HAL
archives-ouvertes.fr

Consensus Needs Broadcast in Noiseless Models but can be Exponentially Easier in the Presence of Noise

Andrea Clementi, Luciano Gualà, Emanuele Natale, Francesco Pasquale, Giacomo Scornavacca, Luca Trevisan

To cite this version:

Andrea Clementi, Luciano Gualà, Emanuele Natale, Francesco Pasquale, Giacomo Scornavacca, et al.. Consensus Needs Broadcast in Noiseless Models but can be Exponentially Easier in the Presence of Noise. [Research Report] CNRS. 2018. hal-01958994

HAL Id: hal-01958994

https://hal.inria.fr/hal-01958994

Submitted on 18 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Consensus Needs Broadcast in Noiseless Models but can be Exponentially Easier in the Presence of Noise

Andrea Clementi
Università di Roma Tor Vergata
Rome, Italy
clementi@mat.uniroma2.it

Luciano Gualà
Università di Roma Tor Vergata
Rome, Italy
guala@mat.uniroma2.it

Emanuele Natale
Max Planck Institute for Informatics
Saarbrücken, Germany
emanuele.natale@mpi-inf.mpg.de

Francesco Pasquale
Università di Roma Tor Vergata
Rome, Italy
pasquale@mat.uniroma2.it

Giacomo Scornavacca
Università degli Studi dell'Aquila
L'aquila, Italy
giacomo.scornavacca@graduate.univaq.it

Luca Trevisan
U.C. Berkeley
Berkeley, CA, United States
luca@berkeley.edu

Abstract

Consensus and Broadcast are two fundamental problems in distributed computing, whose solutions have several applications. Intuitively, Consensus should be no harder than Broadcast, and this can be rigorously established in several models. Can Consensus be *easier* than Broadcast?

In models that allow noiseless communication, we prove a reduction of (a suitable variant of) Broadcast to binary Consensus, that preserves the communication model and all complexity parameters such as randomness, number of rounds, communication per round, etc., while there is a loss in the success probability of the protocol. Using this reduction, we get, among other applications, the first logarithmic lower bound on the number of rounds needed to achieve Consensus in the uniform GOSSIP model on the complete graph. The lower bound is tight and, in this model, Consensus and Broadcast are equivalent.

We then turn to distributed models with noisy communication channels that have been studied in the context of some bio-inspired systems. In such models, only one noisy bit is exchanged when a communication channel is established between two nodes, and so one cannot easily simulate a noiseless protocol by using error-correcting codes. An $\Omega(\varepsilon^{-2}n)$ lower bound on the number of rounds needed for Broadcast is proved by Boczkowski et al. [PLOS Comp. Bio. 2018] in one such model (noisy uniform PULL, where ε is a parameter that measures the amount of noise). We prove an $O(\varepsilon^{-2}\log n)$ upper bound for binary Consensus in such model, thus establishing an exponential gap between the number of rounds necessary for Consensus versus Broadcast. We also prove a new $O(\varepsilon^{-2}n\log n)$ upper bound for Broadcast in this model.

1 Introduction

In this paper we investigate the relation between Consensus and Broadcast, which are two of the most fundamental algorithmic problems in distributed computing [31, 36, 64, 67], and we study how the presence or absence of communication noise affects their complexity.

In the (Single-Source) *Broadcast* problem, one node in a network has an initial message `msg` and the goal is for all the nodes in the network to receive a copy of `msg`.

In the *Consensus* problem, each of the n nodes of a network starts with an input value (which we will also call an *opinion*), and the goal is for all the nodes to converge to a configuration in which they all have the same opinion (this is the *agreement* requirement) and this shared opinion is one held by at least one node at the beginning (this is the *validity* requirement). In the *Binary* Consensus problem, there are only two possible opinions, which we denote by 0 and 1.

In the (binary) *Majority Consensus* problem [6, 35, 65] we are given the promise that one of the two possible opinions is initially held by at least $n/2 + b(n)$ nodes, where $b(n)$ is a parameter of the problem, and the goal is for the nodes to converge to a configuration in which they all have the opinion that, at the beginning, was held by the majority of nodes. Note that Consensus and Majority Consensus are incomparable problems: a protocol may solve one problem without solving the other.[1]

Motivations for studying the Broadcast problem are self-evident. Consensus and Majority Consensus are simplified models for the way inconsistencies and disagreements are resolved in social networks, biological models and peer-to-peer systems [37, 41, 62].[2]

In distributed model that severely restrict the way in which nodes communicate (to model constraints that arise in peer-to-peer systems or in social or biological networks), upper and lower bounds for the Broadcast problem give insights on the effect of the communication constraints on the way in which information can spread in the network. The analysis of algorithms for Consensus often give insights on how to break symmetry in distributed networks, when looking at how the protocol handles an initial opinion vector in which exactly half the nodes have one opinion and half have the other. The analysis of algorithms for Majority Consensus usually hinge on studying the rate at which the number of nodes holding the minority opinion shrinks.

If the nodes are labeled by $\{1, \ldots, n\}$, and each node knows its label, then there is an easy reduction of binary Consensus to Broadcast: node 1 broadcasts its initial opinion to all other nodes, and then all nodes agree on that opinion as the consensus opinion. Even if the nodes do not have known identities, they can first run a *leader election* protocol, and then proceed as above with the leader broadcasting its initial opinion. Even in models where leader election is not trivial, the best known Consensus protocol has, in all the cases that we are aware of, at most the "complexity" (whether it's measured in memory per node, communication per round, number of rounds, etc.) of the best known broadcast protocol.

The question that we address in this paper is whether the converse hold, that is, are there ways of obtaining a Broadcast protocol from a Consensus problem or are there gaps, in certain models, between the complexity of the two problems?

[1] A Consensus protocol is allowed to converge to an agreement to an opinion that was initially in the minority (provided that it was held by at least one node), while a Majority Consensus protocol must converge to the initial majority whenever the minority opinion is held by fewer than $n/2-b$ nodes. On the other hand, a Majority Consensus problem is allowed to converge to a configuration with no agreement if the initial opinion vector does not satisfy the promise, while a Consensus protocol must converge to an agreement regardless of the initial opinion vector.

[2] The Consensus problem is often studied in models in which nodes are subject to malicious faults, and, in that case, one has motivations from network security. In this paper we concentrate on models in which all nodes honestly follow the prescribed protocol and the only possibly faulty devices are the communication channels.

Roughly speaking, we will show that, in the presence of noiseless communication channels, every Consensus protocol can be used to realize a weak form of Broadcast. Since, in many cases, known lower bounds for Broadcast apply also to such weak form, we get new lower bounds for Consensus. In a previously studied, and well motivated, distributed model with noisy communication, however, we establish an exponential gap between Consensus and Broadcast.

1.1 Communication and computational models

In order to state and discuss our results we first introduce some distributed models and their associated complexity measures.

We study protocols defined on a communication network, described by an undirected graph $G = (V, E)$ where V is the set of nodes, each one running an instance of the distributed algorithm, and E is the set of pairs of nodes between which there is a communication link that allows them to exchange data. When not specified, G is assumed to be the complete graph.

In *synchronous parallel* models, there is a global clock and, at each time step, nodes are allowed to communicate using their links.

In the LOCAL model, there is no restriction on how many neighbors a node can talk to at each step, and no restriction on the number of bits transmitted at each step. There is also no restriction on the amount of memory and computational ability of each node. The only complexity measures is the number of rounds of communication. For example, it is easy to see that the complexity of Broadcast is the diameter of the graph G. The CONGEST model is like the LOCAL model but the amount of data that each node can send at each time step is limited, usually to $O(\log n)$ bits.

In the (general) GOSSIP model [30, 52], at each time step, each node v chooses one of its neighbors c_v and *activates* the communication link (v, c_v), over which communication becomes possible during that time step, allowing v to send a message to c_v and, simultaneously, c_v to send a message to v. We will call v the *caller of* c_v. In the PUSH variant, each node v sends a message to its chosen neighbor c_v; in the PULL variant, each node v sends a message to its callers. Note that, although each node chooses only one neighbor, some nodes may be chosen by several others, and so they may receive several messages in the PUSH setting, or send a message to several recipients in the PULL setting. In our algorithmic results for the GOSSIP model, we will assume that each message exchanged in each time step is only one bit, and our negative results for the noiseless setting will apply to the case of messages of unbounded length. In the *uniform* GOSSIP (respectively PUSH or PULL) model, the choice of c_v is done uniformly at random among the neighbors of v. This means that uniform models make sense even in anonymous networks, in which nodes are not aware of their identities nor of the identities of their neighbors.[3]

In this work, we are mainly interested in models like GOSSIP that severely restrict communication [6, 2, 35, 37, 62, 65], both for efficiency consideration and because such models capture aspects of the way consensus is reached in biological population systems, and other domains of interest in network science [5, 7, 36, 19, 37, 39, 41]. Communication capabilities in such scenarios are typically constrained and non-deterministic: both features are well-captured by uniform models.

Asynchronous variants of the GOSSIP model (such as *Population Protocols* [6, 5]) have also been extensively studied [18, 49, 65]. In this variant, no global clock is available to nodes. Instead, nodes are idle until a single node is activated by a (possibly random) scheduler, either in discrete time or in continuous time. When a node wakes up, it activates one of its incident edges and wakes up the corresponding neighbor. Communication happens only between those two vertices, which subsequently go idle again until the next time they wake up.

[3] In the general GOSSIP model in which a node can choose which incident edge to activate, a node must, at least, know its degree and have a way to distinguish between its incident edges.

Previous studies show that, in both PUSH and PULL variants of uniform GOSSIP, (binary) Consensus, Majority Consensus and Broadcast can be solved within logarithmic time (and work per node) in the complete graph, via elementary protocols[4], with high probability (for short *w.h.p.*[5]) [6, 14, 18, 35, 49, 54] (see also Section 1.5). Moreover, efficient protocols have been proposed for Broadcast and Majority Consensus for some restricted families of graphs such as regular expanders and random graphs [1, 24, 23, 29, 50, 61].

However, while for Broadcast $\Omega(\log n)$ time and work are necessary in the complete graph [18, 49, 54], prior to this work, it was still unknown whether a more efficient protocol existed for Consensus and Majority Consensus.

1.2 Our contribution I: Broadcast is "no harder" than Consensus over noiseless communication

Our first result is a reduction of a weak form of Broadcast to Consensus (Theorem 4.2) which establishes, among other lower bounds, tight logarithmic lower bounds for Consensus and Majority Consensus both in the uniform GOSSIP (and hence uniform PULL and PUSH as well) model and in the general PUSH model.

To describe our result, it is useful to introduce the notion of nodes *infected* by a source node in a distributed protocol: if s is a designated source node in the network, then we say that at time 0 the node s is the only infected node and, at time t, a node is infected if and only if either it was infected at time $t-1$ or it received a communication from an infected node at time t.

This notion is helpful in thinking about upper and lower bounds for Broadcast: any successful broadcast protocol from s needs to infect all nodes from source s, and any protocol that is able to infect all nodes from source s can be used to broadcast from s by appending `msg` to each message originating from an infected node. Thus any lower bound for infection is also a lower bound for Broadcast, and any protocol for infection can be converted, perhaps with a small overhead in communication, to a protocol for Broadcast. For example, in the PUSH model (either uniform or general[6]), the number of infected nodes can at most double at each step, because each infected node can send a message to only one other node, and this is the standard argument that proves an $\Omega(\log n)$ lower bound for Broadcast.

In Theorem 4.2 we show that lower bounds for infection *also give lower bounds for Consensus.* More precisely we prove that if we have a Consensus protocol that, for every initial opinion vector, succeeds in achieving consensus with probability $1-o(1/n)$, then there is an initial opinion vector and a source such that the protocol infects all nodes from that source with probability at least $(1-o(1))/n$. Equivalently, if we are in a model in which there is no source for which we can have probability, say, $\geqslant 1/(2n)$ of infecting all nodes with certain resources (such as time, memory, communication per node, etc.), then, in the same model, and with the same resources, every Consensus protocol has probability $\Omega(1/n)$ of failing. For example, by the above argument, we have an $\Omega(\log n)$ lower bound for Consensus in the PUSH model (because, in fewer than $\log_2 n$ rounds, the probability of infecting all nodes is zero).

The proof uses a hybrid argument to show that there are two initial opinion vectors $\mathbf{x}$ and $\mathbf{y}$, which are identical except for the initial opinion of a node s, such that there is at least a $(1-o(1))/n$ difference between the probability of converging to the all-zero configuration starting from $\mathbf{x}$ or from $\mathbf{y}$. Then, we argue that this difference must come entirely from runs of the protocol that fail to achieve consensus (which happens only with $o(1/n)$ probability) or from runs of the protocol in

[4]In the case of Majority Consensus, the initial additive bias must have size $\Omega(\sqrt{n\log n})$.

[5]In this paper, we say that an event $\mathcal{E}_n$ holds *w.h.p.* if $\mathbf{P}(\mathcal{E}_n) \geqslant 1-n^{-\alpha}$, for some $\alpha > 1$.

[6]See Section 3.1 for a formal definition of the two variants.

which s infects all other nodes. Thus the probability that s infects all nodes from the initial vector $\mathbf{x}$ has to be $\geqslant (1 - o(1))/n$.

As for Majority Consensus, we have a similar reduction, but from a variant of the infection problem in which there is an initial set of b infected nodes.[7]

Lower bounds for infection are known in several models in which there where no previous negative results for Consensus. We have not attempted to survey all possible applications of our reductions, but here we enumerate some of them:

- In the uniform GOSSIP model (also known as uniform PUSH-PULL model), and in the general PUSH model, tight analysis (see [52, 54] and Subsection 4.1) show that any protocol $\mathcal{P}$ for the complete graph w.h.p. does not complete Broadcast within less than $\beta \log n$ rounds, where β is a sufficiently small constant. Combining this lower bound with our reduction result above, we get an $\Omega(\log n)$ lower bound for Consensus. This is the first known lower bound for Consensus showing a full equivalence between the complexity of Broadcast and Consensus in such models. Regarding Majority Consensus, we also obtain an $\Omega(\log n)$ lower bound for any initial bias $b = O(n^{\alpha})$, with $\alpha < 1$.

- In a similar way, we are able to prove a lower bound of $\Omega(n \log n)$ number of steps (and hence $\Omega(\log n)$ parallel time) or $\Omega(\log n)$ number of messages per node for Consensus on an asynchronous variant of the GOSSIP model, named *Population Protocols* with uniform/probabilistic scheduler, as defined in [6].

- The last application we mention here concerns the synchronous *Radio Network* model [4, 10, 25, 68]. Several optimal bounds have been obtained on the Broadcast time [10, 28, 55, 56, 58] while only few results are known for Consensus time [25, 68]. In particular, we are not aware of better lower bounds other than the trivial $\Omega(D)$ (where D denotes the diameter of the network). Then, by combining a previous lower bound in [4] on Broadcast with our reduction result, we get a new lower bound for Consensus in this model (see Subsection 4.1).

We also mention that our reduction allows us to prove that some of the above lower bounds hold also for a weaker notion of Consensus, namely δ-Almost Consensus (where δn nodes are allowed to not agree with the rest of the nodes), and even if the nodes have unbounded memory and can send/receive messages of unbounded size. We will expand on these comments in the technical sections.

1.3 Our contribution II: Consensus over noisy communication

We then turn to the study of distributed systems in which the communication links between nodes are noisy. We will consider a basic model of high-noise communication: the binary symmetric channel [60] in which each exchanged bit is flipped independently at random with probability $1/2-\varepsilon$, where $0 \leqslant \varepsilon < 1/2$, and we refer to ε as the *noise* parameter of the model.

In models such as LOCAL and CONGEST, the ability to send messages of logarithmic length (or longer) implies that, with a small overhead, one can encode the messages using error-correcting codes and simulate protocols that assume errorless communication.

In the uniform GOSSIP model with one-bit messages, however, error-correcting codes cannot be used and, indeed, whenever the number of rounds is sublinear in n, most of the pairs of nodes that ever communicate only exchange a single bit.

[7] Recall that b is the value such that we are promised that the majority opinion is held, initially, by at least $n/2+b$ nodes.

The study of fundamental distributed tasks, such as Broadcast and Majority Consensus, has been undertaken in the uniform GOSSIP model with one-bit messages and noisy links [17, 39] as a way of modeling the restricted and faulty communication that takes place in biological systems, and as a way to understand how information can travel in such systems, and how they can repair inconsistencies. Such investigation falls under the agenda of *natural algorithms*, that is, the investigation of biological phenomena from an algorithmic perspective [21, 63].

In [39], the authors prove that (binary) Broadcast and (binary) Majority Consensus can be solved in time $O(\varepsilon^{-2}\log n)$, where ε is the noise parameter, in the uniform PUSH model with one-bit messages. They also prove a matching lower bound assuming that the protocol satisfies a certain symmetry condition, which is true for the protocol of their upper bound. This has been later generalized to non-binary opinions in [40].

In the noisy uniform PULL model however, [17] proves an $\Omega(\varepsilon^{-2}n)$ time lower bound[8]. This lower bound is proved even under assumptions that strengthen the negative result, such as unique node IDs, full synchronization, and shared randomness (see Section 2.4 of [17] for more details on this point).

Such a gap between noisy uniform PUSH and PULL comes from the fact that, in the PUSH model, a node is allowed to decline to send a message, and so one can arrange a protocol in which nodes do not start communicating until they have some confidence of the value of the broadcast value. In the PULL model, instead, a called node must send a message, and so the communication becomes polluted with noise from the messages of the non-informed nodes.

What about Consensus and Majority Consensus in the noisy PULL model? Our reduction in Theorem 4.2 suggests that there could be $\Omega(\varepsilon^{-2}n)$ lower bounds for Consensus and Majority Consensus, but recall that the reduction is to the infection problem, and infection is equivalent to Broadcast only when we have errorless channels.

1.3.1 Upper bounds in noisy uniform PULL

We devise a simple and natural protocol for Consensus for the noisy uniform PULL model having convergence time $O(\varepsilon^{-2}\log n)$, w.h.p., thus exhibiting an exponential gap between Consensus and Broadcast in the noisy uniform PULL model.

The protocol runs in two phases. In the first phase, each node repeatedly collects a batch of $O(1/\varepsilon^2)$ pulled opinions and then updates its opinion to the majority opinion in the batch. This is done $O(\log n)$ times so that the first phase takes $O(\varepsilon^{-2}\log n)$ steps. In the second phase, each node collects a batch of $O(\varepsilon^{-2}\log n)$ pulled opinions and then updates its opinion to the majority opinion within the batch.

The main result of the analysis is that, w.h.p., at the end of the first phase there is an opinion that is held by at least $n/2+\Omega(n)$ nodes, and that if the initial opinions where unanimous then the initial opinion is the majority opinion after the first phase. Then, in the second phase, despite the communication errors, every node has a high probability of seeing the true phase-one majority as the empirical majority in the batch and so all nodes converge to the same valid opinion.

To analyze the first phase, we break it out into two sub-phases (this breakdown is only in the analysis, not in the protocol): in a first sub-phase of length $O(\varepsilon^{-2}\log n)$, the protocol "breaks symmetry" w.h.p. and, no matter the initial vector, reaches a configuration in which one opinion is held by $n/2+\Omega(\sqrt{n\log n})$ nodes. In the second sub-phase, also of length $O(\varepsilon^{-2}\log n)$, a configuration of bias $\Omega(\sqrt{n\log n})$ w.h.p. becomes a configuration of bias $\Omega(n)$. The analysis of this sub-phase for achieving Majority Consensus is similar to that in [39, 40]. If the initial opinion vector is unanimous,

[8] They actually proved a more general result including non-binary noisy channels.

then it is not necessary to break up the first phase into sub-phases, and one can directly see that a unanimous configuration maintains a bias $\Omega(n)$, w.h.p., for the duration of the first phase.

A consequence of our analysis is that, if the initial opinion vector has a bias $\Omega(\sqrt{n \log n})$, then the protocol converges to the majority, w.h.p. So, we get a Majority-Consensus protocol for this model under the above condition on the bias.

We also provide a Broadcast protocol that runs in $O(\varepsilon^{-2} n \log n)$ steps in the noisy uniform PULL model, nearly matching the $\Omega(\varepsilon^{-2} n)$ lower bound mentioned before. This protocol also runs in two phases. In the first phase, which lasts for order of $\varepsilon^{-2} n \log n$ steps, the informed node responds to each PULL request with the message, and other nodes respond to each PULL request with zero. After this phase, each node makes a guess of the value of the message, and with high probability the number of nodes that make a correct guess is at least $n/2 + \Omega(\sqrt{n \log n})$. The second phase is a Majority Consensus protocol applied to the first-phase guesses, which, as discussed above, takes only $O(\varepsilon^{-2} \log n)$ steps.

1.3.2 Lower bounds in noisy PULL models

We prove that any Consensus protocol that has error probability at most δ requires $\Omega\left(\varepsilon^{-2} \log \delta^{-1}\right)$ rounds (Theorem 5.1). This shows that the complexity $O(\varepsilon^{-2} \log n)$ of our protocol described above is tight for protocols that succeed w.h.p. We remark that our result holds for any version (general and uniform) of the noisy PULL model with noise parameter ε, unbounded local memory, even assuming unique node IDs.

In [39], an $\Omega\left(\varepsilon^{-2} \log \delta^{-1}\right)$ round lower bound is proved for Majority Consensus in the uniform PUSH model, for a restricted class of protocols. Their argument, roughly speaking, is that each node needs to receive a bit of information from the rest of the graph (namely, the majority value in the rest of the graph), and this bit needs to be correctly received with probability $1 - \delta$, while using a binary symmetric channel with error parameter ε. It is then a standard fact from information theory that the channel needs to be used $\Omega(\varepsilon^{-2} \log \delta^{-1})$ times.

It is not clear how to adapt this argument to the Consensus problem. Indeed, it is not true that every node receives a bit of information with high confidence from the rest of the graph (consider the protocol in which one node broadcasts its opinion), and it is not clear if there is a distribution of initial opinions such that there is a node v whose final opinion has mutual information close to 1 to the global initial opinion vector given the initial opinion of v (the natural generalization of the argument of [39]).

Instead, we show that there are two initial opinion vectors $\mathbf{x}$ and $\mathbf{y}$, a node v, and a bit b, such that the initial opinion of v is the same in $\mathbf{x}$ and $\mathbf{y}$, but the probability that v outputs b is $\leqslant \delta$ when the initial opinion vector is $\mathbf{x}$ and $\geqslant \Omega(1)$ when the initial opinion vector is $\mathbf{y}$. Thus, the rest of the graph is sending v a bit of information (whether the initial opinion vector is $\mathbf{x}$ or $\mathbf{y}$) and the communication succeeds with probability $\geqslant 1 - \delta$ when the bit has one value and with probability $\geqslant 1/3$ if the bit has the other value. Despite this asymmetry, if the communication takes place over a binary symmetric channel with error parameter ε, a calculation using KL divergence shows that the channel has to be used $\Omega(\varepsilon^{-2} \log \delta^{-1})$ times.

The $\Omega(\varepsilon^{-2} n)$ lower bound of [17] for Broadcast in the uniform PULL model applies to protocols that have constant probability of correctly performing the broadcast operation. In Lemma 5.8 we sketch a way of modifying their proof to derive an $\Omega(\varepsilon^{-2} n \log n)$ for uniform PULL protocols for Broadcast that have high probability of success, matching the $O(\varepsilon^{-2} n \log n)$ round complexity of our protocol mentioned above.

1.4 Two separations that follow from our bounds

We remark that our results establish two interesting separations.

The first, concerns the complexity gap between Consensus and Broadcast in the presence or absence of noise. Informally, we prove that, in the noiseless world, Broadcast and Consensus essentially have the same complexity in several natural models (Corollary 4.3). On the other hand, we show that there is a natural model where the presence of noise has reasonable motivations ([17, 39]), namely the noisy uniform PULL, for which the complexity of the two problems exhibits an exponential gap, since in this model Broadcast requires $\Omega(\varepsilon^{-2}n)$ rounds [17] while we prove that Consensus can be solved in $O(\varepsilon^{-2}\log n)$ time (Theorem 6.1).

The second fact regards a separation between general PULL and PUSH models as far as Consensus is concerned in the noiseless world. Indeed, if we assume unique IDs, in the general PULL model, Consensus can be easily solved in constant time: every node can copy the opinion of a prescribed node by means of a single pull operation. On the other hand, in the general PUSH model, our Broadcast-Consensus reduction shows that $\Omega(\log n)$ rounds are actually necessary for solving Consensus.

1.5 Other related work

Consensus and Broadcast are fundamental algorithmic problems which have been the subject of a huge number of studies focusing on several distributed models and different computational aspects [31, 36, 64, 67]. We here briefly discuss those results which are more relevant w.r.t. to our contribution.

Noiseless communication. Classical results prove that on the uniform PUSH or PULL models, *Rumor Spreading* (Broadcast) takes logarithmic time [42, 54, 66]. Then, a series of recent works has shown that simple uniform PULL protocols can quickly achieve Consensus, Majority Consensus and Broadcast even in the presence of a bounded number of node crashes or Byzantine nodes [15, 11, 14, 13, 32, 35, 45, 54]. The logarithmic bound is known to be tight for Broadcast [54], while, as remarked earlier, no non-trivial lower bounds are known for Consensus in any variant of the GOSSIP model. Further bounds are known for Broadcast and Majority Consensus on graphs having good expansion properties (see for instance [24, 23, 22, 47, 50, 48]). As for the general GOSSIP model with special conditions on node IDs, we mention the upper bound $O(\sqrt{\log n})$ obtained in [9] which has been then improved to $\Theta(\log\log n)$ bound in [53]. A further issue is the minimum amount of *randomness* necessary to solve Broadcast within a given time bound. In the PUSH model, this issue is investigated in [34, 33], where upper bounds and tradeoffs are given.

Noisy communication. In Subsection 1.3 we introduced and motivated the noisy communication model studied in [17, 39, 40] and adopted in this paper. Another model of noisy communication for distributed systems is the one considered in [3, 20]. Departing significantly from the model we adopt in this paper, here there is a (worst-case) adversary that can adaptively flip the bits exchanged during the execution of any protocol and the goal is to provide a robust version of the protocol under the assumption that the adversary has a limited budget on the number of bits it can change. Efficient solutions for such models typically use silent rounds [3] and error-correcting codes [3, 20]. In [38] a different task is studied in a model with noisy interactions: all n nodes of a network hold a bit and they wish to transmit to a single receiver. This line of research culminated in the $\Omega(n\log\log n)$ lower bound on the number of messages shown in [51], matching the upper bound shown in [44].

Other communication models. In [43], Consensus has been studied on a fault-free model. They provide bounds on the message complexity for deterministic protocols. As for Radio Networks, we have already discussed the results for static topologies [4]. We remark here that finding lower bounds for Consensus (and Leader Election) on a rather general model of *dynamic* Radio Networks is an open question posed in [57], where some lower bounds on the *k-Token Dissemination* Problem (a variant of Broadcast) have been derived. Another dynamic model of Radio Networks where lower bounds on Broadcast time have been derived can be found in [27]. Even though we have still not verified the applicability of our reduction result in these contexts, we believe this might be possible. Finally, we mention the works [25, 68] that consider faulty models (some with interference detectors), and provide complexity bounds on Consensus.

1.6 Roadmap of the Paper

The rest of the paper is organized as follows. In Section 3, preliminary definitions are given which will be used all over the paper. Section 4 deals with the noiseless case. In particular, we first describe the general reduction result of Broadcast to Consensus in noiseless communication models and derives the main applications of this result to some specific models, and then, in Section 4.2, we give the simple reduction of (multi)-Broadcast to Majority Consensus showing that the latter requires logarithmic time in any noiseless uniform GOSSIP model. Section 5 provides the lower bound on the noisy PULL model obtained by a reduction to an asymmetric Two-Party Protocol. In Section 6, we propose a simple majority protocol and show it solves Consensus and Majority Consensus in the noisy uniform PULL model within $O(\varepsilon^{-2}\log n)$ rounds. We also describe a protocol for Broadcast in the noisy uniform PULL running in $O(\varepsilon^{-2}n\log n)$ rounds. Finally, some technical tools are located in a separate appendix.

2 Related Work

Consensus and Broadcast are fundamental algorithmic problems which have been the subject of a huge number of studies focusing on several distributed models and different computational aspects. We here briefly discuss those results which are more relevant w.r.t. to our contribution.

2.1 GOSSIP Models

We provide a brief overview of the main related work in the (general) GOSSIP model [30, 52], by distinguishing the noiseless and noisy case. Notice that the GOSSIP model should not be confused with the class of GOSSIP *algorithms* as defined in [69].

Noiseless Communication. Classical results proving that on the uniform PUSH or PULL models, *Rumor Spreading* (Broadcast) takes logarithmic time are given in [42, 54, 66]. Then, a series of recent works has shown that simple uniform PULL protocols can quickly achieve Consensus and Broadcast even in the presence of a bounded number of node crashes or Byzantine nodes [15, 11, 14, 13, 32, 35, 45, 54]. The logarithmic bound is known to be tight for Broadcast [54], while, as remarked earlier, no non-trivial lower bounds are known for Consensus in any variant of the GOSSIP mode. Further bounds are known for Broadcast and Majority Consensus on graphs having good expansion properties (see for instance [24, 23, 22, 47, 50, 48]).

As for generalGOSSIP models with special conditions on node IDs, we mention the upper bound $O(\sqrt{\log n})$ obtained in [9] which has been then improved to $\Theta(\log\log n)$ bounds in [53].

Necessary Randomness for Broadcast. As discussed in Subsection 1.2, a natural question for the Broadcast Problem in the PUSH model is the minimum amount of *randomness* necessary to solve the problem within a given time bound. This question was investigated in works on the *Quasi-Random Rumor Spreading* Protocol [34] and the *Gate* model [33], which provided upper bounds and tradeoffs within their model.

Noisy Communication in Distributed Systems. A model of noisy communication for distributed systems is the one considered in [3, 20]. Departing significantly from the model we adopt in this paper, here there is a (worst-case) adversary that can adaptively flip the bits exchanged during the execution of any protocol and the goal is to provide a robust version of the protocol that worka in the presence of this adversary under the assumption that the latter has a limited budget on the number of bits it can change. Efficient solutions for such models typically use silent rounds [3] and error-correcting codes [3, 20].

2.2 Other Communication Models

Classic research on Consensus focused on faulty models where nodes and/or links are prone to crashes or Byzantine behaviours. This was also the framework the problem has been introduced on [31, 36, 64, 67].

In [43], Consensus has been studied on a fault-free model: the hardness of the task here is determined by the arbitrary, unknown network topology combined with the anonymity of nodes. They provide bounds on the message complexity for deterministic protocols.

As for Radio Networks, we have already discussed some application of our reduction result to this model on static topologies [4, 28]. We remark here that finding lower bounds for Consensus (and Leader Election) on a rather general model of *dynamic* Radio Networks is an open question posed in [57], where some lower bound on the *k-Token Dissemination* Problem (a variant of Broadcast), have been derived. Another dynamic model of Radio Networks where lower bounds on Broadcast time have been derived can be found in [27]. Even though we have still not verify the applicability of our reduction result in these contexts, we believe this might be possible.

Finally, we mention the works [25, 68] that consider faulty models (some with interference detectors), and provide complexity bounds on Consensus.

3 Preliminaries

3.1 Distributed systems and communication models

Let $\mathcal{S}$ be a distributed system formed by a set V of n nodes which mutually interact by exchanging messages over a connected graph $G = (V, E)$ according to a fixed communication model $\mathcal{M}$. The definition of $\mathcal{M}$ includes all features of node communications including, for instance, synchronicity or not and the presence of link faults.

A configuration $\mathbf{c}$ of a distributed system $\mathcal{S}$ is the description of the states of all the nodes of $\mathcal{S}$ at a given time. If we execute a protocol $\mathcal{P}$ for $\mathcal{S}$, the random configuration the system lies in a generic time is denoted as $\mathbf{C}$.

When, in the GOSSIP, PUSH or PULL models, at each round the communication is established with a random neighbor chosen independently and u.a.r., we call the communication model *uniform*. In order to remark the difference with the uniform case, we call the communication model *general* when nodes are equipped with unique IDs which are known to all neighbors, and each node can choose the identity of the neighbor with which to communicate (possibly in a random way).

Finally, we distinguish two main communication scenarios. In the *noiseless* models, every transmitted message on a link of the graph is received safely, without any error.

In the presence of *communication noise*, instead, each bit of any transmitted message is flipped independently at random with probability $1/2 - \varepsilon$, where $\varepsilon \in (0, 1/2]$ is the *noisy* parameter. Then, in the sequel, the version of each model $\mathcal{M}$, in which the presence of communication noise above is introduced, will be shortly denote as *noisy* $\mathcal{M}$. Notice that, in order to capture the role of noise in systems where standard error correcting codes techniques are not feasible, we consider models where each single point-to-point transmission consists of one bit only. In this way, we easily have that, in GOSSIP models, the bit-communication and the convergence time of a Protocol are strongly related.

3.2 Consensus and Broadcast in distributed systems

Several versions of Consensus have been considered in the literature [31, 36, 64, 67]. Since our interest is mainly focused on models having strong constraints on communication (random, limited, and noisy), we adopt some weaker, probabilistic variants of consensus, studied in [6, 13, 35], that well captures this focus.

Formally, we say a protocol guarantees (binary) *Consensus* if, starting from any initial node opinion vector $\mathbf{x} \in \{0, 1\}^n$, the system reaches w.h.p. a configuration where every node has the same opinion (*Agreement*) and this opinion is *valid*, i.e., it was supported by at least one node at the starting time. Moreover, once the system reaches this *consensus* configuration, it is required to stay there for any arbitrarily-large polynomial number of rounds, w.h.p. This *Stability* property somewhat replaces the *Termination* property required by other, classic notions of consensus introduced in stronger distributed models [59].

In order to define Majority Consensus, we need to introduce the notion of bias of an (initial) opinion vector. Given any vector $\mathbf{x} \in \{0, 1\}^n$, the *bias* b associated to $\mathbf{x}$ is the difference between the number of nodes supporting opinion 1 and the number of those supporting 0 in $\mathbf{x}$. The state of each node clearly depends on the specific protocol, however, we can always assume it contains the current opinion of the node, so, we can also define the bias of a configuration $s(\mathbf{c})$. When the opinion vector (or the configuration) is clear from the context, we will just use the term s. The *majority opinion* in a given vector (configuration) $\mathbf{x}$ is the one having the largest number of nodes supporting it. With the term *majority*, we will indicate the number of nodes supporting the majority opinion. A protocol guarantees (binary) *Majority Consensus* if, starting from any initial opinion vector $\mathbf{x} \in \{0, 1\}^n$ with bias $s(\mathbf{x}) > 0$, the system reaches w.h.p. a configuration where every node has the same opinion and this opinion is the initial majority one. Moreover, we require the same stability property we define for Consensus.

Both the notions of Consensus and Majority Consensus above can be further relaxed to those of *δ-Almost Consensus* and *δ-Almost Majority Consensus*, respectively. According to such weaker notions, we allow the system to converge to an almost-consensus regime where *δn outliers* may have a different opinion from the rest of the nodes. In this case, the fraction of outliers is a performance parameter of the protocol we will specify in the statements of our results. Even in this weaker notion, we require the same property of Stability but we remark that the subset of outliers may change during the almost-consensus regime. We emphasize that all lower bounds we obtain in this paper holds for such weaker versions of Almost Consensus, while the upper bound in Section 6 refers to (full) Consensus.

As discussed in the introduction, our work also deals with the (single-source) Broadcast task (a.k.a. *Rumor Spreading*). Given any source node $s \in V$ having an initial message `msg`, a *Broadcast Protocol* $\mathcal{P}$ is a protocol that, after a finite number of rounds, makes every node in V receive a copy

of (and, thus, be *informed* about) `msg`, w.h.p.[9]. Similarly to Consensus, we also consider a weaker version of Broadcast where the final number of informed nodes is required to be at most $(1-\delta)n$, w.h.p.

4 Noiseless Communication: Broadcast vs Consensus

In this section we provide our first main result (Theorem 4.2) which establishes a strong connection between (Almost) Consensus and a suitable, weaker version of (Almost) Broadcast in the noiseless-communication framework. We first describe the result in a rather general setting and then we show its consequences, namely some lower bounds for the (Almost) Consensus problem in specific communication models.

Notice that in Section 4.2, we complement Theorem 4.2 with an analogous lower bound for Almost-Majority Consensus with sub-linear initial bias (see Theorem 4.10).

Let $\mathcal{S}$ be a distributed system formed by a set V of n nodes which mutually interact over a support graph $G = (V, E)$ according to a fixed communication model $\mathcal{M}$. The crucial assumption we make in this section on $\mathcal{M}$ is the absence of communication noise (i.e. message corruption): whenever a node v transmits a message on one of its links, either this message is received with no change or it is fully lost and, in the latter case, both sender and receiver cannot get any information from the state of the corresponding port (no fault detection).

Under the above noiseless framework, the next theorem essentially states that (Almost) Consensus cannot be "easier" than (Almost) Broadcast. As we similarly show in Section 5, much of the technical difficulty in reasoning on the valid-consensus problem arises from the high level of freedom nodes have in agreeing on the final consensus value, since both values are valid solution as long as not all nodes start with inputs that are already identical.

In order to state the reduction, we need to introduce a slightly-different variant of Broadcast where, essentially, it is (only) required that *some* information from the source is spread on the network. Formally

Definition 4.1. *A protocol $\mathcal{P}$ solves the γ-*Infection *problem w.r.t. a source node s if it* infects *at least γn nodes, where we define a node* infected *recursively as follows: initially only s is infected; a node v becomes infected whenever it receives* any *message from an infected node.*

Notice that a protocol $\mathcal{P}$ solving the γ-Infection problem w.r.t. a source node s can be easily turned into a protocol for broadcasting a message `msg` from s to at least γn nodes. Indeed, we give the message `msg` to the source node s, and we simulate $\mathcal{P}$. Every time an infected node sends a message, it appends `msg` to it. Clearly, the size of each message in $\mathcal{P}'$ is increased by the size of `msg`.

The next theorem is the main result of the section. Informally, it states that any protocol for Consensus actually solves the Infection problem (when initialized with a certain opinion vector) in a weak sense: the infection is w.r.t. a source that depends on the consensus protocol in a (possibly) uncontrolled manner; and (ii) the success probability of the infection is quite low. Another intuitive way to look at the result is as follows: any consensus protocol needs to solve the Infection problem from a certain source node when it starts from a certain initial opinion vector.

Theorem 4.2. *Let $\mathcal{P}$ be a protocol reaching δ-Almost Consensus with probability at least $1-o(1/n)$. Then, a source node s and an initial opinion vector $\mathbf{x}$ exist such that $\mathcal{P}$, starting from $\mathbf{x}$, solves the $(1-2\delta)$-Infection problem w.r.t. s with probability at least $(1-o(1))/n$.*

[9]The success probability of the protocol is here defined over both the random choices (if any) of the communication mechanism and the ones of $\mathcal{P}$.

Proof. Let $V = \{1, 2, \ldots, n\}$ be an arbitrary ordering of the vertices and, for any $k = 0, \ldots, n$, let $\mathbf{x}_k$ be the initial opinion vector in which the first k nodes start with 0 and the other $n-k$ with 1. Moreover, let Z_k be the indicator random variable taking value 1 when $\mathcal{P}$, starting from $\mathbf{x}_k$, reaches δ-Almost Consensus on value 1, otherwise Z_k takes value 0 (note that $Z_k = 0$ also when the protocol fails to reach δ-Almost Consensus).

Since the protocol converges to an almost consensus with probability at least $1 - o(1/n)$, it must hold that

$$\mathbf{E}[Z_0] \geqslant 1 - o\left(\frac{1}{n}\right) \quad \text{and} \quad \mathbf{E}[Z_n] \leqslant o\left(\frac{1}{n}\right),$$

and

$$\sum_{i=1}^{n} \left(\mathbf{E}[Z_{i-1}] - \mathbf{E}[Z_i]\right) = \mathbf{E}[Z_0] - \mathbf{E}[Z_n] \geqslant 1 - o\left(\frac{1}{n}\right).$$

Hence, a node k^* exists such that

$$\mathbf{E}[Z_{k^*-1}] - \mathbf{E}[Z_{k^*}] \geqslant \frac{1 - o(1)}{n}. \tag{1}$$

We now show that, when $\mathcal{P}$ starts from opinion vector $\mathbf{x}_{k^*}$, $\mathcal{P}$ is (also) solving the $(1-2\delta)$-Infection problem w.r.t. source node k^*.

First observe that $\mathbf{E}[Z_{k^*-1}] - \mathbf{E}[Z_{k^*}] \leqslant \mathbf{P}(Z_{k^*-1} = 1 \ \wedge \ Z_{k^*} = 0)$ and let us name I_{k^*} the set of nodes infected by node k^*, starting from opinion vector $\mathbf{x}_{k^*}$. We will prove that, if $Z_{k^*-1} = 1$ and $Z_{k^*} = 0$ then either $|I_{k^*}| \geqslant (1-2\delta)n$ or $\mathcal{P}$ fails to reach δ-Almost Consensus starting from $\mathbf{x}_{k^*}$. More formally, if we name $\mathcal{F}$ the event

$$\mathcal{F} = \{\mathcal{P} \text{ fails to reach } \delta\text{-Almost Consensus from } \mathbf{x}_{k^*}\}$$

we claim that

$$\{Z_{k^*-1} = 1 \ \wedge \ Z_{k^*} = 0\} \implies \{\text{“}|I_{k^*}| \geqslant (1-2\delta)n\text{”} \ \vee \ \mathcal{F}\}. \tag{2}$$

In order to obtain (2) we equivalently show that, if $Z_{k^*-1} = 1$, $Z_{k^*} = 0$, and $\mathcal{P}$ does not fail starting from $\mathbf{x}_{k^*}$, it must hold that $|I_{k^*}| \geqslant (1-2\delta)n$. First observe that, if $Z_{k^*-1} = 1$ then $\mathcal{P}$ reaches almost consensus on 1 starting from $\mathbf{x}_{k^*-1}$, therefore the number of nodes which output 1 is at least $(1-\delta)n$. Moreover, if $Z_{k^*} = 0$, and $\mathcal{P}$ does not fail starting from $\mathbf{x}_{k^*}$, then the number of nodes which output 1 is at most δn. Thus, moving the system from input $\mathbf{x}_{k^*-1}$ to input $\mathbf{x}_{k^*}$, the number of nodes switching their opinion from 1 to 0 must be at least $(1-2\delta)n$. Finally observe that, since $\mathbf{x}_{k^*-1}$ and $\mathbf{x}_{k^*}$ differ only at node k^*, the nodes that change their output value must be infected by k^* (according to Definition 4.1). Hence, $|I_{k^*}| \geqslant (1-2\delta)n$, which implies that, when starting from $\mathbf{x}_{k^*}$, $\mathcal{P}$ is also solving the $(1-2\delta)$-Infection problem w.r.t. node k^*.

To conclude the proof, it remains to bound (from below) the probability with which this infection happens. From (1), (2) and the union bound, it follows that

$$\begin{aligned}\frac{1-o(1)}{n} &\leqslant \mathbf{E}[Z_{k^*-1}] - \mathbf{E}[Z_{k^*}] \leqslant \mathbf{P}(Z_{k^*-1} = 1 \wedge Z_{k^*} = 0) \\ &\leqslant \mathbf{P}(\text{“}|I_{k^*}| \geqslant (1-2\delta)n\text{”} \vee \mathcal{F}) \leqslant \mathbf{P}(|I_{k^*}| \geqslant (1-2\delta)n) + o\left(\frac{1}{n}\right).\end{aligned}$$

Thus,

$$\mathbf{P}(|I_{k^*}| \geqslant (1-2\delta)n) \geqslant \frac{1-o(1)}{n}.$$

□

Remark. Observe that factor 2 in the parameter $\gamma = (1 - 2\delta)$ in the statement of Theorem 4.2 is tight. Indeed, consider a protocol in which each node outputs its input value: such protocol is trivially a ($\delta = \frac{1}{2}$)-Almost Consensus protocol, while the number of infected node has size at most 1.

4.1 Specific lower bounds for Consensus

Theorem 4.2 allows us to derive lower bounds for Consensus for specific communication models and resources by using lower bounds for the Infection problem. In fact, by simply restating Theorem 4.2, we obtain the following.

Corollary 4.3. *Let $\mathcal{T}$ be any fixed resource (e.g. time, work, bit-communication, etc.) defined on a distributed system $\mathcal{S}$, and suppose that any protocol, which uses at most τ^{B} units of $\mathcal{T}$, fails to solve the $(1-2\delta)$-Infection problem w.h.p. from any source node. Then, any protocol on this model reaching δ-Almost Consensus w.h.p., must use at least τ^{B} units of $\mathcal{T}$.*

We now apply the above corollary in different settings. Unless differently stated, all results in this subsection refers to the complete graph of n nodes.

The GOSSIP model. The first lower bound is on the Consensus time for the uniform GOSSIP model (and hence for the uniform PUSH and the uniform PULL as well). We first state a simple technical result which will be handy also in the proof of Corollary 4.11. This is a well-known result in the community, however, for the sake of completeness, we give a self-contained proof.

Lemma 4.4. *Consider the uniform* GOSSIP *model and fix any constants α and γ such that $0 < \alpha, \gamma < 1$. Then, there is a sufficiently small constant $\beta_{\alpha,\gamma} > 0$ such that, starting from any subset of infected nodes of size $O(n^{\alpha})$, the γ-*Infection *problem requires at least $\beta \log n$ rounds, w.h.p.*

Proof. The proof shows that, starting from any subset of $O(n^{\alpha})$ infected nodes, the set of infected nodes grows by at most a constant factor at each round,w.h.p. The latter fact easily implies that, if β is sufficiently small, then, within $\beta \log n$ rounds, at most γn nodes are infected w.h.p.

In order to show that the set of infected nodes increases by at most a constant factor, let V be the set of nodes and $I^{(t)}$ the set of infected nodes at time t. At each round the number of (bidirectional) communication-edges between $I^{(t)}$ and the uninfected nodes $V/I^{(t)}$ is $\frac{1}{n}(n - |I^{(t)}|)|I^{(t)}| \leqslant |I^{(t)}|$. By a simple application of Chernoff bounds, it follows that the growth of number of infected nodes is bounded by a constant multiplicative factor η, w.h.p. Since $|I^{(0)}| \leqslant n^{\alpha}$ and w.h.p. $|I^{(t+1)}| \leqslant \eta |I^{(t)}|$, then $|I^{(t)}| \leqslant \eta^t \cdot n^{\alpha}$ and the latter is smaller than δn as long as $t \leqslant (1-\alpha)\log_{\eta}(\delta n)$.

Notice that to get a concentration result over all the process, we just observe that if every event in some family $\{A_i\}_i$ holds w.h.p., then, using the union bound, the intersection of any polylogarithmic number of such events holds w.h.p.

□

We can combine the above lemma (in the case where just one source node is initially infected), with Corollary 4.3, and get the following.

Corollary 4.5. *Consider the uniform* GOSSIP *model and fix any constant δ such that $0 \leqslant \delta < 0$. Then, any protocol reaching δ-Almost Consensus w.h.p. requires $\Omega(\log n)$ communication rounds.*

Next, consider the general PUSH model and fix any (Broadcast) protocol: then, starting from any source node, the set of infected nodes grows by a factor at most 2 at each round. Hence, Corollary 4.3 also implies an $\Omega(\log n)$ lower bound for the general PUSH model.

Corollary 4.6. *Let $0 \leqslant \delta < 0$ be any constant. Then any protocol reaching δ-Almost Consensus w.h.p. on the general* PUSH *model, requires $\Omega(\log n)$ communication rounds, even when the nodes have unique identifiers.*

Remark 4.7. *Corollary 4.6 should be contrasted with the fact that in the general* PULL *model, assuming unique identifiers, (valid) Consensus can be solved in a single round, by having all nodes adopt the input value of a specific node*[10].

Population Protocols. Another interesting model where we can apply our reduction is the Population Protocol one with uniform/probabilistic scheduler as defined in [6, 8]. Broadcast in this model has essentially the same complexity of that in the asynchronous uniform GOSSIP model: in particolar, a similar result to that in Lemma 4.4 holds for Broadcast time (see for instance [49]): any Population Protocol with uniform/probabilistic scheduler on the complete graph cannot infect more than $(1-\delta)n$ nodes w.h.p. within $\beta n \log n$ number of rounds (and hence $\beta \log n$ parallel time) or $\beta \log n$ number of messages per node. Hence, by using Corollary 4.3, we can state the following result.

Corollary 4.8. *Let $0 \leqslant \delta < 0$ be any constant. Then any Population Protocol (with uniform/probabilistic scheduler) reaching δ-Almost Consensus requires $\Omega(n \log n)$ number of steps (and hence $\Omega(\log n)$ parallel time) and $\Omega(\log n)$ number of messages per node.*

Radio Networks. In the synchronous *Radio Network* model [4, 10, 25, 68], the presence of message collisions on the (unique) shared radio frequency is modelled by the following communication paradigm: a node can receive a message at a given round t if and only if exactly one of its neighbors transmits at round $t-1$. We consider the model setting with no *collision detection*, i.e., the nodes of the graph are not able to get any information when a collision occurs.

In [4] the authors derive a lower bound $\Omega(\log^2 n)$ on the radio-broadcast time in networks of constant diameter. In particular, their proof relies on a construction of a family of graphs of n nodes having diameter 2 where every protocol that runs for no more than $\beta \log^2 n$ rounds (where $\beta > 0$ is a sufficiently small constant) cannot infect at least one node w.h.p. (in fact, with probability 1). We observe that the proof can be adapted in order to hold for any choice of the source and when every node knows the graph topology, so for any choice of the initial configuration. This implies that their lower bound also applies on the time required by any protocol to γ-infect all nodes (with $\gamma = 1$) according to Definition 4.1. Then, from Theorem 4.2 we get a lower bound on the Consensus time in Radio Networks.

Corollary 4.9. *Consider the Radio Network model. There is a family of constant-diameter graphs, where any (randomized) protocol reaching Consensus requires $\Omega(\log^2 n)$ time, w.h.p.*

4.2 A lower bound for δ-Almost Majority Consensus

The conditions required by Majority Consensus are much stronger than the validity one and make the relationship between this task and the γ-Infection problem with multiple source nodes rather simple to derive.

[10] On the other hand, if we assume that nodes do not initially share unique identifiers (for example, in the PULL Model with numbered ports), it is easy to see that the broadcast problem cannot be solved w.h.p. in $o(\log n)$ time, since the number of nodes from which a given node v can receive any information from, increases by at most a factor 2 at each round.

Lemma 4.10. *Let $\mathcal{T}$ be any fixed resource defined on a distributed system $\mathcal{S}$ and suppose there is no Infection protocol that, starting from any subset of n^α nodes with $\alpha < 1$, can inform at least $(1-\delta)n$ nodes by using at most τ^{B} units of $\mathcal{T}$, w.h.p. Then, any protocol $\mathcal{P}$ on this model, reaching* δ-Almost Majority Consensus *w.h.p., must use at least τ^{B} units of $\mathcal{T}$.*

Proof. W.l.o.g., let $n-b$ be an even number where b is the initial bias. Consider an arbitrary labeling of the nodes $v_1, \ldots, v_n$ and two initial input vectors $\mathbf{x}_0$ and $\mathbf{x}_1$ such that

$$\mathbf{x}_i = \begin{cases} v_j = 0 & \text{for } j \in \left\{1, \ldots, \frac{n-s}{2}\right\}, \\ v_j = 1 & \text{for } j \in \left\{\frac{n-s}{2}+1, \ldots, n-s\right\}, \\ v_j = i & \text{for } j \in \{n-s+1, \ldots, n\}. \end{cases} \tag{3}$$

In order for a node v to converge to the correct majority opinion, it is necessary that it is able to distinguish between configuration $\mathbf{x}_0$ and $\mathbf{x}_1$. Since $\mathbf{x}_0$ and $\mathbf{x}_1$ are identical for all nodes $v_1, \ldots, v_{n-s}$, it is then necessary for v to be infected by each of the *source* nodes $v_{n-s+1}, \ldots, v_n$ and the proof is completed.

□

A specific lower bound for Majority Consensus. The above lemma allows us to obtain a logarithmic lower bound on the convergence time required by any almost Majority-Consensus protocol on uniform GOSSIP (and, hence, on uniform PULL and uniform PUSH). Notice that the bound holds even for protocols achieving the task with constant probability only.

Corollary 4.11. *Consider* δ-Almost Majority Consensus *in uniform* GOSSIP *starting with initial bias $b \leqslant n^\alpha$, for any positive constant $\alpha < 1$. Then, any protocol that solves the task above with probability at least $2/3$ requires $\Omega_{(1-\alpha)}(\log n)$ rounds.*

Proof. The proof follows from Lemma 4.4 and Lemma 4.10, where $\mathcal{T}$ is the number of rounds in the uniform GOSSIP model. □

5 Lower Bounds in the Noisy Model

The main result of this section is the following lower bound for the δ-Almost Consensus Problem in the noisy general PULL model (and hence in noisy uniform PULL as well).

Theorem 5.1. *Let δ be any real such that $0 < \delta < 1/8$ and consider any protocol $\mathcal{P}$ for the noisy general* PULL *model with noise parameter ε. If $\mathcal{P}$ solves δ-Almost Consensus with probability at least $1-\delta$, then it requires at least $t = \Omega(\varepsilon^{-2} \log \delta^{-1})$ rounds*[11].

Proof Outline. W.l.o.g. we assume that during the execution of $\mathcal{P}$, every node pulls another node at each round. Indeed, if not true, we can simply consider a protocol $\mathcal{P}'$ where this property holds but the extra messages are ignored, obtaining an equivalent protocol.

Suppose that we have a protocol that solves the δ-Almost Consensus with probability at least $\geqslant 1-\delta$. The definition of δ-Almost Consensus implies that each node has an initial one-bit input and produces a one-bit output and

- If the initial opinion vector is all-zeroes, then with probability $\geqslant 1-\delta$ all nodes but at most δn output zero.

[11] We notice the double role parameter δ has in this statement.

- If the initial opinion vector is all-ones, then with probability $\geqslant 1-\delta$ all nodes but at most δn output one.
- For every initial opinion vector, with probability $\geqslant 1-\delta$ all nodes but at most δn agree.

From the above constraints, we derive the existence of (at least) one node v^* which must get from the rest of the system "enough information" in order to decide its output. More in details, in Lemma 5.3, we show that any Almost-Consensus protocol implies a solution to a two-party communication problem over a noisy communication channel, where one of the two parties represents node v^* and it needs to act differently according to the information owned by the other party, namely the rest of the graph.

In Lemma 5.7, we then show that, in the two-party problem above, v^* has to receive at least $\Omega(\varepsilon^{-2}\log\delta^{-1})$ bits (and thus, according to the noisy PULL, performs at least the same number of rounds) in order to recover the information owned by the rest of the graph with a sufficiently large probability, and thus deciding its output. This implies the desired bound. □

5.1 Reduction to the Two-Party Protocol

As outlined in the proof of Theorem 5.1, we start by showing that, if we have a (valid) Almost Consensus protocol, we can convert it into a two-party communication protocol between party A and party B with certain properties. More formally, we give the following definition.

Definition 5.2. *A (t,δ,ε)-Two-Party Protocol is a two party noisy communication protocol between parties A and B such that*

i) B starts with a bit b and at the end A outputs one bit,

ii) A receives t messages, each one of one bit,

iii) Each bit of communication passes through a binary symmetric channel that flips the bit with probability $\frac{1}{2}-\varepsilon$,

iv) If $b=0$ then A outputs 0 with probability $\geqslant 1-O(\delta)$,

v) If $b=1$ then A outputs 0 with probability $\leqslant \frac{3}{4}+O(\delta)$.

We can now state the formal reduction result.

Lemma 5.3. *Let $\mathcal{P}$ be a protocol that solves δ-Almost Consensus problem in t rounds with probability at least $1-\delta$ in the noisy general* PULL *model for some $0<\delta<1/8$. Then, there exists a (t,δ,ε)-Two-Party Protocol, where ε is the noisy parameter.*

Proof. As defined in the previous section, let $\mathbf{x}_j$ be the initial opinion vector such that the first j nodes initially support opinion 0 and the others support opinion 1. Let "$\mathcal{P}\to x$" be the event "$\mathcal{P}$ converges to a consensus where all nodes, but at most δn, output opinion x" where $x\in\{0,1\}$. By hypotheses of the lemma we have that

- During the execution of $\mathcal{P}$, each node v exchanges at most t one-bit messages;
- $\mathbf{P}\left(\mathcal{P}\to 0|\mathbf{x}_n\right)\geqslant 1-\delta$;
- $\mathbf{P}\left(\mathcal{P}\to 1|\mathbf{x}_0\right)\geqslant 1-\delta$;

- For any initial opinion vector $\mathbf{x}_j$, the probability that $\mathcal{P}$ reaches an almost consensus is at least $1-\delta$, namely $\mathbf{P}\left(\mathcal{P} \to 0 \vee \mathcal{P} \to 1 | \mathbf{x}_j\right) \geqslant 1-\delta$.

Thanks to the agreement property we know that $\mathbf{P}(\mathcal{P} \to 0 \vee \mathcal{P} \to 1 \,|\, \mathbf{x}_{\frac{n}{2}}) \geqslant 1-\delta$. Since there are only two possible opinions, then $\mathbf{P}(\mathcal{P} \to 0 \,|\, \mathbf{x}_{\frac{n}{2}}) \geqslant \frac{1-\delta}{2}$ or $\mathbf{P}(\mathcal{P} \to 1 \,|\, \mathbf{x}_{\frac{n}{2}}) \geqslant \frac{1-\delta}{2}$. W.l.o.g. we assume that it holds $\mathbf{P}(\mathcal{P} \to 1 \,|\, \mathbf{x}_{\frac{n}{2}}) \geqslant \frac{1-\delta}{2}$, indeed if not true, we can simply rename opinion 0 with opinion 1. Now we leverage on this property in order to show that, starting from $\mathbf{x}_{\frac{n}{2}}$, a large fraction of nodes have a constant probability to output opinion 1. Let u be any node, we define "$u \to x$" be the event "u outputs opinion x in $\mathcal{P}$", where $x \in \{0,1\}$.

Fact 5.4. *If $\mathbf{P}(\mathcal{P} \to 1 \,|\, \mathbf{x}_{\frac{n}{2}}) \geqslant \frac{1-\delta}{2}$ then a node subset S with $|S| \geqslant (1-2\delta)n$ exists such that for any $u \in S$ it holds $\mathbf{P}(u \to 1 | \mathbf{x}_{\frac{n}{2}}) \geqslant \frac{1}{4} - \frac{\delta}{4}$.*

Proof. Let V' be the set of nodes v such that $\mathbf{P}(v \to 0 \,|\, \mathcal{P} \to 1, \mathbf{x}_{\frac{n}{2}}) \geqslant \frac{1}{2}$, and let $S = V \setminus V'$. Since the expectation of the number Z of nodes that output 0 conditioned to the event "$\mathcal{P} \to 1$" is at most δn, the size of V' is at most $2\delta n$. Indeed,

$$\begin{aligned}
\delta n \geqslant \mathbf{E}\left[Z \,|\, \mathcal{P} \to 1, \mathbf{x}_{\frac{n}{2}}\right] &= \sum_{v \in V} \mathbf{P}\left(v \to 0 \,|\, \mathcal{P} \to 1, \mathbf{x}_{\frac{n}{2}}\right) \\
&\geqslant \sum_{v \in V'} \mathbf{P}\left(v \to 0 \,|\, \mathcal{P} \to 1, \mathbf{x}_{\frac{n}{2}}\right) \geqslant \frac{1}{2}|V'|.
\end{aligned}$$

This implies that $|S| \geqslant (1-2\delta)n$. To conclude the proof, we observe that, for any $u \in S$, we have

$$\begin{aligned}
\mathbf{P}\left(u \to 1 \,|\, \mathbf{x}_{\frac{n}{2}}\right) &\geqslant \mathbf{P}\left(u \to 1 \wedge \mathcal{P} \to 1 \,|\, \mathbf{x}_{\frac{n}{2}}\right) \\
&= \mathbf{P}\left(u \to 1 \,|\, \mathcal{P} \to 1, \mathbf{x}_{\frac{n}{2}}\right) \cdot \mathbf{P}\left(\mathcal{P} \to 1 \,|\, \mathbf{x}_{\frac{n}{2}}\right) \\
&\geqslant \frac{1}{2} \cdot \left(\frac{1}{2} - \frac{\delta}{2}\right) = \frac{1}{4} - \frac{\delta}{4}.
\end{aligned}$$

□

We now consider the initial opinion vector $\mathbf{x}_n$, where all the nodes support opinion 0. Remind that $\mathbf{P}\left(\mathcal{P} \to 0 \,|\, \mathbf{x}_n\right) \geqslant 1-\delta$. Using a similar argument of Fact 5.4, we can prove the following

Fact 5.5. *If $\mathbf{P}\left(\mathcal{P} \to 0 \,|\, \mathbf{x}_n\right) \geqslant 1-\delta$ then a node subset H with $|H| \geqslant \frac{3}{4}n$ exists such that for any $u \in H$, $\mathbf{P}\left(u \to 0 \,|\, \mathbf{x}_n\right) \geqslant 1-5\delta$.*

Proof. Let V' be the set of nodes v such that $\mathbf{P}\left(v \to 1 \,|\, \mathcal{P} \to 0, \mathbf{x}_n\right) \geqslant 4\delta$, and set $H = V \setminus V'$. Since the expected number of nodes that output opinion 1 conditioned to the event "$\mathcal{P} \to 0$" is at most δn, the size of V' is at most $n/4$, and hence $|H| \geqslant \frac{3}{4}n$.

To conclude the proof, observe that, for any $u \in H$,

$$\begin{aligned}
\mathbf{P}\left(u \to 0 | \mathbf{x}_n\right) &\geqslant \mathbf{P}\left(u \to 0 \wedge \mathcal{P} \to 0 | \mathbf{x}_n\right) \\
&= \mathbf{P}\left(u \to 0 | \mathcal{P} \to 0, \mathbf{x}_n\right) \cdot \mathbf{P}\left(\mathcal{P} \to 0 | \mathbf{x}_n\right) \\
&\geqslant (1-4\delta) \cdot (1-\delta) \geqslant 1-5\delta.
\end{aligned}$$

□

By combining Facts 5.4 and 5.5, we obtain the following

Fact 5.6. *Let $0 < \delta < 1/8$. At least one node v^* exists such that:*
(i) its initial opinion is 0 in both $\mathbf{x}_{\frac{n}{2}}$ and $\mathbf{x}_n$, and
(ii) $\mathbf{P}\left(v^* \to 1|\mathbf{x}_{\frac{n}{2}}\right) \geqslant \frac{1}{4} - \frac{\delta}{4}$ and $\mathbf{P}\left(v^* \to 1|\mathbf{x}_n\right) \leqslant 5\delta$.

Proof. From Facts 5.4 and 5.5, if $\delta < \frac{1}{8}$ then $|S| > \frac{3}{4}n$, and thus $|S \cap H| > \frac{1}{2}n$. Since the nodes having initial opinion 0 in both vectors $\mathbf{x}_{\frac{n}{2}}$ and $\mathbf{x}_n$ are exactly $\frac{n}{2}$, it must exist at least one node having Properties (i) and (ii). □

We now realize a (t, δ, ε)-Two-Party Protocol between parties A and B. If B has input 0, then A simulates v^* with initial opinion 0 and B simulates all other nodes as if they had all initial opinion 0. If B has input 1, then A simulates v^* with initial opinion 0 and B simulates all other nodes as if they had as an initial opinion vector of $\frac{n}{2}$ ones and $\frac{n}{2} - 1$ zeroes. In the simulation, A and B need to communicate (via the binary symmetric channel) only when v^* sends or receives messages in $\mathcal{P}$. At the end, A will output 0 or 1, depending on the outcome of the Almost Consensus protocol $\mathcal{P}$.

Note that at each round of simulation of $\mathcal{P}$, v^* receives exactly[12] 1 bit and no other information is available to it. Hence, in the resulting Two-Party Protocol, the only information obtained by A is the bit received in that round from B (corresponding to the rest of the graph).

Thanks to Fact 5.6, for any $\delta < \frac{1}{8}$, if B has input 0, with probability at least $1 - 5\delta$, A will output 0. On the other hand, if B has input 1, with probability at least $\frac{1}{4} - \frac{\delta}{4}$, A will output 1. □

5.2 Lower bound for the Two-Party Protocol

Lemma 5.7. *Any (t, δ, ε)-Two-Party Protocol requires a number of rounds t such that $t = \Omega(\varepsilon^{-2} \log \delta^{-1})$.*

Proof. In any interaction between A and B of t rounds, we name *view* the sequence $\mathbf{w}$ of all 1-bit messages received by A during the interaction. Notice that this sequence determines the sequence of messages sent by A and the final output of A. Let $\mathbf{X}$ be the random variable that represents the (random) view when B's input is 1 and let $\mathbf{Y}$ be the random variable that represents the (random) view when B's input is 0.

Recall that the Kullback–Leibler divergence between $\mathbf{X}$ and $\mathbf{Y}$ is defined as

$$D(\mathbf{X}||\mathbf{Y}) := \sum_{\mathbf{w} \in \{0,1\}^t} \mathbf{P}\left(\mathbf{X} = \mathbf{w}\right) \log \frac{\mathbf{P}\left(\mathbf{X} = \mathbf{w}\right)}{\mathbf{P}\left(\mathbf{Y} = \mathbf{w}\right)},$$

We will prove the following two facts which easily imply the claim of the lemma:

$$D(\mathbf{X}||\mathbf{Y}) \geqslant \Omega\left(\log \frac{1}{\delta}\right), \tag{4}$$

$$D(\mathbf{X}||\mathbf{Y}) \leqslant O\left(t\varepsilon^2\right). \tag{5}$$

To prove (4) we use the *data-processing inequality* which (in particular) states that for every (possibly random) function $f(\cdot)$ we have

$$D(\mathbf{X}||\mathbf{Y}) \geqslant D(f(\mathbf{X})||f(\mathbf{Y})). \tag{6}$$

For a view $\mathbf{w}$, define $f(\mathbf{w})$ to be the output of A for that view. Then $f(\mathbf{X})$ and $f(\mathbf{Y})$ are 0/1 random variables. If we call $p := \mathbf{P}\left(f(\mathbf{X}) = 1\right)$ and $q := \mathbf{P}\left(f(\mathbf{Y}) = 1\right)$ we get

$$D(f(\mathbf{X})||f(\mathbf{Y})) = p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q},$$

[12] At the beginning of the proof we assumed w.l.o.g. that at each round each node pulls another node.

and recalling that $p \geqslant \Omega(1)$ and $q \leqslant O(\delta)$ we get

$$D(f(\mathbf{X})||f(\mathbf{Y})) \geqslant \Omega\left(\log \frac{1}{\delta}\right), \tag{7}$$

which proves (4).

To prove (5) we use the *chain rule*. If we have two pairs of jointly distributed random variables (X, X') and (Y, Y'), then the conditional KL divergence is defined as

$$D(X'|X \mid\mid Y'|Y) = \mathbb{E}_{x\sim X, y\sim Y} D((X'|X = x) \mid\mid (Y'|Y = y)).$$

Let $\oplus$ be the operation that denotes the concatenation between two sequence of random variables, the chain rule is

$$D(X \oplus X'||YY') = D(X||Y) + D((X'|X) \mid\mid (Y'|Y))$$

Since any view has length t, we write $\mathbf{X} = X_1 \oplus \ldots \oplus X_t$ and $\mathbf{Y} = Y_1 \oplus \ldots \oplus Y_t$. Then we can write the KL divergence of $\mathbf{X}$ and $\mathbf{Y}$ as

$$D(\mathbf{X}||\mathbf{Y}) = D(X_1||Y_1) + D((X_2|X_1) \mid\mid (Y_2|Y_1)) + \cdots + D((X_t|X_1 \cdots X_{t-1}) \mid\mid (Y_t|Y_1 \cdots Y_{t-1})).$$

For each i we can easily compute $D((X_i|X_1 \cdots X_{i-1}) \mid\mid (Y_i|Y_1 \cdots Y_{i-1}))$, indeed both "$X_i|X_1 \ldots X_{i-1} = w_1 \ldots w_{i-1}$" and "$Y_i|Y_1 \ldots Y_{i-1} = w'_1 \ldots w'_{i-1}$" are binary random variables such that, for any $w = \{0, 1\}$, it holds

$$\frac{1}{2} - \varepsilon \leqslant \mathbf{P}\left(X_i = w|X_1 \ldots X_{i-1}\right), \mathbf{P}\left(Y_i = w|Y_1 \ldots Y_{i-1}\right) \leqslant \frac{1}{2} + \varepsilon. \tag{8}$$

Thus we have

$$\begin{aligned}
&D((X_i|X_1 \cdots X_{i-1}) \mid\mid (Y_i|Y_1 \cdots Y_{i-1})) \\
&= \mathbf{P}\left(X_i = 0|X_1 \cdots X_{i-1}\right) \log \frac{\mathbf{P}\left(X_i = 0|X_1 \cdots X_{i-1}\right)}{\mathbf{P}\left(Y_i = 0|Y_1 \cdots Y_{i-1}\right)} \qquad (9)\\
&\quad + \mathbf{P}\left(X_i = 1|X_1 \cdots X_{i-1}\right) \log \frac{\mathbf{P}\left(X_i = 1|X_1 \cdots X_{i-1}\right)}{\mathbf{P}\left(Y_i = 1|Y_1 \cdots Y_{i-1}\right)} \\
&< \left|\log \frac{\mathbf{P}\left(X_i = 0|X_1 \cdots X_{i-1}\right)}{\mathbf{P}\left(Y_i = 0|Y_1 \cdots Y_{i-1}\right)}\right| + \left|\log \frac{\mathbf{P}\left(X_i = 1|X_1 \cdots X_{i-1}\right)}{\mathbf{P}\left(Y_i = 1|Y_1 \cdots Y_{i-1}\right)}\right| \\
&\leqslant 2\left|\log \frac{\frac{1}{2} + \varepsilon}{\frac{1}{2} - \varepsilon}\right| = 2\left|\log \frac{1 + 2\varepsilon}{1 - 2\varepsilon}\right| \qquad (10)\\
&= 2\left|\log(1 + 2\varepsilon) - \log(1 - 2\varepsilon)\right| \qquad (11)\\
&\overset{(a)}{=} 2\left|\varepsilon^2 + o(\varepsilon^2)\right| = O(\varepsilon^2),
\end{aligned}$$

where in (a) we use the Taylor approximation. Thus, we can conclude that

$$D(\mathbf{X}||\mathbf{Y}) = \sum_{i=1}^{t} D((X_i|X_1 \cdots X_{i-1}) \mid\mid (Y_i|Y_1 \cdots Y_{i-1})) \leqslant O(t\varepsilon^2).$$

□

5.3 Absence of reliable components in communication mechanism

As discussed in the Introduction, Theorem 5.1 should be contrasted with the result on the noisy uniform PUSH model [39, 40], in which at each round a node may send a bit to a random neighbor and, upon being received, the bit may be flipped by the communication noise with probability $\frac{1}{2} - \varepsilon$.

In [39] it is assumed that the protocol satisfies a *symmetry* hypothesis, i.e. the choice of nodes on whether to communicate or not, cannot depend on the value of the bit that the node wish to communicate. Without such assumption, the action of communicating *anything* can be employed to reliably solve the valid consensus problem (and many others).

More precisely, we can have nodes sending messages at even rounds[13] only if they wish to communicate value 0, and at odd rounds only if they wish to communicate value 1.

Our lower bound is indeed not applicable to the noisy uniform PUSH model, because it is not possible to reduce Consensus to the Two-Party Protocol in this model. Precisely, Definition 5.2 requires that each bit of information passes through the noisy channel (property *iii)*). In the noisy general PULL model, this is verified at the end of the proof of Lemma 5.3. However, this is not the case in the noisy uniform PUSH model since, besides the (noisy) content of the message, the message received by v^* communicates to it the fact that another node *has chosen to communicate something* at the present round.

Hence, the above comparison with the noisy PUSH model essentially suggests that the lower bound in Theorem 5.1 sensibly relies on the fact that no component of the communication model is immune to the noise action.

5.4 A lower bound for Broadcast

As discussed in Section 1.3, in [17] an $\Omega(\varepsilon^{-2}n)$ lower bound for Broadcast in the noisy uniform PULL model is provided. We here show how, with a similar argument to that used in the proof of Lemma 5.7 in Section 5.2, the lower bound of [17] can be strengthen, in the binary case, to an $\Omega(\varepsilon^{-2}n \log n)$ in order for the nodes to solve the problem w.h.p.

Lemma 5.8. *The Broadcast Problem cannot be solved in the noisy uniform* PULL *model w.h.p. in less than* $\Omega(\varepsilon^{-2}n \log n)$ *rounds.*

Sketch of Proof. We show how to adapt the proofs in [16] and [17]. We refer, in particular, to [16][14].

The proof in [16] consists essentially of two parts: a reduction to a *coin distinguishing task* $\mathrm{ACDT}(\varepsilon, \delta)$ (Definition 11), and a lower bound on the number of samples necessary to solve this task with constant probability (Theorem 12).

Our proof is identical to theirs as for the first part (Claim 13 and Lemma 14 in [16]), and for the upper bound to the KL divergence in the second part (equations (6)-(10) in the proof of Theorem 12). Instead, we replace their Theorem 18 with a bound analogous to our (4) in the proof of Lemma 5.7: we make use of the data processing inequality equation (6) where f is, in the language of [16], the *guess function* of the node on the correct opinion of the source.

It is easy to see that the probability that a node guesses the correct opinion of the source is at least a constant; on the other hand, if we require the probability that the node's guess is wrong to be at most[15] δ, we impose the same bound as our (7). Thus, the replacement of Theorem 18 allows

[13] Note that this expedient relies on nodes sharing a synchronous binary clock. In [39], it is shown how such clock can be easily obtained if, for example, nodes are initially inactive and become active upon receiving the first message.

[14] [17] shows a slightly stronger bound than [16] as for the dependency on the alphabet size. However, we refer to [16] because it is more easily accessible.

[15] Notice that some notation in [16] is used differently than us; we keep consistent with our own use.

to obtain, at the end of the proof of Theorem 12, an inequality of the form $T\frac{\varepsilon^2}{n} = \omega(\frac{1}{\delta})$, where T is the number of pulled messages, which for $\delta = \frac{1}{n}$ provides the desired bound. □

6 Upper Bounds in the Noisy Model

In Theorem 5.1, we obtained a lower bound $\Omega(\varepsilon^{-2}\log n)$ on the number of rounds required by any protocol for Almost Consensus and Almost Majority Consensus that works w.h.p., in the noisy general PULL model. In the next section we show that, in this model, that lower bound is tight for both tasks. As discussed in the introduction, combined with the lower bound for Broadcast in [17] this result demonstrates a strong complexity gap between Consensus and Broadcast in the noisy uniform PULL.

6.1 Upper bound for Consensus in noisy PULL

Theorem 6.1. *In the noisy uniform* PULL *model, with noisy parameter ε, a protocol exists that achieves Consensus within $O(\varepsilon^{-2}\log n)$ rounds and communication, w.h.p. The protocol requires $\Theta(\log\log n + \log\varepsilon^{-2})$ local memory.*

Moreover, if the protocol starts from any initial opinion vector with bias $b = \Omega(\sqrt{n\log n})$, then it guarantees Majority Consensus, w.h.p.

The protocol we refer to in the above theorem works in two consecutive phases. Each phase is a simple application of the well-known *k-Majority Dynamics* [13, 14]:

> k-Majority. *At every round, each node samples k neighbours*[16] *independently and u.a.r. (with replacement). Then, the node updates its opinion according to the majority opinion in the sample.*

Notice that k-Majority, as stated above, assumes a *uniform* k-PULL *model* where, at each round, every node can pull one message from each of the k neighbors chosen independently and uniformly at random *with replacement*[17]. However, it is easy to verify that this parallel process can be implemented on the uniform 1-PULL model using additional $\Theta(k)$ local memory and with a slowdown factor k for its convergence time. In the rest of this section, we will thus consider the following two-phase protocol on the uniform k-PULL model.

> Majority Protocol. Let α be a sufficiently large positive constant[18]. Every node performs $\alpha\log n$ rounds of k-Majority with $k = \Theta(1/\varepsilon^2)$, followed by one round of the k-Majority with $k = \Theta(\varepsilon^{-2}\log n)$.

The proof of Theorem 6.1 will proceed according to the following scheme: we will show that

- If $k = \Omega(1/\varepsilon^2)$ then
  - starting from any opinion vector, within $O(\log n)$ rounds of k-Majority the process reaches an opinion vector where the bias is $b = \Omega(\sqrt{n}\log n)$, w.h.p. (Lemma 6.3),

[16] In the binary case when k is odd, the k-Majority is stochastically equivalent to the $k+1$-Majority where ties are broken u.a.r. (see Lemma 17 in [40]). For this reason, in this section we assume that k is odd.

[17] The assumption that neighbors are chosen independently and uniformly at random with replacement is consistent with previous work [12, 46].

[18] The value of α will be fixed later in the analysis.

- starting from an opinion vector with bias $b = \Omega(\sqrt{n \log n})$ then, within $O(\log n)$ rounds of k-Majority the process reaches an opinion vector where the bias is $b = \Theta(n)$ and the majority opinion is preserved (Lemma 6.4),

- If $k = \Omega(\varepsilon^{-2} \log n)$ and the opinion vector has bias $b = \Theta(n)$, then in one round of k-Majority the process reaches consensus on the majority opinion, w.h.p. (Lemma 6.5).

6.2 Proof of Theorem 6.1

Let $\mathbf{C}^{(t)}$ be the random variable indicating the opinion vector at round t of the majority protocol. Let us name $R^{(t)}$ the number of nodes supporting opinion 0 in such opinion vector and let us define the bias at round t as $b^{(t)} = R^{(t)} - \left(n - R^{(t)}\right) = 2R^{(t)} - n$. In the rest of the section we assume w.l.o.g. that the bias is positive.

Since every time a node u pulls the opinion of a node v, node u correctly gets the opinion of node v with probability $\frac{1}{2} + \varepsilon$ and it gets the other opinion with probability $\frac{1}{2} - \varepsilon$, we can write the probability p_r that a node observes opinion 0 after a pull (i.e. after the node has sampled a neighbor and the noise has possibly flipped its opinion), as a function of the bias

$$p_r = \left(\frac{1}{2} + \frac{b^{(t)}}{2n}\right)\left(\frac{1}{2} + \varepsilon\right) + \left(\frac{1}{2} - \frac{b^{(t)}}{2n}\right)\left(\frac{1}{2} - \varepsilon\right) = \frac{1}{2} + \frac{\varepsilon b^{(t)}}{n}.$$

From Lemma 2 in [39] it follows that, if each node samples $\Omega(1/\varepsilon^2)$ neighbors, then the bias b grows exponentially, in expectation, until it reaches linear size.

Lemma 6.2 (Lemma 2 in [39])**.** *Let $k = c/\varepsilon^2$ be an odd integer for a sufficiently large c, and $\mathbf{c}$ be any opinion vector with bias b, then it holds that*

$$\mathbf{P}\left(u \text{ supports } 0 \,|\, \mathbf{C}^{(t)} = \mathbf{c}\right) \geqslant \frac{1}{2} + \min\left\{4b\,,\, \frac{1}{100}\right\}.$$

The above lemma is useful to give high probability results on the behaviour of the protocol when the bias is large enough to guarantee concentration around the expectation. It thus remains to handle the cases in which the bias is so small that its expected multiplicative growth is smaller than its standard deviation. By leveraging on Lemma 4.5 in [26] (see Appendix A.1), the next lemma shows how the variance of the process and the multiplicative drift of the bias break symmetry, when the initial opinion vector has small or no bias.

Lemma 6.3. *Let m be any positive constant. Starting from an opinion vector with bias $0 \leqslant b^{(t)} < m\sqrt{n}\log n$, the system reaches an opinion vector with bias $b^{(t)} \geqslant m\sqrt{n}\log n$ within $O(\log n)$ rounds of the k-Majority dynamics, w.h.p.*

Proof. Let us name Ω the space of all opinion vectors. In order to apply Lemma 4.5 in [26] we need to show that the following two properties hold:

- For any positive constant h, a positive constant $c_1 < 1$ exists such that, for any opinion vector $\mathbf{c} \in \Omega$ with $b^{(t)} < m\sqrt{n}\log n$, it holds that

$$\mathbf{P}\left(b^{(t+1)} < h\sqrt{n} \,|\, \mathbf{C}^{(t)} = \mathbf{c}\right) < c_1;$$

- Two positive constants ε' and c_2 exist such that, for every opinion vector $\mathbf{c} \in \Omega$ with $b^{(t)} < m\sqrt{n}\log n$, it holds that

$$\mathbf{P}\left(b^{(t+1)} < (1+\varepsilon')b \,|\, \mathbf{C}^{(t)} = \mathbf{c}\right) < e^{-c_2 b^2/n}.$$

As for the first property, since $b^{(t)} = 2R^{(t)} - n$ then $\mathbf{Var}\left(b^{(t+1)}\right) = \Theta(\mathbf{Var}\left(R^{(t+1)}\right)) = \Theta(n(\frac{1}{2} + 2\frac{b^{(t)}}{n})(\frac{1}{2} - 2\frac{b^{(t)}}{n}))$. Notice that in this symmetry-breaking phase $\frac{b^{(t)}}{n} = o(1)$, thus $\mathbf{Var}\left(b^{(t+1)}\right) = \Theta(n)$. Hence, a simple application of the Berry-Esseen Theorem (Theorem A.2 in Appendix A.2) shows that, for a sufficiently large n, there is constant probability that the bias becomes $\Theta(\sqrt{n})$.

As for the second property, from Lemma 6.2 we have that

$$\mathbf{E}\left[R^{(t+1)}\right] = \min\left(n\left(\frac{1}{2} + 4\frac{b^{(t)}}{n}\right), n\left(\frac{1}{2} + \frac{1}{100}\right)\right) = n\left(\frac{1}{2} + 4\frac{b^{(t)}}{n}\right).$$

By applying the additive form of the Chernoff Bound (see Appendix A.4) with $\lambda = 2b^{(t)}$ we get

$$\mathbf{P}\left(R^{(t+1)} < n\left(\frac{1}{2} + 4\frac{b}{n}\right) - 2b\right) < e^{-8b^2/n}.$$

Thus, with probability at least $1 - e^{-8b^2/n}$ it holds

$$b^{(t+1)} = 2R^{(t+1)} - n \geqslant 2n\left(\frac{1}{2} + 4\frac{b}{n}\right) - 2b - n = 2b\,.$$

□

Once the bias is large enough, by iteratively using Lemma 6.2 we can also show that the bias reaches $\Theta(n)$ within $O(\log n)$ rounds and that the majority opinion is preserved, w.h.p.

Lemma 6.4. *Let m be any positive constant. Starting from an opinion vector with bias $b^{(t)} \geqslant m\sqrt{n\log n}$, the system reaches an opinion vector with bias $b^{(t)} = \Theta(n)$ within $O(\log n)$ rounds of the k-Majority dynamics, w.h.p., and the sign of the bias is preserved, w.h.p.*

Proof. As long as $4b \leqslant \frac{n}{100}$, similarly to the second point of Lemma 6.3, from Lemma 6.2 it follows that

$$\mathbf{P}\left(b^{(t+1)} < 2b \,|\, \mathbf{C}^{(t)} = \mathbf{c}\right) < e^{-8b^2/n}.$$

Using the fact that $b^{(t)} \geqslant m\sqrt{n\log n}$ we obtain that

$$\mathbf{P}\left(b^{(t+1)} < 2b \,|\, \mathbf{C}^{(t)} = \mathbf{c}\right) < e^{8m\log n} < \frac{1}{n^{8m}} = \frac{1}{n^{\Theta(1)}},$$

Thus, the bias grows of a multiplicative factor at each round, w.h.p., and this implies, by using the Union Bound, that it reaches a value $b^{(t)} \geqslant \frac{n}{400}$ within $O(\log n)$ rounds, w.h.p. □

For many values of ε, the use of the k-Majority dynamics with $k = \Theta(1/\varepsilon^2)$ is not sufficient to reach an opinion vector where all, or at least many, nodes support the same opinion. Indeed, if ε is a constant smaller than $\frac{1}{2}$ and we consider an opinion vector where all the nodes already support opinion 0, it is easy to see that after one round a constant fraction of the nodes will change opinion,

w.h.p.: let X_u be the random variable counting the number of times a node u pulls opinion 1 in one round starting from such opinion vector. It holds that

$$\mathbf{E}\left[X_u\right] = \frac{c}{\varepsilon^2}\left(\frac{1}{2} - \varepsilon\right) = \frac{c}{2\varepsilon^2} - \Theta\left(\frac{1}{\varepsilon}\right)$$

The variance of X_u is $\frac{c}{\varepsilon^2}\left(\frac{1}{2} - \varepsilon\right)\left(\frac{1}{2} + \varepsilon\right) = \Theta\left(1/\varepsilon^2\right)$ and thus the standard deviation is $\Theta\left(1/\varepsilon\right)$. It follows from the Berry-Esseen Theorem (see Appendix A.2) that X_u has constant probability to deviate from his expectation of a quantity $\Theta\left(1/\varepsilon\right)$, that is sufficient to increase X_u above $\frac{c}{2\varepsilon^2}$, making opinion 1 the majority in the sample. Hence, each node has constant probability to change opinion and by a simple application of a Chernoff Bound and the Union Bound it holds that at least a constant fraction of nodes will change opinion with probability $1 - e^{-\Theta(n)}$.

The above example shows that, in order to reach consensus, we need to increase the size of the sample of each node. In the next lemma we show, as an application of a Chernoff Bound and the Union Bound, that considering a sufficiently large sample, in one step the process reaches an opinion vector where only one opinion is present.

Lemma 6.5. *For any positive constant c_3, a suitable constant c_4 exists such that, starting from an opinion vector with bias $b^{(t)} \geqslant \frac{c_3 n}{2}$, in one round of the k-Majority dynamics with $k = c_4\varepsilon^{-2}\log n$, the system reaches an opinion vector with $b = n$, w.h.p.*

Proof. The proof is a straightforward application of the multiplicative form of the Chernoff Bound (see Appendix A.3) and of the Union Bound. Indeed, let X_u be the random variable counting the number of times a node u pulls opinion 1 in one round starting from an opinion vector with bias $b^{(t)} \geqslant \frac{c_3 n}{2}$. It holds that

$$\begin{aligned}
\mathbf{E}\left[X_u\right] &= \frac{c_4 \log n}{\varepsilon^2}\left(\left(\frac{1}{2} + c_3\right)\left(\frac{1}{2} + \varepsilon\right) + \left(\frac{1}{2} - c_3\right)\left(\frac{1}{2} - \varepsilon\right)\right) \\
&= \frac{c_4 \log n}{\varepsilon^2}\left(\frac{1}{2} + 2\varepsilon c_3\right)
\end{aligned}$$

Notice that, if $X_u \geqslant \frac{c_4 \log n}{\varepsilon^2}\left(\frac{1}{2} + \varepsilon c_3\right)$, then the majority opinion is 1. The probability of the complementary event is

$$\begin{aligned}
\mathbf{P}\left(X_u < \frac{c_4 \log n}{\varepsilon^2}\left(\frac{1}{2} + \varepsilon c_3\right)\right) &= \mathbf{P}\left(X_u < \frac{c_4 \log n}{\varepsilon^2}\left(\frac{1}{2} + 2\varepsilon c_3\right) - \frac{c_3 c_4 \log n}{\varepsilon}\right) \\
&= \mathbf{P}\left(X_u < \mathbf{E}\left[X_u\right]\left(1 - \frac{c_3\varepsilon}{\frac{1}{2} + 2\varepsilon c_3}\right)\right) \\
&< \exp\left(-\frac{c_4 \log n}{\varepsilon^2}\left(\frac{1}{2} + 2\varepsilon c_3\right)\left(\frac{c_3\varepsilon}{\frac{1}{2} + 2\varepsilon c_3}\right)^2\right) \\
&= \exp\left(-\frac{c_3^2 c_4 \log n}{\frac{1}{2} + 2\varepsilon c_3}\right) \\
&\leqslant \exp\left(-\frac{c_3^2 c_4 \log n}{\frac{1}{2} + 2}\right) = \frac{1}{n^2}
\end{aligned}$$

where in the last equality we chose $c_4 = \frac{4}{5}c_3^2$. The thesis then follows by applying the Union Bound over the n nodes. □

6.3 A tight bound for Broadcast in noisy uniform PULL

We show an $O(\varepsilon^{-2} n \log n)$ upper bound for Broadcast in the noisy uniform PULL model. This is obtained via a simple protocol NoisyBroadcast we describe below. Observe that the protocol assumes that nodes know the value of the noise parameter ε.

Protocol NoisyBroadcast.

- In the first phase, each non-source node displays 0 (obviously, the source displays its input value), and performs a pull operation for $\Theta(\varepsilon^{-2} n \log n)$ rounds; it then chooses to support value 1 iff the fraction of received messages equal to 1 is at least $\frac{1}{2} - \varepsilon(1 - \frac{1}{2n})$, zero otherwise.
- In the second phase, nodes run the Majority Consensus protocol of Theorem 6.1, starting with the value obtained at the end of the first phase.

We prove the following theorem. Notice that this upper bound is tight, since it matches the lower bound in Lemma 5.8.

Theorem 6.6. *Protocol* NoisyBroadcast *solves the Broadcast problem in the noisy uniform* PULL *model in* $O(\varepsilon^{-2} n \log n)$ *rounds, w.h.p.*

Proof. We prove that at the end of the first phase, the fraction of nodes which have obtained a value equal to the source's input is greater than those that failed by at least $\sqrt{n \log n}$ nodes. The latter fact satisfies the hypothesis of Theorem 6.1 for solving Majority consensus in $O(\varepsilon^{-2} \log n)$, which constitutes the second phase.

Recall that, from the definition of NoisyBroadcast, during the first phase all (non-source) nodes displays value 0. Hence, if the source's input value is also 0, each pulled message is equal to 1 with probability $\frac{1}{2} - \varepsilon$. Otherwise, if the source's input value is 1, each pulled message is equal to 1 with probability[19] $\frac{1}{2} - \varepsilon(1 - \frac{1}{n}) + \frac{\varepsilon}{n}$.

Let us call the two aforementioned configurations $\mathbf{c_0}$ and $\mathbf{c_1}$, respectively.

By the Berry-Esseen Theorem (Theorem A.2), the distribution of the $n \log n$ pulled messages in $\mathbf{c_0}$ and $\mathbf{c_1}$ is given (up to a $O(\frac{1}{\sqrt{n}})$ error in total variation), by the following two normal distribution: the first, with expectation $(\frac{1}{2} - \varepsilon)\frac{n \log n}{\varepsilon^2}$ and variance less than $\frac{1}{2\varepsilon}\sqrt{n \log n}$; the second, with expectation $(\frac{1}{2} - \varepsilon(1 - \frac{1}{n}) + \frac{\varepsilon}{n})\frac{n \log n}{\varepsilon^2}$ and variance less than $\frac{1}{2\varepsilon}\sqrt{n \log n}$.

In order to choose value 1 at the end of the phase, the rule of the protocol requires that, in $\mathbf{c_0}$, the number of received one exceeds the expectation $(\frac{1}{2} - \varepsilon)\frac{n \log n}{\varepsilon^2}$ by an additive factor $\frac{\log n}{2\varepsilon}$. A direct calculation shows that the value of the density $f(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(z-\mu)^2}{2\sigma^2}}$ of a normal distribution $N(\mu, \sigma)$ varies by at most a constant multiplicative factor within a standard deviation σ from its expected value μ. In other words

$$\begin{cases} \int_x^\infty \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(z-\mu)^2}{2\sigma^2}} dz = \frac{1}{2} - \Omega(\frac{x-\mu}{\sigma}) & \text{if } x \in (\mu, \mu + \sigma), \\ \int_x^\infty \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(z-\mu)^2}{2\sigma^2}} dz = \frac{1}{2} + \Omega(\frac{\mu-x}{\sigma}) & \text{if } x \in (\mu - \sigma, \mu). \end{cases}$$

Hence, the probability that, pulling from $\mathbf{c_0}$, a node chooses 1 at the end of the first phase, is $\frac{1}{2} - \Omega(\sqrt{\frac{\log n}{n}})$.

By the same argument, we can show that the probability that, pulling from $\mathbf{c_1}$, a node chooses 1 at the end of the first phase, is $\frac{1}{2} + \Omega(\sqrt{\frac{\log n}{n}})$.

[19] We remark that, in order to simplify calculations, we assume that each node may pull itself with probability $\frac{1}{n}$.

The proof finally follows from a standard application of the Chernoff bounds (Appendix A.3), to show concentration of probability around the expected value $\frac{n}{2} - \Omega(\sqrt{n \log n})$ of nodes supporting 1 when starting from $\mathbf{c_0}$, and the expected value $\frac{n}{2} + \Omega(\sqrt{n \log n})$ of nodes supporting 1 when starting from $\mathbf{c_1}$. □

References

[1] Mohammed Amin Abdullah and Moez Draief. Majority consensus on random graphs of a given degree sequence. *CoRR*, abs/1209.5025, 2012.

[2] Mohammed Amin Abdullah and Moez Draief. Global majority consensus by local majority polling on graphs of a given degree sequence. *Discrete Applied Mathematics*, 180:1–10, 2015.

[3] Abhinav Aggarwal, Varsha Dani, Thomas P. Hayes, and Jared Saia. Distributed computing with channel noise. *IACR Cryptology ePrint Archive*, 2017:710, 2017.

[4] Noga Alon, Amotz Bar-Noy, Nathan Linial, and David Peleg. A lower bound for radio broadcast. *Journal of Computer and System Sciences*, 43(2):290 – 298, 1991.

[5] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and Peralta René. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18(4):235–253, 2006.

[6] Dana Angluin, James Aspnes, and David Eisenstat. A Simple Population Protocol for Fast Robust Approximate Majority. *Distributed Computing*, 21(2):87–102, 2008. (Preliminary version in DISC'07).

[7] Dana Angluin, Michael J. Fischer, and Hong Jiang. Stabilizing consensus in mobile networks. In *Proc. of Distributed Computing in Sensor Systems (DCOSS'06)*, volume 4026 of *LNCS*, pages 37–50, 2006.

[8] James Aspnes and Eric Ruppert. *An Introduction to Population Protocols*, pages 97–120. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[9] Chen Avin and Robert Elsässer. Breaking the log n barrier on rumor spreading. *Distributed Computing*, 2017.

[10] Reuven Bar-Yehuda, Oded Goldreich, and Alon Itai. On the time-complexity of broadcast in multi-hop radio networks: An exponential gap between determinism and randomization. *Journal of Computer and System Sciences*, 45(1):104 – 126, 1992.

[11] Luca Becchetti, Andrea Clementi, Emanuele Natale, Francesco Pasquale, and Riccardo Silvestri. Plurality consensus in the gossip model. In *ACM-SIAM SODA'15*, pages 371–390, 2015.

[12] Luca Becchetti, Andrea Clementi, Emanuele Natale, Francesco Pasquale, and Riccardo Silvestri. Plurality Consensus in the Gossip Model. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, SODA '15, pages 371–390, San Diego, California, 2015. SIAM.

[13] Luca Becchetti, Andrea Clementi, Emanuele Natale, Francesco Pasquale, and Luca Trevisan. Stabilizing consensus with many opinions. In *Proc. of the 27th Ann. ACM-SIAM Symp. on Discrete algorithms*, pages 620–635. SIAM, 2016.

[14] Luca Becchetti, Andrea E.F. Clementi, Emanuele Natale, Francesco Pasquale, Riccardo Silvestri, and Luca Trevisan. Simple dynamics for plurality consensus. In *ACM SPAA'14*, pages 247–256, 2014.

[15] Petra Berenbrink, Andrea Clementi, Peter Kling, Robert Elsässer, Frederik Mallmann-Trenn, and Emanuele Natale. Ignore or comply? on breaking symmetry in consensus. In *ACM PODC'17*, 2017. to appear (Tech. Rep. in arXiv:1702.04921 [cs.DC]).

[16] Lucas Boczkowski, Ofer Feinerman, Amos Korman, and Emanuele Natale. Limits for Rumor Spreading in Stochastic Populations. In Anna R. Karlin, editor, *Proc. of the 9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*, volume 94 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 49:1–49:21, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[17] Lucas Boczkowski, Emanuele Natale, Ofer Feinerman, and Amos Korman. Limits on reliable information flows through stochastic populations. *PLOS Computational Biology*, 14(6):e1006195, June 2018.

[18] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE/ACM Transactions on Networking*, 14:2508–2530, 2006.

[19] L. Cardelli and A. Csikász-Nagy. The cell cycle switch computes approximate majority. *Scientific Reports*, Vol. 2, 2012.

[20] Keren Censor-Hillel, Ran Gelles, and Bernhard Haeupler. Making asynchronous distributed computations robust to noise. *CoRR*, abs/1702.07403, 2017.

[21] Bernard Chazelle. Natural algorithms and influence systems. *Commun. ACM*, 55(12):101–110, December 2012.

[22] F. Chierichetti, S. Lattanzi, and A. Panconesi. Rumour spreading and graph conductance. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, Proceedings, pages 1657–1663. Society for Industrial and Applied Mathematics, January 2010.

[23] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. Almost Tight Bounds for Rumour Spreading with Conductance. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing*, STOC '10, pages 399–408, New York, NY, USA, 2010. ACM.

[24] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. Rumor spreading in social networks. *Theoretical Computer Science*, 412(24):2602 – 2610, 2011. Selected Papers from 36th International Colloquium on Automata, Languages and Programming (ICALP 2009).

[25] Gregory Chockler, Murat Demirbas, Seth Gilbert, Nancy Lynch, Calvin Newport, and Tina Nolte. Consensus and collision detectors in radio networks. *Distributed Computing*, 21(1):55–84, June 2008.

[26] Andrea E. F. Clementi, Luciano Gualà, Francesco Pasquale, Giacomo Scornavacca, Emanuele Natale, and Mohsen Ghaffari. A Tight Analysis of the Parallel Undecided-State Dynamics with Two Colors. In *Proc. of the 43rd Int. Symp. on Mathematical Foundations of Computer Science (MFCS 2018)*, Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018.

[27] Andrea E. F. Clementi, Angelo Monti, Francesco Pasquale, and Riccardo Silvestri. Broadcasting in dynamic radio networks. *J. Comput. Syst. Sci.*, 75(4):213–230, June 2009. Previ0us Version in ACM PODC'07.

[28] Andrea E. F. Clementi, Angelo Monti, and Riccardo Silvestri. Selective families, superimposed codes, and broadcasting on unknown radio networks. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '01, pages 709–718, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics.

[29] C. Cooper, R. Elsasser, and T. Radzik. The power of two choices in distributed voting. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP'14)*, volume 8573 of *LNCS*, pages 435–446. Springer, 2014.

[30] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *ACM PODC'87*, 1987.

[31] E. W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Commun. ACM*, 17(11):643–644, 1974.

[32] Benjamin Doerr, Carola Doerr, Shay Moran, and Shlomo Moran. Simple and optimal randomized fault-tolerant rumor spreading. *Distributed Computing*, 29(2):89–104, Apr 2016.

[33] Benjamin Doerr and Mahmoud Fouz. Quasi-random rumor spreading: Reducing randomness can be costly. *Information Processing Letters*, 111(5):227–230, February 2011.

[34] Benjamin Doerr, Tobias Friedrich, and Thomas Sauerwald. Quasirandom Rumor Spreading. *ACM Trans. Algorithms*, 11(2):9:1–9:35, October 2014.

[35] Benjamin Doerr, Leslie Ann Goldberg, Lorenz Minder, Thomas Sauerwald, and Christian Scheideler. Stabilizing consensus with the power of two choices. In *ACM SPAA'11*, pages 149–158, 2011.

[36] S. Dolev. *Self-Stabilization*. The MIT Press, 2000.

[37] David Doty. Timing in chemical reaction networks. In *ACM-SIAM SODA'14*, pages 772–784, 2014.

[38] A. El Gamal. Open problems. In *Workshop on specific problems in communication and computation*, 1984.

[39] Ofer Feinerman, Bernhard Haeupler, and Amos Korman. Breathe Before Speaking: Efficient Information Dissemination Despite Noisy, Limited and Anonymous Communication. *Distributed Computing*, 30(5):239–355, 2017. Ext. Abs. in ACM PODC'14.

[40] Pierre Fraigniaud and Emanuele Natale. Noisy rumor spreading and plurality consensus. *Distributed Computing*, pages 1–20, June 2018.

[41] Nigel R. Franks, Stephen C. Pratt, Eamonn B. Mallon, Nicholas F. Britton, and David J.T. Sumpter. Information flow, opinion polling and collective intelligence in house–hunting social insects. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 357(1427):1567–1583, 2002.

[42] A. M. Frieze and G. R. Grimmett. The shortest-path problem for graphs with random arc-lengths. *Discrete Applied Mathematics*, 10(1):57–77, January 1985.

[43] Emanuele G. Fusco and Andrzej Pelc. Communication Complexity of Consensus in Anonymous Message Passing Systems. *Fundam. Inf.*, 137(3):305–322, July 2015.

[44] R. G. Gallager. Finding parity in a simple broadcast network. *IEEE Trans. Inf. Theor.*, 34(2):176–180, September 2006.

[45] Mohsen Ghaffari and Johannes Lengler. Tight analysis for the 3-majority consensus dynamics. *CoRR*, abs/1705.05583, 2017.

[46] Mohsen Ghaffari and Merav Parter. A Polylogarithmic Gossip Algorithm for Plurality Consensus. In *Proceedings of the 36th ACM Symposium on Principles of Distributed Computing*, PODC '16, pages 117–126, New York, NY, USA, 2016. ACM.

[47] George Giakkoupis. Tight bounds for rumor spreading in graphs of a given conductance. In *Symposium on Theoretical Aspects of Computer Science (STACS2011)*, volume 9, pages 57–68, 2011.

[48] George Giakkoupis. Tight Bounds for Rumor Spreading with Vertex Expansion. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 801–815, Portland, Oregon, 2014. SIAM.

[49] George Giakkoupis, Yasamin Nazari, and Philipp Woelfel. How asynchrony affects rumor spreading time. In *35th ACM Symposium on Principles of Distributed Computing (PODC 2016)*, 2016.

[50] George Giakkoupis and Thomas Sauerwald. Rumor Spreading and Vertex Expansion. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, SODA '12, pages 1623–1641, Philadelphia, PA, USA, 2012. Society for Industrial and Applied Mathematics.

[51] Navin Goyal, Guy Kindler, and Michael Saks. Lower bounds for the noisy broadcast problem. *SIAM J. Comput.*, 37(6):1806–1841, March 2008.

[52] Bernhard Haeupler. Simple, fast and deterministic gossip and rumor spreading. *J. ACM*, 62(6):47:1–47:18, December 2015.

[53] Bernhard Haeupler and Dahlia Malkhi. Optimal gossip with direct addressing. In *Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing*, PODC '14, pages 176–185, New York, NY, USA, 2014. ACM.

[54] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. Randomized rumor spreading. In *IEEE FOCS'00*, pages 565–574, 2000.

[55] D. R. Kowalski and A. Pelc. Deterministic broadcasting time in radio networks of unknown topology. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 63–72, 2002.

[56] Fabian Kuhn, Nancy Lynch, Calvin Newport, Rotem Oshman, and Andrea Richa. Broadcasting in unreliable radio networks. In *Proceedings of the 29th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, PODC '10, pages 336–345, New York, NY, USA, 2010. ACM.

[57] Fabian Kuhn, Nancy Lynch, and Rotem Oshman. Distributed computation in dynamic networks. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing*, STOC '10, pages 513–522, New York, NY, USA, 2010. ACM.

[58] E. Kushilevitz and Y. Mansour. An $\omega(d \log(n/d))$ lower bound for broadcast in radio networks. *SIAM Journal on Computing*, 27(3):702–712, 1998.

[59] Nancy A Lynch. *Distributed algorithms*. Morgan Kaufmann, 1996.

[60] David J. C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003.

[61] G. B. Mertzios, S. E. Nikoletseas, C. Raptopoulos, and P. G. Spirakis. Determining majority in networks with local interactions and very small local memory. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP'14)*, 2014.

[62] Elchanan Mossel, Joe Neeman, and Omer Tamuz. Majority dynamics and aggregation of information in social networks. *Autonomous Agents and Multi-Agent Systems*, 28(3):408–429, 2014.

[63] Saket Navlakha and Ziv Bar-Joseph. Distributed information processing in biological and computational systems. *Communications of the ACM*, 58(1):94–102, 2015.

[64] Marshall Pease, Robert Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228–234, 1980.

[65] Etienne Perron, Dinkar Vasudevan, and Milan Vojnovic. Using Three States for Binary Consensus on Complete Graphs. In *IEEE INFOCOM'09*, pages 2527–1535, 2009.

[66] Boris Pittel. On Spreading a Rumor. *SIAM J. Appl. Math.*, 47(1):213–223, March 1987.

[67] Michael O. Rabin. Randomized byzantine generals. In *Proc. of the 24th Ann. Symp. on Foundations of Computer Science (SFCS)*, pages 403–409. IEEE, 1983.

[68] N. Santoro and P. Widmayer. Time is Not a Healer. In *Proceedings of the 6th Annual Symposium on Theoretical Aspects of Computer Science on STACS 89*, pages 304–313, New York, NY, USA, 1989. Springer-Verlag New York, Inc.

[69] Devavrat Shah. Gossip Algorithms. *Foundations and Trends® in Networking*, 3(1):1–125, 2007.

Appendix

A Technical Tools

A.1 Symmetry breaking lemma

Lemma A.1 (Lemma 4.5 in [26]). *Let $\{X_t\}_{t\in\mathbb{N}}$ be a Markov Chain with finite state space Ω and let $f : \Omega \mapsto [0, n]$ be a function that maps states to integer values. Let c_3 be any positive constant and let $m = c_3\sqrt{n}\log n$ be a target value. Assume the following properties hold:*

1. *For any positive constant h, a positive constant $c_1 < 1$ exists such that, for any $x \in \Omega$ with $f(x) < m$, it holds that*
$$\mathbf{P}\left(f(X_{t+1}) < h\sqrt{n} \,|\, X_t = x\right) < c_1\,,$$
2. *Two positive constants ε, c_2 exist such that, for any $x \in \Omega$ with $f(x) < m$, it holds that*
$$\mathbf{P}\left(f(X_{t+1}) < (1+\varepsilon)f(X_t) \,|\, X_t = x\right) < e^{-c_2 f(x)^2/n}\,.$$

Then the process reaches a state x such that $f(x) \geqslant m$ within $O(\log n)$ rounds, w.h.p.

A.2 Berry-Esseen Theorem

Theorem A.2 (Berry-Esseen). *Let $X_1, \ldots, X_n$ be independent and identically distributed random variables with mean $\mu = 0$, variance $\sigma^2 > 0$, and third absolute moment $\rho < \infty$. Let $Y_n = \frac{1}{n}\sum_{i=1}^n X_i$; let F_n be the cumulative distribution function of $\frac{Y_n\sqrt{n}}{\sigma}$; let Φ the cumulative distribution function of the standard normal distribution. Then, there exists a positive constant $C < 0.4748$ such that, for all x and for all n,*
$$|F_n(x) - \Phi(x)| \leqslant \frac{C\rho}{\sigma^3\sqrt{n}}.$$

A.3 Chernoff Bound multiplicative form

Let $X_1, \ldots, X_n$ be independent 0-1 random variables. Let $X = \sum_{i=1}^n X_i$ and $\mu \leqslant \mathbf{E}[X] \leqslant \mu'$. Then, for any $0 < \delta < 1$ the following Chernoff bounds hold:

$$\mathbf{P}\left(X \geqslant (1+\delta)\mu\right) \leqslant e^{-\mu\delta^2/3}. \tag{12}$$

$$\mathbf{P}\left(X \leqslant (1-\delta)\mu'\right) \leqslant e^{-\mu'\delta^2/2}. \tag{13}$$

A.4 Chernoff Bound additive form

Let $X_1, \ldots, X_n$ be independent 0-1 random variables. Let $X = \sum_{i=1}^n X_i$ and $\mu = \mathbf{E}[X]$. Then the following Chernoff bounds hold:
for any $0 < \lambda < n - \mu$,
$$\mathbf{P}\left(X \leqslant \mu - \lambda\right) \leqslant e^{-2\lambda^2/n}, \tag{14}$$
for any $0 < \lambda < \mu$,
$$\mathbf{P}\left(X \geqslant \mu + \lambda\right) \leqslant e^{-2\lambda^2/n}. \tag{15}$$