Communication cost of consensus for nodes with limited memory

Giulia Fanti[a], Nina Holden[b], Yuval Peres[c,1], and Gireeja Ranade[d]

[a]Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213; [b]Institute for Theoretical Studies, Swiss Federal Institute of Technology (ETH) Zürich, 8092 Zürich, Switzerland; [c]Microsoft Research, Redmond, WA 98052; and [d]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720

Contributed by Yuval Peres, January 27, 2020 (sent for review July 29, 2019; reviewed by Dan Alistarh and Milan Vojnovic)

Motivated by applications in wireless networks and the Internet of Things, we consider a model of *n* nodes trying to reach consensus with high probability on their majority bit. Each node *i* is assigned a bit at time 0 and is a finite automaton with *m* bits of memory (i.e., 2^m states) and a Poisson clock. When the clock of *i* rings, *i* can choose to communicate and is then matched to a uniformly chosen node *j*. The nodes *j* and *i* may update their states based on the state of the other node. Previous work has focused on minimizing the time to consensus and the probability of error, while our goal is minimizing the number of communications. We show that, when $m > 3 \log\log\log(n)$, consensus can be reached with linear communication cost, but this is impossible if $m < \log\log\log(n)$. A key step is to distinguish when nodes can become aware of knowing the majority bit and stop communicating. We show that this is impossible if their memory is too low.

distributed computing | consensus | communication | memory

Consensus algorithms, which enable distributed parties to make decisions in the absence of a central leader, are widely used in distributed systems that require coordination, including blockchains (1, 2) and sensor networks (3, 4). Such systems are often resource constrained, with participating nodes having limited bandwidth, power, or storage. As such, it is important for consensus algorithms to minimize resource costs, including communication, computation, convergence time, and/or storage. This observation leads to a natural question: what is the minimum resource cost for any consensus protocol? Decades of distributed systems literature have studied how to optimize resource costs like completion time, typically under the assumption that nodes always communicate whenever they are allowed to. However, such an assumption is not representative of, say, wireless networks of battery-powered devices (e.g., the Internet of Things), where remaining silent can preserve battery life. Indeed, many wireless devices go to great lengths to minimize the amount of time spent actively communicating. Low-power wireless devices are also more likely to have limited storage than traditional computers.

In this work, we consider a communication model that is motivated by a wireless network of resource-constrained devices, although similar models have been used to describe physical processes, such as biological computation models (5). We make three primary modeling assumptions: 1) nodes are storage constrained, 2) nodes refrain from communicating whenever possible, and 3) the dominant cost of communication is setting up the connection. Assumption 2 models sleepy end devices that are normally disabled but occasionally wake to poll other nodes; such devices are commonly represented in low-power Internet of Things device networking protocols (6). Assumption 3 is motivated by the observation that, when two mobile devices exchange a message of <1 kB in a line-of-sight setting, the initial Transport Layer Security handshake comprises over 85% of the power overhead (7). As such, our model penalizes the establishment of a communication channel but not the number of bits sent over that channel. Furthermore, although we do not explicitly charge the number of bits sent in our protocol, our protocols transmit well under 1 kB for reasonable network sizes, and therefore, we are operating in a regime where establishing a connection is the energy bottleneck. Our goal is to design probabilistic majority consensus protocols that obey a per-node memory constraint while minimizing the total communication across all nodes.

Model

We summarize our model, which is fully specified in *The Model*. Consider a set of n nodes in a complete graph topology, each of which can be in one of s possible states.* At the beginning of the protocol, each node i is assigned a bit $\mathfrak{b}_i \in \{0, 1\}$, which is stored in its memory. Let $\mathfrak{b}$ be the majority bit, and let $p \in (1/2, 1)$ be the fraction of the nodes for which $\mathfrak{b}_i = \mathfrak{b}$. We assume $p \in [\frac{1}{2} + \epsilon, 1 - \epsilon]$, where $\epsilon \in (0, \frac{1}{4})$ is known to the protocol. We call $p - \frac{1}{2}$ the initial advantage.

Nodes communicate in a pairwise manner, similar to the standard pairwise random interaction model in most population protocols. We consider an asynchronous model where each node i has an independent, unit rate Poisson clock. When i's clock rings, i may choose to either do nothing (which costs zero) or initiate a communication (which costs one). If i chooses to communicate, it will be connected with another node j chosen uniformly at random, and the two nodes update their states based on the state of the other node. The notion of a node choosing whether or not it wants to initiate a communication is an important feature of our model. Our goal is to minimize

Significance

Algorithms that allow a large number, *n*, of processors to reach consensus are of substantial current interest due to applications in sensor networks and blockchains. When each processor is assigned an initial bit, the consensus bit should match the majority of these bits with high probability. We present a consensus algorithm where the total number of communications between all processors grows linearly in *n*, yet each processor uses surprisingly few bits of memory; we also prove a lower bound that shows that this memory requirement is sharp up to a factor of three. Our result contrasts with previous algorithms where the consensus matches the majority with probability one at the cost of a superlinear number of communications.

Author contributions: G.F., N.H., Y.P., and G.R. performed research and wrote the paper.

Reviewers: D.A., Institute of Science and Technology Austria (IST Austria); and M.V., London School of Economics and Political Science.

The authors declare no competing interest.

Data deposition: Code for simulations related to this paper are available in GitHub (https://github.com/gfanti/communication-cost-consensus).

[1]To whom correspondence may be addressed. Email: yperes@gmail.com.

First published March 4, 2020.

*Note that a node needs $\lceil \log_2 s \rceil$ bits of memory to store its state.

the total communication cost (sum over all of the nodes) of the protocol. Note that we do not use the word "asynchronous" in the sense of unbounded communication delays but simply to describe the continuous time communication model. We also assume that the protocol can depend on n; however, nodes need not have enough memory to store n directly.

At any time $t \geq 0$, each node i has an estimate for $\mathfrak{b}$, which we call the belief bit of i. We have reached consensus (sometimes called convergence [to majority]) when all nodes have belief bit equal to $\mathfrak{b}$. We say that a node is in a terminal state if nodes in this state will never change state and never initiate further communications. We say that we have reached terminal consensus if all nodes are in a terminal state and have belief bit equal to $\mathfrak{b}$. The goal is to reach consensus or terminal consensus with high probability (w.h.p.), meaning with probability $1 - o(1)$, while minimizing communication. Notice that this class of protocols can assign nonzero probability to events where the nodes agree on the wrong bit or even where the nodes never come to agreement in the first place. We note that previous work (8, 9) focused on providing tight characterizations of the hitting probabilities of the absorption states. Our notion of terminal consensus is related to the notion of stabilization commonly referred to in the literature; however, stabilization has no requirement on communication initiations.

We say that a state is stable if a node in this state will never change its belief bit. Notice that, when we reach terminal consensus, all nodes are in stable states, while this is not necessarily the case when we reach consensus.

Main Results

It is immediate that any protocol, regardless of the memory constraint s, must incur a communication cost of $\Omega(n)$. Our main results provide upper and lower bounds for the threshold on s above which $\Theta(n)$ communications are sufficient. Earlier literature has studied consensus protocols for the asynchronous model with $\Theta(n \log n)$ communications and $O(1)$ (e.g., $s = 3$) states of memory (9–11). Our results, summarized in Fig. 1, show that these earlier-studied protocols are optimal (up to multiplication by a constant) for the case where $s = O(1)$.

Theorem 1 (Upper Bound). *For any fixed $\epsilon \in (0, 1/4]$, there exists a constant $C_\epsilon > 0$ and an asynchronous consensus protocol such that w.h.p., terminal consensus is achieved with $C_\epsilon n$ communications using $s = \lceil C_\epsilon (\log\log n)^3 \rceil$ states of memory per node if p is in $[1/2 + \epsilon, 1 - \epsilon]$.*

This upper bound is proved by describing and analyzing an explicit consensus protocol described in *Proof Outlines*. Although it is not our goal to minimize running time, we remark that the protocol terminates in time $\widetilde{O}(\log n)$ w.h.p. We also first present a simpler protocol for the asynchronous model, which illustrates the key structure for the protocol that achieves our upper bound.

Proposition (Simpler Upper Bound). *For any fixed $\epsilon \in (0, 1/4]$, there exists a positive constant $C_\epsilon = \Theta(\epsilon^{-2})$ and an asynchronous consensus protocol such that w.h.p., terminal consensus is achieved with $C_\epsilon n$ communications using $s = \lceil (C_\epsilon \log n)^2 \rceil$ states of memory per node if p is in $[1/2 + \epsilon, 1 - \epsilon]$.*

The following theorem provides a lower bound on the communication cost for nodes with a given memory constraint s. In particular, the theorem implies that consensus among nodes with $o(\log\log n)$ states of memory cannot be achieved with $\Theta(n)$ communication cost.

Theorem 2 (Lower Bound). *For any $\epsilon \in (0, 1/4)$, consider an arbitrary asynchronous consensus protocol, which achieves consensus on the correct bit with probability $> 1/2$ for any $n \in \mathbb{N}$, $\mathfrak{b} \in \{0, 1\}$, and $p \in [1/2 + \epsilon, 1 - \epsilon]$. There is a constant $c > 0$ depending only on ϵ such that w.h.p. and for $s < \log\log n - c^{-1}$, the protocol incurs communication cost at least $cns^{-1} \log n$. Furthermore, for $s < \log\log n - c^{-1}$, it holds w.h.p. that no node is ever in a stable state.*

We end this section with a brief heuristic argument explaining the appearance of a memory, which is polynomial in $\log\log n$ in Theorems 1 and 2. We emphasize that the heuristic is very rough and that the reader should consult *Proof Outlines* for details. A synchronous variant of the model where the nodes interact at integer times (rather than the times of a Poisson clock) is analyzed in the full version of this paper (12). In the synchronous model, we get matching (up to a multiplicative constant) upper and lower bounds for the threshold of linear communication cost, which is s of order $\log\log n$. The protocol for the synchronous model is divided into rounds, and in each round, a certain set of $\widetilde{O}(n)$ nodes is called experts. The experts in round k get the belief bit of three randomly chosen experts in round $k - 1$ and update their belief bit to the majority bit of three. The fraction of experts for which the belief bit is different from the majority bit $\mathfrak{b}$ is approximately squared in each round, and therefore, we need $\Theta(\log\log n)$ rounds in order to have no wrong experts since $2^{-2^k} = 1/n$ for $k = \log\log n$. We use a similar construction in the case of the asynchronous model, but the memory need is multiplied by a factor of order $(\log\log n)^2$ due to difficulties arising from the fact that there is no global clock and that the randomness of the Poisson clocks makes it harder for the nodes to keep track of time and know when the majority bit $\mathfrak{b}$ has been found w.h.p. The fact that we get a memory threshold of order $\log\log n$ in our lower bound is due to a somewhat related calculation. We divide the protocol into rounds consisting of one unit of time and argue that, if the frequency of certain states is $> x$ in round $k - 1$, then the frequency is $> \widetilde{c}x^2$ in round k w.h.p. for some constant $\widetilde{c} > 0$, and we deduce from this that after $\log\log n - c^{-1}$ rounds certain states will be present in the system no matter what the majority bit is. From the last statement, we get a memory requirement of at least $\log\log n - c^{-1}$.

Related Work

Prior work on majority consensus can be categorized by network model, consensus problem formulation, and cost metrics. In this work, we do not consider related problems, like leader election and plurality consensus (13–15). There are two dominant communication/timing models in the literature: synchronous (discrete time) and asynchronous (continuous time). Synchronous models may allow nodes to communicate with multiple nodes per time step, whereas asynchronous models assume gossip communication where each node can contact at most one other node per communication event. Asynchronous models are generally more challenging to analyze because it is harder for nodes to coordinate their actions; our work assumes an asynchronous communication model. Cost metrics of interest typically include the probability of consensus, the communication cost, and the time to consensus, while constraints on communication and storage capacity are common. We summarize relevant results in Table 1; we use wall clock time to refer to the global convergence

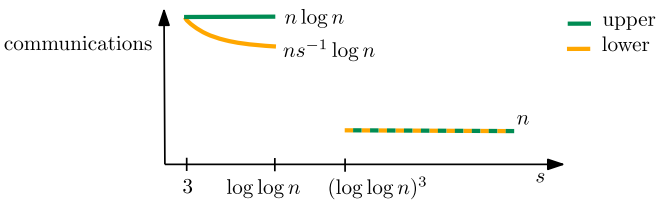

Fig. 1. The figure gives an overview of our upper and lower bounds for the number of communications in the asynchronous model given s states of memory per node.

Table 1. Comparison of related work on majority consensus

Result type and memory (s states)	Communication (message) complexity	Time complexity (wall clock)	Source
Exact upper			
4	$O(n \log n/\epsilon_\star)$	$O(\log n/\epsilon_\star)$	16, 17
$O(n)$	$O(n \log n(\frac{1}{s\epsilon_\star} + \log s))$	$O(\log n(\frac{1}{s\epsilon_\star} + \log s))$	18
$O(\log^2 n)$	$O(n \log^3 n)$	$O(\log^3 n)$	19
$O(\log^2 n)$	$O(n \log^2 n)$	$O(\log^2 n)$	20
$O(\log n)$	$O(n \log^2 n)$	$O(\log^2 n)$	21
$O(\log n)$	$O(n \log^{5/3} n)$	$O(\log^{5/3} n)$	13
Exact lower			
≤ 4	$\Omega(n/\epsilon_\star)$	$\Omega(1/\epsilon_\star)$	18
Any s	$\Omega(n \log n)$	$\Omega(\log n)$	18
$O(\log \log n)$	$\Omega(\frac{n^2}{(K^2+\epsilon_\star n)^2})$	$\Omega(\frac{n}{(K^2+\epsilon_\star n)^2})$	19
$\Omega(\log n)$	$O(n^{2-c}),\ c>0$	$O(n^{1-c}),\ c>0$	21
Approximate upper			
$O(1)$	$O(n \log n)$	$O(\log n)$	9–11
$O(1)$	$O(n \log^3 n)$	$O(\log^3 n)$	22
$O((\log \log n)^3)$	$O(n)$	$\widetilde{O}(\log n)$	This paper
Approximate lower			
$O(\log \log n)$	$\Omega\left(\frac{n \log n}{s}\right)$	—	This paper

We study approximate majority consensus (upper and lower bounds) under an asynchronous communication model. The number of nodes is denoted by n, the initial advantage is $\epsilon_\star = p - \frac{1}{2}$, and K, c are constants. Lower bounds should be interpreted as follows: any protocol consuming $O(\cdot)$ of one resource (e.g., storage) requires $\Omega(\cdot)$ of another (e.g., time); upper bounds imply the existence of a protocol with resource costs in complexity class $O(\cdot)$.

time (expected or w.h.p. depending on the paper). In population protocols, this is often called parallel convergence time, defined as the expected number of interactions needed for consensus divided by n. Since interactions happen concurrently in most population protocols, parallel time is related to wall clock time by a constant factor w.h.p. However, our model and protocols do not require nodes to communicate at each clock ring, and therefore, parallel time and wall clock time need not be proportional.

Much of the relevant work is related to population protocols (23) in which nodes (finite-state automata) engage in random pairwise interactions determined by a random scheduler and update their states according to the state machine. Majority consensus is widely studied under this model in two forms: exact majority refers to protocols that converge to the majority bit with probability 1, whereas approximate majority protocols can converge to the incorrect answer with positive (possibly vanishing) probability. We focus on approximate majority, which has received less attention.

To the best of our knowledge, relevant lower bounds have been proved only for exact consensus. Early papers in this space developed some of the proof techniques that were later used to derive lower time and communication complexity bounds for exact binary and plurality consensus (24, 25). Notably, a series of papers (18–20) culminates in a recent result by Alistarh et al. (21). They consider protocols that satisfy monotonicity and some output conditions and show that, for such protocols to achieve exact consensus in $O(n^{1-c})$ parallel time for some $c>0$, the memory needed is $\Omega(\log n)$ states. We show that, under the assumptions of our model (in particular for $\epsilon_\star$ constant), this is not true for approximate consensus; in a comparable asynchronous model, one can achieve consensus with $\widetilde{O}(\log n)$ parallel time using only $O((\log \log n)^3)$ states of memory and $O(n)$ messages. Nonetheless, notice that our results assume that the initial advantage $\epsilon_\star$ (fraction of nodes with the majority bit) is a constant; this is a stronger assumption than prior work in which the initial advantage can converge to zero. Studying the problem for vanishingly small $\epsilon_\star$ is an interesting question, which we leave for future work.

Outline. We precisely define our model in *The Model* and give proof outlines for our main results in *Proof Outlines*. The full proofs can be found in the extended version of this paper (12).

The Model

Consider a set of n nodes connected in a complete graph topology enumerated by $[n] = \{1, 2, \dots, n\}$. These indices are only for our own bookkeeping and cannot be used by nodes during the protocol. At any point in time, a node $i \in [n]$ has a state chosen from a set $\mathcal{S}$ of cardinality $s \in \{2, 3, \dots\}$. We may assume that each state is a binary string of $\lceil \log_2(s) \rceil$ bits. For a node $i \in [n]$ and a time $t \geq 0$, let $\sigma(i, t) \in \mathcal{S}$ denote the state of node i at time t. All logarithms that we consider throughout the paper will be in base 2 (i.e., $\log x = \log_2 x$ for any $x > 0$).

At the beginning of the protocol, each node i is assigned a bit $\mathfrak{b}_i \in \{0, 1\}$, which is stored in its memory. The state of i at time $t = 0$ can, for example, be represented as a single bit $\mathfrak{b}_i$ followed by $\lceil \log_2(s) \rceil - 1$ bits 0. Let $\mathfrak{b}$ be the majority bit: that is, $\mathfrak{b} = 0$ if and only if[†]

$$\#\{i \in [n] : b_i = 0\} \geq \#\{i \in [n] : b_i = 1\},$$

where $\#A \in \mathbb{N} \cup \{0, \infty\}$ denotes the cardinality of a set A and $\mathbb{N} = \{1, 2, \dots\}$. Let $p \in [1/2, 1]$ be the fraction of nodes for which $\mathfrak{b}_i = \mathfrak{b}$: that is, $p = n^{-1} \cdot \#\{i \in [n] : \mathfrak{b}_i = \mathfrak{b}\}$.

Each node i has an independent unit rate Poisson clock $\mathcal{P}_i$. We identify $\mathcal{P}_i \subset \mathbb{R}_+$ with the set of times that the clock rings. Whenever i's clock rings (i.e., at every time $t \geq 0$ such that $t \in \mathcal{P}_i$), the node is allowed to communicate with another node. The node chooses based on its current state whether to initiate a communication with another node. In other words, there is a set of states $\mathcal{S}' \subset \mathcal{S}$ such that a node $i \in [n]$ initiates a

[†]To resolve draws, we define $\mathfrak{b} = 0$ if there are equally many nodes for $\mathfrak{b}_i = 0$ and $\mathfrak{b}_i = 1$.

communication with another node j at time $t \in \mathcal{P}_i$ if and only if $\sigma(i, t^-) \in \mathcal{S}'$, where $\sigma(i, t^-) \in \mathcal{S}'$ is the state of i infinitesimally before time t. The node j is always chosen uniformly at random from $[n] \setminus \{i\}$ independently of all other randomness. For each $i \in [n]$ and $t \in \mathcal{P}_i$, let $\mathfrak{r}(i, t) \in [n]$ denote the node that i would contact at time t if $\sigma(i, t^-) \in \mathcal{S}'$. The process of initiating a communication has unit cost.

When a connection is established between nodes i and j, each node observes the state of the other node, and the nodes update their states to reflect any new information gained during the interaction. The new states of the nodes are a deterministic function of the state of each node before the communication: that is, there is a function $\Lambda : \mathcal{S}' \times \mathcal{S} \to \mathcal{S}^2$ such that, if i was the initiator of the communication,

$$(\sigma(i, t), \sigma(j, t)) = \Lambda(\sigma(i, t^-), \sigma(j, t^-)).$$

Let $\Lambda_1 : \mathcal{S}' \times \mathcal{S} \to \mathcal{S}$ and $\Lambda_2 : \mathcal{S}' \times \mathcal{S} \to \mathcal{S}$ denote the coordinate functions of Λ such that $\Lambda(\sigma_1, \sigma_2) = (\Lambda_1(\sigma_1, \sigma_2), \Lambda_2(\sigma_1, \sigma_2))$ for all $\sigma_1 \in \mathcal{S}'$ and $\sigma_2 \in \mathcal{S}$. Let $\Theta_i \subset \mathbb{N}$ denote the set of times at which node i initiates a communication: that is, $\Theta_i = \{t \in \mathcal{P}_i : \sigma(i, t^-) \in \mathcal{S}'\}$. A node i that does not initiate a communication at time $t \in \mathcal{P}_i$ may also update its state. More precisely, there is a function[‡] $\Lambda' : \mathcal{S} \to \mathcal{S}$ such that, if $\sigma(i, t^-) \notin \mathcal{S}'$ (so that i does not communicate with any other node at time t), $\sigma(i, t) = \Lambda'(\sigma(i, t^-))$.

At any time $t \geq 0$, each node i has an estimate for $\mathfrak{b}$, which we call the belief bit of i and denote by $\widehat{\sigma}(i, t) \in \{0, 1\}$. We have reached consensus when all nodes have belief bit equal to $\mathfrak{b}$ for the remainder of the protocol: that is, consensus is reached at the time $\tau_{\text{consensus}}$ defined by

$$\tau_{\text{consensus}} = \inf\{t \geq 0 : \widehat{\sigma}(i, t') = \mathfrak{b},\ \forall i \in [n],\ t' \geq t\},$$

where the infimum of an empty set is ∞. For $t \geq 0$, let $N(t)$ denote the number of communications initiated before or at time t: that is, $N(t) = \sum_{i \in [n]} \#(\Theta_i \cap [0, t])$. The cost until consensus is the random variable $N_{\text{consensus}}$ defined by $N_{\text{consensus}} = N(\tau_{\text{consensus}})$, and $N_{\text{consensus}}$ is the number of communications required to reach consensus. This notion is also commonly referred to as convergence in the literature.

Terminal consensus is a stronger notion of consensus. To define this, we first need to introduce the notion of a terminal state. A state $\sigma \in \mathcal{S}$ is a terminal state if a node in this state will never change state and never initiate further communications: that is,

$$\sigma \notin \mathcal{S}' \qquad \text{and} \qquad \Lambda_2(\sigma', \sigma) = \sigma,\ \forall \sigma' \in \mathcal{S}'.$$

Let $\mathcal{S}_\infty \subset \mathcal{S}$ denote the (possibly empty) set of terminal states. We say that we have reached terminal consensus if all nodes are in a terminal state and have belief bit equal to $\mathfrak{b}$: that is, terminal consensus is reached at the time τ_{terminal} defined by

$$\tau_{\text{terminal}} = \inf\{t \geq 0 : \sigma(i, t) \in \mathcal{S}_\infty \text{ and } \widehat{\sigma}(i, t) = \mathfrak{b},\ \forall i \in [n]\},$$

where the infimum of an empty set is ∞. The cost until terminal consensus is the random variable $N_{\text{consensus}}$ defined by $N_{\text{terminal}} = N(\tau_{\text{terminal}})$.

Our goal is to find a protocol that achieves consensus or terminal consensus w.h.p. while minimizing communication cost (i.e., minimizing $N_{\text{consensus}}$ or N_{terminal}). Note that nodes have no perception of time other than the information stored in their memory. Nodes can obtain an estimate for the time by counting their own clock rings or by receiving such estimates from other nodes.

Stability. We say that a state is stable if a node in this state will always keep its belief bit for the remainder of the protocol. In other words, a state $\sigma \in \mathcal{S}$ is stable if a node i in this state at time t satisfies $\widehat{\sigma}(i, s) = \widehat{\sigma}(i, t)$ for all $s \geq t$, no matter which other nodes it communicates with at times $> t$. The set of stable states is a subset of the set of passive states. When we reach consensus (as defined by $\tau_{\text{consensus}}$), all nodes have belief bit equal to the majority bit, but the nodes are not necessarily aware that they have identified the majority bit. A node in a terminal state, on the other hand, never updates its belief bit and is, therefore, stable. Notice that, when we reach terminal consensus, all nodes are in stable states, but this is not necessarily the case when we reach consensus. Not all stable states are terminal states since nodes in stable states may change their state (only the belief bit must stay fixed), and they may initiate communications with other nodes.

Proof Outlines

In *Simple Upper Bound for* $s = C_\epsilon(\log n)^2$ and *Upper Bound for* $s = C(\log \log n)^3$, we describe the consensus protocols used to prove our upper bounds in the proposition and Theorem 1, respectively, along with a proof sketch. We then provide a proof sketch for the lower bound in Theorem 2 in *Lower-Bound Proof Sketch*. Full proofs can be found in the extended version (12).

Both the upper bounds that we present rely on the fact that it is more efficient in terms of communication to have a few nodes learn a good estimate of the true bit and then, have those nodes share this information with the other nodes. This strategy uses less communication than a majority of three protocol, where every node tries to learn the true bit.

Simple Upper Bound for $s = C_\epsilon(\log n)^2$. The upper bound relies on different nodes playing different roles as we describe. All of the nodes are assigned types that describe their behavior: aspirant, expert, regular, or terminal. Aspirants aspire to be experts, and experts are the knowledgeable nodes that first learn and then spread information about the correct bit. A key challenge is ensuring that experts communicate enough to first learn the correct bit and then disseminate it without wasting unnecessary communication. We describe the four phases of the protocol and the behavior of each type of node (Figs. 2–4). The phases are partly overlapping in time due to the asynchronous nature of the communications. Fig. 4 illustrates how the fraction of nodes in each state evolves for a simulation of $n = 1,000$ nodes with initial majority fraction $p = 0.7$. Fig. 2 shows an illustration of the phases, and Fig. 3 shows how the nodes move from one role to the next in different phases of the protocol.

Expert selection phase. Since we are not allowed to designate certain nodes as experts a priori and the nodes do not have

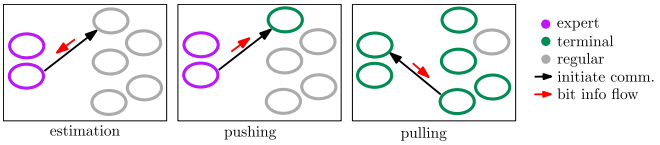

Fig. 2. The figure illustrates three of the phases of the protocol described in *Simple Upper Bound for* $s = C_\epsilon(\log n)^2$: the estimation phase, the pushing phase, and the pulling phase. In the estimation phase, each expert asks $C_\epsilon \log n$ nodes for their bit, and each expert calculates the majority bit among the asked nodes. In the pushing phase, each expert informs $\log n$ nodes about the bit calculated in the estimation phase, and these nodes become terminal nodes. Finally, in the pulling phase, uninformed nodes initiate communications (comm.) until they encounter a terminal node.

[‡]Note that for the asynchronous model defined here it is sufficient to define $\Lambda'|_{\mathcal{S} \setminus \mathcal{S}'}$. However, we choose to let the domain of Λ' be $\mathcal{S}$ since we use the same function for the synchronous model, which is defined later in this section.

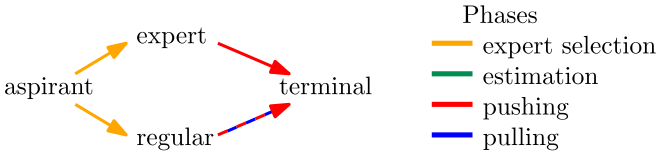

Fig. 3. The figure shows the four types of nodes considered in the proof of the proposition and which types the nodes can move between in the various phases.

internal randomness, the nodes rely on the randomness of the bits assigned to them to choose their roles. When the protocol begins at time $t=0$, all of the nodes are aspirants that may become experts. Each aspirant i repeatedly obtains an ordered tuple of bits (b', b'') by asking two other uniformly chosen nodes for their belief bit in consecutive clock rings. If it observes $\log\log n$ tuples $(0,1)$ before the first tuple $(1,0)$, then it becomes an expert; otherwise, it becomes a regular node.

Note that each time a node obtains a tuple (b', b''), it is equally likely that $(b', b'')=(0,1)$ and that $(b', b'')=(1,0)$ [von Neumann's unbiasing (26)]. Therefore an aspirant turns into an expert with probability $0.5^{\lceil \log\log n\rceil} \approx 1/\log n$, and therefore, we create approximately $n/\log n$ experts w.h.p.

Estimation phase. After the experts are designated, each expert i contacts a uniformly chosen node j at each of $\widehat{C}_\epsilon \log n$ consecutive clock rings and stores the initial bit $\mathfrak{b}_j$ of each node j. At the end of the estimation phase, the expert i calculates the majority bit among the $\mathfrak{b}_j$, and this becomes the new belief bit of i. We choose $\widehat{C}_\epsilon = \frac{10}{\epsilon^2}$. By a Chernoff bound and a union bound, w.h.p. all of the experts estimate the majority bit correctly in the estimation phase.

Pushing phase. Each expert i initiates a communication with a uniformly sampled node j at each of $\log n$ consecutive clock rings. The expert i sends its estimate of the majority bit to j, and j adopts this estimate and becomes a terminal node. Terminal nodes do not initiate any communications and do not change their state if other nodes initiate communications with them. After the $\log n$ clock rings, i also becomes a terminal node. Since there are $\Theta(n/\log n)$ experts and each expert contacts $\log n$ nodes, one can argue that w.h.p. a constant fraction of the nodes becomes a terminal node in this phase.

Pulling phase. Each regular node i initiates a communication with another node every $\log n$ clock rings[§] until it encounters a terminal node j. When i succeeds, it adopts the estimate of j for the majority bit and becomes a terminal node. The protocol ends when all of the nodes are terminal.

The communication cost in this phase is $O(n)$ since a uniformly positive fraction of the nodes is terminal nodes at the beginning of the phase, and therefore, the number of trials of each regular node is stochastically dominated by a geometric random variable with uniformly positive success probability, which has expectation $O(1)$.

Memory usage. Among the four types of nodes that we have introduced, the experts require the most memory. They use $\Theta(\widehat{C}_\epsilon \log n)$ states to count time during the estimation phase, and they use $\Theta(\widehat{C}_\epsilon \log n)$ states to store the initial bits of nodes encountered during this phase, which gives a total of $\Theta((\widehat{C}_\epsilon \log n)^2)$ states of memory.

Simulations. To illustrate this scheme's practical performance, we compare it in simulation with the well-studied best of three polling protocol (11), where each node polls three nodes in successive clock rings and updates its belief bit to the majority vote of the polled bits (code available in ref. 27). Although such protocols have order-optimal convergence times, they are not designed to optimize communication cost. Fig. 5 compares the average number of communications (15 trials) needed to reach stable consensus for both protocols since best of three polling never reaches terminal consensus. We use an initial advantage of $\epsilon=0.2$ for both protocols. Although our proofs use $\widehat{C}_\epsilon = \frac{10}{\epsilon^2}$, we empirically find $\widehat{C}_\epsilon = 7$ to work well (the protocol consistently converges correctly).

Upper Bound for $s = C(\log\log n)^3$. The protocol used to prove Theorem 1 has a more complex structure than the protocol analyzed in the proposition. Like the previous protocol, the protocol proceeds in phases: an expert selection phase followed by an estimation phase, a pushing phase, and a pulling phase. However, in this phase, due to tighter memory constraints, nodes are unable to count high enough to execute the previous protocol; recall that some phases lasted $\Theta(\log n)$ clock rings. To deal with this issue, the estimation phase is subdivided into shorter rounds, with expert nodes in each round. Since each round must use less communication than in the previous protocol, a single round makes only partial progress toward probabilistic consensus.

The majority of three experts at round m is used to create experts in the round $m+1$, similar to the procedure of majority of three protocols (11, 28–30). Since there must be fewer round $m+1$ experts than round m experts, the experts spread their bit during the round using rumor spreading to maintain the fraction of experts. After a sufficient number of rounds has been completed, the experts in the final level push their bit out as in the previous protocol.

Due to the asynchronous nature of the Poisson clocks, the phases (and rounds) are partly overlapping in time; the main technical challenge is designing a scheme that is robust to these overlaps and proving such robustness. At any point in time, each node is one of the following types: aspirant, expert, expert candidate, regular, informed, or terminal.

Expert selection phase. All nodes are aspirants in the beginning of the expert selection phase. The purpose of this phase is to select $\sim n2^{-K}$ level 0 experts for $K=\Theta(\log\log n)$. Nodes that do not become experts become regular nodes. The selection of experts is done by von Neumann unbiasing.

Estimation phase. The estimation phase consists of $M = 2\log\log n$ rounds. Level m is associated with a set of $\sim n2^{-K}$ nodes that we call level m experts. As before, a node may become a level m expert on being contacted by at least three level $m-1$ experts or on being contacted by one level m expert. There are $\sim n2^{-3K}$ level m experts of the former kind, and their belief bit

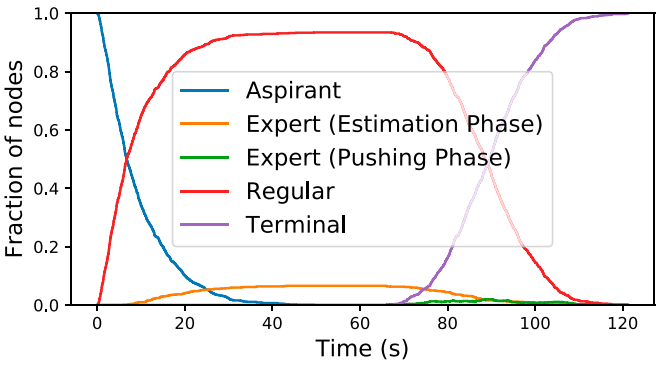

Fig. 4. Evolution of node types as the protocol progresses. All nodes end up in a terminal state, thus limiting the communication cost of the protocol.

[§]In fact, regular nodes initiate a communication with another node every $\log n$ clock rings throughout the full protocol. However, only in the pulling phase and the latter part of the estimation phase are they likely to encounter a terminal node.

is obtained by calculating the majority bit among the three belief bits received from level $m-1$ experts. Each of these level m experts creates $\sim 2^{2K}$ new level m experts by "rumor spreading" their belief bit for $2K$ clock rings. As in the synchronous case, w.h.p. all level M experts will identify the majority bit $\mathfrak{b}$. At the end of the estimation phase, all level M experts become informed nodes.

One substantial challenge is the creation of level m experts from level $m-1$ experts. At first glance, one might think that a level m expert could be created when (for instance) three level $m-1$ experts contact a node. However, since different nodes are at different levels, this event is unlikely and will, therefore, generate too few experts. Hence, we introduce the notion of an expert candidate. A node must remain an expert candidate for a sufficient amount of time to allow other level $m-1$ experts to contact it. However, it should not remain an expert candidate indefinitely. We let an expert candidate convert to a regular node after $\Theta((\log\log n)^2)$ clock rings, which gives sufficient time to be contacted by three level $m-1$ experts since this is the duration of the estimation phase for most nodes.

Pushing phase. Informed nodes spread the bit $\mathfrak{b}$ until a constant fraction of the nodes is terminal nodes with the bit $\mathfrak{b}$. More precisely, every time the clock of an informed node rings, it contacts a uniformly chosen node, and if this node is a regular node, it transforms into an informed node. Similarly as in the synchronous model, the spreading slows down when a sufficiently high fraction of the nodes is terminal nodes since an informed node transforms into a terminal node when it contacts a terminal node.

Pulling phase. Throughout the protocol, each regular node initiates a communication every $\Theta((\log\log n)^2)$ clock rings until it encounters a terminal node, on which it also becomes a terminal node. By comparison with geometric random variables as before, we get that the number of communications in this phase is $O(n)$.

Memory usage. Among the six types of nodes that we have introduced, the expert candidates require the most memory: They use $\Theta((\log\log n)^2)$ states to count down time to the conversion to a regular node and $\Theta(\log\log n)$ states to store the level number for a total of $\Theta((\log\log n)^3)$ states of memory.

Lower-Bound Proof Sketch. In this section, we outline the proof of the lower bound (Theorem 2). The notion of passive and active states plays an essential role. A state $\sigma\in\mathcal{S}$ is passive if a node in this state will not initiate communication prior to another node contacting it. A state is called active if it is not passive.

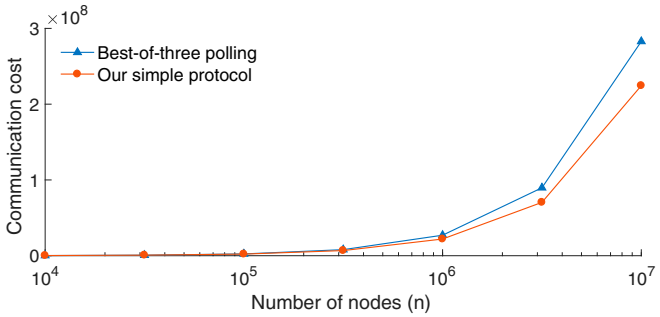

Fig. 5. Number of communications needed to reach stable consensus for our simple protocol (*Simple Upper Bound for* $s=C_\epsilon(\log n)^2$) and best of three polling assuming an initial majority fraction $p=0.7$. As the number of nodes n grows, communication savings become more pronounced, with best of three polling surpassing our proposed protocol around $n=10^5$. Note that with our improved protocol [*Upper Bound for* $s=C(\log\log n)^3$], we expect to see a further improvement in the communication cost for large values of n.

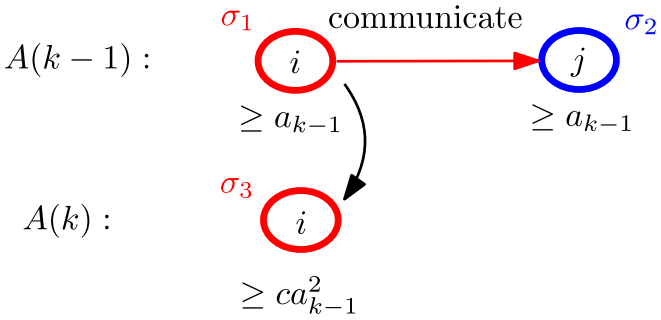

Fig. 6. All states σ_1 and σ_2 in $A(k-1)$ have frequency at least a_{k-1} w.h.p. Thus, all states in $A(k)$ have frequency at least ca_{k-1}^2 w.h.p.

Passive states are essential for reducing the communication cost in *Simple Upper Bound for* $s=C_\epsilon(\log n)^2$ and *Upper Bound for* $s=C(\log\log n)^3$. On the other hand, as we discuss below, it is costly to have many nodes in passive states unless they have a correct estimate for the majority bit.

Summary. Our lower bound considers two cases: 1) no passive state occurs, and 2) at least one passive state occurs with positive probability. The first case is straightforward. If all of the nodes are active, we can lower bound each node's rate of communication and relate this to the overall communication cost by using a simple lower bound on the time to consensus. Since $\#\mathcal{S}=s$, active nodes must be involved in a communication (either as initiator or recipient) at least every s clock rings. This follows because active nodes eventually communicate even without being contacted by another node; the only way for them to know when to communicate is by counting clock rings. Since they can only count up to s, they must communicate at least once every s clock rings.

For the second case, if there is even one node in a passive state, we show that there must be many nodes (at least $n^{0.9}$) in that passive state w.h.p., regardless of the true majority bit. Roughly, this follows because s is (comparatively) small, and therefore, all states that are attained with positive probability will appear in large numbers after s clock rings. The lower bound follows because all nodes in a passive state with the wrong belief bit must be reached by a correct node; the required communication can be bounded with a coupon collector argument.

More detailed sketch. Let $\mathcal{S}_0\subset\mathcal{S}$ be a set of all states that are attained with positive probability: that is,

$$\mathcal{S}_0=\{\sigma_0\in\mathcal{S}:\exists t\geq 0, i\in[n] \text{ such that } \mathbb{P}[\sigma(i,t)=\sigma_0]>0\}.$$

Consider two cases: 1) all nodes are active almost surely (all states in $\mathcal{S}_0$ are active), and 2) nodes in passive states arise with positive probability ($\mathcal{S}_0$ contains at least one passive state).

For case 1, we know that, even if all nodes were to initiate a communication every time their clock rings, w.h.p. there are nodes that do not communicate a single time before time $t=\Omega(\log n)$. Therefore, $\tau_{\text{consensus}}=\Omega(\log n)$ w.h.p. This implies the theorem in case 1 since we have n nodes that communicate for time $\tau_{\text{consensus}}=\Omega(\log n)$ at rate at least $1/s$, and therefore, the total number of communications is $\Omega(ns^{-1}\log n)$.

In case 2, we show that, if $\sigma_0\in\mathcal{S}_0$ is passive, then w.h.p. there are $n^{0.9}$ nodes[¶] in state σ_0 at time s independently of whether the true majority bit $\mathfrak{b}=0$ or $\mathfrak{b}=1$. We first explain how to conclude the proof after we have established this result. If $\mathfrak{b}\neq\widehat{\sigma}(i,t)$ for a node i in state σ_0 at time t, then to reach consensus, all of the $n^{0.9}$ nodes with state

[¶]The exponent 0.9 is arbitrary; we can obtain any fixed power of n by adjusting the constant c.

σ_0 must be reached by other nodes to reach consensus. By a coupon collector argument, $\Theta(n^{0.1}n^{0.9}\log n^{0.9}) = \Theta(n\log n)$ communications are necessary to reach the $n^{0.9}$ nodes (31).

To prove that there are $n^{0.9}$ nodes in state σ_0 at time s, we show that w.h.p. for all $\sigma \in \mathcal{S}_0$ (passive or not) there are at least $n^{0.9}$ nodes in state σ at time s. Let $A(0) \subset \mathcal{S}$ be the set of the two initial states that the nodes can take at time $t = 0$. We define $A(k) \subset \mathcal{S}$ inductively as the set of states that may be attained from states in $A(k-1)$ [i.e., the set of all possible states that may arise from any set of nodes with states in $A(k-1)$ after a single clock ring]. We note that $A(k)$ is obtained deterministically from $A(k-1)$ and does not depend on the realizations of the clock rings, communications, or the majority bit. Also, the number of elements in $A(k)$ is increasing with k since it is always possible that a node does not change state after a clock ring (e.g., if it was not involved with either the clock ring or any resulting communication). We use this fact and the bound on the total number of states, $\#\mathcal{S}_0 \le s$, to show that in fact $A(k) = \mathcal{S}_0$ for all $k \ge s$.

We see that all states in $\mathcal{S}_0$ can be present after the sth clock ring, regardless of whether the majority bit $\mathfrak{b} = 0$ or $\mathfrak{b} = 1$. Indeed, one can show something stronger: not only are all states in $\mathcal{S}_0$ possible after s clock rings, but they are present w.h.p. for n large enough. As a result, we cannot have any states that are stable in $\mathcal{S}_0$ (i.e., states that never change their belief bit). If a stable state σ was present, then we could choose majority bit $\mathfrak{b}$ not equal to the belief bit of σ; since σ appears w.h.p., consensus could not occur. Thus, nodes in a passive state with the incorrect belief bit must be contacted to achieve consensus.

To show that for all $\sigma \in \mathcal{S}_0$, there are at least $n^{0.9}$ nodes in state σ at time s, consider the deterministic set $A(k-1)$ at time $k-1$. Suppose that each of the states in $A(k-1)$ occurs with frequency at least a_{k-1} (i.e., in at least na_{k-1} nodes) at time $k-1$ as illustrated in Fig. 6. Then, w.h.p. all states in $A(k)$ are found in the protocol at time k with frequency at least $c_0 a_{k-1}^2$ for some constant $c_0 > 0$. To see why this is true, we consider all possible interactions between pairs of states in $A(k-1)$ in the unit time interval between $k-1$ and k. Let $\sigma_1 \in A(k-1)$. Then, if the node in state σ_1 initiates communication, the probability that it interacts with some state in $A(k-1)$ is at least a_{k-1}. Therefore, the frequency of states in $A(k)$ that will be present at time k is at least cna_{k-1}^2 w.h.p., where the constant c depends on the various probabilities of communications happening during that unit time interval from k to $k+1$. Applying this bound on the frequency of states from $A(k)$ iteratively and using $s \le \log\log n - c^{-1}$, we get that all states in $\mathcal{S}_0 = A(s)$ are found with frequency at least $(c')^{2^s} > n^{-1} \cdot n^{0.9}$ at time s for a constant $c' > 0$ w.h.p. Since this holds for any state $\sigma \in \mathcal{S}_0$, it holds in particular for passive states, which gives the result.

Data Availability Statement. Code for simulations related to this paper are available in GitHub (27). Complete proofs of all theorems stated in our paper can be found in ref. 12.

ACKNOWLEDGMENTS. We thank Rati Gelashvili for useful references and the reviewers for their helpful suggestions. This work was supported by Microsoft Research, the Simons Institute, Sutardja Center for Entrepreneurship and Technology (SCET) Berkeley, Dr. Max Rössler, the Walter Haefner Foundation, the ETH Zürich Foundation, Distributed Technologies Research, NSF Grant CIF-1705007, Army Research Office Grant W911NF1810332, and Input Output Hong Kong.

1. C. Cachin, "Architecture of the hyperledger blockchain fabric" in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers in Conjunction with ACM Conference on Principles of Distributed Computing (PODC)* (ACM, 2016), vol. 310, p. 4.
2. T. Rocket, Snowflake to avalanche: A novel metastable consensus protocol family for cryptocurrencies. https://pdfs.semanticscholar.org/85ec/19594046bbcfe12137c7c2e3744677129820.pdf. Accessed 7 November 2018.
3. R. Olfati-Saber, J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion"in *Conference on Decision and Control* (IEEE, 2005), pp. 6698–6703.
4. W. Yu, G. Chen, Z. Wang, W. Yang, Distributed consensus filtering in sensor networks. *Trans. Syst. Man Cybern. B* **39**, 1568–1577 (2009).
5. L. Cardelli, A. Csikász-Nagy, The cell cycle switch computes approximate majority. *Sci. Rep.* **2**, 656 (2012).
6. Google, OpenThread. https://openthread.io/. Accessed 7 November 2018.
7. P. Miranda, M. Siekkinen, H. Waris, "TLS and energy consumption on a mobile device: A measurement study" in *Computers and Communications (ISCC)* (IEEE, 2011), pp. 983–989.
8. Y. Hassin, D. Peleg, Distributed probabilistic polling and applications to proportionate agreement. *Inf. Comput.* **171**, 248–268 (2001).
9. E. Perron, D. Vasudevan, M. Vojnovic, "Using three states for binary consensus on complete graphs" in *INFOCOM 2009* (IEEE, 2009), pp. 2527–2535.
10. D. Angluin, J. Aspnes, D. Eisenstat, A simple population protocol for fast robust approximate majority. *Distr. Comput.* **21**, 87–102 (2008).
11. J. Cruise, A. Ganesh, Probabilistic consensus via polling and majority rules. *Queueing Syst.* **78**, 99–120 (2014).
12. G. Fanti, N. Holden, Y. Peres, G. Ranade, Communication cost of consensus for nodes with limited memory. arXiv:1901.01665 (7 January 2019).
13. P. Berenbrink, D. Kaaser, P. Kling, L. Otterbach, "Simple and efficient leader election" in *OASIcs-OpenAccess Series in Informatics*, R. Seidel, Ed. (Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstul, Germany, 2018), vol. 61, pp. 9.1–9.11.
14. L. Becchetti, A. Clementi, E. Natale, F. Pasquale, R. Silvestri, "Plurality consensus in the gossip model" in *Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms* (ACM, 2015).
15. M. Ghaffari, M. Parter, "A polylogarithmic gossip algorithm for plurality consensus" in *Symposium on Principles of Distributed Computing* (ACM, 2016), pp. 117–126.
16. M. Draief, M. Vojnović, Convergence speed of binary interval consensus. *J. Control Optim.* **50**, 1087–1109 (2012).
17. G. B. Mertzios, S. E. Nikoletseas, C. L. Raptopoulos, P. G. Spirakis, "Determining majority in networks with local interactions and very small local memory" in *International Colloquium on Automata, Languages, and Programming* (Springer, 2014), pp. 871–882.
18. D. Alistarh, R. Gelashvili, M. Vojnović, "Fast and exact majority in population protocols" in *Symposium on Principles of Distributed Computing* (ACM, 2015), pp. 47–56.
19. D. Alistarh, J. Aspnes, D. Eisenstat, R. Gelashvili, R. L. Rivest, "Time-space tradeoffs in population protocols" in *ACM-SIAM Symposium on Discrete Algorithms* (ACM, 2017), pp. 2560–2579.
20. A. Bilke, C. Cooper, R. Elsaesser, T. Radzik, Population protocols for leader election and exact majority with $O(\log^2 n)$ states and $O(\log^2 n)$ convergence time. arXiv:1705.01146 (2 May 2017).
21. D. Alistarh, J. Aspnes, R. Gelashvili, "Space-optimal majority in population protocols" in *ACM-SIAM Symposium on Discrete Algorithms* (ACM, 2018), pp. 2221–2239.
22. A. Kosowski, P. Uzna'ski, Population protocols are fast. arXiv:1802.06872 (19 February 2018).
23. D. Angluin, J. Aspnes, Z. Diamadi, M. J. Fischer, R. Peralta, Computation in networks of passively mobile finite-state sensors. *Distr. Comput.* **18**, 235–253 (2006).
24. R. Karp, C. Schindelhauer, S. Shenker, B. Vocking, "Randomized rumor spreading" in *Foundations of Computer Science* (IEEE, 2000), pp. 565–574.
25. D. Doty, "Timing in chemical reaction networks" in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms* (ACM, 2014), pp. 772–784.
26. J. Von Neumann, Various techniques used in connection with random digits. *Nat. Bur. Stand. Appl. Math. Series* **3**, 36–38 (1951).
27. G. Fanti, N. Holden, Y. Peres, and G. Ranade, Simulations from the paper "Communication cost of consensus for nodes with limited memory." GitHub. https://github.com/gfanti/communication-cost-consensus. Deposited 17 January 2020.
28. Y. Kanoria, A. Montanari, Majority dynamics on trees and the dynamic cavity method. *Ann. Appl. Probab.* **21**, 1694–1748 (2011).
29. M. A. Abdullah, M. Draief, Global majority consensus by local majority polling on graphs of a given degree sequence. *Discrete Appl. Math.* **180**, 1–10 (2015).
30. E. Mossel, J. Neeman, O. Tamuz, Majority dynamics and aggregation of information in social networks. *Aut. Agents Multi-Agent Syst.* **28**, 408–429 (2014).
31. M. Mitzenmacher, E. Upfal, *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis* (Cambridge University Press, 2017).