Synchronous t-resilient Consensus in Arbitrary Graphs[⋆,⋆⋆]

Armando Castañeda[a], Pierre Fraigniaud[b], Ami Paz[c], Sergio Rajsbaum[a], Matthieu Roy[d], Corentin Travers[e,*]

[a] *Instituto de Matemáticas, UNAM, Mexico*
[b] *IRIF, CNRS and Université Paris Cité, France*
[c] *LISN, CNRS and Université Paris-Saclay, France*
[d] *LAAS, CNRS, Toulouse, France*
[e] *LIS, Université Aix-Marseille, France*

Abstract

We study the number of rounds needed to solve consensus in a synchronous network G where at most t nodes may fail by crashing. This problem has been thoroughly studied when G is a complete graph, but very little is known when G is arbitrary. We define a notion of $\mathsf{radius}(G,t)$, that extends the standard graph theoretical notion of radius, for considering all the ways in which t nodes may crash, and we present an algorithm that solves consensus in $\mathsf{radius}(G,t)$ rounds. Then we derive a lower bound showing that, among oblivious algorithms, our algorithm is optimal for a large family of graphs including all vertex-transitive graphs.

Keywords: Crash failures, Consensus, Combinatorial topology, Distributed graph algorithms

1. Introduction

The problem. We consider a synchronous message-passing distributed system, where at most t out of n nodes may fail by crashing. The nodes communicate by sending messages to each other over the edges of an undirected graph G known by the nodes. In the *consensus* problem each node is given an input, and after

⋆Supported by ANR Project DUCAT, INRIA Project GANG, UNAM-PAPIIT grants IA102417, IN108720, IN108723, IN109917 and IN106520, Fondation des Sciences Mathématiques de Paris, and Austrian Science Fund (FWF) and netIDEE SCIENCE project P33775-N.

⋆⋆A preliminary version of this work appeared in the proceeding of SSS 2019 [5].

*Corresponding author

Email addresses: `armando.castaneda@im.unam.mx` (Armando Castañeda), `pierref@irif.fr` (Pierre Fraigniaud), `ami.paz@lisn.fr` (Ami Paz), `rajsbaum@im.unam.mx` (Sergio Rajsbaum), `roy@laas.fr` (Matthieu Roy), `corentin.travers@lis-lab.fr` (Corentin Travers)

some number of rounds produces an output, such that all outputs are the same and must be equal to one of the inputs.

One of the earliest and most well-know facts in distributed computing is that the number of rounds needed to solve consensus when G is the complete graph, K_n, is $t+1$. Namely, for $t \leq n-1$, and any algorithm requires this number of rounds in the worst case. The round complexity to solve consensus in K_n has been thoroughly studied, but not for graphs other than the complete graph.

1.1. Results

This paper studies the number of rounds needed to solve consensus, as a function of G and t. It presents two main contributions, inspired by the recently introduced [6] *information flow* perspective.

First, it shows that for any given $(t+1)$-node-connected graph G (i.e., a graph that is connected after the removal of any t nodes), it is possible to solve consensus tolerating t failures, in $\mathsf{radius}(G,t)$ rounds. Roughly, the *eccentricity of v against t failures*, $\mathsf{ecc}(v,t)$, is the smallest number of rounds needed for a node v to broadcast its input value, independently of the failure pattern (when and how nodes crash). Then, $\mathsf{radius}(G,t)$ is equal to the smallest $\mathsf{ecc}(v,t)$, over all nodes v. Both these notions extend the standard notions of eccentricity and radius, and equal to them when $t=0$. For example, $\mathsf{radius}(K_n,t)=t+1$ for the complete graph and $\mathsf{radius}(C_n,1)=n-1$ for the cycle. For the wheel graph W_n, composed of an $(n-1)$-cycle and one extra node connected to all other nodes, $\mathsf{radius}(W_n,2)=n-1$ and $\mathsf{radius}(W_n,1)=1+\lfloor(n-1)/2\rfloor$.

Second, the paper presents a matching lower bound, showing that our algorithm is optimal among oblivious algorithms, in any graph that is vertex-transitive. In an *oblivious* algorithm, the decision value of a node is based solely on the set of input values it has seen so far. Roughly speaking, a graph is *vertex-transitive* if it is highly symmetric. This is a large and well studied class of graphs (see, e.g., [19]). A core difficulty in analyzing our model yields from the "non-clean" crashes, that is, the fact that a node may fail "at the middle" of a round, i.e., it may send messages to some of its neighbors, but not to others. In fact, we show that, for clean crashes that take place initially (i.e., all failing nodes do not perform any round of communication), a faster algorithm exists, and the lower bound does not hold.

Direct generalizations of known upper and lower bound techniques from a complete graph to general graphs seem difficult to obtain. Instead, both our upper and lower bounds use novel ideas, which we discuss next.

1.1.1. Our upper bound techniques.

In a classic algorithm to solve consensus on a complete graph, e.g. [30], nodes repeatedly send all the inputs they know, and at the end of round $t+1$, each node that has not crashed, decides the smallest input value among the values it has seen. The usual agreement argument is that among the $t+1$ rounds there must be at least one in which no node crashes. All nodes that are alive at the end of such a round have seen the same set of inputs, and there is *common knowledge* [15] on a set of inputs. This argument holds only under the assumption that the graph is complete. We use a similar idea on an arbitrary graph, but based on a more general *information flow* argument [6].

Given a node v and its $\mathsf{ecc}(v,t)$, we show that at the end of round $\mathsf{ecc}(v,t)$, either all alive nodes have received v's input, or none has. For the complete graph, $\mathsf{ecc}(v,t) = t+1$ for all nodes v, and indeed, for any node v, either all nodes have received the input of v by round $t+1$, or no node will ever receive it. This implies the correctness of the algorithm for the complete graph described above. Notice that the eccentricity is not less than $t+1$, because the adversary may create a *hidden path*, $v_1, \ldots, v_t$ such that $v_1 = v$ and each v_i, $1 \le i \le t-1$, fails in round i and sends a message to only v_{i+1} before failing.

We use this information flow perspective to derive simple consensus algorithms for arbitrary graphs. Each node repeatedly forwards all the pairs (v, in_v) it knows about, where in_v is the input value of node v. An algorithm is specified by two functions: $\mathsf{R}(G,t)$ which returns the number of rounds to execute, and $\mathsf{D}(G,t)$ which tells a node which value to decide, among the input values it has seen. After $\mathsf{R}(G,t)$ rounds, the active nodes have the same view of the inputs of a carefully chosen *subset* of $t+1$ nodes, thus, after $\mathsf{R}(G,t)$ rounds, $\mathsf{D}(G,t)$ can pick deterministically the input of one of these nodes. Remarkably, our lower bound shows that this is not necessarily the case after fewer rounds.

1.1.2. Our lower bound techniques.

There are several lower bound proofs for the number of rounds to solve consensus under crash failures for the case when G is a complete graph. The classic $t+1$ lower bound proof style proceeds by a rather complex backward induction (a detailed description appears in [26]). Later on, simpler forward induction proofs were discovered [1, 27], following the classical bivalency arguments that were originally developed for proving the impossibility of solving consensus in asynchronous systems [18].

The aforementioned proofs hold for general graphs as well, namely, $t+1$ rounds is a lower bound for solving consensus on any graph G. However, for general graphs this bound is very weak, as it does not take into consideration the structure of the graph. An obvious example is a cycle with $t=1$: our lower bound is $n-1$, while the standard approaches give a lower bound of 2 rounds.

Our lower bound technique is different from both the backward and the forward arguments. It is inspired by the topological techniques for distributed computing [21], though we do not use topology explicitly. Our lower bound technique is similar to the connectivity analysis of the *protocol complex*, the structure of states at the end of executions of an algorithm after a certain number of rounds. However, instead of working with the protocol complex, we consider an *information flow* directed graph version based on failure patterns, without including input values. We prove that consensus is solvable by an oblivious algorithm if and only if all connected components of the information flow graph have a *dominating* node, namely, a node with an edge from it to any other node in its connected component. In [6] we introduced this information flow perspective, and used it to study set agreement and approximate agreement.

The seminal paper [15] shows that, as soon as there is common knowledge of a *clean* round (where a node that crashes does not send any messages), it is also common knowledge that nodes have identical views of the initial configuration. As a consequence, any action that depends on the system's initial configuration can be carried out simultaneously in a consistent way by the set of active nodes at any round $k \geq t+1$, if it can be carried out at all. Our lower bound is larger than $t+1$ on general graphs, and hence shows how the round in which nodes have common knowledge of a subset of the input configuration is affected also by the structure of the graph.

1.2. Related work

Consensus in the failure-prone synchronous model has been thoroughly studied since the beginning of the distributed computing field in the late 1970's [35]. A variety of aspects have been considered, including the number of rounds (in great detail, including worst case, early deciding, simultaneous, unbeatability, etc.), number and size of messages, variants of consensus, in static and dynamic networks, and under various failure models. We only mention some of the most relevant papers, among a vast literature, which is covered only partially even by surveys, e.g. [8, 30] and textbooks on the field, e.g. [4, 26, 31].

For general graphs, since early on there has been an interest in characterizing the graphs where consensus is solvable, initially for Byzantine failures [13, 14, 17]. It was observed early on [25] that $t+1$ connectivity is necessary and an exponential algorithm was described. The algorithms for Byzantine settings also work in our model. However, they have not been optimized for the number of rounds, and furthermore, our setting requires only $t+1$ node-connectivity, while an algorithm tolerating Byzantine failures requires $n \geq 3t+1$, and node-connectivity at least $2t+1$ [13]. Very recently, consensus algorithms for general graphs were designed, for *local broadcast* Byzantine failures [23]. One algorithm works in the local broadcast model on a graph under the weakest requirements—minimum degree $2t$, and $(\lfloor 3t/2+1 \rfloor)$ node-connected; however, it has an exponential time complexity. A different consensus algorithm terminates in $3n$ rounds, but only assuming the graph is $2t$-connected. There has also been work on characterizing the *directed* graphs for which fault tolerant synchronous consensus is solvable, both under crash and under Byzantine failures [33, 34].

We are not aware of any previous lower bound techniques for solving consensus in an arbitrary graph G. A simple lower bound, that can be proven using standard indistinguishability arguments, is the maximum radius among the graphs created by removing at most t nodes from G. However, this yields only a trivial 1-round lower bound for the complete graph. A lower bound of $t+1$ rounds for the complete graph was proven using other methods, specifically crafted for the complete graph case, first for Byzantine failures [16], later for the case were digital signatures can be used [14], and finally to crash failures (see, e.g., [20]).

Our lower bound technique is mainly inspired by the topological techniques for distributed computing [21], and more specifically by the topological structure of the executions of a synchronous algorithm after a certain number of rounds [22]. Indeed, the technique used for deriving our second algorithm is reminiscent of topological existential upper bounds proofs used in the past [3, 9]. Hidden paths have played an important role in the design of *early-deciding* consensus algorithms in the complete graph [7].

Research on *dynamic networks* also characterizes families of networks for which consensus (or a variant of it) is solvable [10, 12, 28, 32, 36]. Interestingly, dynamic networks research and works on synchronous fault-tolerant consensus [33, 34] share the idea of picking a node as a source, and having all nodes deciding on the input of this source. In Theorem 3 we present an information flow characterization for consensus, in terms of such a source. Our notion of a core set (see Section 3.2) can be seen as a refinement of such notions, defined in order to optimize the number of rounds. Interestingly, [28] presents a topological solvability characterization of consensus using the *point set topology* techniques introduced in [2].

2. Preliminaries

Model of Computation. We consider the standard synchronous message-passing model of computation where at most t nodes may fail by crashing. A set of $n \geq 2$ nodes V communicate through reliable bidirectional channels E defining a graph $G = (V, E)$. In the remainder of the paper, we fix G and t, and assume $t < \kappa(G)$, the *node connectivity* of G, i.e., the minimum number of nodes whose deletion disconnects G. Fixing G means that the algorithm performed at each node may depend on the graph G, and on the node's location in it. This assumption allows us to focus solely on the uncertainty caused by crashes, and not by the structure of the network, like it is the case in the classical framework $G = K_n$, the complete graph on n vertices. Each node u of G is identified by a *name*, which is unique in G, that can be viewed as an integer ID in $\{1, \ldots, n\}$. For the sake of simplifying the presentation, we do not make a distinction between the node v itself, and its name. For instance, when referring to the "smallest node", we merely refer to "the node with smallest name".

An *execution* proceeds in a infinite sequence of synchronous rounds, starting in round 1. In every round, each node v first performs some local computation, then sends a message to each of its neighbors in G, denoted $N(v)$, and then receives the messages sent to it from $N(v)$ in that round. When a node crashes in round r, it fails to send its message to some of its neighbors in round r, and sends no message in subsequent rounds. We focus on *full information* algorithms, i.e., each message sent by a node contains all the node's state.

A *failure pattern* φ for G and t specifies, for each node that fails, in which round it fails, and which messages it fails to send. It is a set of triples of the form (v, F_v, f_v), indicating that v crashes in round f_v, in which it does not send the messages to the neighbors in $F_v \subseteq N(v)$, where $F_v \neq \varnothing$. Note that we may have $F_v = N(v)$, in which case the crash is called *clean.* Since at most t nodes can fail, $|\varphi| \leq t$, and since nodes do not recover from a failure, if $(v, F_v, f_v) \in \varphi$ and $(u, F_u, f_u) \in \varphi$, then $v \neq u$.

For an execution with failure pattern φ, the *faulty* nodes are those that appear in a triplet in φ; the others are the *correct* nodes. A node is *active* in round r in φ if it is correct, or if it fails in a round later than r. A node that crashes with $F_v = N(v)$ is said to crash *cleanly* in φ.

Consider any input assignment to the nodes. Our algorithms are of the following form. Initially, for each node v with input in_v, its *view* is $\{(v, in_v)\}$. In each round, each node v sends its *view* to $N(v)$, and at the end of the round it updates its *view* with the new input value-pairs it receives.

Given a failure pattern φ, we say that u *hears from* v *in* φ, if in some round u receives a message containing the input of v. Similarly, we say that u hears from v *by round* r in φ if u receives a message with v's input in round r, or before. In other words, there is a *causal path* from u to v [24] in an execution with failure pattern φ. In more detail, there is a causal path $u = u_0 \to \ldots \to u_\ell = v$ from u to v if there exist $\ell + 1$ distinct nodes $u_0, \ldots, u_\ell$ with $u_0 = u$ and $u_\ell = v$ such that for each $i, 1 \leq i \leq \ell$:

- $u_i \in N(u_{i-1})$ and
- If u_{i-1} fails at round r in φ then either $r > i$, or $r = i$ and u_{i-1} sends a message to u_i in round r, i.e. $u_i \notin F_{u_{i-1}}$.

Clearly, the existence of such a path depends on φ, but not on the input assignment. Thus, to analyze the structure of all possible failure patterns, we ignore the input values. This is what we do next, where we may identify φ with the infinite execution with that failure pattern.

Eccentricity and Radius in Failure Patterns. Let $\mathsf{dist}_G(u, v)$ denote the distance between nodes u and v in $G = (V, E)$. The *eccentricity* of a node $v \in V$ is defined as $\mathsf{ecc}_G(v) = \max_{u \in V} \mathsf{dist}_G(u, v)$. The *diameter* of a graph is defined as $\max_{v \in V} \mathsf{ecc}_G(v)$, and its *radius* as $\min_{v \in V} \mathsf{ecc}_G(v)$. We generalize the notions of eccentricity and radius to the synchronous t-resilient model.

In the following, failure patterns are denoted by lower case Greek letters $\varphi, \psi, \ldots$, and sets of failure patterns are denoted by upper case Greek letters $\Phi, \Psi, \ldots$. We denote by $\Phi^{(t)}_{\text{all}}$ the set of *all* failure patterns for G and t. The failure pattern in which no nodes crash is $\varphi_\varnothing$, and hence $\Phi^{(0)}_{\text{all}} = \{\varphi_\varnothing\}$.

Definition 1. *Given a node $v \in V$ and a failure pattern $\varphi \in \Phi^{(t)}_{\text{all}}$, the* eccentricity $\mathsf{ecc}_G(v, \varphi) \in \mathbb{N} \cup \{\infty\}$ *of v in φ is the minimum number of rounds required for all correct nodes to hear from v (i.e., there is causal path from v to every correct node), or ∞ if not all correct nodes hear from v. If $\mathsf{ecc}_G(v, \varphi) \in \mathbb{N}$, we say that v* floods *to the correct nodes in φ.*

Consider any φ. Notice that since G is at least $(t + 1)$-connected, and at most t nodes crash, if a correct node u hears from v, then every correct node receives a message from v (because a message can get from u to every correct node). We thus have the following claim.

Fact 1. *For every $v \in V$, and every $\varphi \in \Phi^{(t)}_{\text{all}}$, if $\mathsf{ecc}_G(v,\varphi) = \infty$ then* no correct node *hears from v in φ.*

Definition 2. *For $v \in V$ and $\Phi \subseteq \Phi^{(t)}_{\text{all}}$, such that there is at least one $\varphi \in \Phi$ with $\mathsf{ecc}_G(v,\varphi) \in \mathbb{N}$, let*

$$\mathsf{ecc}_G(v,\Phi) = \max\{\mathsf{ecc}_G(v,\varphi) \ : \ \varphi \in \Phi, \mathsf{ecc}_G(v,\varphi) \in \mathbb{N}\}.$$

Notice that, for any Φ containing failure patterns where v is correct, there is at least one $\varphi \in \Phi$ with $\mathsf{ecc}_G(v,\varphi) \in \mathbb{N}$.

Lemma 1. *For $v \in V$ and $\varphi \in \Phi^{(t)}_{\text{all}}$, let A be the set of all active nodes in round $\mathsf{ecc}_G(v,\Phi^{(t)}_{\text{all}})$ under φ. Either all nodes in A hear from v by round $\mathsf{ecc}_G(v,\Phi^{(t)}_{\text{all}})$, or no node in A hears from v by round $\mathsf{ecc}_G(v,\Phi^{(t)}_{\text{all}})$ in φ.*

Proof. Let $\varphi' \in \Phi^{(t)}_{\text{all}}$ be the failure pattern identical to φ in the first $\mathsf{ecc}_G(v,\Phi^{(t)}_{\text{all}})$ rounds, but with all the nodes of A correct in φ'. Then, the nodes in A have the same view in both φ and φ' in round $\mathsf{ecc}_G(v,\Phi^{(t)}_{\text{all}})$.

If $\mathsf{ecc}_G(v,\varphi') \in \mathbb{N}$, by Definition 1, all nodes in A hear from v by time $\mathsf{ecc}_G(v,\varphi')$, which is at most $\mathsf{ecc}_G(v,\Phi^{(t)}_{\text{all}})$, by Definition 2. The same is true for φ, as φ and φ' are identical in the first $\mathsf{ecc}_G(v,\Phi^{(t)}_{\text{all}})$ rounds.

If $\mathsf{ecc}_G(v,\varphi') = \infty$, no node in A hears from v in φ', by Fact 1, and then no node in A hears from v by round $\mathsf{ecc}_G(v,\Phi^{(t)}_{\text{all}})$ in φ because φ and φ' are identical in the first $\mathsf{ecc}_G(v,\Phi^{(t)}_{\text{all}})$ rounds. $\square$

Note that Lemma 1, which holds for the family $\Phi^{(t)}_{\text{all}}$, may not hold for every family Φ of failure patterns. Indeed, the failure pattern φ' constructed from φ in the proof of Lemma 1 needs to belong to Φ, which is to say that Φ must be stable by the transformation changing φ into φ', which is not true for all Φ, but holds for $\Phi^{(t)}_{\text{all}}$.

Definition 3. *Let $\Phi \subseteq \Phi^{(t)}_{\text{all}}$ such that for every $v \in V$ there is at least one $\varphi \in \Phi$ with $\mathsf{ecc}_G(v,\varphi) \in \mathbb{N}$. The* radius *of G with respect to Φ is defined as $\mathsf{radius}(G,\Phi) = \min_{v \in V} \mathsf{ecc}_G(v,\Phi)$.*

For $t = 0$, our notion of eccentricity and radius coincides with the classical graph-theoretic definition, i.e., $\mathsf{ecc}_G(v,\Phi^{(0)}_{\text{all}}) = \mathsf{ecc}_G(v)$ and $\mathsf{radius}(G,\Phi^{(0)}_{\text{all}}) = \mathsf{radius}(G)$. Moreover, in the complete graph K_n, we have $\mathsf{radius}(K_n,\Phi^{(t)}_{\text{all}}) = t+1$, which together with Lemma 1 implies the correctness of the simple algorithm discussed in the Introduction.

3. Consensus Algorithms in Arbitrary Graphs

We consider the usual *consensus* problem in which each node starts with an input value, defined by the following properties.

- **Termination:** Every correct node decides a value
- **Validity:** The decision of a node is equal to the input of some node;
- **Agreement:** The decisions of any pair of nodes are the same.

This version of consensus is sometimes called *uniform* since the agreement property requires that *all* decisions must be the same. In the *nonuniform* version of the problem, it is required that *only* the decisions of correct nodes are the same. In our consensus algorithms all decisions are taken at the same time, and hence they solve both versions of the problem.

Oblivious algorithms. Recall that in our algorithms, a node resends to its neighbors the set of input values it has received, each one together with the name of the node that has the corresponding input value. Thus, to specify a consensus algorithm, we define a function $\mathsf{R}(G,t)$ that returns a round number, stating that all correct nodes decide in round $\mathsf{R}(G,t)$. Also, we define a *decision function* $\mathsf{D}(G,t)$ used by a node to select a consensus value from its view (possibly taking in consideration the names of the nodes that proposed this inputs, and the structure of G and t). Formally, $\mathsf{D}(G,t)$ is a function from the set with all views to the output set. In a t-fault tolerant oblivious consensus algorithm for G, after $\mathsf{R}(G,t)$ rounds of communication (independently of the failure pattern or the input assignment), each node selects a value from its view, as specified by the function $\mathsf{D}(G,t)$. We stress that G is fixed in the paper, and $\mathsf{R}(G,t)$ and $\mathsf{D}(G,t)$ are not computed by the nodes, they are given as part of the algorithm. (Note however that if the nodes "know" G, t, and there relative positions in the graph, then they can compute these functions locally).

3.1. A naive algorithm

We describe a naive algorithm, $\mathsf{P}^{G,t}_{\mathsf{ecc}} = (\mathsf{R}_{\mathsf{ecc}}(G,t), \mathsf{D}_{\mathsf{ecc}}(G,t))$, based on a simple idea. Let us order the n nodes of G as $v_1, \dots, v_n$, with

$$\mathsf{ecc}_G(v_i, \Phi^{(t)}_{\text{all}}) \leq \mathsf{ecc}_G(v_{i+1}, \Phi^{(t)}_{\text{all}}) \tag{1}$$

for $1 \leq i < n$. In particular, we have $\mathsf{radius}(G, \Phi^{(t)}_{\text{all}}) = \mathsf{ecc}_G(v_1, \Phi^{(t)}_{\text{all}})$.

Let $\mathsf{R}_{\mathsf{ecc}}(G,t) = \mathsf{ecc}_G(v_{t+1}, \Phi^{(t)}_{\text{all}})$, and $\mathsf{D}_{\mathsf{ecc}}(G,t)$ be the function that, given a view, returns the input of the smallest[1] node among the nodes in $\{v_1, \dots, v_{t+1}\}$.

Theorem 1. *Algorithm* $\mathsf{P}^{G,t}_{\mathsf{ecc}}$ *solves consensus in* $\mathsf{ecc}_G(v_{t+1}, \Phi^{(t)}_{\text{all}})$ *rounds.*

Proof. The algorithm satisfies termination as all correct nodes run $\mathsf{R}_{\mathsf{ecc}}(G,t) = \mathsf{ecc}_G(v_{t+1}, \Phi^{(t)}_{\text{all}})$ rounds. For validity, the definition of $\mathsf{ecc}_G(v_{t+1}, \Phi^{(t)}_{\text{all}})$ and Equation 1 imply that all nodes receive at least one input of a node in $\{v_1, \dots, v_{t+1}\}$ by round $\mathsf{ecc}_G(v_{t+1}, \Phi^{(t)}_{\text{all}})$, in every $\varphi \in \Phi^{(t)}_{\text{all}}$. For agreement, consider any

[1] Assuming V is a totally ordered set.

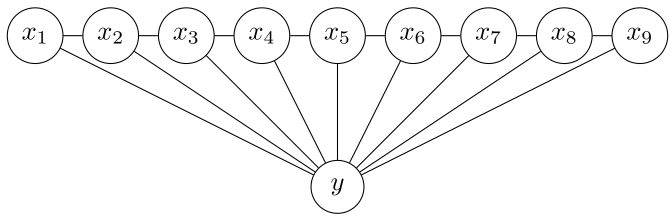

Figure 1: A graph for which $\mathsf{P}_{\mathsf{ecc}}^{G,t}$ is not time optimal.

$\varphi \in \Phi_{\text{all}}^{(t)}$ and the set A of all nodes that are active in round $\mathsf{ecc}_G(v_{t+1}, \Phi_{\text{all}}^{(t)})$ in φ. Lemma 1 and Equation 1 imply that either all nodes in A have received v_i's input, $1 \leq i \leq t+1$, in round $\mathsf{ecc}_G(v_{t+1}, \Phi_{\text{all}}^{(t)})$ in φ, or none of them has received it in that round. Therefore, all nodes in A have the same view of the inputs of the nodes $v_1, \ldots, v_{t+1}$, hence $\mathsf{D}_{\mathsf{ecc}}(G,t)$ returns the same value to all of them. $\square$

It is easy to come up with graphs for which this solution is not optimal, in terms of number of rounds.

Lemma 2. *There is a graph G for which $\mathsf{P}_{\mathsf{ecc}}^{G,t}$ is not time optimal, with $t = 1$.*

Proof. Consider the graph G with $n = 2k+2$ nodes, $k \geq 4$, consisting of a path $(x_1, \ldots, x_{2k+1})$ plus a universal node y connected to every x_i, $i = 1, \ldots, 2k+1$ (See figure 1 for the case $k = 4$.). Set $t = 1$.

Observe that if y does not crash, then for every $i, 1 \leq i \leq 2k+1$, every node hears from x_i in at most 3 rounds, unless x_i crashes cleanly in the first round. If y crashes at the first round, at least one node hears from x_i not before at least k round (the exact number depends on i). As per y, observe that $\mathsf{ecc}_G(y, \Phi_y^{\mathbb{N}}) = 2k+1$, which is reached when y crashes at round 1, sending a message only on the edge $\{y, x_1\}$. A systematic analysis demonstrates that $\mathsf{radius}(G, \Phi_{\text{all}}^{(t)}) = \mathsf{ecc}_G(x_{k+1}, \Phi_{x_{k+1}}^{\mathbb{N}}) = k$, i.e., $v_1 = x_{k+1}$. Similarly, we have $\mathsf{ecc}_G(x_k, \Phi_{x_k}^{\mathbb{N}}) = k+1$, and $v_2 = x_k$. Therefore, the naive algorithm performs in $\mathsf{R}_{\mathsf{ecc}}(G,1) = k+1$ rounds in G, with $\mathsf{D}_{\mathsf{ecc}}(G,1)$ using the set $D = \{x_k, x_{k+1}\}$.

Instead, consider the set $D' = \{y, x_{k+1}\}$, and perform flooding for $k = R-1$ rounds, with the objective of having nodes collecting the inputs of the nodes in D'. If the actual failure pattern φ satisfies $\varphi \in \Phi_{x_{k+1}}^{\mathbb{N}}$, then every correct node receives the input of x_{k+1} by the end of round k, as $\mathsf{ecc}_G(x_{k+1}, \Phi_{x_{k+1}}^{\mathbb{N}}) = k$. Otherwise, i.e., if $\varphi \in \Phi_{x_{k+1}}^{\infty}$, then, by Fact 1, no correct nodes receive the input of x_{k+1}, no matter how many rounds of flooding are performed. On the other hand, we have $\mathsf{ecc}_G(y, \Phi_{x_{k+1}}^{\infty}) = 1$, because y does not crash in any failure pattern in $\Phi_{x_{k+1}}^{\infty}$ as, by Fact 2, x_{k+1} must be the (unique) node that crashes in $\Phi_{x_{k+1}}^{\infty}$. In other words, either (1) all correct nodes receive the input from x_{k+1} in k rounds, or (2) no correct node receives this input, but they all have received the input from y. Therefore, if the nodes adopt the input of x_{k+1} whenever they receive it, or the input of y whenever they have not received the input of x_{k+1}, then consensus is reached, after $k < R$ rounds. $\square$

3.2. An adaptive-eccentricity based algorithm

The algorithm $\mathsf{P}_{\mathsf{ecc}}^{G,t}$ is based on a *core* set of nodes $\{v_1, \ldots, v_{t+1}\}$, consisting of the first $t+1$ nodes in order of ascending eccentricity. We show here that there is a more clever way of selecting a core set of $t+1$ nodes. The corresponding algorithm, $\mathsf{P}_{\mathsf{adapt}}^{G,t} = (\mathsf{R}_{\mathsf{adapt}}(G,t), \mathsf{D}_{\mathsf{adapt}}(G,t))$, is similar, except that, $\mathsf{R}_{\mathsf{adapt}}(G,t) = \mathsf{radius}(G, \Phi_{\text{all}}^{(t)})$. As before, $\mathsf{D}_{\mathsf{adapt}}(G,t)$ returns the input of the smallest node among the core set, but now the core set is $\{s_1, \ldots, s_{t+1}\}$, as defined next.

The first node s_1 is the same v_1 as in $\mathsf{P}_{\mathsf{ecc}}^{G,t}$. To choose the i-th node, we consider all the un-chosen nodes, and their eccentricity *only among the failure patterns where the previously selected nodes have ∞ eccentricity*, and take the node that minimizes this quantity.

Formally, to define the core set of $t+1$ nodes, we construct a sequence of pairs (s_i, Φ_i), with $s_i \in V$, and $\Phi_i \subseteq \Phi_{\text{all}}^{(t)}$, for $i = 1, \ldots, t+1$, inductively, as follows. For every node $v \in V$, let $\Phi_v^{\infty} = \{\varphi \in \Phi_{\text{all}}^{(t)} : \mathsf{ecc}_G(v, \varphi) = \infty\}$ and $\Phi_v^{\mathbb{N}} = \{\varphi \in \Phi_{\text{all}}^{(t)} : \mathsf{ecc}_G(v, \varphi) \in \mathbb{N}\}$.

Let $\Phi_0 = \Phi_{\text{all}}^{(t)}$, and, for $i = 1, \ldots, t+1$, let

$$\left\{ \begin{array}{lcl} s_i & = & \arg\min_{v \in V \smallsetminus \{s_1, \ldots, s_{i-1}\}} \mathsf{ecc}_G(v, \Phi_v^{\mathbb{N}} \cap \Phi_{i-1}), \\ \Phi_i & = & \Phi_{s_i}^{\infty} \cap \Phi_{i-1}, \end{array} \right. \tag{2}$$

where, for $i = 1$, we interpret $\{s_1, \ldots, s_{i-1}\}$ as the empty set. In other words, $\Phi_i = \Phi_{s_1}^{\infty} \cap \cdots \cap \Phi_{s_i}^{\infty}$, and also $\Phi_i = \Phi_{i-1} \smallsetminus \Phi_{s_i}^{\mathbb{N}}$. Observe that, for every $i = 1, \ldots, t+1$, and every $v \in V \smallsetminus \{s_1, \ldots, s_{i-1}\}$, $\Phi_v^{\mathbb{N}} \cap \Phi_{i-1}$ is not empty as it contains the failure pattern in which all nodes $s_1, \ldots, s_{i-1}$ crash cleanly at the first round, and no other node crashes. Also note that $\mathsf{ecc}_G(s_1, \Phi_{s_1}^{\mathbb{N}}) = \mathsf{radius}(G, \Phi_{\text{all}}^{(t)})$.

For example, in K_n, we have $\mathsf{ecc}_{K_n}(s_i, \Phi_{s_i}^{\mathbb{N}}) = t - i + 2$ for $i = 1, \ldots, t+1$ whenever $t < n-1$. For $t = n-1$, we have $\mathsf{ecc}_{K_n}(s_i, \Phi_{s_i}^{\mathbb{N}}) = n - i$ for $i = 1, \ldots, n$. In the cycle C_n with $t = 1$, we have $\mathsf{ecc}_{C_n}(s_1, \Phi_{s_1}^{\mathbb{N}}) = n - 1$ and $\mathsf{ecc}_{C_n}(s_2, \Phi_{s_2}^{\mathbb{N}}) = \lfloor \frac{n-1}{2} \rfloor$. For the graph G in Figure 1, $s_1 = x_5$ and $s_2 = y$, $\mathsf{ecc}_G(s_1, \Phi_{s_1}^{\mathbb{N}}) = \mathsf{radius}(G, \Phi_{all}^{(1)}) = 4$, and $\mathsf{ecc}_G(s_2, \Phi_{s_2}^{\mathbb{N}}) = 1$.

The *core set* for G, t is $\{s_1, \ldots, s_{t+1}\}$, and the *core sequence* for G is the ordered sequence $(s_1, \ldots, s_{t+1})$. A crucial property of this sequence is that, while the sequence $(\mathsf{ecc}_G(v_i, \Phi_{v_i}^{\mathbb{N}}))_{1 \le i \le t+1}$ defined in Eq. (1) is non decreasing, and may even be increasing, the sequence $(\mathsf{ecc}_G(s_i, \Phi_{s_i}^{\mathbb{N}} \cap \Phi_{i-1}))_{1 \le i \le t+1}$ defined in Eq. (2) is non increasing, and is actually always decreasing. Intuitively, this is because the maximization in the computation of $\mathsf{ecc}_G(v, \Phi_v^{\mathbb{N}} \cap \Phi_i)$ for determining s_{i+1} is taken over the set $\Phi_v^{\mathbb{N}} \cap \Phi_i$ which is smaller than the set $\Phi_v^{\mathbb{N}} \cap \Phi_{i-1}$ used for the computation of s_i.

Lemma 3. *Consider the core sequence $(s_1, \ldots, s_{t+1})$ and the pairs (s_i, Φ_i) defined in Eq. (2). Then, $\mathsf{ecc}_G(s_i, \Phi_{s_i}^{\mathbb{N}} \cap \Phi_{i-1})) > \mathsf{ecc}_G(s_{i+1}, \Phi_{s_{i+1}}^{\mathbb{N}} \cap \Phi_i))$, for $i \in \{1, \ldots, t\}$.*

The proof of this lemma uses the following fact:

Fact 2. *For every $v \in V$, and every failure pattern $\varphi \in \Phi^{(t)}_{\text{all}}$, if $\mathsf{ecc}_G(v, \varphi) = \infty$ then v crashes at round 1 in φ.*

Proof. Assume for contradiction that v does not crash an in the first round, but still, $\mathsf{ecc}_G(v, \varphi) = \infty$. As $\deg_G(v) \geq \kappa(G) > t$ and since v does not crash in round 1, there is a correct node that receives the input of v in round 1. By the contrapositive of Fact 1, $\mathsf{ecc}_G(v, \varphi) \in \mathbb{N}$: a contradiction. $\square$

We now are ready to prove Lemma 3.

Proof of Lemma 3. Fix $1 \leq i \leq t$. Recall that s_{i+1} is defined as

$$s_{i+1} = \underset{v \in V \smallsetminus \{s_1, \ldots, s_i\}}{\arg\min} \; \mathsf{ecc}_G(v, \Phi^{\mathbb{N}}_v \cap \Phi_i).$$

Thus, it is enough to identify a node $v \notin \{s_1, \ldots s_i\}$ that satisfies $\mathsf{ecc}_G(s_i, \Phi^{\mathbb{N}}_{s_i} \cap \Phi_{i-1}) > \mathsf{ecc}_G(v, \Phi^{\mathbb{N}}_v \cap \Phi_i)$. We show that a neighbor v of s_i satisfies this. Let $v \notin \{s_1, \ldots s_i\}$ be a neighbor of s_i. Note that such a neighbor v exists, as $\deg_G(s_i) \geq \kappa(G) > t$. Let $\varphi \in \Phi^{\mathbb{N}}_v \cap \Phi_i$, i.e., $\varphi \in \Phi_i$ and $\mathsf{ecc}_G(v, \varphi) < \infty$. For each $(w, F_w, f_w) \in \varphi$, define the triplet (w, F'_w, φ'_w) as follows:

$$F'_w = \begin{cases} F_w & \text{if } w \notin \{s_1, \ldots, s_i\} \\ N(w) & \text{if } w \in \{s_1, \ldots, s_{i-1}\} \\ N(w) \smallsetminus \{v\} & \text{if } w = s_i \end{cases}$$

and

$$f'_w = \begin{cases} f_w + 1 & \text{if } w \notin \{s_1, \ldots, s_i\} \\ 1 & \text{if } w \in \{s_1, \ldots, s_i\} \end{cases}.$$

Let φ' be the failure pattern defined by these triplets. That is, $s_1, \ldots, s_{i-1}$ fail cleanly in the first round, s_i sends a message to v and then fails, and the rest of the nodes fail as in φ, but one round latter.

The crux of the proof lays in the following fact: $\mathsf{ecc}_G(v, \varphi) = \mathsf{ecc}_G(s_i, \varphi') - 1$. To see this, note that the set of correct nodes in φ and φ' is the same, and let u be such a correct node. As $\mathsf{ecc}_G(v, \varphi) < \infty$, there exists a causal path from v to u under φ. By Fact 2, the nodes $s_1, \ldots, s_i$ crash at round 1 in φ, so the path does not go through them. The failure pattern φ' is designed such that the same path exists in φ', even when starting in round 2. Hence, there is a causal path from s_1 to u in φ'. This path starts by a message from s_1 to v in the first round, and continues as the previous path, until u. This implies that $\mathsf{ecc}_G(v, \varphi) \geq \mathsf{ecc}_G(s_i, \varphi') - 1$. The proof of the opposite inequality is almost the same. Namely, any causal path in φ' starting from s_1 must contain a path from v that starts one round later, and it exists in φ as well.

It follows that $\mathsf{ecc}_G(s_i, \varphi') < \infty$, and hence $\varphi' \in \Phi^{\mathbb{N}}_{s_i}$. In addition, $s_1, \ldots, s_{i-1}$ fail cleanly in the first round, so $\varphi' \in \Phi_{i-1}$. Hence, $\varphi' \in \Phi^{\mathbb{N}}_{s_i} \cap \Phi_{i-1}$, and $\mathsf{ecc}_G(s_i, \varphi') \leq \mathsf{ecc}_G(s_i, \Phi^{\mathbb{N}}_{s_i} \cap \Phi_{i-1})$. Thus, $\mathsf{ecc}_G(v, \varphi) = \mathsf{ecc}_G(s_i, \varphi') - 1 < \mathsf{ecc}_G(s_i, \varphi') \leq \mathsf{ecc}_G(s_i, \Phi^{\mathbb{N}}_{s_i} \cap \Phi_{i-1}))$. As this holds for any $\varphi \in \Phi^{\mathbb{N}}_v \cap \Phi_i$, the claim is proved. $\square$

Note that, as for Lemma 1, Lemma 3 may not hold for every family $\Phi \neq \Phi_{\text{all}}^{(t)}$ of failure patterns. Indeed, the failure pattern φ' constructed from φ in the proof of Lemma 3 needs to belong to Φ.

Theorem 2. *Algorithm* $\mathsf{P}_{\mathsf{adapt}}^{G,t}$ *solves consensus in* $\mathsf{radius}(G, \Phi_{\text{all}}^{(t)})$ *rounds.*

The correctness proof of $\mathsf{P}_{\mathsf{adapt}}^{G,t}$ is very similar to that of $\mathsf{P}_{\mathsf{ecc}}^{G,t}$:

Proof. Let $\varphi \in \Phi_{\text{all}}^{(t)}$, and consider an execution of Algorithm $\mathsf{P}_{\mathsf{adapt}}^{G,t}$ with failure pattern φ for R rounds. Let j be the smallest index of a node s_j in the core set such that some correct node v hears from s_j. Such an index j must exist since at least one node in the core set is correct in φ, and it hears from itself. We thus have $\varphi \in \Phi_{s_j}^{\mathbb{N}} \cap \Phi_{j-1}$, which implies $\mathsf{ecc}_G(s_j, \varphi) \leq \mathsf{ecc}(s_j, \Phi_{s_j}^{\mathbb{N}} \cap \Phi_{j-1})$. It then follows from Lemma 3 that $\mathsf{ecc}_G(s_j, \varphi) \leq \mathsf{radius}(G, \Phi_{\text{all}}^{(t)}) = R$. From Fact 1, we deduce that all correct nodes have received the input of s_j in the first R rounds, and the choice of j assures that this is the smallest-indexed node in the core set that any of the correct nodes has received, which completes the proof. $\square$

Finally, observe that $\mathsf{P}_{\mathsf{ecc}}^{\mathsf{G,t}}$ performs in $\mathsf{ecc}_G(v_{t+1}, \Phi_{\text{all}}^{(t)})$ rounds according to the notations of Eq (1), while $\mathsf{P}_{\mathsf{adapt}}^{G,t}$ performs in $\mathsf{radius}(G, \Phi_{\text{all}}^{(t)}) = \mathsf{ecc}_G(v_1, \Phi_{\text{all}}^{(t)})$ rounds according to the same notations.

3.3. Implementing the algorithms with small messages

Our algorithms $\mathsf{P}_{\mathsf{ecc}}^{\mathsf{G,t}}$ and $\mathsf{P}_{\mathsf{adapt}}^{G,t}$ are full information, and hence in every round each node sends all inputs it knows, for a total of $O(n(\log n + \log |U|))$ bits per message, where U is the input space. The algorithms however can be implemented using small messages of only $O(\log n + \log |U|)$ bits. Indeed, in both algorithms, there is a node set S of size $t + 1$ such that, in round $\mathsf{R}(G, t)$, each node decides the input of the smallest node in S it is aware of. Therefore, it is enough that, in every round, each node sends only the pair (v, in_v) with the smallest node $v \in S$ it is aware of. Specifically, if $|U|$ is at most polynomial in n, this gives a simple consensus algorithm exchanging messages on $O(\log n)$ bits.

4. The Lower Bound

In this section we present the notion of information flow graph (Section 4.1), and a solvability characterization for consensus based on this notion (Section 4.2). We then show that $\mathsf{P}_{\mathsf{adapt}}^{G,t}$ is time optimal for vertex-transitive graphs (Section 4.3), among oblivious algorithms. Recall that in an oblivious algorithm, the decision value of a node is based only on the set of input values it has seen so far. Algorithms $\mathsf{P}_{\mathsf{ecc}}^{\mathsf{G,t}}$ and $\mathsf{P}_{\mathsf{adapt}}^{G,t}$ are oblivious. We stress that the notion of information flow graph, the results we prove about it, and our consensus solvability characterization, apply for any graph, not only for vertex-transitive graphs.

4.1. Information flow graph

Recall that the view of a node u in a given round r is the set of all pairs (v, in_v) such that u hears from v by round r. The nodes of the *information flow graph* have the form (v, view_v), meaning that node v has view view_v in round r, and there is a *directed* edge from (v, view_v) to (u, view_u) if and only if $(v, in_v) \in \mathsf{view}_u$, i.e., u hears from v by round r. Of course, these properties are conditioned by the actual failure pattern.

Consider a set of failure patterns $\Phi \subseteq \Phi^{(t)}_{\text{all}}$. Let u be a node that is active in round r in φ, for some $r \geq 1$. Let $\mathsf{view}_G(u, \varphi, r)$ denote the view of u in round r in φ.

Definition 4. *The* information flow *graph in round r with respect to Φ is the directed graph $\mathbb{IF}_{G,\Phi,r}$:*

- $V(\mathbb{IF}_{G,\Phi,r}) = \{(u, \mathsf{view}_G(u, \varphi, r)) : u \in V$ *is active in round r in* $\varphi \in \Phi\}$*;*
- $E(\mathbb{IF}_{G,\Phi,r}) = \big\{\big((u, \mathsf{view}_G(u, \varphi, r)), (v, \mathsf{view}_G(v, \varphi, r))\big) : u \in \mathsf{view}_G(v, \varphi, r)\big\}$.

Note that a node u may have the same view in two distinct failure patterns $\varphi, \psi \in \Phi$ in round r, i.e., $\mathsf{view}_G(u, \varphi, r) = \mathsf{view}_G(u, \psi, r)$, in which case $(u, \mathsf{view}_G(u, \varphi, r))$ and $(u, \mathsf{view}_G(u, \psi, r))$ correspond to the same node of $\mathbb{IF}_{G,\Phi,r}$. Moreover, we have $(u, \mathsf{view}_G(u, \varphi, r)) \neq (v, \mathsf{view}_G(v, \varphi, r))$ for any two distinct nodes u, v, even if $\mathsf{view}_G(u, \varphi, r) = \mathsf{view}_G(v, \varphi, r)$.

The set $\mathsf{config}_G(\varphi, r) = \{(v, \mathsf{view}_G(v, \varphi, r)) : v \in V$ is active in round r in $\varphi\}$ is called the r-round *configuration* for failure pattern φ. See Figure 2 for the information flow graph of the triangle K_3, with one failure, and one communication round.

Lemma 4. *For every failure pattern $\varphi \in \Phi$, and every $r \geq 1$, the set $\mathsf{config}_G(\varphi, r)$ induces a connected subgraph of $\mathbb{IF}_{G,\Phi,r}$.*

Proof. Let u and v be two nodes that are active in round r in φ. Since G is $t+1$-connected, there is a path $w_0 = u, w_1, \ldots, w_k = v$ between u and v in G where all nodes w_i, $i = 0, \ldots, k$, are correct. Since $r > 0$, we have $w_i \in \mathsf{view}_G(w_{i+1}, \varphi, r)$, and thus there is an edge from $(w_i, \mathsf{view}_G(w_i, \varphi, r))$ to $(w_{i+1}, \mathsf{view}_G(w_{i+1}, \varphi, r))$ in $\mathbb{IF}_{G,\Phi,r}$, for every $i = 0, \ldots, k-1$. Therefore, there is a path from $(u, \mathsf{view}_G(u, \varphi, r))$ to $(v, \mathsf{view}_G(v, \varphi, r))$ in the subgraph of $\mathbb{IF}_{G,\Phi,r}$ induced by $\mathsf{config}_G(\varphi, r)$. $\square$

Note that there is an edge from $(u, \mathsf{view}_G(u, \varphi, r))$ to $(v, \mathsf{view}_G(v, \psi, r))$ in $\mathbb{IF}_{G,\Phi,r}$ if and only if there exists $\varrho \in \Phi$ such that u and v are active in round r in ϱ, and $\mathsf{view}_G(u, \varphi, r) = \mathsf{view}_G(u, \varrho, r)$, $\mathsf{view}_G(v, \psi, r) = \mathsf{view}_G(v, \varrho, r)$ and $u \in \mathsf{view}_G(v, \varrho, r)$. Furthermore, if there are two failure patterns φ and ψ yielding the same view for a node v but two different views for a node u, then either the edges from the two views of u to the view of v both exist, or neither exists. This is specified in the following lemma.

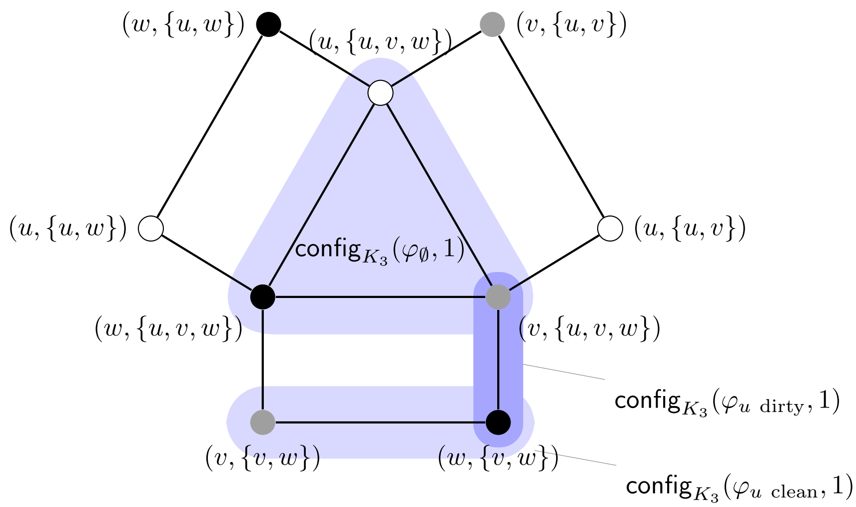

Figure 2: $\mathbb{IF}_{K_3, \Phi^{(1)}_{\text{all}}, 1}$, with the $\mathsf{config}_{K_3}(\varphi, 1)$ sets marked, for some $\varphi \in \Phi^{(1)}_{\text{all}}$; $\varphi_\emptyset$ denotes the failure pattern without failures, $\varphi_{u \text{ clean}}$ the failure patter where u fails cleanly in round 1 and $\varphi_{u \text{ dirty}}$ the failure patter where u fails in round 1 and sends a message only to v.

Lemma 5. *Let $\varphi, \psi \in \Phi$ and $u, v \in V$ such that u and v are active in round r in both φ and ψ. If $\big((u, \mathsf{view}_G(u, \varphi, r)), (v, \mathsf{view}_G(v, \varphi, r))\big) \in E(\mathbb{IF}_{G,\Phi,r})$ and $\mathsf{view}_G(v, \varphi, r) = \mathsf{view}_G(v, \psi, r)$, then $\big((u, \mathsf{view}_G(u, \psi, r)), (v, \mathsf{view}_G(v, \psi, r))\big) \in E(\mathbb{IF}_{G,\Phi,r})$.*

Proof. If $\big((u, \mathsf{view}_G(u, \varphi, r)), (v, \mathsf{view}_G(v, \varphi, r))\big) \in E(\mathbb{IF}_{G,\Phi,r})$, then it must be that $u \in \mathsf{view}_G(v, \varphi, r)$, from which it follows that $u \in \mathsf{view}_G(v, \psi, r)$, and thus $\big((u, \mathsf{view}_G(u, \psi, r)), (v, \mathsf{view}_G(v, \psi, r))\big) \in E(\mathbb{IF}_{G,\Phi,r})$. □

4.2. The solvability characterization

The next result provides a solvability characterization for consensus by oblivious algorithms. In essence, it states that the number r of rounds should be large enough so that every connected component of $\mathbb{IF}_{G,\Phi,r}$ has a *dominating* node. A connected component of $\mathbb{IF}_{G,\Phi,r}$ is a connected component of the underlying, undirected graph of $\mathbb{IF}_{G,\Phi,r}$. We say that a node $v \in V$ of the graph G *dominates* a connected component C of $\mathbb{IF}_{G,\Phi,r}$, if the set $\{(v, \mathsf{view}_G(v, \varphi, r)) : \varphi \in \Phi\}$ dominates C. That is, for every $(w, \mathsf{view}_G(w, \varphi, r))$ in C, there is an arc from the node $(v, \mathsf{view}_G(v, \varphi, r))$ to $(w, \mathsf{view}_G(w, \varphi, r))$.

Theorem 3. *There is an oblivious algorithm solving consensus in r rounds under the set of failure patterns $\Phi \subseteq \Phi^{(t)}_{\text{all}}$ if and only if every connected component C of $\mathbb{IF}_{G,\Phi,r}$ has a dominating node in V.*

The two directions of the theorem are proved by the next two lemmas.

Lemma 6. *For any $\Phi \subseteq \Phi_{\text{all}}^{(t)}$, if every connected component C of $\mathbb{IF}_{G,\Phi,r}$ has a dominating node in V, then there is an oblivious algorithm solving consensus in r rounds under the set of failure patterns Φ.*

Proof. To solve consensus we only need to specify the decision function after r rounds of communication. For every connected component C of $\mathbb{IF}_{G,\Phi,r}$, pick a dominating node $v \in V$ of C. Let w be a node. The view view_w of w determines to which connected component C the node (w, view_w) belongs. The decision of w is the input value of the node v that dominates C.

Clearly, the algorithm satisfies termination and validity. For agreement, consider any $\varphi \in \Phi$. Let w and w' be two nodes that are active in round r in φ. By Lemma 4, the subgraph of $\mathbb{IF}_{G,\Phi,r}$ induced by $\mathsf{config}_G(\varphi, r)$ is connected. Therefore, $(w, \mathsf{view}(w, \varphi, r))$ and $(w', \mathsf{view}(w', \varphi, r))$ belongs to the same connected component C of $\mathbb{IF}_{G,\Phi,r}$, thus w and w' decide the input of the same node. □

Lemma 7. *For any $\Phi \subseteq \Phi_{\text{all}}^{(t)}$, if there is an oblivious algorithm solving consensus in r rounds under the set of failure patterns Φ, then every connected component C of $\mathbb{IF}_{G,\Phi,r}$ has a dominating node in V.*

Proof. For establishing the lemma, we prove the contrapositive. Let $\Phi \subseteq \Phi_{\text{all}}^{(t)}$, and let C be a connected component of $\mathbb{IF}_{G,\Phi,r}$. Assume that, for every $u \in V$, node u does not dominate C. We show that binary consensus in r rounds is impossible. For this purpose, we use a connectivity argument, by proving the existence of a path in the graph of configurations, between the configuration in which all nodes have input 0, and the configuration in which all nodes have input 1.

Let $u_1, \dots, u_n$ be an arbitrary ordering of all the nodes of V. Let $(v_1, \dots, v_n)$ be a sequence of nodes in V, and $(\varphi_1, \dots, \varphi_n)$ be a sequence of failure patterns in Φ, such that $u_i \notin \mathsf{view}_G(v_i, \varphi_i, r)$ and $\mathsf{view}_G(v_i, \varphi_i, r) \in C$ for all $1 \leq i \leq n$. Specifically, v_i is active in round r in φ_i. $(v_i, \varphi_i)_{1 \leq i \leq n}$ exists since no node dominates C. Note that it may be the case that $v_i = v_j$ for $i \neq j$.

Let X_i be the vector composed of $n - i$ 0-entries, follow by i 1-entries, i.e., $X_i(j) = 0$ for $1 \leq j \leq n - i$, and $X_i(j) = 1$ for $n - i < j \leq n$. Specifically, $X_0 = 0^n$ is the all-0 vector, and $X_n = 1^n$ the all-1 vector. For every $0 \leq i \leq n$, let us consider the executions of an alleged r-round algorithm when the inputs of $u_1, \dots, u_n$ are given by X_i, i.e., the input of u_j is $X_i(j)$ for $j = 1, \dots, n$. Let $1 \leq j \leq n$ be the minimum index such that, if the inputs are given by X_j and the failure pattern is φ_j, then v_j decides on 1. Note that such a value must exist, since on X_n, node v_n must decide 1.

Assume first that $j = 1$, i.e., on inputs X_1 and failure pattern φ_1, v_1 decides on 1. Consider the execution of the algorithm with the same failure pattern φ_1, but with inputs X_0. This execution differs from the previous one only by the input of u_1, which is not seen by v_1 as $u_1 \notin \mathsf{view}_G(v_1, \varphi_1, r)$. Hence, v_1 must decide on 1 in this case as well. On the other hand, on the input vector $X_0 = 0^n$, all nodes must decide 0, a contradiction.

Consider now the case of $1 < j \leq n$. In the connected component C, there is a path P connecting $(v_{j-1}, \mathsf{view}_G(v_{j-1}, \varphi_{j-1}, r))$ and $(v_j, \mathsf{view}_G(v_j, \varphi_j, r))$. Let us describe this path P as

$$\begin{aligned}(v_{j-1}, \mathsf{view}_G(v_{j-1}, \varphi_{j-1}, r)) = (w_0, \mathsf{view}_G(w_0, \psi_0, r)),\\ (w_1, \mathsf{view}_G(w_1, \psi_1, r)), \dots, (w_{k-1}, \mathsf{view}_G(w_{k-1}, \psi_{k-1}, r)),\\ (w_k, \mathsf{view}_G(w_k, \psi_k, r)) = (v_j, \mathsf{view}_G(v_j, \varphi_j, r))\end{aligned}$$

By the minimality of j, we know that, on the input vector X_{j-1}, and with failure pattern φ_{j-1}, node v_{j-1} decides on 0. Put differently, on the input vector X_{j-1} and with failure pattern ψ_0, node w_0 decides on 0. Consider now two consecutive nodes in the path P, say $(w_i, \mathsf{view}_G(w_i, \psi_i, r))$ and $(w_{i+1}, \mathsf{view}_G(w_{i+1}, \psi_{i+1}, r))$. As commented earlier, there exists a failure $\varrho \in \Phi$ such that

$$\mathsf{view}_G(w_i, \psi_i, r) = \mathsf{view}_G(w_i, \varrho, r) \text{ and } \mathsf{view}_G(w_{i+1}, \psi_{i+1}, r) = \mathsf{view}_G(w_{i+1}, \varrho, r).$$

So, when running on input vector X_{j-1} (or any other input vector), and with failure pattern ϱ, w_i and w_{1i+1} decide the same. A simple induction on the distance to node $(w_0, \mathsf{view}_G(w_0, \psi_0, r))$ in the path P implies that on X_{j-1}, with failure pattern φ_j, v_j decides on 0. We are now in a case similar to that of $j = 1$: On the input vector X_{j-1}, with failure pattern φ_j, node v_j decides on 0. Instead, on the input vector X_j, with the same failure pattern φ_j, node v_j decides on 1. The only difference between X_{j-1} and X_j is in the input of u_j, which is not seen by v_j as $u_j \notin \mathsf{view}_G(v_j, \varphi_j, r)$. So v_j must decide the same in both cases, a contradiction. □

4.3. Optimality of $\mathsf{P}^{G,t}_{\mathsf{adapt}}$ *for symmetric graphs*

To conclude, we use the characterization in Theorem 3 to show that $\mathsf{P}^{G,t}_{\mathsf{adapt}}$ is time optimal for vertex-transitive graphs, among oblivious algorithms.

An *automorphism* of G is a bijection $\pi : V \to V$ such that, for every two nodes u and v, $\{u, v\} \in E \iff \{\pi(u), \pi(v)\} \in E$. A graph $G = (V, E)$ is *vertex-transitive* if, for every two nodes u and v, there exists an automorphism π of G such that $\pi(u) = v$. For instance, the complete graphs K_n, the cycles C_n, the d-dimensional hypercubes Q_d, the d-dimensional toruses $C_{n_1} \times \cdots \times C_{n_d}$, the Kneser graphs $KG_{n,k}$, and Cayley graphs, are all vertex-transitive. The wheel, composed of a cycle and a center node connected to all cycle nodes, is not vertex-transitive, since the center node has degree $n - 1$ while the cycle nodes have degree 3.

Theorem 4. *If G is vertex-transitive, then there is no oblivious algorithm that solves consensus in fewer than* $\mathsf{radius}(G, \Phi^{(t)}_{\mathrm{all}})$ *rounds.*

Proof. Clearly, the result holds if $\mathsf{radius}(G, \Phi^{(t)}_{\mathrm{all}}) = 0$ (a single-node graph), and if $\mathsf{radius}(G, \Phi^{(t)}_{\mathrm{all}}) = 1$, as consensus is trivially not solvable in zero rounds in any graph with at least 2 nodes, even with no failures. So we assume now that $\mathsf{radius}(G, \Phi^{(t)}_{\mathrm{all}}) \geq 2$.

We will show a result stronger that the one stated in the theorem, namely we show that no oblivious algorithm can solve consensus in a vertex-transitive graph G in a *restricted* set of failure patterns $\Phi \subsetneq \Phi^{(t)}_{\text{all}}$ (or $\Phi' \subsetneq \Phi^{(t)}_{\text{all}}$ in the case of the complete graph K_n with $n-1$ failures). That is, even if the algorithm has only to deal with the $n+1$ failure patterns in $\Phi \subsetneq \Phi^{(t)}_{\text{all}}$ (or Φ' if $G = K_n$ with $t = n-1$ failures), still consensus is not solvable in fewer than $\mathsf{radius}(G, \Phi^{(t)}_{\text{all}})$ rounds.

Both sets of failure patterns Φ and Φ' consist of the empty failure pattern $\varphi_\emptyset$ and one failure pattern φ_s for each node s chosen from a larger set $\tilde{\Phi}$. A failure pattern φ belongs to $\tilde{\Phi}$ if it consists of the union of two, possibly empty, sets of failures: a hidden path φ_h and a set of clean failures φ_c occurring at round 2. A *hidden path* starting at some node s is a failure pattern

$$\varphi_h = \{(v_i, F_{v_i}, i), i = 1, \dots, k\}$$

where $1 \le k \le t$, $v_1 = s$, and $F_{v_i} = N(v_i) \smallsetminus \{v_{i+1}\}$ for every $1 \le i \le k$ with v_{k+1} a correct node. Hence, for any failure pattern $\varphi \in \Phi^{(t)}_{\text{all}}$:

$$\varphi \in \tilde{\Phi} \iff \varphi = \varphi_h \cup \varphi_c$$

where φ_h is either empty or an hidden path starting at some node v and φ_c is empty or has the form:

$$\varphi_c = \{(u_1, N(u_1), 2), \dots, (u_\ell, N(u_\ell), 2)\}$$

for some nodes $u_1, \dots, u_\ell$ and $\ell \le t$.

Remark.. In a vertex-transitive graph G, for every $s \in V$, $\mathsf{radius}(G, \Phi^{(t)}_{\text{all}}) = \mathsf{ecc}_G(s, \Phi^{(t)}_{\text{all}})$. In fact, this is the only property of vertex-transitive graphs we use, and the only way we use vertex-transitivity. Hence our theorem holds for any graph satisfying the above property.

We will now show that $\mathsf{ecc}_G(s, \Phi^{(t)}_{\text{all}}) = \mathsf{ecc}_G(s, \tilde{\Phi})$ and therefore for every $s \in V$, we can assign a failure pattern $\varphi_s \in \tilde{\Phi}$ such that $\mathsf{radius}(G, \Phi^{(t)}_{\text{all}}) = \mathsf{ecc}_G(s, \varphi_s)$. In what follows, let $R = \mathsf{radius}(G, \Phi^{(t)}_{\text{all}})$.

Lemma 8. *For every node s, there exists a failure pattern $\varphi_s \in \tilde{\Phi}$ such that $\mathsf{ecc}_G(s, \varphi_s) = \mathsf{radius}(G, \Phi^{(t)}_{\text{all}}) = R$. Moreover, if φ_s contains an hidden path, it starts in s.*

Proof. Let $\varphi \in \Phi^{(t)}_{\text{all}}$ be a failure pattern such that $\mathsf{ecc}_G(s, \varphi) = R$. As observed above, such a failure pattern exists because G is vertex-transitive. Based on φ, we define another failure pattern $\tilde{\varphi}$. We show that $\tilde{\varphi} \in \tilde{\Phi}$, the hidden path (if any) in $\tilde{\varphi}$ starts in s and $\mathsf{ecc}_G(s, \tilde{\varphi}) = R$, which proves the lemma.

Since $\mathsf{ecc}_G(s, \varphi) = R$, there is a correct node x such that every causal path in φ from s to x has length at least R. Let $u_1 = s \to \ldots \to u_R \to u_{R+1} = x$ be such a path, of length exactly R (such a path must exist as otherwise $\mathsf{ecc}_G(s,\varphi)$ would have been larger). In addition, let

$$\ell = \begin{cases} 0 & s \text{ is correct in } \varphi \\ \max\{i : 1 \leq i \leq R, u_1,\ldots,u_i \text{ fail in } \varphi\} & \text{otherwise.} \end{cases}$$

Note that for each $j, 1 \leq j \leq \ell$, u_j fails at round j or at a later round. For each $(v, F_v, r_v) \in \varphi$, let

$$\tilde{F}_v = \begin{cases} N(v) \setminus \{u_{i+1}\} & \text{if } \exists i \leq \ell : v = u_i \\ N(v) & \text{otherwise,} \end{cases}$$

and

$$\tilde{r}_v = \begin{cases} i & \text{if } \exists i \leq \ell : v = u_i \\ 2 & \text{otherwise.} \end{cases}$$

Finally, we set $\tilde{\varphi} = \{(v, \tilde{F}_v, \tilde{r}_v) : \exists F, r, (v, F, r) \in \varphi\}$. That is, the set of nodes that fail in φ and $\tilde{\varphi}$ is the same, and there is a hidden path from s to u_ℓ in $\tilde{\varphi}$. Each node that fails in $\tilde{\varphi}$ and that is not in the hidden path fails cleanly in round 2. Therefore, $\tilde{\varphi} \in \tilde{\Phi}$. As there is a correct node (namely, u_ℓ) that hears from s, every correct node hears from s in $\tilde{\varphi}$. Hence, $\mathsf{ecc}_G(s,\tilde{\varphi}) \leq \mathsf{radius}(G, \Phi^{(t)}_{\text{all}}) = R$.

Consider a causal path $s = \tilde{u}_1 \to \tilde{u}_2 \to \ldots \to \tilde{u}_m = x$ from s to x in $\tilde{\varphi}$. Nodes $\tilde{u}_1,\ldots,\tilde{u}_\ell$ coincide with nodes $u_1,\ldots,u_\ell$ as for each $i, 1 \leq i \leq \ell-1$, node u_i fails in round i and sends only to node u_{i+1}. Since every faulty nodes not in the hidden path fails cleanly in round 2 in $\tilde{\varphi}$, nodes $\tilde{u}_\ell,\ldots,\tilde{u}_m$ are correct in $\tilde{\varphi}$, and thus also in φ. Therefore, $\tilde{u}_1 \to \ldots \to \tilde{u}_m$ is also a causal path from s to x in φ, from which we conclude that its length is at least $\mathsf{ecc}_G(s,\varphi) = R$. As this holds for any causal path from s to x in $\tilde{\varphi}$, $\mathsf{ecc}_G(s,\tilde{\varphi}) = R$. □

Next, we show that even if that algorithm has to deal with a restricted set of failure patterns consisting in only $n+1$ failure patterns, consensus is not solvable in fewer than $\mathsf{radius}(G, \Phi^{(t)}_{\text{all}})$ rounds. In the general case, this set of failure patterns is called Φ. It follows from Lemma 8 that for every $s \in V$, we can assign a failure pattern $\varphi_s \in \tilde{\Phi}$ such that $\mathsf{radius}(G, \Phi^{(t)}_{\text{all}}) = \mathsf{ecc}_G(s,\varphi_s)$ and whose hidden path (if any) starts at s. Let $\Phi = \{\varphi_s : s \in V\} \cup \{\varphi_\varnothing\}$. These configurations $\mathsf{config}_G(\varphi, t)$ for $\varphi \in \Phi$ are depicted in Figure 3 for the case of $G = K_3$ and $t = 1$.

The case of the complete graph K_n with $t = n-1$ needs special care. In this case, we use a slightly different set of failure pattern denoted Φ' to show that consensus is not solvable in fewer than $\mathsf{radius}(K_n, \Phi^{(n-1)}_{\text{all}}) = n-1 = t$ rounds. Given a node s, let φ'_s be an hidden path of length $n-2$. That is, $\varphi'_s = \{(v_1, N(v) \setminus \{v_2\}, 1), \ldots, (v_{n-2}, N(v) \setminus \{v_{n-1}\}, n-2)\}$ where $v_1 = s$, and v_{n-1} is a correct node. Note that $\mathsf{ecc}_{K_n}(s, \varphi') = n-1$. Indeed, there are two correct nodes in φ' and only one of them, namely v_{n-1} has heard of s at the beginning of round $n-1$. As for any n-nodes graph G, $\mathsf{radius}(G, \Phi^{(t)}_{\text{all}}) \leq n-1$, $\mathsf{ecc}_{K_n}(s, \varphi') = n-1 = \mathsf{radius}(K_n, \Phi^{(n-1)}_{\text{all}})$. We set $\Phi' = \{\varphi'_s : s \in V\} \cup \{\varphi_\varnothing\}$.

Using Theorem 3, it is then sufficient to prove the following lemmas:

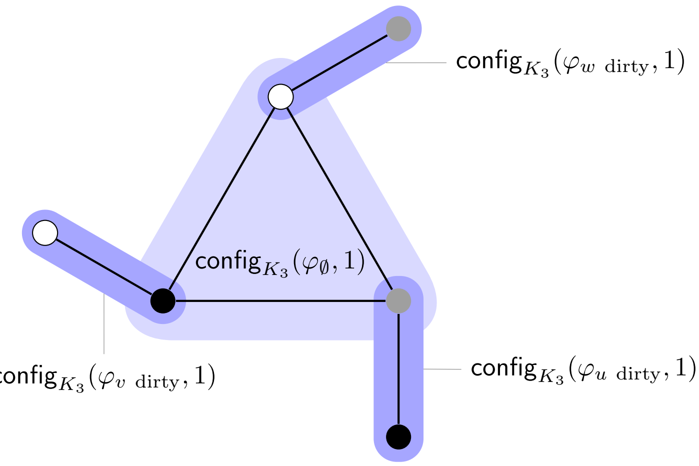

Figure 3: The information flow graph $\mathbb{IF}_{K_3,\Phi,1}$ appearing in the proof of Theorem 4, for K_3 and the failure pattern Φ defined there. $\varphi_{\emptyset}$ denotes the failure pattern without failures, while $\varphi_{x\ \mathrm{dirty}}$ denotes the failure pattern where x fails in round 1, sending a message to only one node.

Lemma 9. *If G is not a complete graph, or $G = K_n$ and $t < n-1$, the information flow graph $\mathbb{IF}_{G,\Phi,R-1}$ is connected and has no dominating node.*

Lemma 10. *If $G = K_n$ and $t = n-1$, the information flow graph $\mathbb{IF}_{K_n,\Phi',n-2}$ is connected and has no dominating node.*

We start with the proof of Lemma 9.

Proof of Lemma 9. We first note that

$$V(\mathbb{IF}_{G,\Phi,R-1}) = \mathsf{config}_G(\varphi_\varnothing, R-1) \cup \Big(\bigcup_{s\in V} \mathsf{config}_G(\varphi_s, R-1)\Big).$$

Now, we prove the following three claims, which together show the connectivity of the underlying graph of $\mathbb{IF}_{G,\Phi,R-1}$:

1. The subgraph of $\mathbb{IF}_{G,\Phi,R-1}$ induced by $\mathsf{config}_G(\varphi_\varnothing, R-1)$ is connected;
2. for every $s \in V$, the subgraph of $\mathbb{IF}_{G,\Phi,R-1}$ induced by $\mathsf{config}_G(\varphi_s, R-1)$ is connected;
3. and finally, $\mathsf{config}_G(\varphi_s, R-1) \cap \mathsf{config}_G(\varphi_\varnothing, R-1) \neq \varnothing$.

The facts that the subgraphs of $\mathbb{IF}_{G,\Phi,R-1}$ induced by $\mathsf{config}_G(\varphi_\varnothing, R-1)$ and $\mathsf{config}_G(\varphi_s, R-1)$ are connected follow directly from Lemma 4.

To show that $\mathsf{config}_G(\varphi_s, R-1) \cap \mathsf{config}_G(\varphi_\varnothing, R-1) \neq \varnothing$ for every node $s \in V$, we show that for every such s there is a node v_s such that $\mathsf{view}_G(v_s, \varphi_s, R-1) = \mathsf{view}_G(v_s, \varphi_\varnothing, R-1)$. To this end, we analyze the possible structures of the failure pattern φ_s. Recall that φ_s is composed of a (possibly empty) hidden path $s = v_1, \ldots, v_{k+1}$ that starts in s and a set of nodes $\{v'_1, \ldots, v'_\ell\}$ that crash cleanly in round 2. We consider two cases, according to the length k of the hidden path in φ_s:

- $k = 0$. If no node crashes cleanly in round 2, $\varphi_s = \varphi_\varnothing$ and every node has the same view at the end of round $R - 1$ in both failure patterns.

  Let us assume that at least one node crashes cleanly in round 2. Let u be a correct neighbor of v'_1, which must exist since $\deg_G(v'_1) \geq \kappa(G) > t$. Assume for contradiction that there is a node u' from which u hears in the first $R - 1$ rounds when there are no failures, but from which it does not hear in φ_s. That is:

  $$u' \in \mathsf{view}_G(u, \varphi_\varnothing, R - 1) \text{ and } u' \notin \mathsf{view}_G(u, \varphi_s, R - 1).$$

  Define a failure pattern φ'_s identical to φ_s, except that the node u' does not crash in φ'_s, and the clean failure of v'_1 is replaced by $(v'_1, N(v_1) \smallsetminus \{u\}, 1)$. In other words, φ'_s is the same as φ_s except that (1) u' is removed from φ_s if it happened that $u' = v'_i$ for some $i \in \{2, \ldots, \ell\}$, and (2) the clean crash of v'_1 at round 2 in φ_s is replaced by a crash in which v'_1 sends to u at round 1. As $\varphi'_s \in \Phi^{(t)}_{\text{all}}$, and $\mathsf{radius}(G, \Phi^{(t)}_{\text{all}}) = R$, there must exist a causal path from v'_1 to u' under φ'_s, which is composed of a message from v'_1 to u, followed by a path P from u to u' of length at most $R - 1$. By the fact that G is undirected, and by the construction of φ'_s, the same path P in the opposite direction is a causal path from u' to u under φ'_s, and also under φ_s. Hence $u' \in \mathsf{view}_G(u, \varphi_s, R - 1)$: a contradiction. As $\mathsf{view}_G(u, \varphi_s, R - 1) \subseteq \mathsf{view}_G(u, \varphi_\emptyset, R-1)$, it follows that $\mathsf{view}_G(u, \varphi_s, R-1) = \mathsf{view}_G(u, \varphi_\emptyset, R-1)$.

- $k = 1$. The analysis of this case is similar to the previous case. Consider node v_2, the neighbor of s that receives a message from s in the first round. As v_2 is the end of a hidden path, it is correct. Assume for contradiction that $\mathsf{view}_G(v_2, \varphi_s, R - 1) \neq \mathsf{view}_G(v_2, \varphi_\emptyset, R - 1)$. Hence, there exists a node $u' : u' \in \mathsf{view}_G(v_2, \varphi_\emptyset, R-1)$ and $u' \notin \mathsf{view}_G(v_2, \varphi_s, R-1)$. Similarly to the previous case, let φ'_s be the failure pattern identical to φ_s, except that u' is correct in φ'_s (φ_s and φ'_s are thus the same if u' is correct in φ_s.). As $\varphi'_s \in \Phi^{(t)}_{\text{all}}$, and $\mathsf{radius}(G, \Phi^{(t)}_{\text{all}}) = R$, there must exist a causal path from $v_1 (= s)$ to u' under φ'_s, which is composed of a message from v_1 to v_2, followed by a path P from v_2 to u' of length at most $R - 1$. As in the previous case, the same path P in the opposite direction is a causal path from u' to v_2 under φ'_s, and also under φ_s. Therefore, $u' \in \mathsf{view}_G(v_2, \varphi_s, R - 1)$: a contradiction.

- $k \geq 2$. In this case, our goal is to show that $\mathsf{view}_G(v_{k+1}, \varphi_s, R - 1) = \mathsf{view}_G(v_{k+1}, \varphi_\emptyset, R - 1)$, where v_{k+1} is the last node of the hidden path starting in s.

  By the end of round k, v_{k+1} has heard from every node $v_1, \ldots, v_k$ in the hidden path. Let $u \neq v_{k+1}$ be a correct node. As $\mathsf{ecc}_G(s, \varphi_s) = R$, u hears from s at the latest at round R. Since v_{k+1} is the only active node that has heard from s at the end of round k, a shortest causal path from s to u consists of the hidden path $v_1(= s), \dots, v_{k+1}$ followed by a causal path P from v_{k+1} to u of length at most $R-k$. Since every faulty node outside the hidden path crashes cleanly in round 2, the path P contains only correct nodes. Hence, the path P in the opposite direction is also a causal path in φ_s, from u to s. Finally, consider a faulty node u' which is not in the hidden path: u' fails cleanly in round 2. As $\deg_G(u') \geq \kappa(G) > t$, u' has a correct neighbor u that hears from it in round 1. As seen above, there is a causal path made of correct nodes and of length at most $R-k$ from u to v_{k+1}. Hence, u hears from u' by the end of round $R-k+1$ at the latest. We conclude that v_{k+1} hears from all the nodes by the end of round $\tau = \max(k, R-k, R-k+1) = \max(k, R-k+1)$ in φ_s. As every causal path under φ_s is also a causal path when there are no failures, $\mathsf{view}_G(v_{k+1}, \varphi_s, \tau) = \mathsf{view}_G(v_{k+1}, \varphi_\emptyset, \tau)$. To conclude the analysis of this case, we consider the following sub-cases depending on the relations between τ and $R-1$:

- $\tau \leq R-1$. As the view of v_{k+1} consists of all the nodes at the end of round τ in both φ_s and $\varphi_\varnothing$, we have $\mathsf{view}_G(v_{k+1}, \varphi_s, R-1) = \mathsf{view}_G(v_{k+1}, \varphi_\emptyset, R-1)$, as desired.
- $\tau > R-1$. We have $\tau = k = R$ since $k \geq 2$ and the length k of the hidden path is at most R. Note that t nodes fail in φ_s. Otherwise, as $\deg_G(v_{k+1}) \geq t$, v_{k+1} has a correct neighbor u. The hidden path can thus be extended by failing v_{k+1} in round $k+1$ with one message sent from v_{k+1} to u in that round. In the resulting failure pattern φ'_s, $\mathsf{ecc}_G(s, \varphi'_s) \geq R+1 > \mathsf{radius}(G, \Phi^{(t)}_{\mathrm{all}}) = R$, which is a contradiction.
  Let us also observe now that the number n of nodes satisfies $n = t+1$. Since $\mathsf{ecc}_G(s, \varphi_s) = R$, every correct node has heard from s in φ_s by the end of round $R = k$. Note that v_{k+1} is the only correct node that hears from s by the end of round R, and thus the only correct node. As t nodes fails, the total number of nodes in G is $n = t+1$. Therefore, since for every node v $\deg_G(v) \geq t = n-1$, G is the complete graph K_n and $t = n-1$, which contradicts the assumptions of the lemma.

Now, we show that, for every $s \in V$, s does not even dominate the subgraph induced by $\mathsf{config}_G(\varphi_s, R-1)$. To see this, let us fix $s \in V$. Since $\mathsf{ecc}_G(s, \varphi_s) = \mathsf{radius}(G, \Phi^{(t)}_{\mathrm{all}}) = R$, there exists a correct node u_s that has not heard from s by the end of round $R-1$ in φ_s. That is, $s \notin \mathsf{view}_G(u_s, \varphi_s, R-1)$. It follows that s does not dominate $\mathsf{config}_G(\varphi_s, R-1)$, and therefore it does not dominate $V(\mathbb{IF}_{G,\Phi,R-1})$, as claimed. □

We now consider the case where $G = K_n$ and $t = n-1$

Proof of Lemma 10. As in the proof of Lemma 9, it follows from Lemma 4 that

1. The subgraph of $\mathbb{IF}_{K_n,\Phi',n-2}$ induced by $\mathsf{config}_G(\varphi_\varnothing, n-2)$ is connected;
2. for every $s \in V$, the subgraph of $\mathbb{IF}_{G,\Phi,n-2}$ induced by $\mathsf{config}_G(\varphi'_s, n-2)$ is connected.

It remains to show that for any node s, $\mathsf{config}_{K_n}(\varphi'_s, n-2) \cap \mathsf{config}_{K_n}(\varphi_\varnothing, n-2) \neq \varnothing$. Recall that φ'_s consists in an hidden path of length $n-2$ starting in $v_1 = s$ and ending in some correct node v_{n-1}. Let u denote the node that is not involved in the hidden path, i.e., the node u such that $\{u\} = V \setminus \{v_1, \dots, v_{n-1}\}$.

By the end of round $n-2$, v_{n-1} has heard from every node in the hidden path $v_1, \dots, v_{n-2}$ and from node u in φ'_s. As the graph is complete, v_{n-1} also hears from every node in the failure-free failure pattern $\varphi_\varnothing$. Therefore, $\mathsf{view}_{K_n}(v_{n-1}, \varphi'_s, n-2) = \mathsf{view}_{K_n}(v_{n-1}, \varphi_\varnothing, n-2)$.

The rest of the proof, namely that no node dominates $\mathbb{IF}_{K_n,\Phi',n-2}$, is the same as in the proof of Lemma 9. $\square$

The theorem directly follows from the previous lemmas and the characterization in Theorem 3. $\square$

Theorem 5. *If G is vertex-transitive, $\mathsf{P}^{G,t}_{\mathsf{adapt}}$ is time optimal among oblivious algorithms.*

We conjecture that $\mathsf{P}^{G,t}_{\mathsf{adapt}}$ is, among oblivious algorithms, time optimal for all graphs and for the class $\Phi^{(t)}_{\mathrm{all}}$ of all failure patterns. This conjecture is grounded on the fact that Lemma 3 holds for all graphs, and not only for those that are vertex-transitive. $\mathsf{P}^{G,t}_{\mathsf{adapt}}$ is however not optimal for specific classes Φ of failure patterns, even in vertex-transitive graphs, as we show in the next section.

5. The Case of Clean Failures

An interesting and well studied type of failures are *clean failures*, i.e., failures where the failing nodes do not send any messages. Here, we focus on *initial* clean failures, i.e., crashes occurring before the failing nodes were able to send any messages. We show that in this case, neither the naive algorithm nor our adaptive algorithm $\mathsf{P}^{G,t}_{\mathsf{adapt}}$ are optimal, and we do so on a vertex-transitive graph. This implies that considering $\Phi^{(t)}_{\mathrm{all}}$ in our algorithm (Theorem 2) and in our lower bound (Theorem 4) is required for these claims to hold.

Consider the graph Q_3, i.e., the 3-dimensional hypercube with nodes marked $x_1x_2x_3 \in \{0,1\}^3$, and edges between two nodes of Hamming distance (i.e., number of different coordinates) equal to 1 — see Figure 4. Interestingly, this graph was also used to prove an impossibility result related to routing with edge failures [11]. The diameter of Q_3 is 3, and its connectivity is 3 as well.

Let $t = 2$, and let us consider the set $\Phi^{(2)}_{\text{clean-init}}$ of clean initial failure patterns, with at most 2 failures. Under this family of failure patterns, each node v has eccentricity $\mathsf{ecc}_{Q_3}(v, \Phi^{(2)}_{\text{clean-init}}) = 4$. To see this, consider, for example, the node 000, and the failure pattern where 001 and 010 fail (initially and cleanly). In this

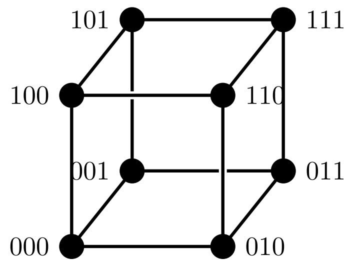

Figure 4: Q_3, the 3-dimensional cube.

case, every path from 000 must start with the edge $(000, 100)$; from 100 to 011, every path take 3 more edges, since this is their Hamming distance, hence the distance between 000 and 011 is 4. Since all nodes have the same eccentricities, the naive algorithm, Algorithm $\mathsf{P}_{\mathsf{ecc}}^{Q_3,4}$, solve consensus in 4 rounds. The radius $\mathsf{radius}(Q_3, \Phi^{(2)}_{\text{clean-init}})$ is also 4, so our algorithm, $\mathsf{P}_{\mathsf{adapt}}^{Q_3,2}$, also takes 4 rounds to reach consensus.

To get a 3-round algorithm under $\Phi^{(2)}_{\text{clean-init}}$, we note that if node 000 is correct, and if it cannot flood in 3 rounds, this is because two nodes of Hamming weight 1 (the nodes $001, 010, 100$) have crashed. Moreover, in this case, the node 111 can flood in 3 rounds. We present the algorithm

$$\mathsf{P}_{\mathsf{clean-init}}^{Q_3,2} = (\mathsf{R}_{\mathsf{clean-init}}(Q_3, 2), \mathsf{D}_{\mathsf{adapt}}(Q_3, 2)),$$

where the flooding time is $\mathsf{R}_{\mathsf{clean-init}}(Q_3, 2) = 3$, and the decision procedure $\mathsf{D}_{\mathsf{adapt}}(Q_3, 2))$ at each node u is as follows.

1. If node u receives at least two nodes of Hamming weight 1, and node u received 000, then return the input of 000;
2. Otherwise, if node u receives 111, return the input of 111;
3. Otherwise, return the input of 001.

Theorem 6. *Algorithm $\mathsf{P}_{\mathsf{clean-init}}^{Q_3,2}$ solves consensus on Q_3 with at most 2 clean initial failures in 3 rounds.*

Proof. The running time of the algorithm is clear from the choice $\mathsf{R}_{\mathsf{clean}}(Q_3, 2) = 3$. The correctness yields from a simple case analysis.

In a failure pattern φ_1 where at most one node of Hamming weight 1 fails, and 000 does not fail, we have $\mathsf{ecc}_{Q_3}(000, \varphi_1) \leq 3$, and all nodes decide on the input of 000, by Instruction 1.

In a failure pattern φ_2 where two nodes of Hamming weight 1 fail, we have that the node 111 does not fail and has $\mathsf{ecc}_{Q_3}(111, \varphi_2) \leq 3$, and all nodes decide on the input of 111 by Instruction 2.

We are left with the case of failure patterns where 000 fails, which leads to two sub-cases. In a failure pattern φ_3 where 000 fails while 111 does not fail, at most one neighbor of 111 fails, 111 has $\mathsf{ecc}_{Q_3}(111, \varphi_3) = 2$ and all nodes decide on the input of 111 by Instruction 2.

Finally, in the failure pattern φ_4 where 000 and 111 fail, node 001 does not fail, has $\mathsf{ecc}_{Q_3}(001, \varphi_4) = 3$, and all nodes decide on the input of 001 by Instruction 3. □

Theorem 6 shows that $\mathsf{P}^{G,t}_{\mathsf{adapt}}$, which was proved optimal for $\Phi^{(t)}_{\text{all}}$ (in vertex-transitive graphs), is not optimal for all families Φ of failure patterns. In particular, $\mathsf{P}^{Q_3,2}_{\mathsf{adapt}}$ is not optimal in Q_3 for $\Phi^{(2)}_{\text{clean-init}}$. We don't know whether $\mathsf{P}^{G,t}_{\mathsf{adapt}}$ is optimal for $\Phi^{(t)}_{\text{clean}}$.

6. Conclusion

We have studied for the first time the number of rounds needed to solve fault-tolerant consensus in a crash prone synchronous network with arbitrary structure. We have defined a notion of *dynamic radius* of a graph G when t nodes may crash, which precisely determines the worst case number of rounds needed to solve oblivious consensus for vertex-transitive networks. The optimality of our algorithm was shown through a novel consensus solvability characterization in arbitrary networks, using the notion of information flow [6]. A second consequence of the characterization is an abstract consensus algorithm that is optimal for all graphs. Our focus has been in the worst-case number of rounds. An interesting challenge would be to design early deciding algorithms; a problem that is well-studied in the case of the complete graph e.g. [8].

An interesting future line of research is to study the case of non-oblivious algorithms (such algorithms have been considered in the past, e.g. [31]). Remarkably, for the case of the complete communication graph, there is no difference between these two types of algorithms: at the end of round $t+1$, every pair of nodes have the same set of pairs (v, in_v) (formally, there is common knowledge on a set of inputs), hence decisions can be taken considering only this set.

Recall that, in our algorithms, $\mathsf{R}(G,t)$ and $\mathsf{D}(G,t)$ are hard-coded for a given G and t. It is worth exploring if our techniques are useful for the case where the graph G is not known to the nodes. Indeed, it is a challenge to combine fault-tolerant arguments with techniques of (failure-free) network computing [29]. Our results for $t = 0$ correspond to network computing. Yet, the case of $t > 0$ for arbitrary or evolving networks is an intriguing and complex research question.

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):155–158, 1999.

[2] Bowen Alpern and Fred B. Schneider. Defining liveness. *Inf. Process. Lett.*, 21(4):181–185, 1985.

[3] Hagit Attiya, Armando Castañeda, Maurice Herlihy, and Ami Paz. Bounds on the step and namespace complexity of renaming. *SIAM J. Comput.*, 48(1):1–32, 2019.

[4] Hagit Attiya and Jenifer Welch. *Distributed computing: fundamentals, simulations, and advanced topics*. Wiley series on parallel and distributed computing. Wiley, 2004.

[5] Armando Castañeda, Pierre Fraigniaud, Ami Paz, Sergio Rajsbaum, Matthieu Roy, and Corentin Travers. Synchronous t-resilient consensus in arbitrary graphs. In *21st International Symposium Stabilization, Safety, and Security of Distributed Systems, SSS*, volume 11914 of *Lecture Notes in Computer Science*, pages 53–68. Springer, 2019.

[6] Armando Castañeda, Pierre Fraigniaud, Ami Paz, Sergio Rajsbaum, Matthieu Roy, and Corentin Travers. A topological perspective on distributed network algorithms. In *26th Int. Colloquium on Structural Information and Communication Complexity, SIROCCO*, volume 11639 of *Lecture Notes in Computer Science*, pages 3–18. Springer, 2019.

[7] Armando Castañeda, Yannai A. Gonczarowski, and Yoram Moses. Unbeatable consensus. In *Distributed Computing - 28th International Symposium, DISC*, pages 91–106, 2014.

[8] Armando Castañeda, Yoram Moses, Michel Raynal, and Matthieu Roy. Early decision and stopping in synchronous consensus: A predicate-based guided tour. In Amr El Abbadi and Benoît Garbinato, editors, *Networked Systems (NETYS), LNCS, vol. 10299*, pages 206–221. Springer, 2017.

[9] Armando Castañeda and Sergio Rajsbaum. New combinatorial topology bounds for renaming: The upper bound. *J. ACM*, 59(1):3:1–3:49, 2012.

[10] Bernadette Charron-Bost and Shlomo Moran. Minmax algorithms for stabilizing consensus. *CoRR*, abs/1906.09073, 2019.

[11] Marco Chiesa, Ilya Nikolaevskiy, Slobodan Mitrovic, Andrei V. Gurtov, Aleksander Madry, Michael Schapira, and Scott Shenker. On the resiliency of static forwarding tables. *IEEE/ACM Trans. Netw.*, 25(2):1133–1146, 2017.

[12] Étienne Coulouma, Emmanuel Godard, and Joseph G. Peters. A characterization of oblivious message adversaries for which consensus is solvable. *Theor. Comput. Sci.*, 584:80–90, 2015.

[13] Danny Dolev. The byzantine generals strike again. *Journal of Algorithms*, 3(1):14–30, 1982.

[14] Danny Dolev and Ray Strong. Authenticated algorithms for byzantine agreement. *SIAM Journal on Computing*, 12(4):656–666, 1983.

[15] Cynthia Dwork and Yoram Moses. Knowledge and common knowledge in a byzantine environment: Crash failures. *Information and Computation*, 88(2):156–186, 1990.

[16] Michael J. Fischer and Nancy A. Lynch. A lower bound for the time to assure interactive consistency. *Information Processing Letters*, 14(4):183 – 186, 1982.

[17] Michael J. Fischer, Nancy A. Lynch, and Michael Merritt. Easy impossibility proofs for distributed consensus problems. *Distributed Computing*, 1(1):26–39, Mar 1986.

[18] Michael J. Fischer, Nancy A. Lynch, and Mike Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2):374–382, 1985.

[19] Chris Godsil and Gordon Royle. *Algebraic Graph Theory*. Graduate Texts in Mathematics, 207. Springer-Verlag, New York, 2001.

[20] Vassos Hadzilacos. A lower bound for Byzantine agreement with fail–stop processors. Technical Report 21–83, Department of Computer Science, Harvard University, Cambridge, MA, July 1983.

[21] Maurice Herlihy, Dmitry Kozlov, and Sergio Rajsbaum. *Distributed Computing Through Combinatorial Topology*. Morgan Kaufmann, 2013.

[22] Maurice Herlihy, Sergio Rajsbaum, and Mark R. Tuttle. An axiomatic approach to computing the connectivity of synchronous and asynchronous systems. *Electr. Notes Theor. Comput. Sci.*, 230:79–102, 2009.

[23] Muhammad Samir Khan, Syed Shalan Naqvi, and Nitin H. Vaidya. Exact byzantine consensus on undirected graphs under local broadcast model. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC*, pages 327–336, 2019.

[24] Fabian Kuhn and Rotem Oshman. Dynamic networks: Models and algorithms. *SIGACT News*, 42(1):82–96, 2011.

[25] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, July 1982.

[26] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996.

[27] Yoram Moses and Sergio Rajsbaum. A layered analysis of consensus. *SIAM J. Comput.*, 31(4):989–1021, 2002.

[28] 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, PODC*, pages 218–227, 2019.

[29] David Peleg. *Distributed Computing: A Locality-Sensitive Approach*. SIAM, Philadelphia, PA, 2000.

[30] Michel Raynal. Consensus in synchronous systems: A concise guided tour. In *9th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pages 221–228, 2002.

[31] Michel Raynal. *Fault-Tolerant Message-Passing Distributed Systems - An Algorithmic Approach*. Springer, 2018.

[32] Nicola Santoro and Peter Widmayer. Agreement in synchronous networks with ubiquitous faults. *Theor. Comput. Sci.*, 384(2-3):232–249, October 2007.

[33] Lewis Tseng and Nitin H. Vaidya. Fault-tolerant consensus in directed graphs. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC*, pages 451–460. ACM, 2015.

[34] Lewis Tseng and Nitin H. Vaidya. A note on fault-tolerant consensus in directed networks. *SIGACT News*, 47(3):70–91, August 2016.

[35] John H. Wensley, Leslie Lamport, Jack Goldberg, Milton W. Green, Karl N. Levitt, P. M. Melliar-Smith, Robert E. Shostak, and Charles B. Weinstock. Sift: Design and analysis of a fault-tolerant computer for aircraft control. In *Proceedings of the IEEE*, volume 66, pages 1240–1255, Oct 1978.

[36] Kyrill Winkler and Ulrich Schmid. An overview of recent results for consensus in directed dynamic networks. *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, 128:41–72, June 2019.