

Agreement Tasks in Fault-Prone Synchronous Networks of Arbitrary Structure

Pierre Fraigniaud   

Institut de Recherche en Informatique Fondamentale (IRIF), CNRS, Université Paris Cité, France

Minh Hang Nguyen   

Institut de Recherche en Informatique Fondamentale (IRIF), CNRS, Université Paris Cité, France

Ami Paz   

Laboratoire Interdisciplinaire des Sciences du Numérique (LISN), CNRS, Université Paris-Saclay, France

Abstract

Consensus is arguably the most studied problem in distributed computing as a whole, and particularly in the distributed message-passing setting. In this latter framework, research on consensus has considered various hypotheses regarding the failure types, the memory constraints, the algorithmic performances (e.g., early stopping and obliviousness), etc. Surprisingly, almost all of this work assumes that messages are passed in a *complete* network, i.e., each process has a direct link to every other process. A noticeable exception is the recent work of Castañeda et al. (Inf. Comput. 2023) who designed a generic oblivious algorithm for consensus running in $\text{radius}(G, t)$ rounds in every graph G , when up to t nodes can crash by irrevocably stopping, where t is smaller than the node-connectivity κ of G . Here, $\text{radius}(G, t)$ denotes a graph parameter called the *radius of G whenever up to t nodes can crash*. For $t = 0$, this parameter coincides with $\text{radius}(G)$, the standard radius of a graph, and, for $G = K_n$, the running time $\text{radius}(K_n, t) = t + 1$ of the algorithm exactly matches the known round-complexity of consensus in the clique K_n .

Our main result is a proof that $\text{radius}(G, t)$ rounds are necessary for oblivious algorithms solving consensus in G when up to t nodes can crash, thus validating a conjecture of Castañeda et al., and demonstrating that their consensus algorithm is optimal for any graph G . We also extend the result of Castañeda et al. to two different settings: First, to the case where the number t of failures is not necessarily smaller than the connectivity κ of the considered graph; Second, to the k -set agreement problem for which agreement is not restricted to be on a single value as in consensus, but on up to k different values.

2012 ACM Subject Classification Theory of computation \rightarrow Distributed algorithms

Keywords and phrases Consensus, set-agreement, fault tolerance, crash failures

Digital Object Identifier 10.4230/LIPIcs.STACS.2025.34

Related Version *Full Version*: <https://arxiv.org/pdf/2410.21538> [16]

Funding *Pierre Fraigniaud*: Additional support from ANR projects DUCAT (ANR-20-CE48-0006), ENEDISC, and QuDATA (ANR-18-CE47-0010).

Minh Hang Nguyen: Additional support from ANR projects DUCAT (ANR-20-CE48-0006), TEMPORAL (ANR-22-CE48-0001), and ENEDISC, and by the European Union's Horizon 2020 program H2020-MSCA -COFUND-2019 Grant agreement n° 945332.

Acknowledgements The authors thank Stephan Felber, Mikaël Rabie, Hugo Rincon Galeana, and Ulrich Schmid for fruitful discussions on this paper.

1 Introduction

For $t \geq 0$, the standard *synchronous t -resilient message-passing* model assumes $n \geq 2$ nodes labeled from 1 to n , and connected as a clique, i.e., as a complete graph K_n . Computation proceeds as a sequence of synchronous rounds, during which every node can send a message



© Pierre Fraigniaud, Minh Hang Nguyen, and Ami Paz;
licensed under Creative Commons License CC-BY 4.0

42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025).

Editors: Olaf Beyersdorff, Michał Pilipczuk, Elaine Pimentel, and Nguyễn Kim Thăng;

Article No. 34; pp. 34:1–34:21



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



to each other node, receive the message sent by each other node, and perform some local computation. Up to t nodes may crash during the execution of an algorithm. When a node v crashes at some round $r \geq 1$, it stops functioning after round r and never recovers. Moreover, some (possibly all) of the messages sent by v at round r may be lost, that is, when v crashes, messages sent by v at round r may reach some neighbors, while other neighbors of v may not hear from v at round r . This model has been extensively studied in the literature (see, e.g., [2, 18, 23, 27]). In particular, it is known that consensus can be solved in $t + 1$ rounds in the t -resilient model [13], and this is optimal for every $t < n - 1$ as far as the worst-case complexity is concerned [1, 13].

It is only very recently that the synchronous t -resilient message-passing model has been extended to the setting in which the complete communication graph K_n is replaced by an arbitrary communication graph G (see [4, 8]). Specifically, the graph G is fixed, but arbitrary, and the concern is to design algorithms for G . It was proved in [4] that if the number of failures is smaller than the connectivity of the graph, i.e., if $t < \kappa(G)$, then consensus in G can be solved in $\text{radius}(G, t)$ rounds in the t -resilient model, where $\text{radius}(G, t)$ generalizes the standard notion of graph radius to the scenarios in which up to t nodes may fail by crashing. For $t = 0$, $\text{radius}(G, 0)$ is the standard radius of the graph G . For the complete graph K_n , the $\text{radius}(K_n, t)$ upper bound from [4] coincides with the seminal $t + 1$ upper bound for consensus in K_n .

To get an intuition of $\text{radius}(G, t)$, let us consider the case of the n -node cycle C_n , for $n \geq 3$. We have $\kappa(C_n) = 2$, so we assume $t \leq 1$. The radius of C_n is $\lfloor \frac{n}{2} \rfloor$, i.e., $\text{radius}(C_n, 0) = \lfloor \frac{n}{2} \rfloor$. For $t = 1$, let v be the node that crashes. We have $\text{radius}(C_n, 1) \geq n - 2$, which is the distance between the two neighbors of v in C_n if v crashes “cleanly” at the first round, preventing them to communicate directly through v . However, we actually have $\text{radius}(C_n, 1) = n - 1$. Indeed, v may crash at the first round, yet be capable to send a message to one of its neighbors, and this message needs $n - 2$ additional rounds to reach the other neighbor of v . That is, computing $\text{radius}(G, t)$ requires to take into account not only which nodes crash, but when and how they are crashing – by “how”, it is meant that, for a node v crashing at some round r , to which neighbors they still succeed to communicate at this round, and to which they fail to communicate.

Importantly, the algorithm of [4] is *oblivious*, that is, the output of a node after $\text{radius}(G, t)$ rounds is solely based on the set of pairs (*node-identifier*, *input-value*) collected by that node during $\text{radius}(G, t)$ rounds (and not, e.g., from whom, when, and how many times it received each of these pairs). There are many reasons why to restrict the study to oblivious algorithms. Among them, oblivious algorithms are simple by design, which is desirable for their potential implementation. Moreover, they are known to be efficient, as illustrated by the case of the complete graphs in which optimal solutions can be obtained thanks to oblivious algorithms. As far as this paper is concerned (and maybe also as far as [4] is concerned) obliviousness is highly desirable for the design of *generic* solutions, that is “meta-algorithms” that apply to each and every graph G . In such algorithms, every node forwards pairs (*node-identifier*, *input-value*) during a prescribed number of rounds (e.g., during $\text{radius}(G, t)$ rounds in the generic algorithm from [4]), and then decides on an output value according to a simple function of the set of input values received during these rounds, without having to track of the sequence of rounds at which each pair was received, and from which neighbor(s). Last but not least, intermediate nodes do not need to send complex information about the history of each piece of information transmitted during the execution, hence reducing the bandwidth requirement of the algorithms.

1.1 Objective

The question of the optimality of the consensus algorithm performing in $\text{radius}(G, t)$ rounds in any fixed graph G for every number $t \leq \kappa(G)$ of failures was however left open in [4]. It was conjectured in [4] that, for every graph G , and for every $0 \leq t < \kappa(G)$, no oblivious algorithms can solve consensus in G in less than $\text{radius}(G, t)$ rounds, but this was only proved for the specific case of *symmetric* (a.k.a. *vertex-transitive*) graphs¹. Although the class of symmetric graphs includes, e.g., the complete graphs K_n , the cycles C_n , and the d -dimensional hypercubes Q_d , a lower bound $\text{radius}(G, t)$ for every graph G in this class does not come entirely as a surprise since all nodes of a symmetric graph have the same eccentricity (i.e., maximum distance to any other node, generalized to include crash failures). The fact that all nodes have the same eccentricity implies that they can merely be ordered according to their identifiers for selecting the output value from the received pairs (*node-identifier*, *input-value*). Instead, if the graph is not symmetric, a node that received a pair (*node-identifier*, *input-value*) after $\text{radius}(G, t)$ rounds does not necessarily know whether all the nodes have received this pair, and thus the choice of the output value from the set of received pairs is more subtle. Not only the design of an upper bound is made harder, but it also makes the determination of a strong lower bound more involved. The main question addressed in this paper is therefore the following: For every graph G , and every non-negative integer $t < \kappa(G)$, is there an oblivious algorithm solving consensus in G in less than $\text{radius}(G, t)$ rounds under the t -resilient model (i.e., when up to t nodes may fail by crashing)?

Moreover, the study in [4] let aside the design of a generic (oblivious) algorithm for solving the standard important relaxation of consensus, namely k -set agreement. (Recall that, in k -set agreement, the set of all values outputted by the nodes must be of cardinality at most k .) In fact, several tools developed in [4] do not extend to k -set agreement. Our next step is therefore to question the ability to design a generic algorithm for solving k -set agreement in arbitrary graphs G , for every $k > 1$.

Last but not least, the study in [4] assumed that the number t of failures is smaller than the connectivity $\kappa(G)$ of the graph G at hand. We question what can be said about the case where the number of failures may be larger, that is when $t \geq \kappa(G)$, for both consensus and k -set agreement?

1.2 Our Results

We extend the investigation of the t -resilient model in arbitrary graphs, in various complementary directions.

Lower Bounds for Consensus. We affirmatively prove the conjecture from [4] that their consensus algorithm is indeed optimal (among oblivious algorithms) for *every* graph G , and not only for symmetric graphs. That is, we show that, for every graph G , no oblivious algorithms can solve consensus in G in less than $\text{radius}(G, t)$ rounds under the t -resilient model. This result is achieved by revisiting the notion of *information flow graph* defined in [4] for fixing some inaccuracies in the original definition. We present a more robust (an accurate) definition of information flow graph, and we provide a characterization of the number of rounds required to solve consensus as a function of some structural property of

¹ A graph $G = (V, E)$ is vertex-transitive if, for every two nodes $u \neq v$, there exists an automorphism f of G (i.e., a permutation $f : V \rightarrow V$ preserving the edges and the non-edges of G) such that $f(u) = v$.

that graph. With this characterization at hand, we establish the optimality of the algorithm in [4] by showing that $\text{radius}(G, t)$ rounds are necessary for the information flow graph to satisfy the desired property required for consensus solvability.

Beyond the Connectivity Threshold. Inspired by [8], we extend the study of consensus in the t -resilient model in arbitrary graphs to the case where the number t of crash failures is arbitrary, i.e., not necessarily lower than the connectivity $\kappa(G)$ of the considered graph G . We show that the generic algorithm from [4] can be extended to this framework, at the mere cost of relaxing consensus to impose agreement to hold within each connected component of the graph resulting from removing the faulty nodes from G . Under this somehow unavoidable relaxation, we present extension of the consensus algorithm from [4] to t -resilient models for $t \geq \kappa(G)$, and express the round complexities of these algorithms in term of a non-trivial extension of the radius notion to disconnected graphs.

Extension to Set Agreement. Finally, we extend the study of consensus under the t -resilient model in arbitrary graphs to k -set agreement, for an arbitrary fixed $k \geq 1$. We show that, for every integer $k \geq 1$, and every graph G , there exists an oblivious k -set agreement algorithm performing in $\text{radius}(G, t, k)$ rounds, where $\text{radius}(G, t, k)$ denotes a parameter extending $\text{radius}(G, t)$ to the case where k nodes broadcast instead of just one, with the objective that each (non faulty) node receives a pair (*node-identifier*, *input-value*) from at least one of these k nodes. This extension holds for every t . For $t \geq \kappa(G)$ the k -set agreement tasks must however be relaxed similarly to consensus, so that agreement hold within each connected component of the graph separately. Due to lack of space, our results on k -set agreement are not included in this extended abstract, but they can be found in the full version of the paper (see [16]).

1.3 Related Work

Distributed computing in synchronous networks has a long tradition, including the early studies of the message complexity and round complexity of various tasks such as leader election, spanning tree constructions, BFS and DFS traversals, etc. (see, e.g., [2, 23]). The topic has then flourished in the 2000s under the umbrella of the so-called LOCAL and CONGEST models [19, 25], with the study of numerous graph problems such as coloring, maximal independent set, minimum-weight spanning tree, etc.

Distributed computing in synchronous *fault-prone* networks has also a long history, but it remained for a long time mostly confined to the special case of the message-passing model in the complete networks. That is, n nodes subject to *crash* or *malicious* (a.k.a. Byzantine) failures are connected as a complete graph K_n in which every pair of nodes has a private reliable link allowing them to exchange messages. In this setting, a significant amount of effort has been dedicated to narrowing down the complexity of solving agreement tasks such as consensus and, more generally, k -set agreement for $k \geq 1$. This includes in particular the issue of *early stopping* algorithms whose performances depend on the actual number of failures f experienced during the execution of the algorithm, and not on the upper bound t on the number of failures. We refer to a sequence of surveys on the matter [5, 26, 28].

In the Byzantine case, general communication graphs were studied early on [12], and are still being investigated [20]. In the stop-fault case, on the other hand, it is only recently that this approach has been extended to arbitrary networks, beyond the case of the complete graph K_n [4, 8]. Our paper is carrying on the preliminary investigations in [4], by extending them from consensus to k -set agreement, establishing various lower bounds including one

demonstrating the optimality of the consensus algorithm in [4], and extending the analysis to the case where the number of crashes may exceed the connectivity threshold. The original work in [4] has been extended to solving consensus when *links* are subject to crash failures [8]. Several consensus algorithms were proposed in [8], but their round complexities are expressed as a function of the so-called *stretch*, defined as the number of connected components of the graph after removing the faulty links, plus the sum of the diameters of the connected components. Instead, the round-complexity of the algorithm in [4] is expressed in term of the *radius*, which is a more refined measure. Indeed, we show that the upper bound in [4] is tight (no multiplicative constants, nor even additive constants). The consensus algorithms in [8] however extend to the case where failures may disconnect the graph, and the task is then referred to as “disconnected agreement”. Again, the complexities of the algorithms are expressed in term of the stretch, while we shall express the complexity of our local consensus algorithm as a function of the more refined radius parameter. We actually conjecture that our local consensus algorithm is optimal (with no multiplicative nor additive constants) for all t , no matter whether $t < \kappa(G)$ or $t \geq \kappa(G)$. On the other hand, some consensus algorithms proposed in [8] are early stopping, but the one with round-complexity close to the stretch of the actual failure pattern is not oblivious, and it uses messages with size significantly larger than the size of the messages in oblivious algorithms.

The case of *omission* failures has also attracted a lot of attention. In this context, nodes are reliable but messages may be lost. This is modeled as a sequence $\mathcal{S} = (G_i)_{i \geq 1}$ of directed graphs, where G_i captures the connections that are functioning at round i . The *oblivious message adversary* model allows an adversary to choose each communication graph G_i from a set \mathcal{G} and independently of its choices for the other graphs. The nodes know the set \mathcal{G} a priori, but not the actual graph picked by the adversary at each round). We refer to [9, 24, 29] for recent advances in this domain, including solving consensus. We also refer to the *heard-of* model [6, 7], which bears similarities with the oblivious message adversary model.

The case of *transient* failures is addressed in the context of *self-stabilizing* algorithms [14]. As opposed to most distributed algorithms for networks, which start from a given specific initial configuration, self-stabilizing algorithms must be able to start from any initial configuration (which may result from a corruption of the internal variables of the nodes). Under the synchronous scheduler, a self-stabilizing algorithm performs in a sequence of synchronous rounds, just that it must be able to cope with an arbitrary initial state of the system.

Last but not least, we underline the recent trend related to modeling communication between nodes (under the full-information paradigm) as a topological deformation of the input simplicial complex, and the computation (i.e., the decision of each node regarding its output value) as a simplicial map from the deformed input complex to the output simplicial complex [18]. The KNOW-ALL model [3] has been designed as a first attempt to understand the LOCAL model through the lens of algebraic topology. In particular, it was shown that k -set agreement in a graph G known to all the nodes a priori requires r rounds, where r is the smallest integer such that there exists a k -node dominating set in the r -th transitive closure of G . A follow-up work [17] minimized the involved simplicial complexes, and extended the framework to handle graph problems such as finding a proper coloring.

The study of *anonymous* networks, in which nodes may not be provided with distinct identifiers, and of *asynchronous* communication and computing, is beyond the scope of this paper, and we merely refer the reader to [10, 11, 15, 21, 22] for recent advances in these domains, as far as computing in (non-necessarily complete) networks is concerned.

2 Model and definitions

In this section, we recall the definition of the (synchronous) t -resilient model for networks, and the graph theoretical notions related to this model, all taken from [4], as well as the consensus algorithm presented there.

2.1 The Model

Let $G = (V, E)$ be an n -node undirected graph, which is also connected and simple (i.e., no multiple edges, nor self-loops). Each node $v \in V$ is a computing entity modeled as an infinite state machine. The nodes of G have distinct identifiers, which are positive integers. For the sake of simplifying the notations, we shall not distinguish a node v from its identifier; for instance, by “the smallest node” we mean “the node with the smallest identifier”. Initially, every node knows the graph G , that is, it knows the identifiers of all nodes, and how the nodes are connected. The uncertainty is thus not related to the initial structure of the connections, but is only due to the presence of potential failures, in addition to the fact that, of course, every node is not a priori aware of the inputs of the other nodes.

Computation in G proceeds as a sequence of synchronous rounds. All nodes start simultaneously, at round 1. At each round, each node sends a message to each of its neighbors in G , receives the messages sent by its neighbors, and performs some local computation. Each node may however fail by crashing – when a node crashes, it stops functioning and never recover. However, if a node v crashes at round r , it may still send a message to a non-empty subset of its set $N(v)$ of neighbors during round r . For every positive integer $t \geq 0$, the t -resilient model assumes that at most t nodes may crash. A *failure pattern* is defined as a set

$$\varphi = \{(v, F_v, f_v) \mid v \in F\}$$

where $F \subset V$ is the set of faulty nodes in φ , with $0 \leq |F| \leq t$, and, for each node $v \in F$, we use f_v to specify the round at which v crashes, and $F_v \subseteq N(v)$ to specify the non-empty set of neighbors to which v fails to send messages at round f_v .

A node $v \in F$ such that $F_v = N(v)$ is said to crash *cleanly* in φ (at round f_v). All the nodes in $V \setminus F$ are the correct nodes in φ . The failure pattern in which no nodes fail is denoted by φ_\emptyset . The set of all failure patterns in which at most t nodes fail is denoted by $\Phi_{\text{all}}^{(t)}$. In any execution of an algorithm in graph G under the t -resilient model, the nodes know t and G , but they do not know in advance to which failure pattern they may be exposed. This absence of knowledge is the source of uncertainty in the t -resilient model.

2.2 Eccentricity, connectivity, and radius

The *eccentricity* of a node v in G with respect to a failure pattern φ , denoted by $\text{ecc}(v, \varphi)$, is defined as the minimum number of rounds required for broadcasting a message from v to all *correct* nodes in φ . The broadcast protocol is by flooding, i.e., when a node receives a message at round r , it forwards it to all its neighbors at round $r + 1$. That is $\text{ecc}(v, \varphi)$ is the maximum, taken over all correct nodes v' , of the length of a shortest causal path from v to v' , where a *causal* path with respect to a failure pattern φ from a node v to a node v' is a sequence of nodes u_1, \dots, u_q with $u_1 = v$, $u_q = v'$, and, for every $i \in \{1, \dots, q - 1\}$, $u_{i+1} \in N(u_i)$, u_i has not crashed in φ during rounds $1, \dots, i - 1$, and if u_i crashes in φ at round i , i.e., if $(u_i, F_i, i) \in \varphi$ for some non-empty set $F_i \subseteq N(u_i)$, then $u_{i+1} \notin F_i$.

Note that $\text{ecc}(v, \varphi)$ might be infinite, in case v cannot broadcast to all correct nodes in G under φ . A typical example is when v crashes cleanly at the first round in φ , before sending any message to any of its neighbors. A more elaborate failure pattern φ in which v fails to broadcast is $\varphi = \{(v, N(v) \setminus \{w\}, 1), (w, N(w), 2)\}$ where v crashes at round 1, and sends the message only to its neighbor w , which crashes cleanly at round 2.

The node-connectivity of G , denoted $\kappa(G)$, is the smallest integer q such that removing q nodes disconnects the graph G (or reduces it to a single node whenever G is the complete graph K_n). The following was established in [4].

► **Proposition 1** (Lemma 1 in [4]). *For every graph G , every $t < \kappa(G)$, every node v , and every failure pattern φ in the t -resilient model, $\text{ecc}(v, \varphi) < \infty$ if and only if there exists at least one correct node that becomes aware of the message broadcast from v .*

Note that, in particular, thanks to proposition 1, if v is correct then $\text{ecc}(v, \varphi) < \infty$. Let

$$\Phi_v^* = \{\varphi \in \Phi_{\text{all}}^{(t)} \mid \text{ecc}(v, \varphi) < \infty\}$$

denote the set of failure patterns in the t -resilient model in which v eventually manages to broadcast to all correct nodes. The t -resilient radius is a key parameter defined in [4]:

► **Definition 2.** *The t -resilient radius of G is $\text{radius}(G, t) = \min_{v \in V} \max_{\varphi \in \Phi_v^*} \text{ecc}(v, \varphi)$.*

2.3 Consensus, oblivious algorithms, and the information flow graph

This section defines consensus, and survey the results in [4] regarding the round-complexity of oblivious consensus algorithms, which uses the notion of information flow graph. Note that this latter notion will be revisited, later in our paper.

2.3.1 Oblivious consensus algorithms

In the consensus problem, every node $v \in V$ receives an input value x_v from a set I of cardinality at least 2, and every correct node must decide on an output value $y_v \in I$ such that (1) $y_u = y_v$ for every pair $\{u, v\}$ of correct nodes, and (2) for every correct node $v \in V$, there exists $u \in V$ (not necessarily correct) such that $y_v = x_u$.

Assuming that every node $u \in V$ starts broadcasting the pair (u, x_u) at round 1, we let $\text{view}(v, \varphi, r)$ be the *view* of node v after $r \geq 0$ rounds in failure pattern φ , that is, the set of pairs (u, x_u) received by v after r rounds. An algorithm solving consensus is said to be *oblivious* if the output y_v of every correct node v depends only on the set of values received by v during the execution of the algorithm. That is, in an r -round oblivious algorithm executed under failure pattern φ , every node v outputs a value based solely on the set of pairs $(u, x_u) \in \text{view}(v, \varphi, r)$ (and not, say, on when each value was first received, or from which neighbor it was received). The following result was proved in [4].

► **Proposition 3** (Theorem 2 in [4]). *For every graph G and every $t < \kappa(G)$, consensus in G can be solved by an oblivious algorithm running in $\text{radius}(G, t)$ rounds under the t -resilient model.*

That is, consensus can be solved in the minimal time it takes for a *fixed* node to broadcast in all failure patterns (in which it manages to broadcast). Note that $\text{radius}(G, t)$ might be much larger than $\max_{\varphi \in \Phi_{\text{all}}^{(t)}} \min_{v \in V} \text{ecc}(v, \varphi)$. For instance, the radius of the clique K_n is $t + 1$: consider a path (v_1, \dots, v_{t+1}) in which $v_1 = v$, and, for every $i \in \{1, \dots, t\}$, v_i crashes at round i while sending only to v_{i+1} . On the other hand, $\max_{\varphi \in \Phi_{\text{all}}^{(t)}} \min_{v \in V} \text{ecc}(v, \varphi) = 1$ because,

for every failure pattern φ , there is a (correct) node v that broadcasts to all correct nodes in a single round. Similarly, the cycle C_n has radius $n - 1$, whereas $\max_{\varphi \in \Phi_{\text{all}}^{(t)}} \min_{v \in V} \text{ecc}(v, \varphi)$ is roughly $n/2$.

The consensus algorithm in [4] works as follows. It selects an ordered set of $t + 1$ nodes s_1, \dots, s_{t+1} according to the following rules. Node s_1 is a node with smallest eccentricity, i.e., a node that broadcasts the fastest among all nodes. However, there are failure patterns for which s_1 fails to broadcast (e.g., if s_1 crashes cleanly at round 1). Node s_2 is a node that broadcasts the fastest for all failure patterns in which s_1 fails to broadcast, that is node s_2 is a node that broadcasts the fastest for all failure patterns in $\Phi_{\text{all}}^{(t)} \setminus \Phi_{s_1}^*$. Similarly, node s_3 is a node that broadcasts the fastest for all failure patterns in which s_1 and s_2 fail to broadcast, that is node s_3 is a node that broadcasts the fastest for all failure patterns in $\Phi_{\text{all}}^{(t)} \setminus (\Phi_{s_1}^* \cup \Phi_{s_2}^*)$. And so on, for every $1 < i \leq t + 1$, s_i is a node that broadcasts the fastest for all failure patterns in

$$\Phi_{\text{all}}^{(t)} \setminus \bigcup_{j=1, \dots, i-1} \Phi_{s_j}^*.$$

A key property of the sequence s_1, \dots, s_{t+1} defined as above is that, for all $1 < i \leq t + 1$, the worst-case broadcast time of s_i over all failure patterns in

$$\Phi_{\text{all}}^{(t)} \setminus \bigcup_{j=1, \dots, i-1} \Phi_{s_j}^*$$

is at most the worst-case broadcast time of s_{i-1} over all failure patterns in

$$\Phi_{\text{all}}^{(t)} \setminus \bigcup_{j=1, \dots, i-2} \Phi_{s_j}^*.$$

As a consequence, for every $i \in \{1, \dots, t + 1\}$, the worst-case broadcast time of s_i over all failure patterns in $\Phi_{\text{all}}^{(t)} \setminus \bigcup_{j=1, \dots, i-1} \Phi_{s_j}^*$ is at most $\text{radius}(G, t)$ rounds.

The algorithm in [4] merely consists of letting all nodes s_1, \dots, s_{t+1} broadcast the pairs (s_i, x_{s_i}) by flooding during $\text{radius}(G, t)$ rounds. Every node u then selects as output the input x_{s_i} of the node s_i with smallest index i such that the pair (s_i, x_{s_i}) was received by node u . It was shown that this choice guarantees agreement.

2.4 Information flow graph

The lower bound from [4] on the number of rounds for achieving consensus in vertex-transitive graphs used the core notion of *information flow digraph*. The (directed) graph $\text{IF}(G, r)$ captures the state of mutual knowledge of the nodes at the end of round $r \geq 1$, assuming every node u broadcasts the pair (u, x_u) by flooding throughout the graph G , starting at round 1.

- The vertices of $\text{IF}(G, r)$ are all pairs $(v, \text{view}(v, r, \varphi))$ for $v \in V$ and $\varphi \in \Phi_{\text{all}}^{(t)}$ in which v does not crash in φ during the first r rounds. Note that a same vertex of $\text{IF}(G, r)$ can represent both $(v, \text{view}(v, r, \varphi))$ and $(v, \text{view}(v, r, \psi))$ if v has the same view after r rounds in φ and ψ .
- There is an arc from $(u, \text{view}(u, r, \varphi))$ to $(v, \text{view}(v, r, \varphi))$ whenever $(u, x_u) \in \text{view}(v, r, \varphi)$, where x_u is the input of u .

The *connected components* of $\text{IF}(G, r)$ play an important role, where by connected component we actually refer to the vertices of a connected component of the undirected graph resulting from $\text{IF}(G, r)$ by ignoring the directions of the arcs. A node $v \in V$ of the communication graph $G = (V, E)$ is said to *dominate* a connected component C of $\text{IF}(G, r)$ if,

for every vertex $(u, \text{view}(u, r, \varphi)) \in C$ with $u \neq v$ there is a vertex $(v, \text{view}(v, r, \varphi)) \in C$ with an arc from $(v, \text{view}(v, r, \varphi))$ to $(u, \text{view}(u, r, \varphi))$ in $\text{IF}(G, r)$. The following result characterizes the round-complexity of consensus in G .

► **Proposition 4** (Theorem 3 in [4]). *For every graph $G = (V, E)$ and every $t < \kappa(G)$, consensus in G can be solved by an oblivious algorithm running in r rounds under the t -resilient model if and only if every connected component of $\text{IF}(G, r)$ has a dominating node in V .*

It was proved in [4] that, if G is a symmetric graph then no node in V dominates $\text{IF}(G, \text{radius}(G, t) - 1)$. Property 4 immediately implies that consensus in G cannot be solved by an oblivious algorithm running in less than $\text{radius}(G, t)$ rounds under the t -resilient model. Their proof, however, holds only for symmetric graphs, and does not extend to general graphs.

Remark. The definition of the information flow *digraph* in [4] actually suffers from inconsistencies, and Theorem 3 there is formally incorrect. Roughly, it overlooks the possibility of deciding on an input of a process that already stopped. The “spirit” of the definition and the theorem is nevertheless plausible, and the specific consequences mentioned there are correct. For establishing our lower bound, we had to fix the inaccuracy in the definition of the information flow digraph, and the bugs in the proof of Theorem 3 of [4]. Concretely, we introduce a new information flow *graph* instead of the digraph of [4], and establish a correct version of Theorem 3 using that definition (cf. Theorem 9). See Section 4 for more details.

3 Detailed description of our results

In this section, we survey our results on consensus in detail, and, as already mentioned before, we refer to the full version [16] for our results on k -set agreement.

3.1 Lower bounds for consensus

We show that the consensus algorithm in [4] is optimal for every graph G , and not only for symmetric graphs. Specifically, we establish the following in Section 4.

► **Theorem 5.** *For every graph G and every $t < \kappa(G)$, consensus in G cannot be solved in less than $\text{radius}(G, t)$ rounds by an oblivious algorithm in the t -resilient model.*

This result was conjectured in [4], but only proved to be true for symmetric graphs. The class of symmetric graphs includes cliques, cycles and hypercubes, but remains limited. Moreover, in symmetric graphs, for every two nodes u and v ,

$$\text{ecc}(u, \Phi_{\text{all}}^{(t)}) = \text{ecc}(v, \Phi_{\text{all}}^{(t)}) = \text{radius}(G, t),$$

which implies that a naive algorithm for consensus in which every node outputs the input received from the node with smallest identifier performs in $\text{radius}(G, t)$ rounds. The fact that $\text{radius}(G, t)$ is a tight upper bound for consensus is thus not surprising for the family of symmetric graphs because, essentially, the choice of the $t + 1$ nodes s_1, \dots, s_{t+1} defined in Section 2.3.1 does not matter.

Instead, for an arbitrary graph G , two different nodes may have different eccentricities, which may differ by a multiplicative factor 2 at least. As a consequence, the choice of the source nodes s_1, \dots, s_{t+1} whose input can be adopted as output by the other nodes matters, as well as the ordering of these nodes (in case a node receives the input of two different source nodes).

3.1.1 A naive lower bound

A naive lower bound for the round-complexity of consensus is the maximum, over all failure patterns, of the time it takes *some* node to broadcast in the given pattern, obtained by switching the min and max operator in the definition of $\text{radius}(G, t)$, i.e.,

$$\max_{\varphi \in \Phi_{\text{all}}^{(t)}} \min_{v \in V} \text{ecc}(v, \varphi). \quad (1)$$

Indeed, for every failure pattern φ , even binary consensus under failure pattern φ cannot be solved in less than $R(\varphi) = \min_{v \in V} \text{ecc}(v, \varphi)$ rounds. The proof of this claim is by a standard indistinguishability argument. Specifically, let us assume, for the purpose of contradiction, that there is an algorithm ALG solving consensus in $G = (V, E)$ under failure pattern φ in $R(\varphi) - 1$ rounds. Let us order the nodes of G as v_1, \dots, v_n arbitrarily. Let us consider the input configuration I_0 in which all nodes have input 0. For every $i = 1, \dots, n$, we gradually change the input configuration as follows (see Figure 1).

$$\begin{array}{ccccccccc} 000..0 & \cdots & 100..0 & \cdots & 110..0 & \cdots & 1..100 & \cdots & 1..110 & \cdots & 1..111 \\ I_0 & & I_1 & & I_2 & & I_{n-2} & & I_{n-1} & & I_n \end{array}$$

(Note: In the original image, dots above the transitions indicate nodes $w_1, w_2, \dots, w_{n-1}, w_n$ that do not distinguish the configurations.)

■ **Figure 1** Input configurations I_0, \dots, I_n of a graph $G = (V, E)$, where $V = \{v_1, \dots, v_n\}$.

Since $\text{ecc}(v_i, \varphi) > R(\varphi)$, there exists a node w_i that does not receive the input of v_i in ALG. Let us then switch the input of v_i from 0 to 1, and denote by I_i the resulting input configuration. Note that I_n is the input configuration in which all nodes have input 1. Note also that, for every $i \in \{1, \dots, n\}$, node w_i does not distinguish I_{i-1} from I_i , and therefore ALG must output the same at w_i in both input configurations. Since, for every $i \in \{1, \dots, n\}$, all nodes must output the same value for input configuration I_i , we get that the consensus value returned by ALG for I_0 is the same as for I_n , which contradicts the validity condition.

It was conjectured in [4] that, in the t -resilient model, consensus needs longer time than $\max_{\varphi \in \Phi_{\text{all}}^{(t)}} \min_{v \in V} \text{ecc}(v, \varphi)$, and cannot be solved by an oblivious algorithm in less than $\text{radius}(G, t)$ rounds, i.e., the time it takes a fixed node to broadcast. As said before, this conjecture was however proved only for vertex-transitive graphs.

3.1.2 Sketch of proof of Theorem 5

To show that the consensus algorithm in [4] is optimal, i.e., to establish Theorem 5, we use the characterization of Proposition 4. In fact, we first fix the aforementioned bugs in [4] by defining the information flow graph, and then establish Proposition 4, a correct version of their theorem, using this new definition. With this new definition and new proposition at hand, we show that for every graph $G = (V, E)$ and every $t < \kappa(G)$, there exists a connected component of $\text{IF}(G, \text{radius}(G, t) - 1)$ that has no dominating node in V . To achieve this fact, we show that for every node $v \in V$ there exists a failure pattern φ_v such that

$$\text{ecc}(v, \varphi_v) \geq \text{radius}(G, t),$$

with some additional desirable properties. Then, we define a notion of *successor* of any failure pattern satisfying these desirable properties, which satisfies two key features.

- First, a failure pattern and its successor are in the same connected component of $\text{IF}(G, \text{radius}(G, t) - 1)$. Here we abuse terminology since the vertices of the information flow graph are not failure patterns, but pairs $(\text{node}, \text{view})$. What we formally mean is that the two subgraphs of $\text{IF}(G, \text{radius}(G, t) - 1)$ induced by all the views in the two failures patterns are both in the same connected component of $\text{IF}(G, \text{radius}(G, t) - 1)$.

- Second, for every node $v \in V$, there exists a sequence of failure patterns $\varphi_0, \varphi_1, \dots, \varphi_\ell$ such that $\varphi_0 = \varphi_v$, $\varphi_\ell = \varphi_\emptyset$ (the failure pattern in which no failures occur), and for every $i \in \{0, \dots, \ell - 1\}$, φ_{i+1} is the successor of φ_i .

It follows from these two features that, for every node $v \in V$, φ_v and φ_\emptyset are in the same connected component of $\text{IF}(G, \text{radius}(G, t) - 1)$, namely the connected component of $\text{IF}(G, \text{radius}(G, t) - 1)$ containing φ_\emptyset . Let C be this connected component. For every node $v \in V$, since $\text{ecc}(v, \varphi_v) \geq \text{radius}(G, t)$, we have that v does not dominate C . Therefore, no nodes dominate C , and our new Proposition 4 thus implies that no oblivious algorithm can solve consensus in less than $\text{radius}(G, t)$ rounds.

3.2 Beyond the connectivity threshold

The algorithm from [4] for consensus in the t -resilient model is under the assumption that $t < \kappa(G)$ in graph G , that is, the number of failing nodes is (strictly) smaller than the connectivity of the graph. This assumption is motivated by the mere observation that a set of $\kappa(G)$ nodes that, e.g., fails cleanly at the very first round, might disconnect the graph G , preventing tasks such as consensus to be solved. We show that, by slightly relaxing consensus and set-agreement, one can still consider the case where $t \geq \kappa(G)$, in a meaningful way, in the sense that if the t failing nodes do not disconnect the graph, then the standard consensus and set agreement tasks are solved.

3.2.1 Local consensus

For any given failure pattern φ , let $\text{comp}(G, \varphi)$ be the set of connected components of G resulting by removing from G all nodes that fail in φ . If $t \geq \kappa(G)$, then the nodes in a connected component $C \in \text{comp}(G, \varphi)$ of G may never hear from the nodes in a connected component $C' \neq C$, and vice versa, regardless of the number of rounds. To study consensus and k -set agreement for $t \geq \kappa(G)$, we merely relax the agreement condition: *Agreement must hold component-wise*, i.e., for each connected component separately, in the spirit of [8].

In other words, under φ , for any connected components C and C' of $\text{comp}(G, \varphi)$ all nodes in C must agree (on a single value for consensus, or on at most k values for k -set agreement), and all nodes in C' must agree, but no conditions are imposed the two sets of agreement values corresponding to C and C' . In particular, for consensus, the nodes in C may agree on x , but the nodes in C' may agree on $x' \neq x$.

The validity condition remains unchanged, that is, every output value must be the input value of some node. Note however that a node can return an output value which was the input value of a node from a different connected component.

We refer to these variants of consensus and k -set agreement as *local*, because agreement must hold “locally”, i.e., inside each connected component.

Remark. When $t < \kappa(G)$, consensus and local consensus are the same tasks, and, for every $k \geq 1$, k -set agreement and local k -set agreement are the same tasks. More generally, for every graph G , and for every failure pattern $\varphi \in \Phi_{\text{all}}^{(t)}$, if the nodes failing in φ do not disconnect G , and an algorithm solving local consensus (resp., local k -set agreement) does solve standard consensus (resp., standard k -set agreement).

3.2.2 Consensus beyond the connectivity threshold

We design a local consensus algorithm for an arbitrary graph G in the t -resilient model, for every given t , which does not need to be less than the connectivity $\kappa(G)$ of G . This algorithm satisfied the following property (see proof in the full version [16]).

► **Theorem 6.** *For every connected graph $G = (V, E)$, and every $t \geq 0$, local consensus in G can be solved by an oblivious algorithm running in $\text{radius}(G, t)$ rounds under the t -resilient model.*

In the statement above, $\text{radius}(G, t)$ denotes an extension of the notion of t -resilient radius to the case where $t \geq \kappa(G)$, which coincide to the aforementioned notion of radius whenever $t < \kappa(G)$. For the purpose of extending the notion of radius beyond the connectivity threshold, we revisit the notion of eccentricity entirely. Indeed, given a failure pattern φ , a node v may succeed to broadcast in some connected components but not in all of them. The control of the way information flow through the graph G with respect to the connected components is complex, as the connected components for one failure pattern are typically different from the connected components for another failure pattern.

Despite these difficulties, we are able to design and analyse an oblivious (and hence generic) local consensus algorithm. Given a graph $G = (V, E)$, our algorithm performs in $\text{radius}(G, t) = \min_{v \in V} \text{ecc}(v, \Phi_v^*)$ rounds, where the notion of eccentricity has been redefined and extended for allowing an arbitrary number t of failures. Again, for $t < \kappa(G)$, the extended notion of eccentricity coincides with the notion of eccentricity defined for consensus in [4], which itself coincide with the graph-theoretical notions of eccentricity for $t = 0$. We also note that our extended notion of radius, for all $t \geq 0$, provides a fine grain analysis of our local consensus algorithm, more refined than the notion of stretch defined in [8].

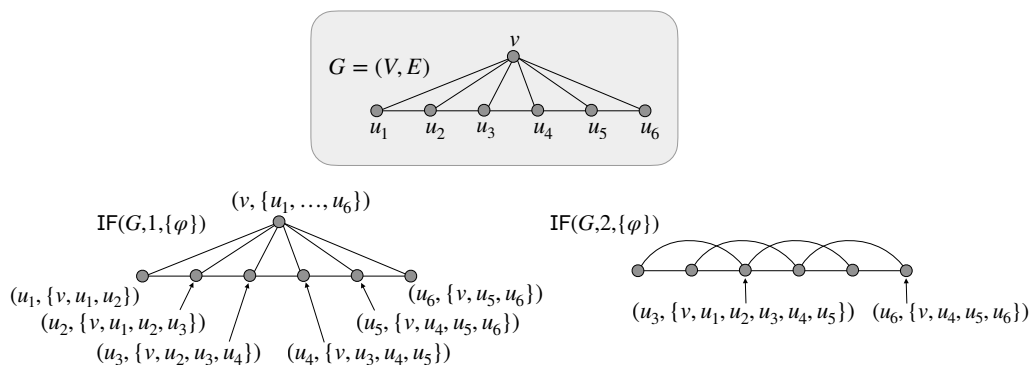
4 Lower Bound for Consensus

This section is entirely devoted to the proof of Theorem 5, that is, we show that, for every graph G and every $t < \kappa(G)$, consensus in G cannot be solved in less than $\text{radius}(G, t)$ rounds by an oblivious algorithm in the t -resilient model. We first establish a consistent notion of information flow graph, which can then be used to characterize consensus solvability, and we fix the bugs in the proof of Theorem 3 in [4] (see Proposition 4) resulting from inconsistencies in the original definition of the information flow digraph. Using our new characterization, we establish our lower bound.

4.1 Information flow graph revisited

The main issue with the notion of information flow *digraph* $\text{IF}(G, r)$ as defined in [4] comes from the fact that this directed graph includes only vertices $(v, \text{view}(v, r, \varphi))$ where v has not crashed in φ during rounds $1, \dots, r$. The main issue is related to the concept of domination, as defined in [4]. A vertex v dominates a connected component C of $\text{IF}(G, r)$ if the set $\{(v, \text{view}(v, r, \varphi)) \mid \varphi \in \Phi_{\text{all}}^{(t)}\}$ dominates C . This is too restrictive, as the correct nodes may agree on the input value of a node v that has already crashed. It follows that, for some failure pattern φ , the vertex $(v, \text{view}(v, r, \varphi))$ may not be present in $\text{IF}(G, r)$ (and therefore cannot dominate any other vertices of $\text{IF}(G, r)$), whereas the nodes that are correct in φ may agree on the input value of v . The characterization of Theorem 3 in [4] is therefore incorrect, even if the “spirit” of the characterization remains conceptually valid, as we shall show in this section.

To provide an illustration of the problems resulting from the original definition of information flow digraph in [4], let us clarify that this definition was aiming for capturing any subset $\Phi \subseteq \Phi_{\text{all}}^{(t)}$ of failure patterns (for instance the subset Φ of failure patterns in which nodes crash cleanly), in which case only the failure patterns $\varphi \in \Phi$ are considered. Let us



■ **Figure 2** The information flow graph $\text{IF}(G, r, \{\varphi\})$ as defined in [4] for $r = 1$ and $r = 2$, where φ is the failure pattern in which v crashes cleanly at the second round. No node dominates $\text{IF}(G, 2, \{\varphi\})$ (right), even though consensus is solvable in G under φ in 2 rounds.

then consider the scenario displayed on Fig. 2. The graph G is a 6-node path plus a universal node v . The set $\Phi = \{\varphi\}$ contains a single failure pattern φ in which v crashes cleanly at the second round.

Fig. 2 displays $\text{IF}(G, 1, \{\varphi\})$ and $\text{IF}(G, 2, \{\varphi\})$ as defined in [4] (the direction of the arcs are omitted, each edge corresponding to two symmetric arcs). A vertex $(v, \text{view}(v, r, \varphi))$ is present in the former but not in the latter, and thus, as opposed to what one might expect since nodes acquire more and more information as time passes, $\text{IF}(G, 2, \{\varphi\})$ is not a denser super graph of $\text{IF}(G, 1, \{\varphi\})$ nor it includes more vertices (with larger views), as some vertices present in $\text{IF}(G, 1, \{\varphi\})$ may disappear in $\text{IF}(G, 2, \{\varphi\})$. In fact, node v dominates $\text{IF}(G, 1, \{\varphi\})$, but it does not dominate $\text{IF}(G, 2, \{\varphi\})$. Therefore, when analyzing G with the set $\{\varphi\}$ of failure patterns using the characterization theorem in [4], consensus should be solvable in 1 round but not in 2 rounds!

We propose below a more robust notion of information flow *graph* (which is not directed anymore). The reader familiar with the algebraic topology interpretation of distributed computing [18] will recognize the mere 1-skeleton of the protocol complex after r rounds. For the purpose of fixing the issues in [4], we introduce $\text{IF}(G, r, \Phi)$ for an arbitrary set of failure patterns $\Phi \subseteq \Phi_{\text{all}}^{(t)}$.

► **Definition 7.** *The information flow graph of a communication graph $G = (V, E)$ after $r \geq 0$ rounds for a set $\Phi \subseteq \Phi_{\text{all}}^{(t)}$, $t \geq 0$, of failure patterns is the graph $\text{IF}(G, r, \Phi)$ defined as follows.*

- *The vertices of $\text{IF}(G, r, \Phi)$ are all pairs $(v, \text{view}(v, r, \varphi))$ for $v \in V$ and $\varphi \in \Phi$, where v is correct in φ .*
- *There is an edge between (v_1, w_1) and (v_2, w_2) in $\text{IF}(G, r, \Phi)$ whenever there exists $\varphi \in \Phi$ such that $w_1 = \text{view}(v_1, r, \varphi)$ and $w_2 = \text{view}(v_2, r, \varphi)$*

Remark. Unlike the definition of [4], this new notion of information-flow graph is not limited to $t \leq \kappa(G)$.

Note that a same vertex (v, ω) of $\text{IF}(G, \Phi, r)$ can represent both $(v, \text{view}(v, r, \varphi))$ and $(v, \text{view}(v, r, \psi))$ if v has the same view after r rounds in $\varphi \in \Phi$ and $\psi \in \Phi$. Note also that, for every $\varphi \in \Phi$, the set

$$\text{config}(G, r, \varphi) = \{(v, \text{view}(v, r, \varphi)) \in \text{IF}(G, r, \Phi) \mid v \in V\}$$

is a clique in $\text{IF}(G, r, \Phi)$. The *connected components* of $\text{IF}(G, r, \Phi)$ play an important role, w.r.t. the following concept of *domination*.

34:14 Agreement Tasks in Fault-Prone Synchronous Networks of Arbitrary Structure

► **Definition 8.** A node $v \in V$ of the communication graph $G = (V, E)$ is said to dominate a connected component C of $\text{IF}(G, r, \Phi)$ if, for every $\varphi \in \Phi$ and every $u \in V$,

$$(u, \text{view}(u, r, \varphi)) \in C \implies (v, x_v) \in \text{view}(u, r, \varphi).$$

Note that only correct nodes need to be dominated, as

$$(u, \text{view}(u, r, \varphi)) \in C \subseteq \text{IF}(G, r, \Phi)$$

implies that u is correct at round r . On the other hand, any node may be dominating. The following result characterizes the round-complexity of consensus in G by fixing the aforementioned inaccuracies in the definition of the information flow graph in [4], with impact on the proof of their characterization theorem (Theorem 3 in [4]).

► **Theorem 9.** For every graph $G = (V, E)$, every $t \geq 0$, and every set of failure patterns $\Phi \subseteq \Phi_{\text{all}}^{(t)}$, consensus in G can be solved by an oblivious algorithm running in r rounds under the t -resilient model with failure patterns in Φ if and only if every connected component of $\text{IF}(G, r, \Phi)$ has a dominating node in V .

Proof. Let us first show that if every connected component of $\text{IF}(G, r, \Phi)$ has a dominating node in V then consensus in G can be solved by an oblivious algorithm running in r rounds. For every connected component C of $\text{IF}(G, r, \Phi)$, let $v_C \in V$ be a node of G that dominates C . The algorithm proceeds as follows. Every node v_C broadcasts by flooding during r rounds. After r rounds, every correct node u considers its view, denoted by $\text{view}(u)$. A crucial point is that $\text{view}(u)$ may not be sufficient for u to determine what is the actual failure pattern $\varphi \in \Phi$ experienced during the execution, merely because one may have

$$\text{view}(u) = \text{view}(u, r, \varphi) = \text{view}(u, r, \psi)$$

for two different failure patterns φ, ψ in Φ . However, $\text{view}(u)$ is sufficient to determine the connected component C of $\text{IF}(G, r, \Phi)$ to which $(u, \text{view}(u))$ belongs. Node u outputs the input x_{v_C} of node v_C .

To establish correctness of this algorithm, observe first that (v_C, x_{v_C}) belongs to the view of node u . To see why, let $\varphi \in \Phi$, and let us consider the execution of the algorithm under φ . Let C be the connected component of $(u, \text{view}(u, r, \varphi))$. Since v_C dominates C , the mere definition of domination implies that $(v_C, x_{v_C}) \in \text{view}(u, r, \varphi)$. As a consequence, the algorithm is well defined. To show agreement, let $u' \neq u$ be another correct node in φ . By definition of the information flow graph, there is an edge between $(u, \text{view}(u, r, \varphi))$ and $(u', \text{view}(u', r, \varphi))$, and thus these two vertices belong to the same connected component C , and both output the same value x_{v_C} .

For the other direction, we show the contrapositive. That is, we let C be a connected component of $\text{IF}(G, r, \Phi)$ that is not dominated, and we aim at showing that there are no oblivious consensus algorithms in G running in r rounds. Let us assume, for the purpose of contradiction, that there exists an oblivious consensus algorithm ALG in G running in r rounds.

▷ **Claim 10.** Let $(u, \text{view}(u, r, \varphi))$ and $(u', \text{view}(u', r, \varphi'))$ be two vertices of C , where u and u' need not be different, nor do φ and φ' . For the same input configuration, node u outputs the same value in ALG under φ as node u' under φ' .

To see why this claim holds, observe that, since $(u, \text{view}(u, r, \varphi))$ and $(u', \text{view}(u', r, \varphi'))$ belong to the same connected component C , there is a sequence

$$(v_0, \text{view}(v_0, r, \psi_0)), \dots, (v_k, \text{view}(v_k, r, \psi_k))$$

of vertices of C such that

$$(v_0, \text{view}(v_0, r, \psi_0)) = (u, \text{view}(u, r, \varphi)), \quad (v_k, \text{view}(v_k, r, \psi_k)) = (u', \text{view}(u', r, \varphi')),$$

and, for every $i \in \{0, \dots, k-1\}$, there is an edge between the two vertices $(v_i, \text{view}(v_i, r, \psi_i))$ and $(v_{i+1}, \text{view}(v_{i+1}, r, \psi_{i+1}))$ in $\text{IF}(G, r, \Phi)$. Note that, for every $i \in \{0, \dots, k\}$, node v_i is correct in ψ_i since $(v_i, \text{view}(v_i, r, \psi_i))$ belongs to the information flow graph. For every $i \in \{0, \dots, k-1\}$, the presence of an edge between $(v_i, \text{view}(v_i, r, \psi_i))$ and $(v_{i+1}, \text{view}(v_{i+1}, r, \psi_{i+1}))$ implies that there exists $\chi \in \Phi$ such that

$$(v_i, \text{view}(v_i, r, \psi_i)) = (v_i, \text{view}(v_i, r, \chi)),$$

and

$$(v_{i+1}, \text{view}(v_{i+1}, r, \psi_{i+1})) = (v_{i+1}, \text{view}(v_{i+1}, r, \chi)).$$

As a consequence, since ALG is a consensus algorithm, ALG outputs the same value at v_{i+1} under ψ_{i+1} as it outputs at v_i under ψ_i , which is the value outputted by ALG under χ . Since this holds for every $i \in \{0, \dots, k-1\}$, we get that, in particular, u outputs the same value in φ as u' in φ' , as claimed.

For establishing a contradiction, let us enumerate the n nodes of G as u_0, \dots, u_{n-1} in arbitrary order. Since C is not dominated, for every node u_i , $i \in \{0, \dots, n-1\}$, there exists a vertex $(v_i, \text{view}(v_i, r, \varphi_i))$ of C such that $(u_i, x_{u_i}) \notin \text{view}(v_i, r, \varphi_i)$, where v_i is correct in φ_i . For $i \in \{0, \dots, n\}$, let us denote by I_i the input configuration in which the $n-i$ nodes $u_0, \dots, u_{n-(i+1)}$ have input 0, and all the other nodes have input 1. Thus, in particular, I_0 is the configuration in which all nodes have input 0, and I_n is the configuration in which all nodes have input 1. Since, for every $i \in \{0, \dots, n-1\}$, $(u_i, x_{u_i}) \notin \text{view}(v_i, r, \varphi_i)$, node u_i does not distinguish I_i from I_{i+1} under φ_i , and thus ALG must output the same at u_i for both configurations.

Since consensus imposes that all (correct) nodes output the same value, this means that, for every $i \in \{0, \dots, n-1\}$, all nodes output the same in ALG for I_i and I_{i+1} under φ_i . By Claim 10, all nodes output the same for I_i under φ_i as they do for I_{i+1} under φ_{i+1} . It follows that all nodes output the same for I_0 under φ_0 as for I_n under φ_n . This is a contradiction as all nodes must output 0 for I_0 , whereas all nodes must output 1 for I_n . \blacktriangleleft

Notation. For a fixed upper bound t on the number of failures, for every graph G , and for every integer $r \geq 0$, we denote by $\text{IF}(G, r)$ the information flow graph for the set of all failure patterns in the t -resilient model, that is, $\text{IF}(G, r) = \text{IF}(G, r, \Phi_{\text{all}}^{(t)})$.

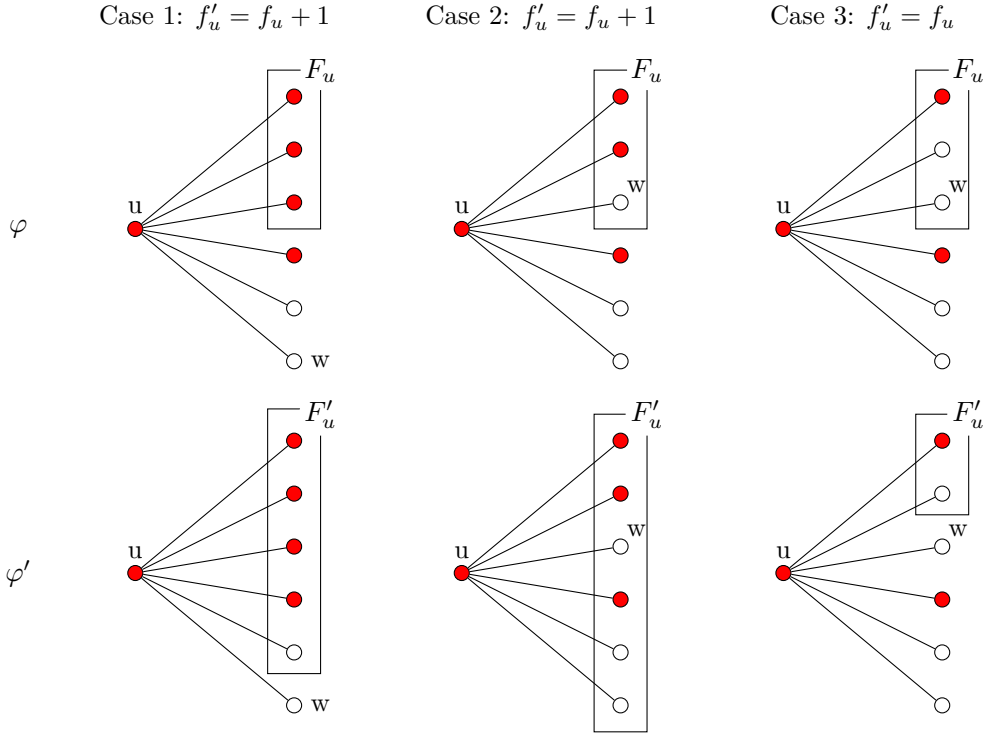
4.2 Proof of Theorem 5

To prove Theorem 5, we define the notion of *successor* of a failure pattern. Given $\varphi \in \Phi_{\text{all}}^{(t)}$, we say that a node u is *crashing last* in φ if there exists a triple $(u, F_u, f_u) \in \varphi$ (i.e., u crashes in φ), and, for every $(v, F_v, f_v) \in \varphi$, $f_u \geq f_v$.

► **Definition 11.** Let $\varphi \in \Phi_{\text{all}}^{(t)}$, let $(u, F_u, f_u) \in \varphi$, and assume that u is crashing last in φ . A successor of φ with respect to u is a failure pattern

$$\text{succ}(\varphi, u) = \left(\varphi \setminus \{(u, F_u, f_u)\} \right) \cup \{(u, F'_u, f'_u)\}$$

where F'_u and f'_u are defined as follows (see Fig. 3):



■ **Figure 3** A successor φ' of a failure pattern φ with respect to node u . Red nodes are faulty in φ and white nodes are correct in it.

1. If F_u contains only faulty nodes in φ , then $f'_u = f_u + 1$, and $F'_u = N(u) \setminus \{w\}$ for some arbitrary correct neighbor w of u .
2. If F_u contains exactly one correct node w in φ , then $f'_u = f_u + 1$, and $F'_u = N(u)$.
3. If F_u contains at least two correct nodes in φ , then $f'_u = f_u$, and $F'_u = F_u \setminus \{w\}$ for some arbitrary correct node $w \in F_u$.

Note that the correct node w in Definition 11 is well defined as the number of failures satisfies $t < \kappa(G) \leq \delta(G) \leq \deg(u)$, where $\delta(G)$ is the minimum degree of the nodes in G . Intuitively, $\text{succ}(\varphi, u)$ is identical to φ , except that u fails at round $f_u + 1$, or it still fails at round f_u but sends its message to one more correct neighbor before crashing.

Note also that a failure pattern may have different successors, which depends on the choice of the node u that crashes last, and on the choice of the correct neighbor w of u in the first and third cases of Definition 11. A correct neighbor w of u in Definition 11 is called a *witness* of the pair (φ, φ') .

Still using the notations of Definition 11, let us set $f''_u = f'_u$ in case 1, and $f''_u = f_u$ in cases 2 and 3. At the end of round f''_u , there is at most one correct node with different views in φ and $\text{succ}(\varphi, u)$. The only correct node may have different views in φ and $\varphi' = \text{succ}(\varphi, u)$ at the end of round f''_u is the *witness* of the pair (φ, φ') . Before applying the notion of successor to derive our lower bound, let us observe the following.

► **Lemma 12.** *For every node v , there exists a failure pattern $\varphi \in \Phi_v^*$ such that no node $u \neq v$ fails at round 1 in φ , and $\text{ecc}(v, \varphi) \geq \text{radius}(G, t)$.*

Proof. By definition of the radius, for every $v \in V$, there exists $\psi \in \Phi_v^*$ such that $\text{ecc}(v, \psi) \geq \text{radius}(G, t)$. The failure pattern φ is identical to ψ , except that, for every node $u \neq v$ that crashes at round 1 in ψ , u crashes cleanly at round 2 in φ . We have $\text{ecc}(v, \varphi) = \text{ecc}(v, \psi)$ because every node that crashes later in φ than in ψ does not send any message to their neighbors after round 1 which may contain information received from v . Thus $\text{ecc}(v, \varphi) \geq \text{radius}(G, t)$. ◀

The premises of the following lemma are justified by Lemma 12.

► **Lemma 13.** *Let $\varphi \in \Phi_{\text{all}}^{(t)}$ such that (1) at most one node crashes at round 1, and (2) if there exists a node v that crashes at round 1 in φ , then $\varphi \in \Phi_v^*$ (i.e., v broadcasts despite the fact that it crashes at round 1). For every successor φ' of φ , the following holds:*

- *at most one node crashes at round 1 in φ' ;*
- *if there is a node v that crashes at round 1 in φ' , then v crashes at round 1 in φ as well;*
- *there exists a correct node with the same view in φ and φ' at the end of round $\text{radius}(G, t) - 1$.*

Proof. Let φ' be a successor of φ , such that the entry (u, F_u, f_u) of φ is replaced by the entry (u, F'_u, f'_u) in φ' . Let w be a witness for the pair (φ, φ') with respect to u . Using the notations from Definition 11, let $f''_u = f'_u$ in Case 1, and $f''_u = f_u$ in Cases 2 and 3.

After f''_u rounds, the only correct node that may have different views in φ and φ' is w . Since u is a node crashing last in φ , we get that, after round f''_u , w needs the same number of rounds in φ and φ' for broadcasting to all correct nodes. Indeed, all nodes that have not crashed in φ nor in φ' up to round f''_u included satisfy: (1) they are correct nodes in both φ and φ' , (2) they have the same view in both φ and φ' , and (3) the subgraph of G induced by the correct nodes in φ is identical to the subgraph of G induced by the correct nodes in φ' .

Let $R = \text{radius}(G, t)$. We consider two cases, depending on whether w broadcasts or not.

Let us first consider the case where, assuming that w starts broadcasting at round $f''_u + 1$, w cannot broadcast to all correct nodes during rounds $f''_u + 1, \dots, R - 1$ under the failure patterns φ' and φ . That is, under φ' , some node s does not receive $\text{view}(w, f''_u, \varphi')$ during rounds $f''_u + 1, \dots, R - 1$. As a consequence, this node s does not detect any difference between $\text{view}(w, f''_u, \varphi)$ and $\text{view}(w, f''_u, \varphi')$. It follows that s has the same view in φ and φ' at the end of $R - 1$ rounds.

Consider now the case where, assuming that w starts broadcasting at round $f''_u + 1$, w does succeed to broadcast to all correct nodes during rounds $f''_u + 1, \dots, R - 1$ under the failure patterns φ' and φ . Since no node fails after round f''_u in both φ and φ' , a causal path from w to a node s in rounds $f''_u + 1, \dots, R - 1$ is also a causal path from s to w in rounds $f''_u + 1, \dots, R - 1$. At the end of round $R - 1$, every correct node can thus send to w its view at the end of round f''_u . Since no node $s \neq v$ fails at round 1, every node $s \neq v$ does send its input to some correct neighbor during round 1. Therefore, $s \in \text{view}(w, R - 1, \varphi)$ and $s \in \text{view}(w, R - 1, \varphi')$. Since $\varphi \in \Phi_v^*$, we get that, at the end of round f''_u , there exists a correct node x that heard from v , i.e., such that $v \in \text{view}(x, f''_u, \varphi)$. At the end of round $R - 1$, this node x will send $\text{view}(x, f''_u, \varphi)$ to w , so $v \in \text{view}(w, R - 1, \varphi)$. Similarly, $v \in \text{view}(w, R - 1, \varphi')$. As a consequence, $\text{view}(w, R - 1, \varphi) = \text{view}(w, R - 1, \varphi')$, and w has a same view in both failure patterns after $R - 1$ rounds, as claimed.

Furthermore, at most one node v crashes at round 1 in φ' , and $\varphi' \in \Phi_v^*$, as desired. ◀

Using the characterization of Theorem 9 of consensus solvability based on the information-flow graph, it is sufficient to prove the following result for establishing our lower bound.

► **Lemma 14.** *The information-flow graph $\text{IF}(G, \text{radius}(G, t) - 1)$ has a connected component that is not dominated by any node of V .*

Proof. Let $R = \text{radius}(G, t)$. For every node $v \in V$, we denote by φ_v a failure pattern in Φ_v^* such that φ_v contains no node $u \neq v$ that fails at round 1, and $\text{ecc}(v, \varphi_v) \geq R$. The existence of φ_v is guaranteed by Lemma 12. Borrowing the notation from [4], for every failure pattern φ , and every $r \geq 1$, let

$$\text{config}(\varphi, r) = \{(v, \text{view}(v, \varphi, r)) \in V(\text{IF}(G, r)) \mid v \in V \text{ is active in } \varphi \text{ at round } r\},$$

where by v is active in φ at round r , we mean that v has not crashed in φ during rounds $1, \dots, r$. It was proved in [4] (see Lemma 4 in there) that, for every failure pattern φ , and every $r \geq 1$, the subgraph of $\text{IF}(G, r)$ induced by the vertices of $\text{config}(\varphi, r)$ is connected.

We now show that, for every $v \in V$, $\text{config}(\varphi_v, R - 1)$ and $\text{config}(\varphi_\emptyset, R - 1)$ are contained in the same connected component of $\text{IF}(G, R - 1)$. Roughly, we shall construct a sequence of intermediate failure patterns from φ_v to φ_\emptyset such that, for every two consecutive failure patterns ψ and ψ' in the sequence, there is a correct node with the same view in ψ and ψ' . Note that the existence of this node implies that the subgraph of $\text{IF}(G, R - 1)$ induced by $\text{config}(\psi, R - 1)$, and the subgraph of $\text{IF}(G, R - 1)$ induced by $\text{config}(\psi', R - 1)$ are included in the same connected component of $\text{IF}(G, R - 1)$.

Let us order the crashing nodes in φ_v in a decreasing order of the rounds at which they crash where ties are broken arbitrarily, and let u_1, \dots, u_{t_v} be the resulting sequence. We have $t_v \leq t$ and, for every $i \in \{1, \dots, t_v - 1\}$, $f_{u_i} \geq f_{u_{i+1}}$. Let us construct a sequence $S = \psi_0, \dots, \psi_\ell$ of failure patterns, where $\psi_0 = \varphi_v$, and $\psi_\ell = \varphi_\emptyset$. This sequence is itself the concatenation of sub-sequences S_i for $i = 1, \dots, t_v$ such that $S_1 = \psi_0, \dots, \psi_{\ell_1}$, and, for every $i \in \{2, \dots, t_v\}$, $S_i = \psi_{\ell_{i-1}+1}, \dots, \psi_{\ell_i}$ with $0 \leq \ell_1 \leq \ell_2 \leq \dots \leq \ell_{t_v} = \ell$. For every sub-sequence S_i , $i \in \{1, \dots, t_v\}$, and for every $j \in \{\ell_{i-1} + 1, \dots, \ell_i - 1\}$, we set

$$\psi_{j+1} = \text{succ}(\psi_j, u_i).$$

Moreover, the first failure pattern $\psi_{\ell_{i-1}+1}$ in the sequence S_i is obtained from φ_v by removing the crashing nodes u_1, \dots, u_{i-1} , i.e., these nodes are correct in $\psi_{\ell_{i-1}+1}$. The last failure pattern ψ_{ℓ_i} of the sequence S_i is when the node u_i that crashes last in ψ_{ℓ_i} fails at round R .

▷ **Claim 15.** For any two consecutive failure patterns ψ_j and ψ_{j+1} in S , there exists a correct node w_j with the same view in both patterns after $R - 1$ rounds, that is,

$$\text{view}(w_j, \psi_j, R - 1) = \text{view}(w_j, \psi_{j+1}, R - 1).$$

To see why the claim holds, let us first assume that ψ_j and ψ_{j+1} belong to a same sub-sequence S_i . In this case, the claim directly follows from Lemma 13. If ψ_j and ψ_{j+1} do not belong to a same sub-sequence S_i , then ψ_j is the last element of a sub-sequence S_i , and ψ_{j+1} is the first element of sub-sequence S_{i+1} , then the claim follows from the fact that the sets of nodes crashing in ψ_j and ψ_{j+1} during round r are the same, for every $r \in \{1, \dots, R - 1\}$. This completes the proof of Claim 15.

From Claim 15, for any two consecutive failure patterns ψ_j and ψ_{j+1} in S , $\text{config}(\psi_j)$ and $\text{config}(\psi_{j+1})$ belong to the same connected component of $\text{IF}(G, R - 1)$. To wrap up, we have shown that, for every $v \in V$, there exists a connected component of $\text{IF}(G, R - 1)$ containing both $\text{config}(\varphi_\emptyset)$ and $\text{config}(\varphi_v)$. Recall that φ_v is a failure pattern in Φ_v^* satisfying that it contains no node different from v that fails at round 1, and $\text{ecc}(v, \varphi_v) \geq R$. At the end of round $R - 1$, no node dominates the component that contains $\text{config}(\varphi_\emptyset)$ because, for every node $v \in V$, v cannot dominates $\text{config}(\varphi_v, R - 1)$. ◀

5 Conclusion

In this paper, we have completed the picture for consensus in the t -resilient model for arbitrary graphs. That is, we have proved that the consensus algorithm in [4] is optimal, i.e., for every graph G and $t < \kappa(G)$, consensus can be solved by an oblivious algorithm performing in $\text{radius}(G, t)$ rounds under the t -resilient model, and no oblivious algorithms can solve consensus in G in less than $\text{radius}(G, t)$ rounds under the t -resilient model. Moreover, we have extended the study of consensus beyond the connectivity threshold. Specifically, we defined the *local* consensus task, a generalization of consensus. We designed and analyzed a generic algorithm for this task, which we believe to be optimal among oblivious algorithms. The technical difficulty of establishing optimality of our algorithm for the local variant of consensus yields from the fact that we miss an analog of our characterization theorem (cf. Theorem 9 in Section 4) for local consensus. Finally, we have generalized the algorithm in [4] for consensus, as well as our algorithm for local consensus, to k -set agreement.

Our results open a vast domain for further investigations. In particular, what could be said for sets of failure patterns Φ distinct from $\Phi_{\text{all}}^{(t)}$? The case Φ_{clean} of clean failures, for which there are no known generic consensus algorithms applying to arbitrary graphs, is particularly intriguing. Another intriguing and potentially challenging area for further research is exploring scenarios where no upper bounds on the number of failing nodes are assumed, by concentrating solely on the set Φ_{connect} of failure patterns that do not result in disconnecting the graph. The main difficulty is that basic results such as Lemma 1 in [4] (cf. Proposition 1) do not hold anymore in this framework. Indeed, some ill behaviors that do not occur when the number of failures is bounded from above by the connectivity of the graph, or when the problems are considered in each connected component separately, pop up when the number of failures is arbitrarily large yet preserving connectivity. Finally, the design of early-stopping algorithms in the t -resilient model for arbitrary graphs is also highly desirable. The early-stopping algorithms in [8] are very promising, but their analysis must be refined to a grain finer than the stretches of the failure patterns, by focusing on, e.g., eccentricities and radii.

References

- 1 Marcos Kawazoe Aguilera and Sam Toueg. A simple bivalency proof that t -resilient consensus requires $t+1$ rounds. *Information Processing Letters*, 71(3-4):155–158, 1999. doi:10.1016/S0020-0190(99)00100-3.
- 2 Hagit Attiya and Jennifer Welch. *Distributed computing: fundamentals, simulations, and advanced topics*, volume 19. John Wiley & Sons, 2004.
- 3 Armando Castañeda, Pierre Fraigniaud, Ami Paz, Sergio Rajsbaum, Matthieu Roy, and Corentin Travers. A topological perspective on distributed network algorithms. *Theoretical Computer Science*, 849:121–137, 2021. doi:10.1016/J.TCS.2020.10.012.
- 4 Armando Castañeda, Pierre Fraigniaud, Ami Paz, Sergio Rajsbaum, Matthieu Roy, and Corentin Travers. Synchronous t -resilient consensus in arbitrary graphs. *Inf. Comput.*, 292:105035, 2023. doi:10.1016/J.IC.2023.105035.
- 5 Armando Castañeda, Yoram Moses, Michel Raynal, and Matthieu Roy. Early decision and stopping in synchronous consensus: A predicate-based guided tour. In *5th International Conference on Networked Systems – (NETYS)*, volume 10299 of *LNCS*, pages 206–221, 2017. doi:10.1007/978-3-319-59647-1_16.
- 6 Bernadette Charron-Bost and Stephan Merz. Formal verification of a consensus algorithm in the heard-of model. *Int. J. Softw. Informatics*, 3(2-3):273–303, 2009. URL: http://www.ijsi.org/ch/reader/view_abstract.aspx?file_no=273&flag=1.

- 7 Bernadette Charron-Bost and André Schiper. The heard-of model: computing in distributed systems with benign faults. *Distributed Comput.*, 22(1):49–71, 2009. doi:10.1007/S00446-009-0084-6.
- 8 Bogdan S. Chlebus, Dariusz R. Kowalski, Jan Olkowski, and Jędrzej Olkowski. Disconnected agreement in networks prone to link failures. In *25th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*, volume 14310 of *LNCS*, pages 207–222. Springer, 2023. doi:10.1007/978-3-031-44274-2_16.
- 9 Étienne Coulouma and Emmanuel Godard. A characterization of dynamic networks where consensus is solvable. In *International Colloquium on Structural Information and Communication Complexity*, pages 24–35. Springer, 2013. doi:10.1007/978-3-319-03578-9_3.
- 10 Carole Delporte-Gallet, Hugues Fauconnier, Sergio Rajsbaum, and Nayuta Yanagisawa. A characterization of t -resilient colorless task anonymous solvability. In *25th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, volume 11085 of *LNCS*, pages 178–192. Springer, 2018. doi:10.1007/978-3-030-01325-7_18.
- 11 Carole Delporte-Gallet, Hugues Fauconnier, and Andreas Tielmann. Fault-tolerant consensus in unknown and anonymous networks. In *29th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 368–375, 2009. doi:10.1109/ICDCS.2009.36.
- 12 Danny Dolev. The byzantine generals strike again. *J. Algorithms*, 3(1):14–30, 1982. doi:10.1016/0196-6774(82)90004-9.
- 13 Danny Dolev and H. Raymond Strong. Authenticated algorithms for byzantine agreement. *SIAM Journal on Computing*, 12(4):656–666, 1983. doi:10.1137/0212045.
- 14 Shlomi Dolev. *Self-Stabilization*. MIT Press, 2000.
- 15 Pierre Fraigniaud, Patrick Lambein-Monette, and Mikaël Rabie. Fault tolerant coloring of the asynchronous cycle. In *36th International Symposium on Distributed Computing (DISC)*, volume 246 of *LIPICs*, pages 23:1–23:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.DISC.2022.23.
- 16 Pierre Fraigniaud, Minh Hang Nguyen, and Ami Paz. Agreement tasks in fault-prone synchronous networks of arbitrary structure. *CoRR arXiv*, abs/2410.21538, 2024. doi:10.48550/arXiv.2410.21538.
- 17 Pierre Fraigniaud and Ami Paz. The topology of local computing in networks. In *47th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 168 of *LIPICs*, pages 128:1–128:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.ICALP.2020.128.
- 18 Maurice Herlihy, Dmitry N. Kozlov, and Sergio Rajsbaum. *Distributed Computing Through Combinatorial Topology*. Morgan Kaufmann, 2013.
- 19 Juho Hirvonen and Jukka Suomela. *Distributed Algorithms*. Aalto University, Finland, 2023.
- 20 Muhammad Samir Khan, Syed Shalan Naqvi, and Nitin H. Vaidya. Exact byzantine consensus on undirected graphs under local broadcast model. In *PODC*, pages 327–336. ACM, 2019. doi:10.1145/3293611.3331619.
- 21 Giuseppe Antonio Di Luna and Giovanni Viglietta. Computing in anonymous dynamic networks is linear. In *63rd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1122–1133, 2022. doi:10.1109/FOCS54457.2022.00108.
- 22 Giuseppe Antonio Di Luna and Giovanni Viglietta. Optimal computation in leaderless and multi-leader disconnected anonymous dynamic networks. In *37th International Symposium on Distributed Computing (DISC)*, volume 281 of *LIPICs*, pages 18:1–18:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICs.DISC.2023.18.
- 23 Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.
- 24 Thomas Nowak, Ulrich Schmid, and Kyrill Winkler. Topological characterization of consensus under general message adversaries. In *Proceedings of the 2019 ACM symposium on principles of distributed computing*, pages 218–227, 2019. doi:10.1145/3293611.3331624.
- 25 David Peleg. *Distributed Computing: A Locality-sensitive Approach*. SIAM, 2000.

- 26 Michel Raynal. Consensus in synchronous systems: A concise guided tour. In *9th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pages 221–228. IEEE, 2002. doi:10.1109/PRDC.2002.1185641.
- 27 Michel Raynal. *Fault-tolerant Agreement in Synchronous Message-passing Systems*. Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool Publishers, 2010. doi:10.2200/S00294ED1V01Y201009DCT003.
- 28 Michel Raynal and Corentin Travers. Synchronous set agreement: A concise guided tour. In *12th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC)*, pages 267–274, 2006.
- 29 Kyrill Winkler, Ami Paz, Hugo Rincon Galeana, Stefan Schmid, and Ulrich Schmid. The time complexity of consensus under oblivious message adversaries. *Algorithmica*, pages 1–32, 2024.