

## **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

**HAL Id: hal-01958994**

**<https://hal.inria.fr/hal-01958994v2>**

Submitted on 29 Nov 2019

**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.

# 1 Consensus vs Broadcast, with and without Noise 2 (Extended Abstract)

3 **Andrea Clementi**

4 Università *Tor Vergata* di Roma  
5 clementi@mat.uniroma2.it

6 **Luciano Gualà**

7 Università *Tor Vergata* di Roma  
8 guala@mat.uniroma2.it

9 **Emanuele Natale**

10 Université Côte d'Azur  
11 emanuele.natale@univ-cotedazur.fr

12 **Francesco Pasquale**

13 Università *Tor Vergata* di Roma  
14 pasquale@mat.uniroma2.it

15 **Giacomo Scornavacca**

16 Università degli Studi di Sassari  
17 giacomoscornavacca@gmail.com

18 **Luca Trevisan**

19 Università Bocconi  
20 l.trevisan@unibocconi.it

## 21 — Abstract —

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

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

32 We then turn to distributed models with noisy communication channels that have been studied  
33 in the context of some bio-inspired systems. In such models, only one noisy bit is exchanged when  
34 a communication channel is established between two nodes, and so one cannot easily simulate a  
35 noiseless protocol by using error-correcting codes. An  $\Omega(\varepsilon^{-2}n)$  lower bound is proved by Boczkowski  
36 et al. [PLOS Comp. Bio. 2018] on the convergence time of binary Broadcast in one such model  
37 (noisy uniform PULL), where  $\varepsilon$  is a parameter that measures the amount of noise).

38 We prove an  $O(\varepsilon^{-2} \log n)$  upper bound on the convergence time of binary Consensus in such  
39 model, thus establishing an exponential complexity gap between Consensus versus Broadcast.  
40 We also prove our upper bound above is tight and this implies, for binary Consensus, a further  
41 strong complexity gap between noisy uniform PULL and noisy uniform PUSH. Finally, we show a  
42  $\Theta(\varepsilon^{-2}n \log n)$  bound for Broadcast in the noisy uniform PULL.

43 **2012 ACM Subject Classification** Theory of computation → Distributed algorithms; Theory of  
44 computation → Random walks and Markov chains; Theory of computation → Random network  
45 models.

46 **Keywords and phrases** Distributed Computing, Consensus, Broadcast, Gossip Models, Noisy Com-  
47 munication Channels.



© Andrea Clementi, Luciano Gualà, Emanuele Natale, Francesco Pasquale, Giacomo Scornavacca and  
Luca Trevisan;  
licensed under Creative Commons License CC-BY

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

48 Digital Object Identifier 10.4230/LIPIcs.ITCS.2020.42

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

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

## 53 1 Introduction

54 In this paper we investigate the relation between Consensus and Broadcast, which are two of  
55 the most fundamental algorithmic problems in distributed computing [21, 23, 39, 41], and  
56 we study how the presence or absence of communication noise affects their complexity.

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

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

65 In the (binary) *Majority Consensus* problem [5, 22, 40] we are given the promise that  
66 one of the two possible opinions is initially held by at least  $n/2 + b(n)$  nodes, where  $b(n)$  is a  
67 parameter of the problem, and the goal is for the nodes to converge to a configuration in  
68 which they all have the opinion that, at the beginning, was held by the majority of nodes.  
69 Note that Consensus and Majority Consensus are incomparable problems: a protocol may  
70 solve one problem without solving the other.<sup>1</sup> Both the notions of Consensus and Majority  
71 Consensus above can be further relaxed to those of  $\delta$ -*Almost Consensus* and  $\delta$ -*Almost*  
72 *Majority Consensus*, respectively. According to such weaker notions, we allow the system to  
73 converge to an almost-consensus regime where  $\delta n$  *outliers* may have a different opinion from  
74 the rest of the nodes.

75 Motivations for studying the Broadcast problem are self-evident. Consensus and Majority  
76 Consensus are simplified models for the way inconsistencies and disagreements are resolved  
77 in social networks, biological models and peer-to-peer systems [24, 27, 37].<sup>2</sup>

78 In distributed model that severely restrict the way in which nodes communicate (to model  
79 constraints that arise in peer-to-peer systems or in social or biological networks), upper and  
80 lower bounds for the Broadcast problem give insights on the effect of the communication  
81 constraints on the way in which information can spread in the network. The analysis of  
82 algorithms for Consensus often give insights on how to break symmetry in distributed  
83 networks, when looking at how the protocol handles an initial opinion vector in which exactly

---

<sup>1</sup> 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.

<sup>2</sup> 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.

84 half the nodes have one opinion and half have the other. The analysis of algorithms for  
85 Majority Consensus usually hinge on studying the rate at which the number of nodes holding  
86 the minority opinion shrinks.

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

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

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

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

## 109 Roadmap of the paper and a remark

110 In order to formally state and discuss our results, in the next section, we introduce the  
111 distributed models and their associated complexity measures our results deal with. In Section  
112 3, we describe our results and their consequences for noiseless communication models and  
113 compare them with the related previous work. Section 4 is devoted to our results for the noisy  
114 communication models and to their comparison with the related previous work. Finally, in  
115 Section 5 we provide a short summary of the obtained results and discuss some related open  
116 questions. We remark that, in this version, we only sketch the main ideas of the technical  
117 proofs: detailed proofs are given in the full version of the paper [17].

## 118 **2** Communication and computational models

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

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

126 In the LOCAL model, there is no restriction on how many neighbors a node can talk to at  
127 each step, and no restriction on the number of bits transmitted at each step. There is also  
128 no restriction on the amount of memory and computational ability of each node. The only

129 complexity measures is the number of rounds of communication. For example, it is easy to  
 130 see that the complexity of Broadcast is the diameter of the graph  $G$ . The CONGEST model is  
 131 like the LOCAL model but the amount of data that each node can send at each time step is  
 132 limited, usually to  $O(\log n)$  bits.

133 In the (general) GOSSIP model [20, 30], at each time step, each node  $v$  chooses one of its  
 134 neighbors  $c_v$  and *activates* the communication link  $(v, c_v)$ , over which communication becomes  
 135 possible during that time step, allowing  $v$  to send a message to  $c_v$  and, simultaneously,  $c_v$  to  
 136 send a message to  $v$ . We will call  $v$  the *caller of  $c_v$* . In the PUSH variant, each node  $v$  sends  
 137 a message to its chosen neighbor  $c_v$ ; in the PULL variant, each node sends a message to its  
 138 callers (if any). Note that, although each node chooses only one neighbor, some nodes may  
 139 be chosen by several others, and so they may receive several messages in the PUSH setting, or  
 140 send a message to several recipients in the PULL setting. In our algorithmic results for the  
 141 GOSSIP model, we will assume that each message exchanged in each time step is only one  
 142 bit, while our negative results for the noiseless setting will apply to the case of messages of  
 143 unbounded length. In the *uniform* GOSSIP (respectively PUSH or PULL) model, the choice of  
 144  $c_v$  is done uniformly at random among the neighbors of  $v$ . This means that uniform models  
 145 make sense even in anonymous networks, in which nodes are not aware of their identities nor  
 146 of the identities of their neighbors.<sup>3</sup>

147 In this work, we are mainly interested in models like GOSSIP that severely restrict  
 148 communication [5, 2, 22, 24, 37, 40], both for efficiency consideration and because such  
 149 models capture aspects of the way consensus is reached in biological population systems,  
 150 and other domains of interest in network science [4, 6, 23, 12, 24, 25, 27]. Communication  
 151 capabilities in such scenarios are typically constrained and non-deterministic: both features  
 152 are well-captured by uniform models.

153 Asynchronous variants of the GOSSIP model (such as *Population Protocols* [5, 4]) have also  
 154 been extensively studied [11, 28, 40]. In this variant, no global clock is available to nodes.  
 155 Instead, nodes are idle until a single node is activated by a (possibly random) scheduler,  
 156 either in discrete time or in continuous time. When a node wakes up, it activates one of  
 157 its incident edges and wakes up the corresponding neighbor. Communication happens only  
 158 between those two vertices, which subsequently go idle again until the next time they wake  
 159 up.

160 Previous studies show that, in both PUSH and PULL variants of uniform GOSSIP, (binary)  
 161 Consensus, Majority Consensus and Broadcast can be solved within logarithmic time (and  
 162 work per node) in the complete graph, via elementary protocols<sup>4</sup>, with high probability (for  
 163 short *w.h.p.*<sup>5</sup>) [5, 9, 11, 22, 28, 31]. Moreover, efficient protocols have been proposed for  
 164 Broadcast and Majority Consensus for some restricted families of graphs such as regular  
 165 expanders and random graphs [1, 15, 14, 19, 29, 36].

166 However, while for Broadcast  $\Omega(\log n)$  time and work are necessary in the complete graph  
 167 [11, 28, 31], prior to this work, it was still unknown whether a more efficient protocol existed  
 168 for Consensus and Majority Consensus.

---

<sup>3</sup> 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.

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

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

### 169 **3 Our contribution I: Noiseless communication**

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

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

177 ► **Definition 1.** *A protocol  $\mathcal{P}$  solves the  $\gamma$ -Infection problem w.r.t. a source node  $s$  if it infects*  
 178 *at least  $\gamma n$  nodes, where we define a node infected recursively as follows: initially only  $s$  is*  
 179 *infected; a node  $v$  becomes infected whenever it receives any message from an infected node.*

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

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

194 In the next theorem, we show that lower bounds for Infection *also give lower bounds for*  
 195 *Consensus. More precisely we prove that if we have a Consensus protocol that, for every*  
 196 *initial opinion vector, succeeds in achieving almost consensus with probability  $1 - o(1/n)$ ,*  
 197 *then there is an initial opinion vector and a source such that the protocol infects many nodes*  
 198 *from that source with probability at least  $(1 - o(1))/n$ .*

199 ► **Theorem 2.** *Let  $\mathcal{P}$  be a protocol reaching  $\delta$ -Almost Consensus with probability at least*  
 200  *$1 - o(1/n)$ . Then, a source node  $s$  and an initial opinion vector  $\mathbf{x}$  exist such that  $\mathcal{P}$ , starting*  
 201 *from  $\mathbf{x}$ , solves the  $(1 - 2\delta)$ -Infection problem w.r.t.  $s$  with probability at least  $(1 - o(1))/n$ .*

202 Notice that the above result implies that any protocol for Consensus actually solves  
 203 the Infection problem (when initialized with a certain opinion vector) in a weak sense: the  
 204 infection is w.r.t. a source that depends on the consensus protocol in a (possibly) uncontrolled  
 205 manner; and (ii) the success probability of the infection is quite low. However, if we are  
 206 in a model in which there is no source for which we can have probability, say,  $\geq 1/(2n)$  of  
 207 infecting all nodes with certain resources (such as time, memory, communication per node,  
 208 etc.), then, in the same model, and with the same resources, the above theorem implies  
 209 that every Consensus protocol has probability  $\Omega(1/n)$  of failing. For example, by the above  
 210 argument, we have an  $\Omega(\log n)$  lower bound for Consensus in the PUSH model (because, in  
 211 fewer than  $\log_2 n$  rounds, the probability of infecting all nodes is zero).

212 In case of Consensus problem (i.e.  $\delta = 0$ ), our proof for Theorem 2 makes use of a hybrid  
 213 argument to show that there are two initial opinion vectors  $\mathbf{x}$  and  $\mathbf{y}$ , which are identical

214 except for the initial opinion of a node  $s$ , such that there is at least a  $(1 - o(1))/n$  difference  
 215 between the probability of converging to the all-zero configuration starting from  $\mathbf{x}$  or from  $\mathbf{y}$ .  
 216 Then, we prove that this difference must come entirely from runs of the protocol that fail to  
 217 achieve consensus (which happens only with  $o(1/n)$  probability) or from runs of the protocol  
 218 in which  $s$  infects all other nodes. Thus the probability that  $s$  infects all nodes from the  
 219 initial vector  $\mathbf{x}$  has to be  $\geq (1 - o(1))/n$ . Then, to extend the above approach for the Almost  
 220 Consensus problem (i.e.  $\delta > 0$ ), some additional care and a suitable counting argument are  
 221 required to manage the unknown set of outliers.

222 As for Majority Consensus, we have a similar reduction, but from a variant of the infection  
 223 problem in which there is an initial set of  $b$  infected nodes.<sup>6</sup> Formally:

224 ► **Theorem 3.** *Let  $\mathcal{T}$  be any fixed resource defined on a distributed system  $\mathcal{S}$  and suppose*  
 225 *there is no Infection protocol that, starting from any subset of  $n^\alpha$  nodes with  $\alpha < 1$ , can*  
 226 *inform at least  $(1 - \delta)n$  nodes by using at most  $\tau^B$  units of  $\mathcal{T}$ , w.h.p. Then, any protocol  $\mathcal{P}$*   
 227 *on this model, reaching  $\delta$ -Almost Majority Consensus w.h.p., must use more than  $\tau^B$  units*  
 228 *of  $\mathcal{T}$ .*

### 229 3.1 Some applications

230 Lower bounds for infection are known in several models in which there are no previous  
 231 negative results for Consensus. We have not attempted to survey all possible applications  
 232 of our reductions, but here we enumerate some of them (see the full version for the formal  
 233 statements of such results):

- 234 ■ In the uniform Gossip model (also known as uniform PUSH-PULL model), and in the  
 235 general PUSH model, tight analysis (see [30, 31]) show that any protocol  $\mathcal{P}$  for the complete  
 236 graph w.h.p. does not complete Broadcast within less than  $\beta \log n$  rounds, where  $\beta$  is a  
 237 sufficiently small constant. Combining this lower bound with our reduction result above,  
 238 we get an  $\Omega(\log n)$  lower bound for Consensus. This is the first known lower bound for  
 239 Consensus showing a full equivalence between the complexity of Broadcast and Consensus  
 240 in such models. Regarding Majority Consensus, we also obtain an  $\Omega(\log n)$  lower bound  
 241 for any initial bias  $b = O(n^\alpha)$ , with  $\alpha < 1$ .
- 242 ■ In a similar way, we are able to prove a lower bound of  $\Omega(n \log n)$  number of steps  
 243 (and hence  $\Omega(\log n)$  parallel time) or  $\Omega(\log n)$  number of messages per node for Con-  
 244 sensus on an asynchronous variant of the Gossip model, the *Population Protocols* with  
 245 uniform/probabilistic scheduler, as defined in [5].
- 246 ■ The last application we mention here concerns the synchronous *Radio Network* model  
 247 [3, 7, 16, 42]. Several optimal bounds have been obtained on the Broadcast time [7, 18,  
 248 32, 33, 34] while only few results are known for Consensus time [16, 42]. In particular, we  
 249 are not aware of better lower bounds other than the trivial  $\Omega(D)$  (where  $D$  denotes the  
 250 diameter of the network). Then, by combining a previous lower bound in [3] on Broadcast  
 251 with our reduction result, we get a new lower bound for Consensus in this model.

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

---

<sup>6</sup> 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  $0 \leq \varepsilon < 1/2$ , and we refer to  $\varepsilon$  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  $\varepsilon$  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<sup>7</sup>. 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

<sup>7</sup> They actually proved a more general result including non-binary noisy channels.



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

## 301 4.1 Upper bounds in noisy uniform Pull

### 302 4.1.1 A protocol for Consensus and its analysis

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

306 ► **Theorem 4.** *In the noisy uniform PULL model, with noisy parameter  $\varepsilon$ , a protocol exists  
307 that achieves Consensus within  $O(\varepsilon^{-2} \log n)$  rounds and communication, w.h.p. The protocol  
308 requires  $\Theta(\log \log n + \log \varepsilon^{-2})$  local memory.  
309 Moreover, if the protocol starts from any initial opinion vector with bias  $b = \Omega(\sqrt{n \log n})$ ,  
310 then it guarantees Majority Consensus, w.h.p.*

311 The protocol we refer to in the above theorem works in two consecutive phases. Each  
312 phase is a simple application of the well-known *k-Majority Dynamics* [8, 9]:

313 *k-MAJORITY. At every round, each node samples  $k$  neighbours<sup>8</sup> independently and  
314 u.a.r. (with replacement). Then, the node updates its opinion according to the majority  
315 opinion in the sample.*

316 The protocol is thus the following:

317 **MAJORITY PROTOCOL.** Let  $\alpha$  be a sufficiently large positive constant<sup>9</sup>. Every node  
318 performs  $\alpha \log n$  rounds of *k-Majority* with  $k = \Theta(1/\varepsilon^2)$ , followed by one round of the  
319 *k-Majority* with  $k = \Theta(\varepsilon^{-2} \log n)$ .

320 Our analysis shows that, w.h.p., at the end of the first phase there is an opinion that is held  
321 by at least  $n/2 + \Omega(n)$  nodes, and that if the initial opinions were unanimous then the initial  
322 opinion is the majority opinion after the first phase (notice that the latter fact guarantees the  
323 validity property, w.h.p.). Then, in the second phase, despite the communication errors, we  
324 show every node has a high probability of seeing the true phase-one majority as the empirical  
325 majority in the batch and so all nodes converge to the same valid opinion. To analyze the  
326 first phase, we break it out into two sub-phases (this breakdown is only in the analysis, not  
327 in the protocol): in a first sub-phase of length  $O(\varepsilon^{-2} \log n)$ , we prove the protocol “breaks  
328 symmetry” w.h.p. and, no matter the initial vector and the presence of communication  
329 noise, reaches a configuration in which one opinion is held by  $n/2 + \Omega(\sqrt{n \log n})$  nodes. In  
330 the second sub-phase, also of length  $O(\varepsilon^{-2} \log n)$ , a configuration of bias  $\Omega(\sqrt{n \log n})$  w.h.p.  
331 becomes a configuration of bias  $\Omega(n)$ . The analysis of the first sub-phase is our main technical  
332 novelty while the analysis of the second sub-phase for achieving Majority Consensus is similar  
333 to that in [25, 26]. If the initial opinion vector is unanimous, then it is not necessary to break  
334 up the first phase into sub-phases, and one can directly see that a unanimous configuration  
335 maintains a bias  $\Omega(n)$ , w.h.p., for the duration of the first phase.

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

<sup>8</sup> 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.

<sup>9</sup> The value of  $\alpha$  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  $\delta n$  outliers and with error probability at most  $\delta$  requires  $\Omega(\varepsilon^{-2} \log \delta^{-1})$  rounds. Formally:

► **Theorem 6.** *Let  $\delta$  be any real such that  $0 < \delta < 1/8$  and consider any protocol  $\mathcal{P}$  for the noisy general PULL model with noise parameter  $\varepsilon$ . If  $\mathcal{P}$  solves  $\delta$ -Almost Consensus with probability at least  $1 - \delta$ , then it requires at least  $t = \Omega(\varepsilon^{-2} \log \delta^{-1})$  rounds<sup>10</sup>.*

This shows that the complexity  $\mathcal{O}(\varepsilon^{-2} \log n)$  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  $\varepsilon$ , 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  $\varepsilon$ ), 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  $\Omega(\varepsilon^{-2} \log \delta^{-1})$  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  $\varepsilon$ . 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

<sup>10</sup>We notice the double role parameter  $\delta$  has in this statement.

380 graph (consider the protocol in which one node broadcasts its opinion), and it is not clear if  
 381 there is a distribution of initial opinions such that there is a node  $v$  whose final opinion has  
 382 mutual information close to 1 to the global initial opinion vector given the initial opinion of  
 383  $v$  (the natural generalization of the argument of [25]). Instead, we prove that there are two  
 384 initial opinion vectors  $\mathbf{x}$  and  $\mathbf{y}$ , a node  $v$ , and a bit  $b$ , such that the initial opinion of  $v$  is the  
 385 same in  $\mathbf{x}$  and  $\mathbf{y}$ , but the probability that  $v$  outputs  $b$  is  $\leq \delta$  when the initial opinion vector  
 386 is  $\mathbf{x}$  and  $\geq \Omega(1)$  when the initial opinion vector is  $\mathbf{y}$ . Thus, the rest of the graph is sending  $v$   
 387 a bit of information (whether the initial opinion vector is  $\mathbf{x}$  or  $\mathbf{y}$ ) and the communication  
 388 succeeds with probability  $\geq 1 - \delta$  when the bit has one value and with probability  $\geq 1/3$  if  
 389 the bit has the other value. Despite this asymmetry, if the communication takes place over a  
 390 binary symmetric channel with error parameter  $\varepsilon$ , we use KL divergence to show that the  
 391 channel has to be used  $\Omega(\varepsilon^{-2} \log \delta^{-1})$  times.

### 392 4.2.1 An improved lower bound for Broadcast

393 The  $\Omega(\varepsilon^{-2}n)$  lower bound of [10] for Broadcast in the uniform PULL model applies to  
 394 protocols that have constant probability of correctly performing the broadcast operation.  
 395 With the following theorem we show a way of modifying their proof (in particular, to derive  
 396 an  $\Omega(\varepsilon^{-2}n \log n)$  for uniform PULL protocols for Broadcast that have high probability of  
 397 success, matching the  $O(\varepsilon^{-2}n \log n)$  round complexity of Theorem 5.

398 ► **Theorem 7.** *The Broadcast Problem cannot be solved in the noisy uniform PULL model*  
 399 *w.h.p. in less than  $\Omega(\varepsilon^{-2}n \log n)$  rounds.*

## 400 5 Conclusions

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

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

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

## 419 — References —

- 420 1 Mohammed Amin Abdullah and Moez Draief. Majority consensus on random graphs of a  
 421 given degree sequence. *CoRR*, abs/1209.5025, 2012. URL: <http://arxiv.org/abs/1209.5025>,  
 422 [arXiv:1209.5025](https://arxiv.org/abs/1209.5025).

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

- 423 2 Mohammed Amin Abdullah and Moez Draief. Global majority consensus by local majority  
424 polling on graphs of a given degree sequence. *Discrete Applied Mathematics*, 180:1–10, 2015.
- 425 3 Noga Alon, Amotz Bar-Noy, Nathan Linial, and David Peleg. A lower bound for radio  
426 broadcast. *Journal of Computer and System Sciences*, 43(2):290 – 298, 1991. URL: [http://](http://www.sciencedirect.com/science/article/pii/002200009190015W)  
427 [www.sciencedirect.com/science/article/pii/002200009190015W](http://www.sciencedirect.com/science/article/pii/002200009190015W), doi:[https://doi.org/](https://doi.org/10.1016/0022-0000(91)90015-W)  
428 [10.1016/0022-0000\(91\)90015-W](https://doi.org/10.1016/0022-0000(91)90015-W).
- 429 4 Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and Peralta René. Computation  
430 in networks of passively mobile finite-state sensors. *Distributed Computing*, 18(4):235–253,  
431 2006.
- 432 5 Dana Angluin, James Aspnes, and David Eisenstat. A Simple Population Protocol for Fast  
433 Robust Approximate Majority. *Distributed Computing*, 21(2):87–102, 2008. (Preliminary  
434 version in DISC’07).
- 435 6 Dana Angluin, Michael J. Fischer, and Hong Jiang. Stabilizing consensus in mobile networks.  
436 In *Proc. of Distributed Computing in Sensor Systems (DCOSS’06)*, volume 4026 of *LNCS*,  
437 pages 37–50, 2006.
- 438 7 Reuven Bar-Yehuda, Oded Goldreich, and Alon Itai. On the time-complexity of broadcast  
439 in multi-hop radio networks: An exponential gap between determinism and randomiza-  
440 tion. *Journal of Computer and System Sciences*, 45(1):104 – 126, 1992. URL: [http://](http://www.sciencedirect.com/science/article/pii/002200009290042H)  
441 [www.sciencedirect.com/science/article/pii/002200009290042H](http://www.sciencedirect.com/science/article/pii/002200009290042H), doi:[https://doi.org/](https://doi.org/10.1016/0022-0000(92)90042-H)  
442 [10.1016/0022-0000\(92\)90042-H](https://doi.org/10.1016/0022-0000(92)90042-H).
- 443 8 Luca Becchetti, Andrea Clementi, Emanuele Natale, Francesco Pasquale, and Luca Trevisan.  
444 Stabilizing consensus with many opinions. In *Proc. of the 27th Ann. ACM-SIAM Symp. on*  
445 *Discrete algorithms*, pages 620–635. SIAM, 2016.
- 446 9 Luca Becchetti, Andrea E.F. Clementi, Emanuele Natale, Francesco Pasquale, Riccardo  
447 Silvestri, and Luca Trevisan. Simple dynamics for plurality consensus. In *ACM SPAA’14*,  
448 pages 247–256, 2014.
- 449 10 Lucas Boczkowski, Emanuele Natale, Ofer Feinerman, and Amos Korman. Limits on re-  
450 liable information flows through stochastic populations. *PLOS Computational Biology*,  
451 14(6):e1006195, June 2018. URL: <http://dx.plos.org/10.1371/journal.pcbi.1006195>,  
452 doi:[10.1371/journal.pcbi.1006195](https://doi.org/10.1371/journal.pcbi.1006195).
- 453 11 Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip  
454 algorithms. *IEEE/ACM Transactions on Networking*, 14:2508–2530, 2006. URL: [http://](http://dx.doi.org/10.1109/TIT.2006.874516)  
455 [dx.doi.org/10.1109/TIT.2006.874516](http://dx.doi.org/10.1109/TIT.2006.874516), doi:[10.1109/TIT.2006.874516](https://doi.org/10.1109/TIT.2006.874516).
- 456 12 L. Cardelli and A. Csikász-Nagy. The cell cycle switch computes approximate majority.  
457 *Scientific Reports*, Vol. 2, 2012.
- 458 13 Bernard Chazelle. Natural algorithms and influence systems. *Commun. ACM*, 55(12):101–  
459 110, December 2012. URL: <http://doi.acm.org/10.1145/2380656.2380679>, doi:[10.1145/](https://doi.org/10.1145/2380656.2380679)  
460 [2380656.2380679](https://doi.org/10.1145/2380656.2380679).
- 461 14 Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. Almost Tight Bounds for  
462 Rumour Spreading with Conductance. In *Proceedings of the Forty-second ACM Symposium*

- 463        *on Theory of Computing*, STOC '10, pages 399–408, New York, NY, USA, 2010. ACM. URL:  
464        <http://doi.acm.org/10.1145/1806689.1806745>, doi:10.1145/1806689.1806745.
- 465 **15** Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. Rumor spreading in social  
466 networks. *Theoretical Computer Science*, 412(24):2602 – 2610, 2011. Selected Papers from 36th  
467 International Colloquium on Automata, Languages and Programming (ICALP 2009). URL:  
468        <http://www.sciencedirect.com/science/article/pii/S0304397510006122>, doi:<https://doi.org/10.1016/j.tcs.2010.11.001>.
- 470 **16** Gregory Chockler, Murat Demirbas, Seth Gilbert, Nancy Lynch, Calvin Newport, and Tina  
471 Nolte. Consensus and collision detectors in radio networks. *Distributed Computing*, 21(1):55–  
472 84, June 2008. URL: <https://link.springer.com/article/10.1007/s00446-008-0056-2>,  
473        doi:10.1007/s00446-008-0056-2.
- 474 **17** Andrea E. F. Clementi, Luciano Gualà, Emanuele Natale, Francesco Pasquale, Giacomo  
475 Scornavacca, and Luca Trevisan. Consensus needs broadcast in noiseless models but can  
476 be exponentially easier in the presence of noise. *CoRR*, abs/1807.05626, 2018. URL: <http://arxiv.org/abs/1807.05626>, arXiv:1807.05626.
- 478 **18** Andrea E. F. Clementi, Angelo Monti, and Riccardo Silvestri. Selective families, superimposed  
479 codes, and broadcasting on unknown radio networks. In *Proceedings of the Twelfth Annual*  
480 *ACM-SIAM Symposium on Discrete Algorithms*, SODA '01, pages 709–718, Philadelphia,  
481 PA, USA, 2001. Society for Industrial and Applied Mathematics. URL: [http://dl.acm.org/](http://dl.acm.org/citation.cfm?id=365411.365756)  
482        [citation.cfm?id=365411.365756](http://dl.acm.org/citation.cfm?id=365411.365756).
- 483 **19** C. Cooper, R. Elsasser, and T. Radzik. The power of two choices in distributed voting. In  
484 *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming*  
485 *(ICALP'14)*, volume 8573 of *LNCS*, pages 435–446. Springer, 2014.
- 486 **20** A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart,  
487 and D. Terry. Epidemic algorithms for replicated database maintenance. In *ACM PODC'87*,  
488 1987.
- 489 **21** E. W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Commun. ACM*,  
490 17(11):643–644, 1974. URL: <http://doi.acm.org/10.1145/361179.361202>, doi:10.1145/  
491 361179.361202.
- 492 **22** Benjamin Doerr, Leslie Ann Goldberg, Lorenz Minder, Thomas Sauerwald, and Christian  
493 Scheideler. Stabilizing consensus with the power of two choices. In *ACM SPAA'11*, pages  
494 149–158, 2011.
- 495 **23** S. Dolev. *Self-Stabilization*. The MIT Press, 2000.
- 496 **24** David Doty. Timing in chemical reaction networks. In *ACM-SIAM SODA'14*, pages 772–784,  
497 2014.
- 498 **25** Ofer Feinerman, Bernhard Haeupler, and Amos Korman. Breathe Before Speaking: Effi-  
499 cient Information Dissemination Despite Noisy, Limited and Anonymous Communication.  
500 *Distributed Computing*, 30(5):239–355, 2017. Ext. Abs. in ACM PODC'14.
- 501 **26** Pierre Fraigniaud and Emanuele Natale. Noisy rumor spreading and plurality consensus.  
502 *Distributed Computing*, pages 1–20, June 2018. URL: [https://link.springer.com/article/](https://link.springer.com/article/10.1007/s00446-018-0335-5)  
503        [10.1007/s00446-018-0335-5](https://link.springer.com/article/10.1007/s00446-018-0335-5), doi:10.1007/s00446-018-0335-5.
- 504 **27** Nigel R. Franks, Stephen C. Pratt, Eamonn B. Mallon, Nicholas F. Britton, and David J.T.  
505 Sumpter. Information flow, opinion polling and collective intelligence in house-hunting social  
506 insects. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*,  
507 357(1427):1567–1583, 2002.
- 508 **28** George Giakkoupis, Yasamin Nazari, and Philipp Woelfel. How asynchrony affects rumor  
509 spreading time. In *35th ACM Symposium on Principles of Distributed Computing (PODC*  
510 *2016)*, 2016.
- 511 **29** George Giakkoupis and Thomas Sauerwald. Rumor Spreading and Vertex Expansion. In  
512 *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, SODA  
513 '12, pages 1623–1641, Philadelphia, PA, USA, 2012. Society for Industrial and Applied  
514 Mathematics. URL: <http://dl.acm.org/citation.cfm?id=2095116.2095245>.

- 515 30 Bernhard Haeupler. Simple, fast and deterministic gossip and rumor spreading. *J. ACM*,  
516 62(6):47:1–47:18, December 2015. URL: <http://doi.acm.org/10.1145/2767126>, doi:10.  
517 1145/2767126.
- 518 31 R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. Randomized rumor spreading. In  
519 *IEEE FOCS'00*, pages 565–574, 2000.
- 520 32 D. R. Kowalski and A. Pelc. Deterministic broadcasting time in radio networks of unknown  
521 topology. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002.*  
522 *Proceedings.*, pages 63–72, 2002. doi:10.1109/SFCS.2002.1181883.
- 523 33 Fabian Kuhn, Nancy Lynch, Calvin Newport, Rotem Oshman, and Andrea Richa. Broadcasting  
524 in unreliable radio networks. In *Proceedings of the 29th ACM SIGACT-SIGOPS Symposium*  
525 *on Principles of Distributed Computing*, PODC '10, pages 336–345, New York, NY, USA,  
526 2010. ACM. URL: <http://doi.acm.org/10.1145/1835698.1835779>, doi:10.1145/1835698.  
527 1835779.
- 528 34 E. Kushilevitz and Y. Mansour. An  $\omega(d \log(n/d))$  lower bound for broadcast in radio  
529 networks. *SIAM Journal on Computing*, 27(3):702–712, 1998. URL: [https://doi.org/](https://doi.org/10.1137/S0097539794279109)  
530 [10.1137/S0097539794279109](https://doi.org/10.1137/S0097539794279109), arXiv:<https://doi.org/10.1137/S0097539794279109>, doi:  
531 [10.1137/S0097539794279109](https://doi.org/10.1137/S0097539794279109).
- 532 35 David J. C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge  
533 University Press, 2003.
- 534 36 G. B. Mertzios, S. E. Nikolettseas, C. Raptopoulos, and P. G. Spirakis. Determining majority  
535 in networks with local interactions and very small local memory. In *Proceedings of the 41st*  
536 *International Colloquium on Automata, Languages, and Programming (ICALP'14)*, 2014.
- 537 37 Elchanan Mossel, Joe Neeman, and Omer Tamuz. Majority dynamics and aggregation of  
538 information in social networks. *Autonomous Agents and Multi-Agent Systems*, 28(3):408–429,  
539 2014.
- 540 38 Saket Navlakha and Ziv Bar-Joseph. Distributed information processing in biological and  
541 computational systems. *Communications of the ACM*, 58(1):94–102, 2015.
- 542 39 Marshall Pease, Robert Shostak, and Leslie Lamport. Reaching agreement in the presence of  
543 faults. *Journal of the ACM*, 27(2):228–234, 1980.
- 544 40 Etienne Perron, Dinkar Vasudevan, and Milan Vojnovic. Using Three States for Binary  
545 Consensus on Complete Graphs. In *IEEE INFOCOM'09*, pages 2527–2535, 2009.
- 546 41 Michael O. Rabin. Randomized byzantine generals. In *Proc. of the 24th Ann. Symp. on*  
547 *Foundations of Computer Science (SFCS)*, pages 403–409. IEEE, 1983.
- 548 42 N. Santoro and P. Widmayer. Time is Not a Healer. In *Proceedings of the 6th Annual*  
549 *Symposium on Theoretical Aspects of Computer Science on STACS 89*, pages 304–313, New  
550 York, NY, USA, 1989. Springer-Verlag New York, Inc. URL: [http://dl.acm.org/citation.](http://dl.acm.org/citation.cfm?id=73228.73254)  
551 [cfm?id=73228.73254](http://dl.acm.org/citation.cfm?id=73228.73254).