Consensus vs Broadcast, with and without 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 vs Broadcast, with and without Noise. ITCS 2020 - 11th Annual Innovations in Theoretical Computer Science, Jan 2020, Seattle, United States. pp.42 - 43, 10.4230/LIPIcs.ITCS.2020.42 . hal-01958994v2

Consensus vs Broadcast, with and without Noise (Extended Abstract)

Andrea Clementi
Università *Tor Vergata* di Roma
clementi@mat.uniroma2.it

Luciano Gualà
Università *Tor Vergata* di Roma
guala@mat.uniroma2.it

Emanuele Natale
Université Côte d'Azur
emanuele.natale@univ-cotedazur.fr

Francesco Pasquale
Università *Tor Vergata* di Roma
pasquale@mat.uniroma2.it

Giacomo Scornavacca
Università degli Studi di Sassari
giacomoscornavacca@gmail.com

Luca Trevisan
Università Bocconi
l.trevisan@unibocconi.it

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 is proved by Boczkowski et al. [PLOS Comp. Bio. 2018] on the convergence time of binary Broadcast 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 on the convergence time of binary Consensus in such model, thus establishing an exponential complexity gap between Consensus versus Broadcast. We also prove our upper bound above is tight and this implies, for binary Consensus, a further strong complexity gap between noisy uniform PULL and noisy uniform PUSH. Finally, we show a $\Theta(\varepsilon^{-2}n\log n)$ bound for Broadcast in the noisy uniform PULL.

2012 ACM Subject Classification Theory of computation → Distributed algorithms; Theory of computation → Random walks and Markov chains; Theory of computation → Random network models.

Keywords and phrases Distributed Computing, Consensus, Broadcast, Gossip Models, Noisy Communication Channels.

11th Innovations in Theoretical Computer Science Conference (ITCS 2020).
Editor: Thomas Vidick; Article No. 42; pp. 42:1–42:13
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Digital Object Identifier 10.4230/LIPIcs.ITCS.2020.42

Related Version A full version of the paper is available at `https://arxiv.org/abs/1807.05626`.

Funding *Luca Trevisan*: LT was supported by the NSF under grant CCF 1815434 and his work on this project has received funding from the European Research Council (ERC) under the European Union Horizon 2020 research and innovation programme (grant agreement No. 834861).

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 [21, 23, 39, 41], 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 [5, 22, 40] 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] 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.

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 [24, 27, 37].[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

---

[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.

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.

A first major 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?

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, namely the noisy Gossip, however, we establish an exponential gap between Consensus and Broadcast.

As a second major question, we investigate the impact of the communication noise on the Consensus problem. More in detail, does this impact strongly depend on the particular noisy Gossip model we adopt? We will give a positive answer to this question by establishing a strong complexity separation between the two most popular versions of the Gossip model, namely, the Pull model and the Push one.

Roadmap of the paper and a remark

In order to formally state and discuss our results, in the next section, we introduce the distributed models and their associated complexity measures our results deal with. In Section 3, we describe our results and their consequences for noiseless communication models and compare them with the related previous work. Section 4 is devoted to our results for the noisy communication models and to their comparison with the related previous work. Finally, in Section 5 we provide a short summary of the obtained results and discuss some related open questions. We remark that, in this version, we only sketch the main ideas of the technical proofs: detailed proofs are given in the full version of the paper [17].

2 Communication and computational models

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 [20, 30], 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 sends a message to its callers (if any). 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, while 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 [5, 2, 22, 24, 37, 40], 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 [4, 6, 23, 12, 24, 25, 27]. 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* [5, 4]) have also been extensively studied [11, 28, 40]. 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.

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]) [5, 9, 11, 22, 28, 31]. 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, 15, 14, 19, 29, 36].

However, while for Broadcast $\Omega(\log n)$ time and work are necessary in the complete graph [11, 28, 31], prior to this work, it was still unknown whether a more efficient protocol existed for Consensus and Majority Consensus.

---

[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.

[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 $\alpha > 1$.

3 Our contribution I: Noiseless communication

Our main result is a reduction of a weak form of Broadcast to Consensus 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.

In order to formally 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.

▶ **Definition 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`.

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, 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 the next theorem, 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 almost consensus with probability $1 - o(1/n)$, then there is an initial opinion vector and a source such that the protocol infects many nodes from that source with probability at least $(1 - o(1))/n$.

▶ **Theorem 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 $(1 - o(1))/n$.*

Notice that the above result implies 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. However, 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, the above theorem implies that 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).

In case of Consensus problem (i.e. $\delta = 0$), our proof for Theorem 2 makes use of 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 prove 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 which s infects all other nodes. Thus the probability that s infects all nodes from the initial vector $\mathbf{x}$ has to be $\geqslant (1-o(1))/n$. Then, to extend the above approach for the Almost Consensus problem (i.e. $\delta > 0$), some additional care and a suitable counting argument are required to manage the unknown set of outliers.

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.[6] Formally:

▶ **Theorem 3.** *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 more than τ^{B} units of $\mathcal{T}$.*

3.1 Some applications

Lower bounds for infection are known in several models in which there are 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 (see the full version for the formal statements of such results):

- In the uniform GOSSIP model (also known as uniform PUSH-PULL model), and in the general PUSH model, tight analysis (see [30, 31]) 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, the *Population Protocols* with uniform/probabilistic scheduler, as defined in [5].
- The last application we mention here concerns the synchronous *Radio Network* model [3, 7, 16, 42]. Several optimal bounds have been obtained on the Broadcast time [7, 18, 32, 33, 34] while only few results are known for Consensus time [16, 42]. 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 [3] on Broadcast with our reduction result, we get a new lower bound for Consensus in this model.

We remark that our reduction allows us to prove that some of the above lower bounds hold even if the nodes have unbounded memory and can send/receive messages of unbounded size.

---

[6] 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.

4 Our contribution II: Noisy communication

We now 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 [35] 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. Then, in the sequel, the version of each model $\mathcal{M}$, in which the presence of communication noise above is introduced, will be shortly denoted as *noisy* $\mathcal{M}$.

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.

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 [10, 25] 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 [13, 38].

As for the uniform Push model with one-bit messages, we first notice that there is a simple local strategy that solves both (binary) Broadcast and Consensus in the noisy Push (this strategy holds even assuming that agents share only a *binary* synchronous clock). For instance, consider binary Consensus: let every node with initial opinion 0 start a broadcast process at even rounds, while the same task is started in odd rounds by nodes with initial opinion 1. When a node receives a bit in any even (odd) round, this bit is always interpreted as 0 (1). Then, at every round, each node updates its output with, for instance, the minimum value it has seen so far (any round).

In [25], the authors consider a restricted, natural class of *symmetric* algorithms where the action of the nodes cannot depend on the value of the exchanged bits. In this setting, they prove that (binary) Broadcast and (binary) Majority Consensus can be solved in time $O(\varepsilon^{-2}\log n)$, where ε is the noise parameter. They also prove a matching lower bound for this class of algorithms. This has been later generalized to non-binary opinions in [26].

In the noisy uniform Pull model however, [10] proves an $\Omega(\varepsilon^{-2}n)$ time lower bound[7]. 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 [10] 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 2 suggests that there could be $\Omega(\varepsilon^{-2}n)$ lower bounds for Consensus and Majority

---

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

Consensus, but recall that the reduction is to the infection problem, and infection is equivalent to Broadcast only when we have errorless channels.

4.1 Upper bounds in noisy uniform Pull

4.1.1 A protocol for Consensus and its analysis

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.

▶ **Theorem 4.** *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* [8, 9]:

> k-Majority. *At every round, each node samples* k *neighbours*[8] *independently and u.a.r. (with replacement). Then, the node updates its opinion according to the majority opinion in the sample.*

The protocol is thus the following:

> Majority Protocol. Let α be a sufficiently large positive constant[9]. 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 $k = \Theta(\varepsilon^{-2}\log n)$.

Our analysis shows 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 (notice that the latter fact guarantees the validity property, w.h.p.). Then, in the second phase, despite the communication errors, we show 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 $O(\varepsilon^{-2}\log n)$, we prove the protocol "breaks symmetry" w.h.p. and, no matter the initial vector and the presence of communication noise, 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 the first sub-phase is our main technical novelty while the analysis of the second sub-phase for achieving Majority Consensus is similar to that in [25, 26]. If the initial opinion vector is unanimous, 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.

---

[8] 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 [26]). For this reason, in this section we assume that k is odd.

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

4.1.2 A protocol for Broadcast

We provide a simple two-phases Broadcast protocol that runs in the noisy uniform PULL model.

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 4, starting with the value obtained at the end of the first phase.

We prove the following performance of the protocol, nearly matching the $\Omega(\varepsilon^{-2} n)$ lower bound mentioned before [10]:

▶ **Theorem 5.** *Protocol* NoisyBroadcast *solves the Broadcast problem in the noisy uniform* Pull *model in $\mathcal{O}(\varepsilon^{-2} n \log n)$ rounds, w.h.p.*

Our proof shows 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 4 for solving Majority Consensus in $\mathcal{O}(\varepsilon^{-2} \log n)$, which constitutes the second phase.

4.2 Lower bounds in noisy Pull models

We prove that any Almost Consensus protocol with at most δn outliers and with error probability at most δ requires $\Omega\left(\varepsilon^{-2} \log \delta^{-1}\right)$ rounds. Formally:

▶ **Theorem 6.** *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*[10]*.*

This shows that the complexity $O(\varepsilon^{-2} \log n)$ of our protocol described in Subsection 4.1.1 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. Recalling the $\Theta(\log n)$ bound that holds for (general) Consensus protocols in the noisy uniform Push (for any value of ε), our lower bound above thus implies a strong separation result between noisy uniform Pull and noisy uniform Push.

The proof of Theorem 6 is one of the main technical contributions of this work and below we provide a short discussion.

In [25], 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

---

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

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 [25]). Instead, we prove 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, we use KL divergence to show that the channel has to be used $\Omega(\varepsilon^{-2}\log\delta^{-1})$ times.

4.2.1 An improved lower bound for Broadcast

The $\Omega(\varepsilon^{-2}n)$ lower bound of [10] for Broadcast in the uniform PULL model applies to protocols that have constant probability of correctly performing the broadcast operation. With the following theorem we show a way of modifying their proof (in particular, 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 Theorem 5.

▶ **Theorem 7.** *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.*

5 Conclusions

Figure 1 shows the two main separation results that follow from a comparison between some previous bounds and the bounds we obtain in this paper: The complexity gap between Consensus and Broadcast in the presence or absence of noise and the different complexity behaviour of Consensus between noisy uniform Pull and noisy uniform Push. The figure also shows our new lower bounds for Consensus in the noiseless Gossip models.

A further consequence 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.

We considered noisy communication models that assume the presence of a global clock: nodes work in parallel sharing the value of the current round. Our protocols definitely exploit this important property of the model. Then, an interesting open issue is to analyse fundamental tasks, such as Consensus and Broadcast, in asynchronous versions of the Push and Pull models where, as in our setting, communication is noisy and takes place via binary messages only. A further interesting future work we plan to consider is to introduce a (strong) bound on the local memory of the nodes.

References

1 Mohammed Amin Abdullah and Moez Draief. Majority consensus on random graphs of a given degree sequence. *CoRR*, abs/1209.5025, 2012. URL: `http://arxiv.org/abs/1209.5025`, `arXiv:1209.5025`.

	ε-noisy models		Noiseless models	
	Uniform Pull	**Uniform Push**	**Uniform Pull**	**Uniform Push**
Broadcast	$\Omega\left(n/\varepsilon^2\right)$ [10] $\Theta\left(\frac{n\log n}{\varepsilon^2}\right)$	$\Theta\left(\log n\right)^*$	$\Theta(\log n)$ [30, 31]	$\Theta(\log n)$ [30, 31]
Consensus	$\Theta\left(\frac{\log n}{\varepsilon^2}\right)$	$\Omega\left(\log n\right)$ $\mathcal{O}(\log n)^*$	$\Omega\left(\log n\right)$ $\mathcal{O}(\log n)$ [30, 31]	$\Omega\left(\log n\right)$ $\mathcal{O}(\log n)$ [30, 31]

Figure 1 The updated state-of-art for Broadcast and Consensus in GOSSIP models with and w/o noise. The results of this paper are in red, while "folklore" results are marked with * (see also the preliminary discussion in Section 4).

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 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. URL: `http://www.sciencedirect.com/science/article/pii/002200009190015W`, `doi:https://doi.org/10.1016/0022-0000(91)90015-W`.

4 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.

5 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).

6 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.

7 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. URL: `http://www.sciencedirect.com/science/article/pii/002200009290042H`, `doi:https://doi.org/10.1016/0022-0000(92)90042-H`.

8 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.

9 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.

10 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. URL: `http://dx.plos.org/10.1371/journal.pcbi.1006195`, `doi:10.1371/journal.pcbi.1006195`.

11 Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE/ACM Transactions on Networking*, 14:2508–2530, 2006. URL: `http://dx.doi.org/10.1109/TIT.2006.874516`, `doi:10.1109/TIT.2006.874516`.

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

13 Bernard Chazelle. Natural algorithms and influence systems. *Commun. ACM*, 55(12):101–110, December 2012. URL: `http://doi.acm.org/10.1145/2380656.2380679`, `doi:10.1145/2380656.2380679`.

14 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. URL: `http://doi.acm.org/10.1145/1806689.1806745`, `doi:10.1145/1806689.1806745`.

15 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). URL: `http://www.sciencedirect.com/science/article/pii/S0304397510006122`, `doi:https://doi.org/10.1016/j.tcs.2010.11.001`.

16 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. URL: `https://link.springer.com/article/10.1007/s00446-008-0056-2`, `doi:10.1007/s00446-008-0056-2`.

17 Andrea E. F. Clementi, Luciano Gualà, Emanuele Natale, Francesco Pasquale, Giacomo Scornavacca, and Luca Trevisan. Consensus needs broadcast in noiseless models but can be exponentially easier in the presence of noise. *CoRR*, abs/1807.05626, 2018. URL: `http://arxiv.org/abs/1807.05626`, `arXiv:1807.05626`.

18 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. URL: `http://dl.acm.org/citation.cfm?id=365411.365756`.

19 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.

20 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.

21 E. W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Commun. ACM*, 17(11):643–644, 1974. URL: `http://doi.acm.org/10.1145/361179.361202`, `doi:10.1145/361179.361202`.

22 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.

23 S. Dolev. *Self-Stabilization*. The MIT Press, 2000.

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

25 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.

26 Pierre Fraigniaud and Emanuele Natale. Noisy rumor spreading and plurality consensus. *Distributed Computing*, pages 1–20, June 2018. URL: `https://link.springer.com/article/10.1007/s00446-018-0335-5`, `doi:10.1007/s00446-018-0335-5`.

27 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.

28 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.

29 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. URL: `http://dl.acm.org/citation.cfm?id=2095116.2095245`.

30 Bernhard Haeupler. Simple, fast and deterministic gossip and rumor spreading. *J. ACM*, 62(6):47:1–47:18, December 2015. URL: `http://doi.acm.org/10.1145/2767126`, `doi:10.1145/2767126`.

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

32 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. `doi:10.1109/SFCS.2002.1181883`.

33 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. URL: `http://doi.acm.org/10.1145/1835698.1835779`, `doi:10.1145/1835698.1835779`.

34 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. URL: `https://doi.org/10.1137/S0097539794279109`, `arXiv:https://doi.org/10.1137/S0097539794279109`, `doi:10.1137/S0097539794279109`.

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

36 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.

37 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.

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

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

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

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

42 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. URL: `http://dl.acm.org/citation.cfm?id=73228.73254`.