given $x_{t+1} = y$, for $y_0 \le y < 1$. Let $\sigma = \delta \cdot \frac{1-y_0}{y(1-y)} > \delta$. We have

$$\begin{aligned} 1 - y^2 + \sigma y(1-y) &\le 1 - y_0^2 + \delta(1 - y_0) \\ &= (1-y_0)(1+y_0+\delta) \le (x^2 + \delta x)(2 - x^2 + \delta), \end{aligned}$$

where for the last inequality we used that $1 - x^2 - \delta x \le y_0 \le 1 - x^2$. The rightmost side above is

$$\begin{aligned} (x^2 + \delta x)&(2 - x^2 + \delta) \\ &\le x(2x - x^3 + 4\delta) \\ &= x(2(\beta - z) - (\beta - z)^3 + 4\delta) \\ &= x(1 - z(3\beta - 1) - z^2(3\beta - z) + 4\delta) \\ &\le x(1 - z(3\beta - 1) + 4\delta) \\ &= x(1 - z\beta), \end{aligned}$$

where the third line is obtained using $\beta^2 = 1 - \beta$. It follows

$$\begin{aligned} \Pr[x_{t+2} &> x(1 - z\beta) \mid x_{t+1} = y] \\ &\le \Pr[x_{t+2} > 1 - y^2 + \sigma y(1-y) \mid x_{t+1} = y] \\ &< e^{-\sigma^2 ny(1-y)/(2+\sigma)}, \quad \text{by (B.3)} \\ &= e^{-\sigma ny(1-y)\cdot\sigma/(2+\sigma)} \\ &\le e^{-\delta n(1-y_0)\cdot\delta/(2+\delta)} \\ &\le e^{-\delta^2 n(1-y_0)/3} \\ &\le e^{-c_2 nxz^2}, \end{aligned} \tag{B.6}$$

where $c_2 = (\delta/z)^3\beta(1-\beta)/3$, and the last inequality is obtained by substituting

$$\begin{aligned} 1 - y_0 &= x(x + \delta(1-x)) \\ &\ge x(x + \delta(1-\beta)) \\ &= x(x \cdot [(\delta/z)(1-\beta)]^{-1} + z) \cdot [(\delta/z)(1-\beta)] \\ &\ge x(x+z) \cdot [(\delta/z)(1-\beta)] \\ &= x\beta \cdot (\delta/z)(1-\beta). \end{aligned}$$

Equation (B.6) is also true when $y = 1$, as $x_{t+2} = 0$ in that case. Finally, we have

$$\begin{aligned} \Pr[x_{t+2} &> x(1 - z\beta) \mid x_t = x] \\ &\le \Pr[x_{t+1} < y_0 \mid x_t = x] \\ &\quad + \sum_{y \ge y_0} \Big(\Pr[x_{t+2} > x(1 - z\beta) \mid x_{t+1} = y] \\ &\qquad \cdot \Pr[x_{t+1} = y \mid x_t = x] \Big) \\ &\le e^{-c_1 nxz^2} + e^{-c_2 nxz^2} \\ &\qquad \cdot \Pr[x_{t+1} \ge y_0 \mid x_t = x], \text{ by (B.5),(B.6)} \\ &\le e^{-c_1 nxz^2} + e^{-c_2 nxz^2}. \end{aligned}$$

The claim then follows. ∎

LEMMA B.7. *There is a constant $c > 0$, such that for any $\beta < x < 1$,*

$$\Pr[x_{t+1} \le 2\beta - x \mid x_t = x] \ge 1 - e^{-cn(1-x)(x-\beta)^2}.$$

Proof. Let $z = x - \beta$, and $\delta = z(2\beta - 1)$. We bound x_{t+1} using (B.3). We have

$$1 - x^2 + \delta x(1-x) \le 1 - (\beta+z)^2 + \delta \le 1 - \beta^2 - 2\beta z + \delta = \beta - z,$$

where for the last equality we used that $\beta^2 = 1 - \beta$. Then,

$$\begin{aligned} \Pr[x_{t+1} &> \beta - z \mid x_t = x] \\ &\le \Pr[x_{t+1} > 1 - x^2 + \delta x(1-x) \mid x_t = x] \\ &< e^{-\delta^2 nx(1-x)/(2+\delta)}, \quad \text{by (B.3)} \\ &\le e^{-cn(1-x)z^2}, \end{aligned}$$

for $c = (\delta/z)^2\beta/3$. ∎

B.3 Proof of Theorem B.1. We partition interval $(0, 1)$ into several subintervals, and analyze how x_t evolves in each of them. We start by defining the partition. Let $c > 0$ be a (small enough) constant that satisfies the statements of both Lemmas B.6 and B.7. Let

$$\gamma = \beta^2/2, \quad \alpha = 2\ln 8/(c\beta\gamma),$$
$$w_1 = (\ln n)^2/n, \qquad w_0 = z_0 = \beta/2,$$
$$z_1 = (\alpha \ln n/n)^{1/2}, \quad z_2 = (\alpha/n)^{1/2},$$

$$\begin{aligned} i_1 &= \max\{i \colon w_0(1-\gamma)^i \ge w_1\} = \lfloor \ln(w_1/w_0)/\ln(1-\gamma) \rfloor, \\ i_2 &= \max\{i \colon z_0/(1+\gamma)^i \ge z_1\} = \lfloor \ln(z_0/z_1)/\ln(1+\gamma) \rfloor, \\ i_3 &= \max\{i \colon z_0/(1+\gamma)^i \ge z_2\} = \lfloor \ln(z_0/z_2)/\ln(1+\gamma) \rfloor, \end{aligned}$$

$$w_1' = w_0 \cdot (1-\gamma)^{i_1} \in [w_1, w_1/(1-\gamma)),$$
$$z_2' = z_0/(1+\gamma)^{i_3} \in [z_2, z_2(1+\gamma)).$$

We partition interval $(0, 1)$ into the following intervals A, B_i, D_i, G:

$$\begin{aligned} A &= (0, w_1'], \\ B_i &= \left(w_0 \cdot (1-\gamma)^i,\, w_0 \cdot (1-\gamma)^{i-1}\right],\ 1 \le i \le i_1, \\ D_i &= \left(\beta - z_0/(1+\gamma)^{i-1},\, \beta - z_0/(1+\gamma)^i\right],\ 1 \le i \le i_3, \\ G &= (\beta - z_2', 1). \end{aligned}$$

We write $\mathsf{left}(I)$ to denote the left endpoint of interval I, e.g., $\mathsf{left}(B_i) = w_0 \cdot (1-\gamma)^i$.

Let $\mathcal{T}$ be a subset of $\mathbb{N} = \{0, 1, \ldots\}$ defined recursively as follows: $0 \in \mathcal{T}$, and for each $t \in \mathbb{N}$, if $t \in \mathcal{T}$ and $0 < x_t \le \beta - z_2'$ then $t+1 \notin \mathcal{T}$, otherwise $t+1 \in \mathcal{T}$. Later, in the proof of Lemma B.8, we will show an upper bound on the number of rounds $t \in \mathcal{T}$ for which $x_t \notin \{0, 1\}$. Since, $\mathcal{T}$ contains at least every other round $t \in \mathbb{N}$, the above bound (multiplied by 2) yields an upper bound on the total number of rounds before $x_t \in \{0, 1\}$. In preparation for Lemma B.8, we prove a series of claims, for the different classes of intervals defined above.

In the next claim, a is the first $t \in \mathcal{T}$ for which $x_t \in A$, or $a = \infty$ if no such t exists.

Claim B.1. *Let $a = \min\{t \in \mathcal{T} : x_t \in A\} \cup \{\infty\}$. Then,*

$$\Pr[\{a = \infty\} \cup \{x_{a+2} = 0\}] = 1 - O(\ln^4 n/n).$$

Proof. Suppose that $a < \infty$. Then, from Lemma B.3, for any $x \in A$,

$$\begin{aligned} \Pr[x_{a+1} = 1 \mid x_a = x] \ge 1 - nx^2 &\ge 1 - n(w_1')^2 \\ &= 1 - O(\ln^4 n/n). \end{aligned}$$

The claim then follows. ∎

Claim B.2. *For $1 \le i \le i_1$, let $b_i = \min\{t \in \mathcal{T} : x_t \in B_i\} \cup \{\infty\}$. Then,*

$$\Pr[\{b_i = \infty\} \cup \{x_{b_i+2} \le \mathsf{left}(B_i)\}] = 1 - o(1/n).$$

Proof. Suppose that $b_i < \infty$, and recall that $\mathsf{left}(B_i) = w_0 \cdot (1-\gamma)^i$. Then, for any $x \in B_i$,

$$\begin{aligned} &\Pr[x_{b_i+2} \le w_0 \cdot (1-\gamma)^i \mid x_{b_i} = x] \\ &\qquad \ge \Pr[x_{b_i+2} \le x \cdot (1-\gamma) \mid x_{b_i} = x] \\ &\qquad \ge \Pr[x_{b_i+2} \le x \cdot (1 - \beta(\beta - x)) \mid x_{b_i} = x] \\ &\qquad \ge 1 - 2e^{-cnx(\beta-x)^2}, \quad \text{by Lemma B.6} \\ &\qquad \ge 1 - 2e^{-cnx_1(\beta-w_0)^2} \\ &\qquad = 1 - o(1/n). \end{aligned}$$

The claim then follows. ∎

Claim B.3. *For $1 \le i \le i_2$, let $d_i = \min\{t \in \mathcal{T} : x_t \in D_i\} \cup \{\infty\}$. Then,*

$$\Pr[\{d_i = \infty\} \cup \{x_{d_i+2} \le \mathsf{left}(D_i)\}] = 1 - o(1/n).$$

Proof. Suppose that $d_i < \infty$, and recall that $\mathsf{left}(D_i) = \beta - z_0/(1+\gamma)^{i-1}$. Then, for any $x \in D_i$,

$$\begin{aligned} &\Pr[x_{d_i+2} \le \beta - z_0/(1+\gamma)^{i-1} \mid x_{d_i} = x] \\ &\qquad \ge \Pr[x_{d_i+2} \le \beta - (\beta - x) \cdot (1+\gamma) \mid x_{d_i} = x] \\ &\qquad \ge \Pr[x_{d_i+2} \le x - \beta x(\beta - x) \mid x_{d_i} = x] \\ &\qquad \ge 1 - 2e^{-cnx(\beta-x)^2}, \quad \text{by Lemma B.6} \\ &\qquad \ge 1 - 2e^{-cn(\beta-z_0)(z_1)^2} \\ &\qquad = 1 - o(1/n), \end{aligned}$$

where the second inequality holds because

$$[\beta - (\beta - x)\cdot(1+\gamma)] - [x - \beta x(\beta - x)] = (\beta - x)(\beta x - \gamma) \ge 0.$$

The claim then follows. ∎

In the next claim, $d_{i,j}$ is the jth smallest $t \in \mathcal{T}$ for which $x_t \in D_i$, or ∞ if no such t exists.

Claim B.4. *For $i_2 < i \le i_3$ and $j \ge 1$, let $d_{i,j} = \min\{t \in \mathcal{T} : t > d_{i,j-1},\, x_t \in D_i\} \cup \{\infty\}$, where $d_{i,0} = -1$. Let also*

$$f_i = |\{1 \le j \le \log n : d_{i,j} < \infty,\, x_{d_{i,j}+2} > \mathsf{left}(D_i)\}|.$$

Then, for $s_i = \lceil \gamma \log n/(1+\gamma)^{2(i_3-i)} \rceil$,

$$\Pr[f_i < s_i] = 1 - O(1/n).$$

Proof. Suppose that $d_{i,j} < \infty$, and recall that $\mathsf{left}(D_i) = \beta - z_0/(1+\gamma)^{i-1}$. As in the proof of Claim B.3, for any $x \in D_i$,

$$\begin{aligned} \Pr[x_{d_{i,j}+2} \le \beta - z_0/(1+\gamma)^{i-1} &\mid x_{d_{i,j}} = x] \\ &\ge 1 - 2e^{-cnx(\beta-x)^2}. \end{aligned}$$

Since $x > \beta/2$ and $\beta - x \ge z_0/(1+\gamma)^i = z_2'(1+\gamma)^{i_3-i} \ge z_2(1+\gamma)^{i_3-i}$,

$$\begin{aligned} 2e^{-cnx(\beta-x)^2} &\le 2e^{-cn(\beta/2)(1+\gamma)^{2(i_3-i)}\alpha/n} \\ &= 2e^{-(1+\gamma)^{2(i_3-i)}\ln 8/\gamma} \le 4^{-(1+\gamma)^{2(i_3-i)}/\gamma}. \end{aligned}$$

Therefore,

$$\Pr[x_{d_{i,j}+2} \le \mathsf{left}(D_i) \mid x_{d_{i,j}} = x] \ge 1 - 4^{-(1+\gamma)^{2(i_3-i)}/\gamma}.$$

For $1 \le j \le \log n$, let $Y_j = 1$, if $d_{i,j} < \infty$ and $x_{d_{i,j}+2} > \mathsf{left}(D_i)$; $Y_j = 0$ otherwise. From the above,

$$\begin{aligned} &\Pr[Y_j = 1 \mid Y_1, \ldots, Y_{j-1}] \\ &\quad \le \Pr[x_{d_{i,j}+2} > \mathsf{left}(D_i) \mid Y_1, \ldots, Y_{j-1}; d_{i,j} < \infty] \\ &\quad \le 4^{-(1+\gamma)^{2(i_3-i)}/\gamma}. \end{aligned}$$

It follows that $Y = \sum_{1\le j\le \log n} Y_j$ is dominated by the binomial $B\big(\log n, 4^{-(1+\gamma)^{2(i_3-i)}/\gamma}\big)$. Thus

$$\Pr[Y \ge s_i] \le \binom{\log n}{s_i} \cdot \Big(4^{-(1+\gamma)^{2(i_3-i)}/\gamma}\Big)^{s_i} \le 2^{\log n} \cdot 4^{-\log n} = 1/n.$$

The claim then follows, since $f_i = Y$. ☐

Claim B.5. *For $j \ge 1$, let $g_j = \min\{t \in \mathcal{T} \colon t > g_{j-1}, x_t \in G\} \cup \{\infty\}$, where $g_0 = -1$. Let also*

$$h_\kappa = |\{1 \le j \le \kappa \log n \colon g_j < \infty,\, x_{g_j+1} > \mathsf{left}(G)\}|.$$

Then, there is a constant κ such that

$$\Pr[h_\kappa \le (\kappa-1)\log n] = 1 - O(1/n).$$

Proof. Suppose that $g_j < \infty$, and recall $\mathsf{left}(G) = \beta - z_2'$. We consider two cases, $x_{g_j} \ge \beta + z_2'$ and $x_{g_j} < \beta + z_2'$, and use Lemma B.7 and Lemma B.5, respectively. For any $\beta + z_2' \le x < 1$,

$$\begin{aligned}\Pr[x_{g_j+1} &\le \beta - z_2' \mid x_{g_j} = x]\\ &\ge \Pr[x_{g_j+1} \le 2\beta - x \mid x_{g_j} = x]\\ &\ge 1 - e^{-cn(1-x)(x-\beta)^2}, \quad \text{by Lemma B.7}\\ &\ge c_1,\end{aligned}$$

for some constant $c_1 > 0$, because in the range $\beta + z_2' \le x < 1$, the value of $f(x) = cn(1-x)(x-\beta)^2$ is minimized at one of the two extreme points, $x = \beta + z_2'$ or $x = 1 - 1/n$. For these points, $f(\beta+z_2') = cn(1-\beta+z_2')(z_2')^2 \ge c(1-\beta+z_2')\alpha$, and $f(1-1/n) = c(1-1/n-\beta)^2$, thus at both points, $f(x)$ is bounded away from 0.

Next, for any $\beta - z_2' < x < \beta + z_2'$, and for $\delta = (2\beta-1)(1+\gamma)\alpha^{1/2} \ge (2\beta-1)z_2' n^{1/2}$,

$$\begin{aligned}1 - x^2 - \delta n^{-1/2} &\le 1 - (\beta - z_2')^2 - \delta n^{-1/2}\\ &\le 1 - \beta^2 + 2\beta z_2' - \delta n^{-1/2} \le \beta - z_2'.\end{aligned}$$

Then

$$\begin{aligned}\Pr[x_{g_j+1} &\le \beta - z_2' \mid x_{g_j} = x]\\ &\ge \Pr[x_{g_j+1} \le 1 - x^2 - \delta n^{-1/2} \mid x_{g_j} = x]\\ &\ge 0.62 \cdot \delta e^{-48\delta^2}, \quad \text{by Lemma B.5}\\ &= c_2,\end{aligned}$$

where $c_2 > 0$ is a constant. Combining the two cases above we obtain that, for any $x \in G$,

$$\Pr[x_{g_j+1} \le \mathsf{left}(G) \mid x_{g_j} = x] \ge c_3 = \min\{c_1, c_2\}.$$

Let $\kappa = 4/c_3$. For $1 \le j \le \kappa$, let $Y_j = 1$, if $g_j = \infty$ or $x_{g_j+1} \le \beta - z_2'$; $Y_j = 0$, otherwise. Then

$$\begin{aligned}&\Pr[Y_j = 1 \mid Y_1, \dots, Y_{j-1}]\\ &\ge \Pr[x_{g_j+1} \le \mathsf{left}(G) \mid Y_1, \dots, Y_{j-1}; g_j < \infty] \ge c_3.\end{aligned}$$

It follows that $Y = \sum_{1\le j\le \kappa \log n} Y_j$ dominates the binomial distribution $B(\kappa \log n, c_3)$. Thus

$$\Pr[Y \ge \log n] \ge \Pr[B(\kappa \log n, 4/\kappa) \ge \log n] \ge 1 - O(1/n),$$

by a standard Chernoff bound. The claim then follows, as $h_k = \kappa \log n - Y$. ☐

Combining the previous claims we show the following lemma, which bounds the convergence time, i.e., the number of rounds before $x_t \in \{0,1\}$.

Lemma B.8. *There is a constant $\hat{c} > 0$, such that for any $x \in (0,1)$ and $k \ge \hat{c} \ln n$,*

$$\Pr[x_{t+k} \in \{0,1\} \mid x_t = x] = 1 - O(\ln^4 n/n).$$

Proof. W.l.o.g., we assume that $t = 0$, and $x_0 = x$. The following event $\mathcal{E}$ is the intersection of all events considered in Claims B.1 to B.5,

$$\begin{aligned}\mathcal{E} = &\big(\{a = \infty\} \cup \{x_{a+2} = 0\}\big)\\ &\cap \bigcap_{1\le i\le i_1} \big(\{b_i = \infty\} \cup \{x_{b_i+2} \le \mathsf{left}(B_i)\}\big)\\ &\cap \bigcap_{1\le i\le i_2} \big(\{d_i = \infty\} \cup \{x_{d_i+2} \le \mathsf{left}(D_i)\}\big)\\ &\cap \bigcap_{i_2< i\le i_3} \{f_i < s_i\}\\ &\cap\ \{h_\kappa \le (\kappa-1)\log n\}.\end{aligned}$$

From the claims above and a union bound, we get

$$\Pr[\mathcal{E}] = 1 - O(\ln^4 n/n).$$

To complete the proof it suffices to show that: $\mathcal{E}$ implies $x_t \in \{0,1\}$ for all $t \ge \hat{c} \ln n$, for some $\hat{c}$.

Let $\mathcal{I}$ be the set of the intervals in which we partitioned $(0,1)$ at the beginning of the analysis, i.e., $\mathcal{I} = \{A\} \cup \{B_i \colon 1 \le i \le i_1\} \cup \{D_i \colon 1 \le i \le i_3\} \cup \{G\}$. For each $t \in \mathcal{T}$, either $x_t \in \{0,1\}$, or $x_t \in I$ for some $I \in \mathcal{I}$; and $t+1 \notin \mathcal{T}$ if and only if $t \in I$ for some $I \in \mathcal{I} \setminus \{G\}$. We use the following terminology. If $x_t \in I$ and $I \in \mathcal{I} \setminus \{G\}$, we say t is a *success* at I if $x_{t+2} \le \mathsf{left}(I)$, and a *failure* if $x_{t+2} > \mathsf{left}(I)$. Similarly, if $x_t \in G$, t is a *success* at G if $x_{t+1} \le \mathsf{left}(G)$, and a *failure* if $x_{t+1} > \mathsf{left}(G)$. For each $I \in \mathcal{I}$, $S(I)$ and $F(I)$ denote the total number of successes and failures at I, respectively, and $F^-(I) = \sum_{I'<I} F(I')$, where $I' < I$ denotes that $x' < x$ for any $x' \in I'$, $x \in I$.

CLAIM B.6. *For any $I \in \mathcal{I}$,*

(a) $S(I) \le F^-(I) + 1$.

(b) If $S(I) = F^-(I) + 1$ *and* $F(I) > 0$, *then the last failure at* I *precedes the last success at* I.

Proof. The claim follows from the next two observations. First, if $x_t \in \{0,1\}$ then $x_{t'} \in \{0,1\}$ for all $t' > t$. Second, if t is a success at interval I', then $x_{t'} \le \mathsf{left}(I')$ for the next point $t' \in \mathcal{T}$, i.e., for $t' = t+1$ if $I' = G$, or $t' = t+2$ otherwise. Therefore, if t_1 is a success at I, and t_2 is the next success or failure at I, then there exists a failure $t_3 \in (t_1, t_2)$ at some $I' < I$. ◻

CLAIM B.7. *Suppose that event $\mathcal{E}$ holds.*

(a) For any $I \in \{A\} \cup \{B_i; 1 \le i \le i_1\} \cup \{D_i; 1 \le i \le i_2\}$, $F(I) = 0$.

(b) For $i_2 < i \le i_3$, $F(D_i) \le s_i - 1$, *where s_i is defined in Claim B.4.*

(c) $F(G) \le (\kappa - 1)\log n$, *where κ is the constant from Claim B.5.*

Proof. We prove (a) by contradiction. Suppose (a) does not hold, and let I be the leftmost interval (i.e., the one closest to 0) for which $F(I) > 0$. Then, $F^-(I) = 0$. Let t be the first failure at I, and let $t^* \le t$ be the first success or failure at I. From $\mathcal{E}$, t^* is a success, thus $t > t^*$. From Claim B.6(a), $S(I) \le F^-(I) + 1 = 1$, thus $S(I) = 1$ since there is at least one success at I, namely t^*. Then, from Claim B.6(b), the last failure at I must precede t^*, which contradicts $t > t^*$.

The proof of (b) is similar. Suppose, for contradiction, that (b) does not hold, and let $i > i_2$ be the smaller index for which $F(D_i) \ge s_i$. Then,

$$
\begin{aligned}
F^-(D_i) &\le \sum_{i_2 < i' < i} (s_{i'} - 1) \\
&\le \sum_{i_2 < i' < i} \gamma \log n/(1+\gamma)^{2(i_3 - i')} \\
&\le \gamma \log n \cdot \sum_{j \ge 1} 1/(1+\gamma)^{2j} \\
&= \gamma \log n \cdot \frac{1}{(1+\gamma)^2 - 1} < \log n/2.
\end{aligned}
$$

From Claim B.6(a),

$$
\begin{aligned}
S(D_i) &\le F^-(D_i) + 1 < \log n/2 + 1 \\
&< \log n - \lceil \gamma \log i \rceil \le \log n - s_i.
\end{aligned}
$$

But, from $\mathcal{E}$, we have that $S(D_i) < \log n - s_i$ implies $F(D_i) < s_i$, which is a contradiction.

Finally, for (c) we have from Claim B.6(a),

$$
\begin{aligned}
S(G) &\le F^-(G) + 1 = F^-(D_{i_3}) + F(D_{i_3}) + 1 \\
&\le \log n/2 + s_{i_3} + 1 < \log n,
\end{aligned}
$$

and from $\mathcal{E}$, $S(G) < \log n$ implies that $F(G) \le (1-\kappa) \cdot \log n$. ◻

Let $T = \min\{t \colon x_t \in \{0,1\}\}$.

CLAIM B.8. *If event $\mathcal{E}$ holds, then $T = O(\log n)$.*

Proof. Since $\mathcal{T}$ contains at least every other $t \in \mathbb{N}$,

$$
\begin{aligned}
T &\le 2 \cdot \sum_{I \in \mathcal{I}} (S(I) + F(I)) \le 2 \cdot \sum_{I \in \mathcal{I}} (F^-(I) + F(I) + 1) \\
&= 2 \cdot \sum_{I \in \mathcal{I}} F(I) \cdot (|\{I' \colon I < I'\}| + 1) + 2 \cdot |\mathcal{I}|,
\end{aligned}
$$

where the second inequality was obtained using Claim B.6. From Claim B.7,

$$
\begin{aligned}
&\sum_{I \in \mathcal{I}} F(I) \cdot (|\{I' \colon I < I'\}| + 1) \\
&\quad = \sum_{i_2 < i \le i_3} F(D_i) \cdot (i_3 - i + 1) + F(G) \\
&\quad \le \sum_{i_2 < i \le i_3} (s_i - 1) \cdot (i_3 - i + 1) + (\kappa - 1)\log n.
\end{aligned}
$$

Also

$$
\begin{aligned}
&\sum_{i_2 < i \le i_3} (s_i - 1) \cdot (i_3 - i + 1) \\
&\quad \le \sum_{i_2 < i \le i_3} \gamma \log n/(1+\gamma)^{2(i_3 - i)} \cdot (i_3 - i + 1) \\
&\quad = \gamma \log n \cdot \sum_{0 \le j < i_3 - i_2} (j+1)/(1+\gamma)^{2j} \\
&\quad = O(\log n).
\end{aligned}
$$

Last, we have $|\mathcal{I}| = 2 + i_1 + i_3 = O(\log n)$. Combining all the above yields $T = O(\log n)$. ◻

Since $x_t \in \{0,1\}$ for all $t \ge T$, it follows from Claim B.8 that if $\mathcal{E}$ occurs, then $x_t \in \{0,1\}$ for all $t \ge \hat{c} \ln n$, for some constant $\hat{c}$. This completes the proof of Lemma B.8. ◻

Finally, to complete the proof of Theorem B.1, we just apply Lemma B.8 repeatedly, for a constant number of times, to obtain that, with the desired high probability, $x_t \in \{0,1\}$ for all $t \ge c' \log n$, for a large enough constant c'.

Acknowledgements. We are deeply indebted to Emanuele Natale for introducing the problem to us, for helping us to devise the binary clock protocol, for pointing out related work, and for helpful discussions through the course of this project.

References

[1] Y. Afek, N. Alon, Z. Bar-Joseph, A. Cornejo, B. Haeupler, and F. Kuhn. Beeping a maximal independent set. *Distributed Comput.*, 26(4):195–208, 2013.

[2] D. Aldous and J. A. Fill. Reversible markov chains and random walks on graphs, 2002. Unfinished monograph, http://www.stat.berkeley.edu/users/aldous/RWG/book.html.

[3] D. Alistarh, J. Aspnes, and R. Gelashvili. Space-optimal majority in population protocols. In *ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 2221–2239, 2018.

[4] T. Amir, J. Aspnes, D. Doty, M. Eftekhari, and E. E. Severson. Message complexity of population protocols. In *International Symposium on Distributed Computing, DISC*, pages 6:1–6:18, 2020.

[5] D. Angluin, J. Aspnes, Z. Diamadi, M. J. Fischer, and R. Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Comput.*, 18(4):235–253, 2006.

[6] D. Angluin, J. Aspnes, and D. Eisenstat. A simple population protocol for fast robust approximate majority. *Distributed Comput.*, 21(2):87–102, 2008.

[7] D. Angluin, J. Aspnes, D. Eisenstat, and E. Ruppert. The computational power of population protocols. *Distributed Comput.*, 20(4):279–304, 2007.

[8] D. Angluin, J. Aspnes, M. J. Fischer, and H. Jiang. Self-stabilizing population protocols. *ACM Trans. Auton. Adapt. Syst.*, 3(4):13:1–13:28, 2008.

[9] A. Arora, S. Dolev, and M. G. Gouda. Maintaining digital clocks in step. *Parallel Process. Lett.*, 1:11–18, 1991.

[10] P. Bastide, G. Giakkoupis, and H. Saribekyan. Self-stabilizing clock synchronization with 1-bit messages. Research report, Inria, Nov. 2020. https://hal.inria.fr/hal-02987598.

[11] L. Becchetti, A. E. F. Clementi, E. Natale, F. Pasquale, and R. Silvestri. Plurality consensus in the gossip model. In *ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 371–390, 2015.

[12] L. Becchetti, A. E. F. Clementi, E. Natale, F. Pasquale, and L. Trevisan. Stabilizing consensus with many opinions. In *ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 620–635, 2016.

[13] M. Ben-Or, D. Dolev, and E. N. Hoch. Fast self-stabilizing byzantine tolerant digital clock synchronization. In *ACM Symposium on Principles of Distributed Computing, PODC*, pages 385–394, 2008.

[14] P. Berenbrink, A. E. F. Clementi, R. Elsässer, P. Kling, F. Mallmann-Trenn, and E. Natale. Ignore or comply?: On breaking symmetry in consensus. In *ACM Symposium on Principles of Distributed Computing, PODC*, pages 335–344, 2017.

[15] L. Boczkowski, A. Korman, and E. Natale. Minimizing message size in stochastic communication patterns: Fast self-stabilizing protocols with 3 bits. In *ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 2540–2559, 2017.

[16] L. Boczkowski, A. Korman, and E. Natale. Minimizing message size in stochastic communication patterns: Fast self-stabilizing protocols with 3 bits. *Distributed Comput.*, 32(3):173–191, 2019.

[17] C. Boulinier, F. Petit, and V. Villain. Synchronous vs. asynchronous unison. *Algorithmica*, 51(1):61–80, 2008.

[18] B. Doerr, L. A. Goldberg, L. Minder, T. Sauerwald, and C. Scheideler. Stabilizing consensus with the power of two choices. In *ACM Symposium on Parallelism in Algorithms and Architectures, SPAA*, pages 149–158, 2011.

[19] D. Dolev, K. Heljanko, M. Järvisalo, J. H. Korhonen, C. Lenzen, J. Rybicki, J. Suomela, and S. Wieringa. Synchronous counting and computational algorithm design. *J. Comput. Syst. Sci.*, 82(2):310–332, 2016.

[20] D. Dolev and E. N. Hoch. On self-stabilizing synchronous actions despite byzantine attacks. In *International Symposium on Distributed Computing, DISC*, pages 193–207, 2007.

[21] D. Dolev and R. Reischuk. Bounds on information exchange for byzantine agreement. *J. ACM*, 32(1):191–204, 1985.

[22] S. Dolev and J. L. Welch. Self-stabilizing clock synchronization in the presence of byzantine faults. *J. ACM*, 51(5):780–799, 2004.

[23] B. Dudek and A. Kosowski. Universal protocols for information dissemination using emergent signals. In *ACM SIGACT Symposium on Theory of Computing, STOC*, pages 87–99, 2018.

[24] Y. Emek and R. Wattenhofer. Stone age distributed computing. In *ACM Symposium on Principles of Distributed Computing, PODC*, pages 137–146, 2013.

[25] O. Feinerman, B. Haeupler, and A. Korman. Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication. *Distributed Comput.*, 30(5):339–355, 2017.

[26] M. Feldmann, A. Khazraei, and C. Scheideler. Time- and space-optimal discrete clock synchronization in the beeping model. In *ACM Symposium on Parallelism in Algorithms and Architectures, SPAA*, pages 223–233, 2020.

[27] L. Gasieniec and G. Stachowiak. Fast space optimal leader election in population protocols. In *ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 2653–2667, 2018.

[28] M. Ghaffari and J. Lengler. Nearly-tight analysis for 2-choice and 3-majority consensus dynamics. In *ACM Symposium on Principles of Distributed Computing, PODC*, pages 305–313, 2018.

[29] M. G. Gouda and T. Herman. Stabilizing unison. *Inf.*

Process. Lett., 35(4):171–175, 1990.

[30] R. M. Karp, C. Schindelhauer, S. Shenker, and B. Vöcking. Randomized rumor spreading. In *Symposium on Foundations of Computer Science, FOCS*, pages 565–574, 2000.

[31] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Symposium on Foundations of Computer Science, FOCS*, pages 482–491, 2003.

[32] A. Kosowski and P. Uznanski. Population protocols made easy. *CoRR*, abs/1802.06872, 2018.

[33] C. Lenzen and J. Rybicki. Near-optimal self-stabilising counting and firing squads. *Distributed Comput.*, 32(4):339–360, 2019.

[34] C. Lenzen, J. Rybicki, and J. Suomela. Efficient counting with optimal resilience. *SIAM J. Comput.*, 46(4):1473–1500, 2017.

[35] T. M. Liggett. *Interacting Particle Systems*. Springer Berlin Heidelberg, 1985.

[36] R. E. Mirollo and S. H. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM J. Appl. Math.*, 50(6):1645–1662, 1990.