k-Set Agreement in Communication Networks with Omission Faults

Emmanuel Godard[1] and Eloi Perdereau[2]

1 **Aix Marseille University, CNRS, LIF, Marseille, France**
emmanuel.godard@lif.univ-mrs.fr

2 **Aix Marseille University, CNRS, LIF, Marseille, France**
eloi.perdereau@lif.univ-mrs.fr

Abstract

We consider an arbitrary communication network G where at most f messages can be lost at each round, and consider the classical k-set agreement problem in this setting. We characterize exactly for which f the k-set agreement problem can be solved on G.

The case with $k = 1$, that is the Consensus problem, has first been introduced by Santoro and Widmayer in 1989 [20], the characterization is already known from [10]. As a first contribution, we present a detailed and complete characterization for the 2-set problem. The proof of the impossibility result uses topological methods. We introduce a new subdivision approach for these topological methods that is of independent interest.

In the second part, we show how to extend to the general case with $k \in \mathbb{N}$. This characterization is the first complete characterization for this kind of synchronous message passing model, a model that is a subclass of the family of oblivious message adversaries.

1998 ACM Subject Classification C.2.4 Distributed Systems

Keywords and phrases k-set agreement, message passing, dynamic networks, message adversary, omission faults

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2016.8

1 Introduction

We consider communication networks with arbitrary topology where some messages can be lost. The system evolves in rounds, and at each round, at most f messages can be lost, the unreliable links can be different in each round. We consider the classical k-set agreement problem that has been introduced in 1993 by Chaudhuri [9]. The problem, for n processes, is defined informally as follows. Given $k + 1$ possible initial values, each process must decide a final value among the proposed values in such a way that there are at most k different decided values. Note that when $n = k + 1$, this problem is also defined as the set agreement problem.

The system can be modeled by an adversary that in every round can choose f messages to be "omitted". The corresponding faulty communication links are not necessarily the same at each round and can be changed later by the adversary, it is an oblivious adversary. Such faulty communication actually induces a sub-directed graph of G (or sub-digraph). Such directed graphs are called "instant graphs". The message adversaries terminology was introduced in [1] despite this model being introduced a long time ago, see Subsection 1.3. Since a general characterization of the solvability of the k-set agreement is still elusive for synchronous message passing (whereas the minimal failure detector to solve this problem is known for shared memory [11]), it is of interest to consider such special cases. This communication model is an important case of oblivious message adversaries.

20th International Conference on Principles of Distributed Systems (OPODIS 2016).
Editors: Panagiota Fatourou, Ernesto Jiménez, and Fernando Pedone; Article No. 8; pp. 8:1–8:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1.1 Our Result

We give a complete characterization of the k-set agreement problem for networks with arbitrary topology in the omission faults model. We introduce a new combinatorial parameter for any graph G, denoted $c_k(G)$, that is the number of edge removals that G can withstand while having at most k connected components. This parameter is an extension of the classical edge-connectivity of a graph (that corresponds to $c_1(G)$ with our notation).

We prove that k-set agreement is solvable in G despite at most f message losses per round if and only if $f \leq c_k(G)$. The necessary condition is first proved for graphs of size $k + 1$ using a reduction to the Sperner's lemma. For general graphs, we then show how to reduce to one of these graphs with $k + 1$ vertices.

Interestingly, while the case of the $k + 1$-clique corresponds to the standard chromatic subdivision found in topological proofs for distributed computability, we had to introduce a new technique, that is called a round diagram, in order to solve other non-complete topology cases. This new technique could be of independent interest in distributed computability.

1.2 Related Works

The k-set agreement problem is a classical paradigm of coordination problems. It is also a theoretical benchmark for distributed computability in numerous models. A recent review by Raynal can be found in [17].

The solvability of the 1-set agreement problem, that is, the Consensus problem, in the context of communication networks with arbitrary topology has been introduced by Santoro and Widmayer in 1989 [20, 21]. It has been fully characterized for arbitrary oblivious message adversaries in [10] answering the same problem as this paper for $k = 1$, a problem that was open since [20]. In this setting, the Consensus problem is equivalent to the Broadcast problem, that is the network should be connected at each round, i.e. f must be less than the connectivity of the underlying graph to be able to solve 1-set agreement.

The solvability of the k-set problem has been considered in the omission context by [12]. The communication graph is the complete graph and the omissions are counted in the whole execution. By contrast, our work present models that can withstand an infinite number of omissions.

The k-set agreement problem has been investigated in the context of dynamic networks in [2, 3], where the adversaries are non-oblivious. We have been recently made aware of an independent work of Biely et al, [4] under submission, that presents an algorithm that would solve k-set agreement in the sufficient condition of Section 5. Like in the $k = 1$ case, where Consensus algorithms are actually simple flooding algorithms, note that the main contribution here is for the impossibility results.

In the shared memory model, the impossibility of wait-free k-set agreement for more than $k + 1$ processes is one of the crowning achievements of topological methods in distributed computing [15, 19, 6].

1.3 Related Models

The failure model considered here is very relevant in many ways. This model of synchronous communication has actually been introduced numerous times under different names. We present briefly the mobile omissions model [20] then the more recent "Heard-of" model [8], the iterated snapshot model [5] and its final evolution as the *message adversary* model [1]. Some equivalences have been proved between these synchronous presentations and asynchronous models in the case of non-coloured tasks [7]. Note also that in the case of dynamic networks, whenever the communication primitive is a broadcast (to the current neighbours), this model can also be used.

Mobile Omissions / Omission Faults. This is the model originally used in [20] and [21] by Santoro and Widmayer. At a given moment, there are at most f omission faults, that is f arcs missing from the underlying graph G. In the following step, it is possible that omissions have "moved" at other arcs of the network. Hence the name. Note that in [20, 21], other kind of failures are also considered like byzantine failures. These are oblivious adversaries.

Heard-of Model. The "Heard-of" model has been presented by Charron-Bost and Schiper in [8] to model what the authors have called "benign faults", i.e. transient faults like omission faults. The presentation is mostly done in the logical form, where a special predicate HO describes for every node the set of nodes it received a message from in the current step. The system is evolving synchronously except that nodes do not start the algorithm at the same round. In this model, numerous families were considered, some are oblivious, other not.

Iterated Write Snapshot. This is a shared memory model. Single-writers/multi-readers registers are accessible by processes. There is usually as many registers as processes and the registers are arranged in a one-shot array. It can be assumed, as in [5], that there is a `writeSnapshot()` primitive that enables processes to atomically write values to their register and read the content of the full array. Each concurrent access can read the values corresponding to the calling process and also the values previously written by other processes in the same round. In a given round, all possible interleaving of calls to `writeSnapshot()` are allowed. Process may never fail, however, it is possible that a correct process never sees another correct process (e.g. if it always writes first).
The main interest of this model is that it has a simple synchronous and regular structure and that it was proved in [7] that a bounded colourless task can be wait-free solved in the classical read-write model if and only if it is solvable in the Iterated Write Snapshot model. So this model has the same computing power as shared memory, but using topological tools is simpler in this model (see the tutorial in [14] and the thorough coverage of [13]).

Message Adversaries. The communication structure that one gets with a shared memory model is usually edge-transitive. Considering message passing systems, this condition is actually not necessary, and in [1], where the terminology of "message adversaries" is introduced, this condition is dropped by considering various families of graphs where the instant graphs are not transitive.
In [1], Afek and Gafni show that the same tasks can be solved as in standard asynchronous shared memory if the instant digraphs are tournaments. So some message adversaries with specific non-complete topologies have the same computing power as classical shared memory.
Subsequently, Raynal and Stainer have shown in [18] that it is possible to consider various message adversaries (they are not all oblivious) where further restrictions on the set of possible scenarios, i.e. weaker adversaries, correspond to well known asynchronous shared-memory models enriched with failures detectors. It appears that the message adversary model is a very rich, but also very convenient (some have very simple protocol complex) model to describe distributed systems from the point of view of computability.

1.4 Outline

We first present our notations and formally define the communication model. Section 3 considers G to be one of the graphs K_3 and P_3 (the 3-clique and the 3-path), and using two different subdivisions of the 2-dimensional simplex (one for each graph), we show that the number of possible failures is less than $c_2(G)$ for both cases. We show this is optimal for the set agreement problem. Then, in Section 4, we consider graphs of arbitrary size and show how to reduce the solvability of the 2-set agreement problem to one of the previous cases.

In the last section we investigate the general case $k \in \mathbb{N}$. We first show how to prove the impossibility of the set agreement problem for graphs of size $k+1$ using the chromatic subdivision technique with a twist, some instant graphs can appear more than one time in the subdivision. We then show that the reduction of the case $k = 2$ easily extends to arbitrary k and prove the necessary condition. Finally, a simple k-set agreement algorithm demonstrates that the condition $f \leq c_k(G)$ is sufficient, the characterization is complete.

2 Model and Definitions

2.1 Graphs and digraphs

Let $G = (V, E)$ be an undirected graph, we note $dir(E) = \bigcup_{\{u,v\} \in E} \{(u, v), (v, u)\}$ the set where each edge is replaced by two symmetric arcs ; we then have $|dir(E)| = 2|E|$. By extension of notation, for every undirected graph $G = (V, E)$, we note $dir(G) = (V, dir(E))$ the corresponding directed graph.

Given a graph G, a sub-digraph of $dir(G)$ is called an *instant graph*.

Let $D = (V, A)$ be a directed graph and $p \in V$ a vertex. The in-neighbourhood of each vertex $p \in V$ is denoted by $N^-_D(p)$ and corresponds to the set of sources of each arc reaching p in A:

$$\forall p \in V \quad N^-_D(p) = \{q \in V \mid (q, p) \in A\}$$

A directed path, or dipath, from p to q in D is a sequence $p_0, \ldots, p_t$ where $p_0 = p, p_t = q$ and $\forall i \; 0 \geq i \geq t-1 \quad (p_i, p_{i+1}) \in A$. The Boolean predicate $\text{path}_D(p, q)$ indicates the existence of a dipath from p to q in D.

We define now the set of vertices of D *reachable from* $p \in V$: $\text{Reach}_D(p) = \{q \in V \mid \text{path}_D(p, q) = 1\}$. And for all $U \subseteq V$ and $p \in V$, $AllReach_D(p, U) = \{D' = (V, A') \mid A' \subseteq A \wedge U \subseteq \text{Reach}_{D'}(p)\}$ is the set of all sub-graphs of D in which every vertices q in U are reachable from p.

We say that D is a *strongly connected graph* if there is a path between every pair of vertices. In other words $\forall p \in V \quad \text{Reach}_D(p) = V$. Note that if an undirected graph G is connected, $dir(G)$ is strongly connected.

Let $S \subseteq V$, we note $D_{|S}$ the graph induced from D by the vertices S. And we say that S is a *strongly connected component* (or SCC) if $D_{|S}$ is a maximal strongly connected subgraph of D.

2.2 Message Adversaries

In the general case, a message adversary is simply a set of infinite sequences of instant graphs. We only consider here “oblivious” or “iterated” message adversaries where instant graphs can be chosen in the same set which remains fixed all along the execution. In other words, the adversary does not choose by looking at the past. So given a fixed set of digraphs $\mathbf{M}$, we consider only the infinite sequences of elements of $\mathbf{M}$. Such a sequence is denoted $D^1, D^2, \cdots$ or $(D^i)_{i=1}^{\infty}$. The set of such infinite sequences is denoted $\mathbf{M}^\omega$. The set of finite sequences of length $r \in \mathbb{N}$ is denoted $\mathbf{M}^r$. Oblivious adversaries are simply the sets of digraphs $\mathbf{M}$.

Let $G = (V, E)$ be a graph and note $A = dir(E)$. We introduce two message adversaries. The f-omission message adversary $\mathbf{O}_f(G)$ is the set of all possible sub-graphs when at most f arcs can be removed from A:

▶ **Definition 1** (f-omission message adversary).

$$\mathbf{O}_f(G) = \{D' = (V, A') \text{ digraph} \mid A' \subseteq A \wedge |A| - |A'| \leq f\} .$$

The f-half-duplex message adversary forbids the removal of two symmetric arcs between two vertices:

▶ **Definition 2** (f-half-duplex message adversary).

$$\mathbf{HD}_f(G) = \{D' = (V, A') \mid D' \in \mathbf{O}_f(G) \wedge \forall p, q \in V \quad \{p, q\} \in E \wedge (p, q) \notin A' \Rightarrow (q, p) \in A'\} .$$

Note that by construction $\mathbf{HD}_f(G) \subseteq \mathbf{O}_f(G)$. Moreover, $\mathbf{HD}_f(G)$ contains all the tournament graphs with base G.

2.3 Execution of a Distributed Algorithm

A scenario is a sequence of instant graphs. We explain how to relate executions to scenarios. Given a oblivious message adversary $\mathbf{M}$, we define what is an execution of a given algorithm $\mathcal{A}$ with a given initial configuration ι. Every process can execute the following communication primitives:

- $send(msg)$ to send the same message msg to all out-neighbours,
- $recv()$ to get the messages from all in-neighbours.

An *execution*, or *run*, of algorithm $\mathcal{A}$ *subject to* scenario $\sigma \in \mathbf{M}^\omega$ is the following. Consider process u and one of its out-neighbours v in the underlying communication network. During round $r \in \mathbb{N}$, a message msg is sent from u to all its neighbours according to the instructions in algorithm $\mathcal{A}$. The node v will receive the corresponding message msg only if H, the r-th element of σ, is such that $(u, v) \in H$. All messages sent in a round can only be received in the same round. After sending and receiving messages, all processes update their states according to $\mathcal{A}$ and the messages they received. Given that all nodes have unique identities, when a message is received, it is known from which neighbour it is received. A *configuration* corresponds to the set of local states at the end of a given round.

Given $w \in \mathbf{M}^r$, and an initial configuration ι, let $s^p_\iota(w)$ denote the state of process p at the end of the r-th round of algorithm $\mathcal{A}$ subject to scenario w, with initial configuration ι. The initial state of p is therefore $\iota(p) = s^p_\iota(\varepsilon)$ where ε is the empty run. When ι is clear from the context, we might omit it and simply write $s^p(w)$. An *execution* of $\mathcal{A}$ subject to scenario $\sigma \in \rho(\mathbf{M})$ is the (possibly infinite) sequence of such message exchanges and corresponding configurations.

2.4 The k-Set Agreement Problem

We give the formal definition of the k-set agreement problem. This problem has been introduced in 1993 by [9].

Consider a system with n vertices. Given that each process has an initial value, each of them must decide a final value among the proposed values in such a way that there are at most k different decided values.

Validity: Any final value was the initial value of some process,
k-agreement: The set of final values is of size at most k,
Termination: Every process outputs a final value.
Note that when $n = k + 1$, this problem is also defined as the set agreement problem.

f \ k	1	2	3
1	yes	yes	yes
2	no	**yes**	yes
3	no	**no**	yes

(a) in $dir(K_3)$

f \ k	1	2	3
1	no	**yes**	yes
2	no	**no**	yes
3	no	no	yes

(b) in $dir(P_3)$

Figure 1 Solvability of the k-set agreement according to the number of omission faults f in $dir(K_3)$ and $dir(P_3)$.

3 2-Set Agreement in the K_3 and P_3 Topology

In this section we characterize the solvability of the 2-set agreement problem according to the number of possible omission faults f in graphs $dir(K_3)$ and $dir(P_3)$. We note $\Pi = \{p_1, p_2, p_3\} = \{\bullet, \bullet, \circ\}$ the set of processes in the network.

Figure 1 depicts both $dir(K_3)$ and $dir(P_3)$ and the solvability of the k-set agreement with respect to the number of omission faults f when $k \leq 3$ and $f \leq 3$. The $k = 3$ case is trivially solvable but highlights the border between the solvability and unsolvability of the k-set agreement. We prove below the four results highlighted in bold font in the table. Other results of the table are either straightforward or already well known, see e.g. [10]. This table completely characterizes the 2-set agreement for K_3 and P_3.

▶ **Proposition 3.** *The* 2*-set agreement problem is solvable in* $\boldsymbol{HD}_f(K_3)$ *if and only if* $f \leq 2$.

Proof. For the necessary part, we show that the 2-set agreement is impossible in $\mathbf{HD}_3(K_3)$. To this end, we extract from $\mathbf{HD}_3(K_3)$ a subset $\mathcal{S}$ of digraphs (its poset by arc inclusion is shown in Figure 2) such that we can construct a subdivision of the triangle by gluing graphs together by identifying the views of the processes (the in-neighbourhood of the vertices). More formally, let two digraphs $D_1 = (\Pi, A_1)$ and $D_2 = (\Pi, A_2)$ of $\mathcal{S}$. D_1 and D_2 are glued together by identifying p_i and p_j of Π if $N^-_{D_1}(p_i) = N^-_{D_2}(p_i)$ and $N^-_{D_1}(p_j) = N^-_{D_2}(p_j)$. The newly created object after identifying all such views is depicted in Figure 3 and corresponds indeed to a subdivision of the triangle.

Suppose there exists a 2-set agreement algorithm. From a distributed system point of view, the triangles which form the subdivision correspond to all possible 1-round execution. And we can iterate the subdivision to represent all possible executions of a certain round. Now, if we look closer to the subdivision and the represented digraphs, we see that the processes standing in the corner of the subdivided triangle doesn't receive any messages, they thus decide their own value. Moreover, those on the edges receive messages from processes in the corresponding corners, and those inside receive messages from every processes.

Consider the values decided by a all executions of the 2-set agreement algorithm. What we get here is a Sperner colouring where the colour of a vertex in the subdivision is the decision value of the associated process: every algorithm run on $\mathcal{S} \subseteq \mathbf{HD}_3(K_3)$ that terminates leads to a decision of each process as described, and thus to a Sperner colouring on the iterated subdivided figure. Now, by Sperner's Lemma (see e.g. [19]) we have that for any Sperner colouring on a subdivided triangle, there exists at least one 2-dimensional simplex coloured

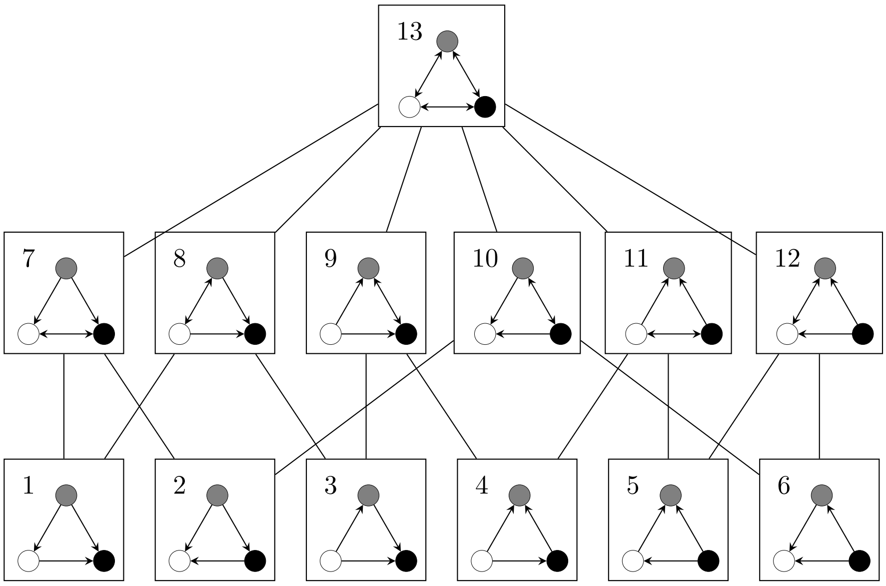

Figure 2 Poset of $\mathcal{S}$.

by three different colours, i.e. an execution where the three processes decide three different values. The 2-agreement condition of the problem is thus violated.

For the sufficient part, we give a simple flooding algorithm that solves the 2-set agreement in a single round in K_3 with 2 or less omission faults: we fix a priority order (which can be arbitrary) that is known by all processes $p \in \Pi$. Now, each process decides the value of its candidate after one round (i.e. the process with the highest priority that it is aware of).

To prove the correctness of this algorithm, fix the priority order π: $\forall p_i \in \Pi \quad \pi(p_i) = i$. For the algorithm to work, all it takes is that a message is received by a process having a lower priority than the sender: if A is the set of arcs of the digraph in the first round, it is sufficient that there exists $(p_i, p_j) \in A$ such that $\pi(p_i) > \pi(p_j)$ because in that case p_j will have p_i as candidate and so p_i and p_j will have the same candidate after one round, thus deciding the same value. In $dir(K_3)$, there are 3 such arcs (whatever π), and as $f \leq 2$, there is at least one in every $D \in \mathbf{HD}_2(K_3)$. ◀

▶ **Proposition 4.** *The 2-set agreement problem is solvable in* $\boldsymbol{HD}_f(P_3)$ *if and only if* $f \leq 1$.

Proof. As for $dir(K_3)$, for the necessary part, we extract from $\mathbf{HD}_2(dir(P_3))$ a subset of instant digraphs that forms a subdivision of the triangle (Figure 4). This subset actually consist of all graphs of $\mathbf{HD}_2(dir(P_3))$.

The proof argument is the same as for Proposition 3. Yet, we notice that the vertices $\circ$ and $\bullet$ in the inside doesn't receive the messages from all other vertices ; and three inside triangles (the topmost ones) correspond to the same instant digraph. This is not a problem because as the states of the processes constituting these triangles are the same, they will be coloured the same way, i.e. the association between an execution and a triangle in the graph still stands. Moreover, the processes colours (decision value) satisfies a Sperner colouration: nodes on an edge are coloured by one of the colours of its end points and nodes in the interior are coloured by one of the colours of the triangle. Thus, we can apply Sperner's lemma as in the classical proof (see [13, Chap. 9]).

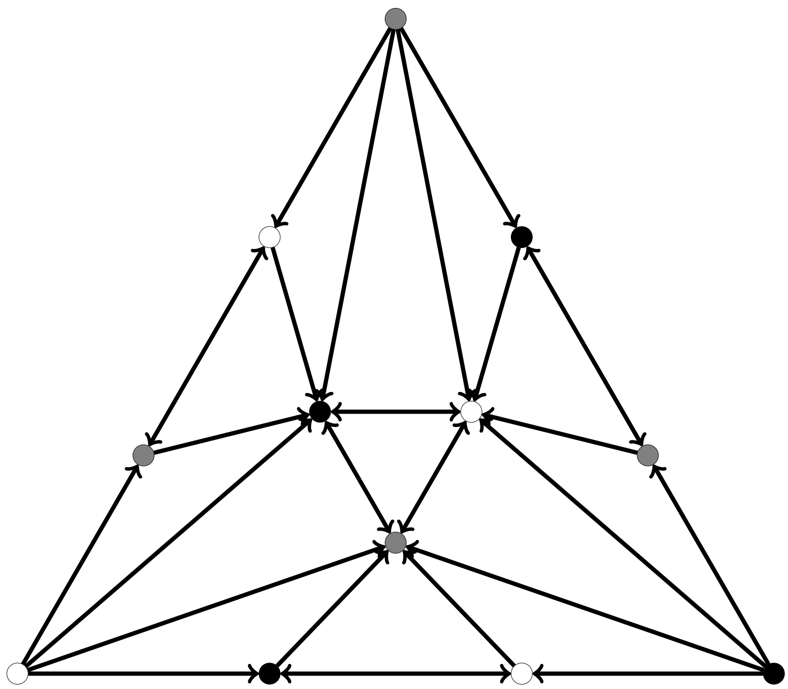

Figure 3 Subdivision constructed from $\mathcal{S} \subseteq \mathbf{HD}_3(\mathcal{K}_3)$.

The necessary part uses the same algorithm as in the proof of Proposition 3: in $dir(P_3)$ there are 2 arcs (p_i, p_j) with $\pi(p_i) > \pi(p_j)$, and $f \leq 1$ so there is at least one in each $D \in \mathbf{HD}_1(P_3)$. ◀

4 Solvability of 2-Set Agreement for Arbitrary Graphs

Before stating the characterization for graphs of arbitrary size, let's introduce first some notations and an important lemma.

4.1 Notations and the Causal Influence Lemma

We generalize the standard notion of *edge-connectivity* by introducing a new parameter ℓ allowing us to define the number of connected components.

▶ **Definition 5.** An undirected graph $G = (V, E)$ is (k, ℓ)-*edge-connected* ($\ell > 0$) if and only if $\forall E' \subseteq E \quad |E'| \leq k \Rightarrow G' = (V, E \setminus E')$ have at most ℓ connected components.

In other words, G cannot be cut into $\ell + 1$ connected components with only k edges removed.

▶ **Definition 6.** The ℓ-edge-connectivity of a graph G, denoted by $c_\ell(G)$, is the largest possible k such that G is (k, ℓ)-edge-connected.

In particular, the classical edge-connectivity $conn(G)$ corresponds to $c_1(G) + 1$: it is the minimal number of edges we need to remove to be able to disconnect the graph in two connected components.

For $dir(G) = (V, A)$, we remark that all sub-digraphs $dir(G') = (V, A')$ such that $|A| - |A'| \leq c_\ell(G)$ have at most ℓ strongly connected components.

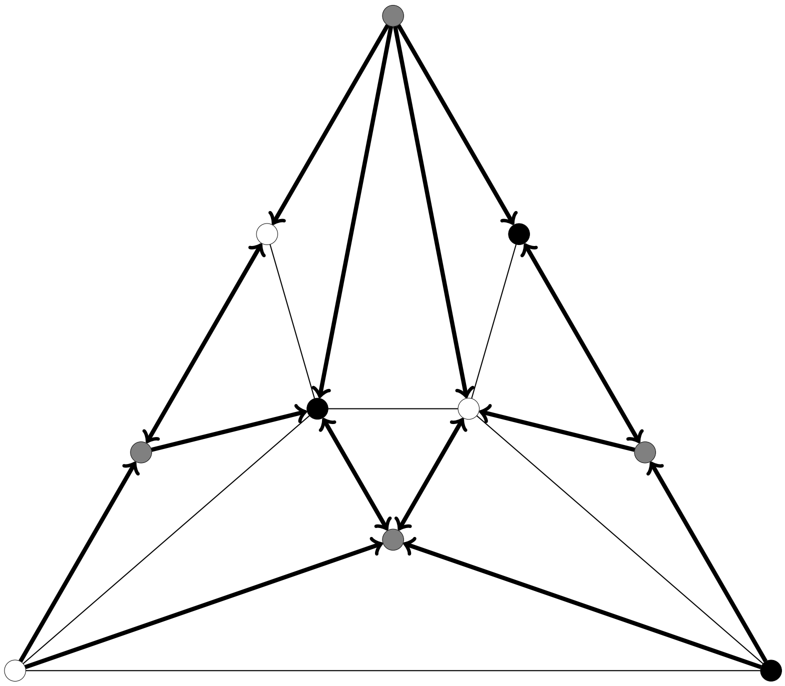

Figure 4 Subdivision constructed from $\mathbf{HD}_2(P_3)$.

Indeed, assume the opposite, this means that there exists $c_\ell(G)$ arcs that can be removed in $dir(G)$ to form more than ℓ strongly connected components in $dir(G)$. Now, remove the corresponding set of edges in G, there are no more than $c_\ell(G)$ of them, and one would get more than ℓ connected components in G. A contradiction with the definition of $c_\ell(G)$.

Given $p \in \Pi$, we note x_p the initial value of process p, and $\mathrm{Info}_p(r)$ the set of initial values known by p at round r. Let $N_p^r = N^-_{D^r}(p)$ be the set of source processes of incoming arcs of p at round r. In a full-information protocol, Info_p is defined as:

$$\begin{aligned} &\mathrm{Info}_p(0) = \{x_p\}\,, \\ \forall r > 0 \quad &\mathrm{Info}_p(r) = \mathrm{Info}_p(r-1) \cup \{x_q \mid q \in N_p^r\}\,. \end{aligned}$$

Remark that Info_p can only grow or stay fixed from round to round, i.e. the processes do not forget the values they have seen in the past.

Denote $\mathcal{I}^a_{p,b}$ the set of processes that know at round a at least the information that p had at round b. Formally:

$$\mathcal{I}^a_{p,b} = \{q \in \Pi \mid \mathrm{Info}_p(b) \subseteq \mathrm{Info}_q(a)\}\,.$$

If p or b can be deduced from the context, we will simply write $\mathcal{I}^a_p$ or even $\mathcal{I}^a$.

Recall that $AllReach_D(p, S)$ is the set of all sub-graphs of D in which every vertices $q \in S$ are reachable from p.

The following lemma expresses the fact that after $n-1$ occurrences of digraphs in which some processes are reachable from $p \in V$, they will have the information that p had.

▶ **Lemma 7** (Causal Influence Lemma). *Let $\sigma = (D^i)_{i=1}^{\infty}$ be a sequence of instant digraphs and $t \geq n-1$. Let $S \subseteq V$; if there exists an increasing sequence of indices $1 \leq i_1 < \ldots < i_t$ such that $\forall 1 \leq j \leq t \quad D^{i_j} \in AllReach_D(p, S)$, then in a full information protocol, we have:*

$$\forall q \in S \quad \forall i_{t'} > i_t \quad \mathrm{Info}_p(i_1) \subseteq \mathrm{Info}_q(i_{t'})\,.$$

Proof. Note $\forall r \geq 0 \quad \mathcal{I}^r = \mathcal{I}^r_{p,i_1}$. Consider the worst case where $\mathcal{I}^{i_1} = \{p\}$. Let $1 < a \leq t$; if $\mathcal{I}^{i_a} \not\subseteq S$, then because $D^{i_a} \in \mathit{AllReach}_D(p,S)$, we necessarily have an arc from a vertex $q \in \mathcal{I}^{i_a}$ to $q' \in V \setminus \mathcal{I}^{i_a}$, and so $|\mathcal{I}^{i_{a+1}}| \geq |\mathcal{I}^{i_a}| + 1$, i.e. at each occurrence of a digraph of $\mathit{AllReach}_D(p,S)$, the number of processes informed by $\mathrm{Info}_p(i_1)$ increases by at least 1. Thus, as $|S| \leq n$, $t \geq n-1$ and in the worst case $|\mathcal{I}^{i_1}| = 1$, we have $S \subseteq \mathcal{I}^{i_{t'}}$ for all $t' > t$. ◀

Let π be a priority order over Π, we define $\forall p \in \Pi \quad \forall r \geq 0 \quad \mathrm{cand}^\pi_p(r) \in \mathrm{Info}_p(r)$ the *candidate* of $p \in \Pi$ at round r as the process with the highest priority known by p, with respect to π:

$$\mathrm{cand}^\pi_p(r) = \underset{q \in \mathrm{Info}_p(r)}{\arg\max}\, \pi(q)\,.$$

When π can be easily deduced from the context, we will sloppily write $\mathrm{cand}_p(r)$.

4.2 2-Set Agreement Characterization for Arbitrary Graphs

▶ **Theorem 8.** *$\forall G = (V,E)$ such that $|V| = n \geq 3$, the 2-set agreement problem is solvable despite f omission faults if and only if $f \leq c_2(G)$.*

Proof.

Sufficient part

The algorithm for the sufficient part acts as a full information protocol and processes decide after a sufficiently long time T (to be bounded later). It is based, once again, on a priority order π over V and it solves the 2-set agreement in $\mathbf{HD}_f(G)$ for all G if $f \leq c_2(G)$.

Denote $p^* \in V$ the process with the highest priority: $p^* = \underset{p \in V}{\arg\max}\, \pi(p)$ and note $\mathcal{I}^r = \mathcal{I}^r_{p,0}$ the set of processes "informed" by the value of p^* at round r and $\overline{\mathcal{I}^r} = V \setminus \mathcal{I}^r$ its complement. By definition, we have that $\forall r \geq 0 \quad \forall p \in \mathcal{I}^r \ \mathrm{cand}_p(r) = x_{p^*}$. We define the stability property of $\mathcal{I}$:

$$\forall a \leq b \quad \mathrm{stable}^b_a = \begin{cases} 1 & \text{if } \forall a \leq r, r' \leq b \quad \mathcal{I}^r = \mathcal{I}^{r'} \\ 0 & \text{otherwise} \end{cases}$$

The following lemma expresses the fact that after a stability period of n rounds, there are at most two candidates among all processes.

▶ **Lemma 9.** *Applying a full-information protocol, if there exist $a, b \geq 0$ such that $b - a \geq n$ and if* $\mathrm{stable}^b_a = 1$ *then*

$$\forall r \geq b \quad \left| \bigcup_{p \in V} \mathrm{cand}_p(r) \right| \leq 2\,.$$

Proof. $\forall a \leq b$, a stability period between a and b (i.e. $\mathrm{stable}^b_a = 1$) necessarily implies that we are in one of the following two configurations:

1. Either $\mathcal{I}^a$ is "saturated", i.e. $\mathcal{I}^a = V$, and $\forall r \geq a \quad \mathcal{I}^r = V$
2. Or there are no communication from $\mathcal{I}^r$ to $\overline{\mathcal{I}^r}$: $\forall (p,q) \in A$ and $\forall a \leq r \leq b$, $p \in \mathcal{I}^r$ implies that $q \in \mathcal{I}^r$. In other words, there are no messages from any process $p \in \mathcal{I}^r$ to a process $q \in \overline{\mathcal{I}^r}$. Indeed, the contrary would imply that q receive x_{p^*} and thus would be in $\mathcal{I}^r$.

In the first case, by definition of $\mathcal{I}$, we have that $x_{p^*} \in \mathrm{Info}_p(b)$ for all processes $p \in V$, thus the only candidate after round b is x_{p^*}.

Now for the second case, consider the sequence $(D^1, D^2, \ldots)$ of instant digraphs subgraphs of $dir(G)$. We have that in every round $a \le r \le b$ the D^r has more than one strongly connected component. And in fact, it has exactly two SCC because $f \le c_2(G)$ which by definition means that D^r has at most 2 SCC, which can only be $\mathcal{I}^r$ and $\overline{\mathcal{I}^r}$. In other words, the sub-graph $D^r_{|\overline{\mathcal{I}^r}}$ is a strongly connected graphs, so $\forall p \in \overline{\mathcal{I}^r}$, $D^r \in \mathit{AllReach}_{dir(G)_{|\overline{\mathcal{I}^r}}}(p, \overline{\mathcal{I}^r})$. In particular, consider $p^+ = \arg\max_{p\in\overline{\mathcal{I}^r}} \mathrm{cand}_p(a)$ the process that has the best candidate over all $\overline{\mathcal{I}^r}$ at round a. Now, applying Lemma 7 with $i_1 = a, i_2 = a+1, \ldots, i_t = b-1$ and $S = \overline{\mathcal{I}^r}$ gives us that $\forall q \in \overline{\mathcal{I}^r} \quad \forall r \le b \quad \mathrm{Info}_{p^+}(a) \subseteq \mathrm{Info}_q(r)$, i.e. by round b, all processes of $\overline{\mathcal{I}^b}$ have $\mathrm{cand}_{p^+}(a)$ as candidate. Now, $\mathrm{cand}_{p^+}(a)$ combined with x_{p^*} known by all processes of $\mathcal{I}^r$, leads to at most two different candidates for all processes at round b.

And this holds for all the following rounds $r \ge b$ because $\forall p \in \overline{\mathcal{I}^b}$, a change of candidate at round r $\mathrm{cand}_p(r) \ne \mathrm{cand}_p(b)$ necessarily means that $\mathrm{cand}_p(r) = x_{p^*}$: a message to from a process $q \in \mathcal{I}^{r-1}$ was received by p. ◀

Such a period of stability of n rounds must occurs before round n^2, or all processes are in $\mathcal{I}^{n^2}$: A "non stable" period from a to b of n rounds ($\mathrm{stable}_a^b = 0$ with $b - a = n$) necessarily implies that $\mathcal{I}^b \ge \mathcal{I}^a + 1$, i.e. at least one process get informed by x_{p^*} during this period. So if n consecutive such periods occur then all processes are informed; otherwise a period of n rounds has occurred and Lemma 9 tells us there are at most two different candidates. Thus, all processes $p \in V$ deciding value $x_{\mathrm{cand}_p(n^2)}$ respect the agreement property of the 2-set agreement and the problem is solved.

Necessary part

For the impossibility part, we proceed by reduction to $dir(P_3)$ and $dir(K_3)$: we suppose there exists an algorithm $\mathcal{A}$ that solves the 2-set agreement despite $f = c_2(G) + 1$ omission faults, and we construct an adversary for $dir(G)$ corresponding to an adversary for $dir(K_3)$ and $dir(P_3)$ for which the 2-set agreement is impossible to solve.

Let's first consider a decomposition of the vertices of G in 3 sets: $V = V_1 \cup V_2 \cup V_3$ corresponding to the definition of $c_2(G)$. Denote $d_{i,j}$ the number of edges between V_i and V_j in G; and $A_{i,j}$ the set of arcs in $dir(G)$ from V_i to V_j. By definition, we have that $|A_{i,j}| = |A_{j,i}| = d_{i,j} = d_{j,i}$. Let $H_i = (V_i, A_{i,i}) = G_{|V_i}$ $(1 \le i \le 3)$ the associated components in $dir(G)$. The decomposition is chosen such that $d_{1,2} + d_{2,3} + d_{3,1} = c_2(G) + 1 = f$. Consequently, we have:

$$\left(\bigcup_{1\le i\le 3} V_i, \bigcup_{1\le i,j\le 3} A_{i,j} \right) = dir(G)$$

Suppose the problem is solvable for $f = c_2(G) + 1$, i.e. there exists an algorithm $\mathcal{A}$ such that for each sequence of instant digraphs of $\mathbf{O}_{c_2(G)+1}(G)$, $\mathcal{A}$ terminates and solves the 2-set agreement problem. In particular, consider the message adversary $\mathbf{D}$ that removes only one given set $A_{i,j}$ of arcs between two components H_i and H_j $(i \ne j)$, three times.

Formally:

$$\begin{aligned}\mathbf{D} = \{&(V, dir(E) \setminus \{A_{i_1,j_1} \cup A_{i_2,j_2} \cup A_{i_3,j_3}\}) \mid \\ &\forall 1 \le a, b \le 3 \quad i_a \ne j_a \text{ and } a \ne b \Rightarrow ((i_a, j_a) \ne (i_b, j_b) \wedge (i_a, j_a) \ne (j_b, i_b))\} \ .\end{aligned}$$

We indeed have the inclusion $\mathbf{D} \subseteq \mathbf{O}_f(G)$ because the f-half-duplex condition necessarily implies that $|A_{i_1,j_1} \cup A_{i_2,j_2} \cup A_{i_3,j_3}| \leq f$. Note that if the decomposition in H_i of G forms a chain and not a clique, then one of the A_{i_a,j_a} is empty.

We now show that solving with $\mathcal{A}$ for $\mathbf{D}$ implies a solution for $\mathbf{HD}_3(\Gamma)$ where Γ has three processes: $\Pi = \{p_1, p_2, p_3\} = \{\circ, \bullet, \bullet\}$, with either $\Gamma = K_3$, or $\Gamma = P_3$ depending on the decomposition of G. To do so, we define a new algorithm $\mathcal{A}_\Gamma$ that simulates $\mathcal{A}$ in Γ such that every sequence $\sigma \in \mathbf{D}$ corresponds to a sequence $\sigma' \in \mathbf{HD}_3(\Gamma)$ where p_i simulates the processes V_i of H_i.

For every variable var used in $\mathcal{A}$, we note var_u^r the value of var for process $u \in V$ at round r. For each of these variables, the processes $p_i \in \Pi$ hold an array of values $\overline{var}_{p_i}^r$ for every round r where $\forall u \in V \quad \overline{var}_{p_i}^r[u]$ corresponds to the simulated value var_u^r for process u. For all $u, v \in V$, we note $m^r(u, v)$ the message possibly sent from u to v at round r by $\mathcal{A}$. Every process $p_i \in \Pi$ simulates the execution of $\mathcal{A}$ by V_i as such:

At round r, $p_i \in \Pi$ performs the two following steps:

1. Compute and locally record the values $\overline{var}_{p_i}^r[u]$ for all processes $u \in V_i$ by simulating their local behaviour and the content of messages $m^r(u, v)$ for all $u, v \in V_i$.
2. Send to process $p_j \in \Pi$ the concatenation of messages $m^r(u, v)$ where $u \in V_i$ and $v \in V_j$.

The step 1 corresponds to local computations in V_i and to the sending by u and recieving by v (update of v's state) of the messages transiting on $A_{i,j}$ by algorithm $\mathcal{A}$. Step 2 corresponds to messages transiting on $A_{i,j}$ by $\mathcal{A}$. We thus have the equality $\overline{var}_{p_i}^r[u] = var_u^r$ that stands for all $r \geq 0$ and all $u \in V_i$.

By our assumption on $\mathcal{A}$, all processes of V decide an output value that satisfies the 2-agreement property required. When the termination occurs, $p_i \in \Pi$ chooses the smallest value among the ones of its simulated processes V_i. This guarantees that the total number of different decided values in Γ isn't larger than those decided with $\mathcal{A}$. So this solves the 2-set agreement problem on Γ in $\mathbf{HD}_3(\Gamma)$, which is a contradiction with Proposition 3 (K_3) or 4 (P_3). ◀

5 General Case

5.1 Impossibility Proof for Graphs of Size $k + 1$

In this section, we assume $|G| = k + 1$. Let Δ^k be the standard k-dimensional simplex and let $G = (V, E)$ be a communication graph with $k + 1$ vertices. The vertices of G and Δ^k are indexed by the elements of $[k] = \{0, \ldots, k\}$. Using the description of the standard chromatic subdivision used in [16], we define a subdivision of the simplex Δ^k corresponding to G, which we denote $Chr(G)$.

For this purpose, we will use the author's notation, for a full description, see [16]. Our construction of $Chr(G)$ is in many points similar to the construction of [16], $\Sigma_{\mathcal{P}}(K)$ with $K = \Delta^k$ and $\mathcal{P}$ being the infinite family of cross-polytopes. The difference lies in subdivided simplices considered at each step: at step j, instead of replacing each $k - j + 1$-dimensional simplices of K (denoted $K^{(k-j+1)}$) by the corresponding Schlegel diagram, we only replace those in which their corresponding sub-graph has at least one edge. More formally we simply replace the definition of $K^{(i)}$ as such:

$$K_G^{(i)} = \{\sigma \in K \mid |\sigma| = i \wedge \mathbb{P}_2(\sigma) \cap E \neq \emptyset\}$$

where $\mathbb{P}_2(\sigma) = \{\tau \subseteq \sigma \mid |\tau| = 2\}$ is the set of all possible edges between the vertices of σ. Now, using the author's inductive steps we construct $X_1, \ldots, X_k$ and set $Chr(G) = X_k$.

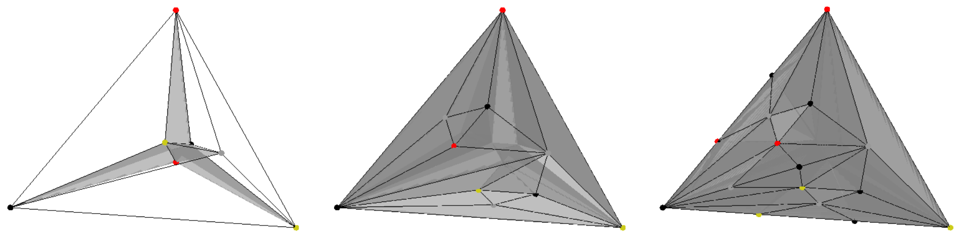

Figure 5 Construction steps of $Chr(S_3)$.

To visualize how $Chr(G)$ is constructed, we give an example for a graph with 4 vertices: the star S_3 (with • at the center). The subdivision is constructed in three steps ($\mathcal{P}_d$ is the Schlegel diagram of the cross-polytope of dimension d):

1. replace Δ^3 with $\mathcal{P}_3$;
2. for each faces σ of dimension 2 such that its corresponding sub-graph has at least one edge, replace σ by $\mathcal{P}_2$ and join the result with the previous step, thus subdividing a top-dimensional simplex;
3. same as 2 with faces of dimension 1, namely edges.

This construction is presented in Figure 5. For visualization comfort, we didn't grayed all 3-dimensional simplices for the first step.

The Figure 4 in Section 3 is another example, it represents $Chr(P_3)$.

▶ **Proposition 10.** *$Chr(G)$ is a simplicial subdivision of Δ^k.*

Proof. Comparing to [16], we only prevented the subdivision of some $k - i + 1$-simplices of Δ^k at step i. Yet, in the construction of [16], the transition of X_{i-1} to X_i preserves the fact that the resulting complex is a subdivision. Thus, by *not* subdividing simplices, the resulting complex is still a subdivision of Δ^k. ◀

Now, for each k-dimensional simplex σ of $Chr(G)$, we associate a directed graph γ_σ. To define it, recall the combinatorial description of the k-simplices in X_ℓ presented in [16]: they are in the form $\sigma = ((i_1, A_1), \ldots, (i_{k+1}, A_{k+1}))$ with $\{i_1, \ldots, i_{k+1}\} = [k]$ and satisfies some conditions on the A_is. Our construction only changes this description by adding a condition on the intersection of the edges of G and the A_is, preventing the creation of undesirable k-simplices.

We define corresponding graphs as follows:

$$\forall \sigma \in Chr(G) \quad \gamma_\sigma = ([k], \{(i_a, i_b) \mid \{i_a, i_b\} \in E \wedge A_a \subseteq A_b\}).$$

And let $\Gamma(G) = \{\gamma_\sigma \mid \sigma \in Chr(G)\}$ be the set of all such digraphs.

▶ **Proposition 11.** *For all communication graph $G = (V, E)$ with $|E| = m$, we have*

$$\Gamma(G) \subseteq \boldsymbol{HD}_m(G).$$

Proof. Every γ_σ has the same set of vertices than G, namely $[k]$ and for each edge $\{i_a, i_b\} \in E$, we add one or two arcs depending on the inclusion order of A_a and A_b. Thus γ_σ is a sub-digraph of G and has at least m arcs, so it indeed lies in $\mathbf{HD}_m(G)$. ◀

Before stating the main lemma of this section, we need to remark a useful fact about ℓ-edge-connectivity.

▶ **Lemma 12.** *For any communication graph $G = (V, E)$ with $|V| = k + 1$ and $|E| = m$, we have*

$$c_k(G) = m - 1\,.$$

Proof. In order to have $|V| = k + 1$ connected components, we need to remove every edge. Now, removing $m - 1$ edges guaranties us to have exactly one connected components with two vertices, thus we have no more than k connected components. ◀

▶ **Lemma 13.** *For any communication graph $G = (V, E)$ with $|V| = k + 1$ and $|E| = m$, the k-set agreement problem is not solvable if $c_k(G) + 1$ or more omission faults can occur.*

Proof. First, note following what we just proved above, $c_k(G) + 1 = m$. Now, we roughly use the same technique we used in Section 3 to prove the impossibility of the 2-set agreement in the K_3 and P_3 topology. The difference is that instead of extracting a subset of $\mathbf{HD}_m(G)$ that can form a subdivision of Δ^k, we rather construct the subdivision $Chr(G)$ which is indeed a subdivision of Δ^k. Sperner's lemma tells us that at least one k-dimensional simplex has its vertices coloured with $k + 1$ different colours. Thus k-set agreement is not solvable for the adversary consisting of the graphs corresponding to the k-dimensional simplices of $Chr(G)$ – namely $\Gamma(G)$. And Proposition 11 yields that $\Gamma(G) \subseteq \mathbf{HD}_m(G)$ and by definition $\mathbf{HD}_m(G) \subseteq \mathbf{O}_m(G)$; thus the result follows. ◀

5.2 General Proof

We can now state the main theorem of the paper which fully characterize the solvability of the k-set agreement problem in the model of the paper.

▶ **Theorem 14.** *Let $k \in \mathbb{N}$ and $G = (V, E)$ be any communication network. The k-set agreement problem is solvable despite f omission faults if and only if $f \leq c_k(G)$.*

Proof. To show the impossibility part, i.e. the k-set agreement is not solvable if $f \geq c_k(G)+1$, we reduce G to the case of a smaller graph that has $k + 1$ vertices and we use Lemma 13 to prove the impossibility.

Let $\mathcal{G}(n)$ be the set of all undirected graphs with n vertices.

Let's first consider a partition of the vertices of G in $k + 1$ non-empty sets: $V = V_1 \cup \ldots \cup V_{k+1}$ associated with $c_k(G)$. Denote $d_{i,j}$ the number of edges between V_i and V_j in G; and $A_{i,j}$ the set of arcs in $dir(G)$ from V_i to V_j (some may empty):

$$A_{i,j} = \{(u, v) \mid u \in V_i \wedge v \in V_j\}\,.$$

By definition, we have that $|A_{i,j}| = |A_{j,i}| = d_{i,j} = d_{j,i}$. The partition is chosen such that $\sum_{1 \leq i,j \leq k+1} d_{i,j} = c_k(G) + 1 = f$.

Let G' be the communication graph obtained by contracting each of the $k + 1$ sets V_i into a single process p_i (and removing redundant edges). Denote Π the set of G''s processes and m its number of edges. Remark that $|\{\{i, j\} \mid 1 \leq i, j \leq k + 1 \text{ and } d_{i,j} \neq 0\}| = m$.

Suppose the problem solvable for $f = c_k(G) + 1$, i.e. there exists an algorithm $\mathcal{A}$ such that for each sequence of instant digraphs of $\mathbf{O}_{c_k(G)+1}(G)$, $\mathcal{A}$ terminates and solves the k-set agreement. In particular, consider the message adversary $\mathbf{D}$ that removes – at most m times – only one given set $A_{i,j}$ of arcs between two components V_i and V_j ($i \neq j$). Formally, we define $\mathcal{F}$ as the family of sets of arcs that satisfies this condition:

$$\mathcal{F} = \left\{ \bigcup_{1 \le a \le m} A_{i_a, j_a} \mid 1 \le i_a, j_a \le k+1 \text{ and } i_a \neq j_a \text{ and } \forall 1 \le b \le m \quad a \neq b \Rightarrow (i_a, j_a) \neq (j_b, i_b) \right\} .$$

Now $\mathbf{D}$ is defined as

$$\mathbf{D} = dir(G) \cup \{(V, dir(E) \setminus F) \mid F \in \mathcal{F}\} .$$

We indeed have the inclusion $\mathbf{D} \subseteq \mathbf{O}_f(G)$ because of the choice of the V_is and the half-duplex condition, we necessarily have for each $F \in \mathcal{F}$, $|F| \le f$.

We now show that solving the k-set agreement problem with $\mathcal{A}$ for $\mathbf{D}$ implies a solution for $\mathbf{HD}_m(G')$. To do so, we define a new algorithm $\mathcal{A}_{G'}$ that simulates $\mathcal{A}$ in G' such that every sequence $\sigma \in \mathbf{D}$ corresponds to a sequence $\sigma' \in \mathbf{HD}_m(G')$ where p_i simulates the processes V_i.

For every variable var used in $\mathcal{A}$, we note var_u^r the value of var for every process $u \in V$ at round r. In $\mathcal{A}_{G'}$, each process $p_i \in \Pi$ holds an array of values $\overline{var}_{p_i}^r$ for every round r where $\forall u \in V_i \quad \overline{var}_{p_i}^r[u]$ corresponds to the simulated value var_u^r for process u. For all $u, v \in V$, we note $m^r(u,v)$ the message possibly sent from u to v at round r by $\mathcal{A}$. Every process $p_i \in \Pi$ simulates the execution of $\mathcal{A}$ by V_i as such:

At round r, $p_i \in \Pi$ performs the two following steps:

1. According to $\mathcal{A}$, compute and locally record the values $\overline{var}_{p_i}^r[u]$ for all processes $u \in V_i$ by simulating their local behaviour and the content of messages $m^r(u,v)$ for all $u, v \in V_i$.
2. Send to process $p_j \in \Pi$ the concatenation of messages $m^r(u,v)$ where $u \in V_i$ and $v \in V_j$.

The step 1 corresponds to local computations in V_i and to the expedition and reception of the messages transiting on $A_{i,i}$ by algorithm $\mathcal{A}$. Step 2 corresponds to messages transiting on $A_{i,j}$ by $\mathcal{A}$. We thus have the equality $\overline{var}_{p_i}^r[u] = var_u^r$ that stands for all $r \ge 0$ and all $u \in V_i$.

By our assumption on $\mathcal{A}$, all processes of V decide an output value that satisfies the agreement condition of the k-set agreement problem. When this termination occurs, in $\mathcal{A}_{G'}$ $p_i \in \Pi$ chooses the smallest value among the ones of its simulated processes V_i. This guarantees that the total number of different decided values in $\mathcal{A}_{G'}$ isn't larger than those decided with $\mathcal{A}$. So this solves k-set agreement problemin $\mathbf{HD}_m(G')$, which is a contradiction with Lemma 13.

For the sufficient part, the k-set agreement algorithm is the exact extension of what is presented in Section 4.2. ◀

6 Conclusion

In this note, we give a complete characterization of the number of omission faults a communication network with arbitrary topology can withstand when solving the k-set agreement problem for any given $k \in \mathbb{N}$. Introducing a new combinatorial parameter c_k that is an extension of the classical edge-connectivity of graphs, we have shown that, given a graph G, the k-set agreement problem can be solved if and only if the number of omission faults per round is less than $c_k(G)$. The general solvability of the k-set agreement problem in oblivious message adversaries is still open but an important sub-class has been solved in this paper.

Note that we proved the necessary condition using a new twist for topological methods in distributed computability: we allow some one-step events to correspond to more than one simplex in the round diagram. This new technique is of independent interest and could open new impossibility proofs for other distributed problems. We recently found a way to prove (part of) the same result using a reduction from the well-known complete topology case. However, the relationship between this new topological technique and general reduction techniques is yet to investigate.

Acknowledgement. The authors wish to thanks Damien Imbs for fruitful discussions on the 2-set agreement case.

References

1 Yehuda Afek and Eli Gafni. *Asynchrony from Synchrony*, pages 225–239. Number 7730 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013.

2 Martin Biely, Peter Robinson, and Ulrich Schmid. Easy impossibility proofs for k-set agreement in message passing systems. In *OPODIS*, volume 7109 of *Lecture Notes in Computer Science*, pages 299–312. Springer, 2011.

3 Martin Biely, Peter Robinson, Ulrich Schmid, Manfred Schwarz, and Kyrill Winkler. Gracefully degrading consensus and k-set agreement in directed dynamic networks. In Ahmed Bouajjani and Hugues Fauconnier, editors, *Networked Systems – Third International Conference, NETYS 2015, Agadir, Morocco, May 13-15, 2015, Revised Selected Papers*, volume 9466 of *Lecture Notes in Computer Science*, pages 109–124. Springer, 2015.

4 Martin Biely, Peter Robinson, Ulrich Schmid, Manfred Schwarz, and Kyrill Winkler. Gracefully degrading consensus and k-set agreement in directed dynamic networks, 2016. submitted.

5 E. Borowsky and E. Gafni. Immediate atomic snapshots and fast renaming. In *Proc. of the 12th Annual ACM Symposium on Principles of Distributed Computing*, 1993.

6 Elizabeth Borowsky and Eli Gafni. Generalized flp impossibility result for t-resilient asynchronous computations. In *STOC'93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 91–100, New York, NY, USA, 1993. ACM Press. doi:10.1145/167088.167119.

7 Elizabeth Borowsky and Eli Gafni. A simple algorithmically reasoned characterization of wait-free computation (extended abstract). In *Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing*, PODC'97, pages 189–198. ACM, 1997. doi:10.1145/259380.259439.

8 Bernadette Charron-Bost and André Schiper. The heard-of model: computing in distributed systems with benign faults. *Distributed Computing*, 22(1):49–71, 2009. doi:10.1007/s00446-009-0084-6.

9 S. Chaudhuri. More choices allow more faults: Set consensus problems in totally asynchronous systems. *Information and Computation*, 105(1):132–158, Jul 1993.

10 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. doi:10.1016/j.tcs.2015.01.024.

11 Eli Gafni and Petr Kuznetsov. The weakest failure detector for solving k-set agreement. In *Proceedings of the 28th ACM Symposium on Principles of Distributed Computing*, PODC'09, pages 83–91, New York, NY, USA, 2009. ACM. doi:10.1145/1582716.1582735.

12 Rachid Guerraoui, Petr Kouznetsov, and Bastian Pochon. A note on set agreement with omission failures. *Electr. Notes Theor. Comput. Sci.*, 81, 2003.

13 Maurice Herlihy, Dmitry N. Kozlov, and Sergio Rajsbaum. *Distributed Computing Through Combinatorial Topology*. Morgan Kaufmann, 1 edition edition, 2013.

14 Maurice Herlihy, Sergio Rajsbaum, and Michel Raynal. Computability in distributed computing: A tutorial. *SIGACT News*, 43(3):88–110, 2012.

15 Maurice Herlihy and Nir Shavit. The topological structure of asynchronous computability. *J. ACM*, 46(6):858–923, 1999.

16 Dmitry N. Kozlov. Chromatic subdivision of a simplicial complex. *Homology Homotopy Appl.*, 14(2):197–209, 2012. URL: `http://projecteuclid.org/euclid.hha/1355321488`.

17 Michel Raynal. *Set Agreement*, pages 1956–1959. Springer New York, New York, NY, 2016. `doi:10.1007/978-1-4939-2864-4_367`.

18 Michel Raynal and Julien Stainer. Synchrony weakened by message adversaries vs asynchrony restricted by failure detectors. In Panagiota Fatourou and Gadi Taubenfeld, editors, *PODC*, pages 166–175. ACM, 2013.

19 M. Saks and F. Zaharoglou. Wait-free k-set agreement is impossible: The topology of public knowledge. *SIAM J. on Computing*, 29:1449–1483, 2000.

20 Nicola Santoro and Peter Widmayer. Time is not a healer. In *STACS*, pages 304–313, 1989.

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