Tight Bounds for Connectivity and Set Agreement in Byzantine Synchronous Systems

Hammurabi Mendes[1] and Maurice Herlihy[2]

1 **Mathematics and Computer Science Department, Davidson College, Davidson, NC, USA**
`hamendes@davidson.edu`

2 **Computer Science Department, Brown University, Providence, RI, USA**
`mph@cs.brown.edu`

Abstract

In this paper, we show that the protocol complex of a Byzantine synchronous system can remain $(k-1)$-connected for up to $\lceil t/k \rceil$ rounds, where t is the maximum number of Byzantine processes, and $t \geq k \geq 1$. This topological property implies that $\lceil t/k \rceil + 1$ rounds are necessary to solve k-set agreement in Byzantine synchronous systems, compared to $\lfloor t/k \rfloor + 1$ rounds in synchronous crash-failure systems. We also show that our connectivity bound is tight as we indicate solutions to Byzantine k-set agreement in exactly $\lceil t/k \rceil + 1$ synchronous rounds, at least when n is suitably large compared to t. In conclusion, we see how Byzantine failures can potentially require one extra round to solve k-set agreement, and, for n suitably large compared to t, at most that.

1998 ACM Subject Classification F.1.1 Models of Computation

Keywords and phrases Byzantine, synchronous, k-set agreement, topology, connectivity

Digital Object Identifier 10.4230/LIPIcs.DISC.2017.35

1 Introduction

A *task* is a distributed coordination problem where multiple processes start with private inputs, communicate among themselves (by shared memory or message passing), and halt with outputs consistent with the task specification. There are *crash-failure* systems [1], where processes can fail only by permanent, unannounced halting, or *Byzantine-failure* systems [18], where processes can fail arbitrarily, even maliciously. Communication among processes can be *synchronous* or *asynchronous*. In *synchronous* systems, communication and computation are organized in discrete rounds. In each round, each non-faulty process performs as follows, in order:

(i) sends a message;
(ii) receives all messages sent in the current round by the other processes; and
(iii) performs internal computation.

In *asynchronous* systems, processes may have different relative speeds, and communication is subject to unbound, finite delays.

The problem of consensus in the synchronous Byzantine message-passing model was among the earliest to be investigated, and upper and lower consensus bounds in that model are well-understood. In this paper, we turn our attention to the bounds for problems such as k-set agreement, using concepts and techniques adapted from combinatorial topology. We can capture all possible information dissemination patterns permitted by this model in a single combinatorial structure called a *simplicial complex* (or just *complex*). A classical topological property of a simplicial complex is its level of *connectivity*, which is, roughly speaking, the

31st International Symposium on Distributed Computing (DISC 2017).
Editor: Andréa W. Richa; Article No. 35; pp. 35:1–35:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

dimension below which it has no holes. Many classical proofs of consensus impossibility can be reformulated as showing that certain complexes are 0-connected (also called *path-connected*), and all known impossibility proofs for k-set agreement rely on showing that certain complexes are $(k-1)$-connected. Very informally, the higher the degree of connectivity imposed by the adversary, the weaker the model's computational power. Here, we present the first tight bounds on connectivity for the synchronous Byzantine message-passing model.

Prior work using topological techniques is discussed in Section 2. Our operational setting is detailed in Section 3, and our topological model is formalized in Section 4.

Our **first contribution** comes in Section 5. We show that, in a Byzantine synchronous system, the protocol complex can remain $(k-1)$-*connected* for $\lceil t/k \rceil$ rounds, where t is an upper bound on the number of Byzantine processes. Perhaps surprisingly, this is only *one* more round than the upper bound for crash-failure systems ($\lfloor t/k \rfloor$, shown in [8]). In order to show that, as part of our **second contribution**, we conceive a combinatorial operator modeling the ability of Byzantine processes to *equivocate* – that is, to transmit ambiguous state information – without revealing their Byzantine nature. We compose this operator with regular crash-failure operators, extending the protocol complex connectivity for one extra round. As noted before, connectivity is of interest because a $(k-1)$-connected protocol complex prevents important problems such as k-set agreement [7, 9] from having solutions.

Our **third contribution** comes in Section 6. We show that the above connectivity bound is *tight* in certain settings (described in Section 6), by solving k-set agreement in $\lceil t/k \rceil + 1$ rounds. We do so with a full-information protocol that assumes n suitably large compared to t. The protocol suits well our purpose of tightening the $\lceil t/k \rceil$ bound, and also exposes clearly *the reason why* $\lceil t/k \rceil + 1$ rounds is enough to solve k-set agreement.

These results give new insight into the power of Byzantine adversaries for problems beyond consensus. Although Byzantine adversaries seem much more powerful than crash-failure ones, we show that a Byzantine adversary can impose at most *one* additional synchronous round beyond that imposed by a crash-failure adversary. In terms of solvability vs. number of rounds, the penalty for moving from crash to Byzantine failures, captured by $(k-1)$-connectivity in the protocol complex, can be *quite limited* in synchronous systems, particularly when n is relatively large compared to t.

2 Related Work

The Byzantine failure model was initially introduced by Lamport, Shostak, and Pease [18]. The use of simplicial complexes to model distributed computations was introduced by Herlihy and Shavit [15]. The asynchronous computability theorem for general tasks in [16] details the approach for asynchronous wait-free computation in the crash-failure model. This model was recently generalized by Gafni, Kuznetsov, and Manolescu [10]. Computability in Byzantine asynchronous systems, where tasks are constrained in terms of non-faulty inputs, was recently considered in [19].

The k-set agreement problem was originally defined by Chaudhuri [7]. Alternative formulations with different validity notions, or failure/communication settings, are discussed in [22, 9]. A full characterization of optimal translations between different failure settings is given in [2, 23], which requires different number of rounds depending on the relation between the number of faulty processes, and the number of participating processes.

The relationship between connectivity and the impossibility of k-set agreement is described explicitly or implicitly in [8, 16, 24]. Recent work by Castañeda, Gonczarowski, and Moses [6] considers an issue of chains of hidden values, a concept loosely explored here. The approach based on shellability and layered executions for lower bounds in connectivity has been used by Herlihy, Rajsbaum, and Tuttle [14, 13, 12], assuming crash-failure systems, synchronous or asynchronous.

3 Operational Model

We have $n+1$ processes[1] $\mathbb{P} = \{P_0, \ldots, P_n\}$ communicating by message-passing via pairwise, reliable channels (*authenticated channels* in the literature [5]). Technically, all transmitted messages are delivered uniquely, and with sender reliably identified.

At most t processes are *faulty* or *Byzantine* [18], and may display arbitrary, even malicious behavior, at any point in the execution. The actual behavior of Byzantine processes is defined by an *adversary*. Byzantine processes may execute the protocol correctly or incorrectly, at the discretion of the adversary. Processes that perform internal state transitions and message exchanges in strict accordance to the protocol for rounds 1 up to some r (inclusive) are called *non-faulty processes up to round* r, and are denoted by $\mathbb{G}^r$. Also, *faulty processes up to round* r are denoted by $\mathbb{B}^r = \mathbb{P} \setminus \mathbb{G}^r$. A non-faulty process up to any round $r \geq 1$ is called simply *non-faulty* or *correct*, which we denote by $\mathbb{G}$.

We model processes as state machines. The input value (resp. output value) of a non-faulty process P_i is written I_i (resp. O_i). Byzantine processes may have *apparent* inputs, denoted as above, and defined as one of the valid input values transmitted to other processes in the first round of computation. Each non-faulty process P_i has an internal state called *view*, which we denote by $\text{view}(P_i)$. In the beginning of the protocol, $\text{view}(P_i)$ is I_i. At any round r, any non-faulty process:

(1) sends its internal state to all other processes;
(2) receives the state information from other processes;
(3) concatenates that information to its own internal state.

After completing some number of iterations, each process applies a decision function δ to its current state in order to decide O_i. Thus, we assume that processes follow a *full-information* protocol [13].

For simplicity of notation, we define a round 0 where processes are simply assigned their inputs. Without losing generality, all processes are assumed non-faulty up to round 0: $\mathbb{G}^0 = \mathbb{P}$ and $\mathbb{B}^0 = \emptyset$. For any round $r \geq 0$, a *global state up to round* r formally specifies:

(1) the non-faulty processes up to round r; and
(2) the view of all non-faulty processes up to round r.

4 Topological Model

We now sketch the required concepts from combinatorial topology. For details, please refer to Munkres [20], Kozlov [17], or Herlihy *et al.* [11].

4.1 Basics

A *simplicial complex* $\mathcal{K}$ consists of a finite set V along with a collection of subsets of V closed under containment. An element of V is called a *vertex* of $\mathcal{K}$. The set of vertices of $\mathcal{K}$ is referred by $V(\mathcal{K})$. Each set in $\mathcal{K}$ is called a *simplex*, usually denoted by lower-case Greek letters: σ, τ, etc. The *dimension* $\dim(\sigma)$ of a simplex σ is $|\sigma| - 1$.

[1] Choosing $n+1$ processes rather than n simplifies the topological notation, but slightly complicates the computing notation. Choosing n processes has the opposite trade-off. We choose $n+1$ for compatibility with prior work.

A subset of a simplex is called a *face*. The collection of faces of σ with dimension exactly x is called $\text{Faces}^x(\sigma)$. A face τ of σ is called *proper* if $\dim(\tau) = \dim(\sigma) - 1$. We use "$k$-simplex" as shorthand for "k-dimensional simplex", analogously in "k-face." The dimension $\dim(\mathcal{K})$ of a complex is the maximal dimension of its simplexes, and a *facet* of $\mathcal{K}$ is any simplex having maximal dimension in $\mathcal{K}$. A complex is said *pure* if all facets have dimension $\dim(\mathcal{K})$. In a pure complex, we define the *codimension* of σ in $\mathcal{K}$, denoted $\text{codim}_{\mathcal{K}}(\sigma)$, as $\dim(\mathcal{K}) - \dim(\sigma)$. The set of simplexes of $\mathcal{K}$ having dimension at most ℓ is a subcomplex of $\mathcal{K}$, which is called *ℓ-skeleton* of $\mathcal{K}$, denoted by $\text{skel}^{\ell}(\mathcal{K})$.

4.2 Maps

Let $\mathcal{K}$ and $\mathcal{L}$ be complexes. A *vertex map f* carries vertices of $\mathcal{K}$ to vertices of $\mathcal{L}$. If f additionally carries simplexes of $\mathcal{K}$ to simplexes of $\mathcal{L}$, it is called a *simplicial* map. A *carrier map* Φ from $\mathcal{K}$ to $\mathcal{L}$ takes each simplex $\sigma \in \mathcal{K}$ to a subcomplex $\Phi(\sigma) \subseteq \mathcal{L}$, such that for all $\sigma, \tau \in \mathcal{K}$, we have $\Phi(\sigma \cap \tau) \subseteq \Phi(\sigma) \cap \Phi(\tau)$. If additionally $\Phi(\sigma \cap \tau) = \Phi(\sigma) \cap \Phi(\tau)$, we say that the carrier map is *strict*. A simplicial map $\phi : \mathcal{K} \to \mathcal{L}$ is *carried by the carrier map* $\Phi : \mathcal{K} \to 2^{\mathcal{L}}$ if, for every simplex $\sigma \in \mathcal{K}$, we have $\phi(\sigma) \subseteq \Phi(\sigma)$.

Although we defined simplexes and complexes in a purely combinatorial way, they can also be interpreted geometrically. An n-simplex can be identified with the convex hull of $(n + 1)$ affinely-independent points in the Euclidean space of appropriate dimension. This geometric realization can be extended to complexes. The point-set that underlies such *geometric complex* $\mathcal{K}$ is called the *polyhedron* of $\mathcal{K}$, denoted by $|\mathcal{K}|$. For any simplex σ, the *boundary* of σ, which we denote $\partial\,\sigma$, is the simplicial complex of $(\dim(\sigma) - 1)$-faces of σ. The *interior* of σ is defined as $\text{Int}\,\sigma = |\sigma| \setminus |\,\partial\,\sigma|$.

We can define simplicial/carrier maps between geometrical complexes. Given a simplicial map $\phi : \mathcal{K} \to \mathcal{L}$ (resp. carrier map $\Phi : \mathcal{K} \to 2^{\mathcal{L}}$), the polyhedrons of every simplex in $\mathcal{K}$ and $\mathcal{L}$ induce a continuous simplicial map $\phi_c : |\mathcal{K}| \to |\mathcal{L}|$ (resp. continuous carrier map $\Phi_c : |\mathcal{K}| \to |2^{\mathcal{L}}|$). We say ϕ (resp. ϕ_c) is carried by Φ if, for any $\sigma \in \mathcal{K}$, we have $|\phi(\sigma)| \subseteq |\Phi(\sigma)|$ (resp. $\phi_c(|\sigma|) \subseteq \Phi_c(|\sigma|)$).

4.3 Connectivity

In light of topology, two geometrical objects A and B are *homeomorphic* if, there is a bicontinuous map from A into B. In other words, there exists a continuous map between those objects, with a continuous inverse [21, 20].

▶ **Fact 1.** *[20] For any k-simplex σ, the boundary of σ is homeomorphic to a $(k-1)$-sphere, and σ is homeomorphic to a k-disk.*

We say that a simplicial complex $\mathcal{K}$ is *x-connected*, $x \geq 0$, if every continuous map of a subset of $|\mathcal{K}|$ homeomorphic to an x-sphere in $|\mathcal{K}|$ can be extended into a subset of $|\mathcal{K}|$ homeomorphic to an $(x + 1)$-disk in $|\mathcal{K}|$. In analogy, think of the extremes of a pencil as a 0-disk, and the pencil itself as a 1-sphere (the extension is possible if 0-connected); the rim of a coin as a 1-sphere, and the coin itself as a 2-disk (the extension is possible if 1-connected); the outer layer of a billiard ball as a 2-sphere, and the billiard ball itself as a 3-disk (the extension is possible if 2-connected). For us, (-1)-connected is understood as *non-empty*, and (-2)-connected or lower imposes no restriction.

4.4 Pseudospheres & Shellability

▶ **Definition 2.** Let $\mathbb{S} = \{(P_i, S_i) : P_i \in \mathbb{P}'\}$, where each S_i is an arbitrary set and $\mathbb{P}' \subseteq \mathbb{P}$. A *pseudosphere* $\Psi(\mathbb{P}', \mathbb{S})$ is a simplicial complex where $\sigma \in \Psi(\mathbb{P}', \mathbb{S})$ if $\sigma = \{(P_i, V_i) : P_i \in \mathbb{P}', V_i \in S_i\}$.

Essentially, a pseudosphere is a simplicial complex formed by independently assigning values to all the specified processes. If $S_i = S$ for all $P_i \in \mathbb{P}'$, we simply write $\Psi(\mathbb{P}', S)$.

▶ **Definition 3.** A pure, simplicial complex $\mathcal{K}$ is *shellable* if we can arrange the facets of $\mathcal{K}$ in a linear order $\phi_0 \ldots, \phi_t$ such that $\left(\bigcup_{0 \leq i < k} \phi_i\right) \cap \phi_k$ is a pure $(\dim(\phi_k) - 1)$-dimensional simplicial complex for all $0 < k \leq t$. We call the above linear order $\phi_0, \ldots, \phi_t$ a *shelling order*.

Intuitively, a simplicial complex is shellable if it can be built by gluing its x-simplexes along their $(x-1)$ faces only, where x is the dimension of the complex. Note that $\phi_0, \ldots, \phi_t$ is a shelling order if any $\phi_i \cap \phi_j$ $(0 \leq i < j \leq t)$ is contained in a $(\dim(\phi_k) - 1)$-face of ϕ_k $(0 \leq k < j)$. Hence,

$$\text{for any } i < j \text{ exists } k < j \text{ where } (\phi_i \cap \phi_j) \subseteq (\phi_k \cap \phi_j) \text{ and } |\phi_j \setminus \phi_k| = 1. \tag{1}$$

Shellability and pseudospheres are important tools to characterize connectivity in simplicial complexes. The following lemmas are proved in [12] and [11] (pp. 252–253).

▶ **Lemma 4.** *Any pseudosphere $\phi(\mathbb{P}', \mathbb{S})$ is shellable, considering arbitrary $\mathbb{S} = \{(P_i, S_i) : \forall P_i \in \mathbb{P}'\}$.*

▶ **Lemma 5.** *For any $k \geq 1$, if the simplicial complex $\mathcal{K}$ is shellable and $\dim(\mathcal{K}) \geq k$ then $\mathcal{K}$ is $(k-1)$-connected.*

4.5 Nerve Theorem

Let $\mathcal{K}$ be a simplicial complex with a *cover* $\{\mathcal{K}_i : i \in I\} = \mathcal{K}$, where I is a finite index set. The *nerve* $\mathcal{N}(\{\mathcal{K}_i : i \in I\})$ is the simplicial complex with vertexes I and simplexes $J \subseteq I$ whenever $\mathcal{K}_J = \bigcap_{j \in J} \mathcal{K}_j \neq \emptyset$. We can characterize the connectivity of $\mathcal{K}$ in terms of the connectivity of the intuitively simpler nerve of $\mathcal{K}$ with the next theorem.

▶ **Theorem 6** (Nerve Theorem [17, 3])**.** *If for any $J \subseteq I$ denoting a simplex of $\mathcal{N}(\{\mathcal{K}_i : i \in I\})$ (thus, $\mathcal{K}_J \neq \emptyset$) we have that $\mathcal{K}_J$ is $(k - |J| + 1)$-connected, then $\mathcal{K}$ is k-connected if and only if $\mathcal{N}(\{\mathcal{K}_i : i \in I\})$ is k-connected.*

4.6 Protocol Complexes

We represent the evolution of the global state of the system throughout the rounds by simplicial complexes that we call *protocol complexes*. The first, round-0 protocol complex $\mathcal{K}_0$, represents the possible inputs attributed to processes. After each round r, the round-r protocol complex $\mathcal{K}^r$ represents all possible global states of the system at round r. We also call $\mathcal{K}_0$ the *input complex*, also denoted $\mathcal{I}$.

▶ **Definition 7.** For $r \geq 0$, a *name-view* simplex σ is such that:
1. $\sigma = \{(P_i, \text{view}^r(P_i)) : \forall P_i \in \mathbb{G}^r\}$, where $\text{view}^r(P_i)$ denotes P_i's view at round r; and
2. if $(P_i, \text{view}^r(P_i))$ and $(P_j, \text{view}^r(P_j))$ are both in σ, then $P_i \neq P_j$.

Unless otherwise noted, all of our simplicial and carrier maps f are such that $\text{names}(\sigma) = \text{names}(f(\sigma))$, that is, they map between vertices associated with the same processes.

▶ **Definition 8.** For any name-view simplex σ, define

1. $\text{names}(\sigma) = \{P_i : \exists V \text{ such that } (P_i, V) \in \sigma\}$; and
2. $\text{views}(\sigma) = \{V_i : \exists P \text{ such that } (P, V_i) \in \sigma\}$.

The round-0 protocol complex $\mathcal{K}^0$ has name-view n-simplexes $\sigma_I = \{(P_i, I_i) : \forall P_i \in \mathbb{G}^0\}$, representing all the possible process inputs in the beginning of the protocol. The round-r protocol complex $\mathcal{K}^r$, for any $r \geq 0$, is defined as follows: if $\sigma \in \mathcal{K}^r$, then $\sigma = \{(P_i, \text{view}^r(P_i)) : \forall P_i \in \mathbb{G}^r\}$, representing a possible global state of the system for round r.

5 Connectivity Upper Bound

Informally, if the adversary displays Byzantine behavior early in the execution, then in a synchronous, full-information protocol, subsequent communication among the non-faulty processes can reveal the identities of the Byzantine processes, using simple techniques inspired from [2, 4, 25]. Instead, it behooves the adversary to postpone malicious behavior to the very last round, where it cannot be detected.

Say that non-faulty processes start the computation with inputs in $V = \{v_0, \dots, v_d\}$, *arbitrarily* assigned, with some $d \geq k$ and $t \geq k \geq 1$. To prove our upper bound, we show how the adversary can impose a particular *admissible* execution that preserves high connectivity in the protocol complex: by admissible, we mean an execution where at most t processes fail, with other processes behaving in accordance with the protocol.

Let $r = \lfloor t/k \rfloor$ and $m = t \bmod k$. We have r *crash rounds*, where in each round k processes fail by crashing, but display no Byzantine behavior. If $m > 0$, we have an extra *equivocation round*, where a single Byzantine process sends different views to different processes, causing extra confusion. This round-by-round execution produces a sequence of protocol complexes $\mathcal{K}^0, \dots, \mathcal{K}^{r+1}$, related by carrier maps $\mathcal{C}^i : \mathcal{K}^{i-1} \to 2^{\mathcal{K}^i}$, for $1 \leq i \leq r$, and $\mathcal{E} : \mathcal{K}^r \to 2^{\mathcal{K}^{r+1}}$.

$$\mathcal{K}^0 \xrightarrow{\mathcal{C}^1} \mathcal{K}^1 \dots \xrightarrow{\mathcal{C}^r} \mathcal{K}^r \underbrace{\xrightarrow{\mathcal{E}} \mathcal{K}^{r+1}}_{\text{only if } m>0}. \tag{2}$$

5.1 A Quick Background Detour: The Tools of the Trade

In each of the first r rounds, exactly k processes are failed by the adversary. The crash-failure carrier maps are defined as follows [12, 11]:

▶ **Definition 9.** For any $1 \leq i \leq r$, the crash-failure operator $\mathcal{C}^i : \mathcal{K}^{i-1} \to 2^{\mathcal{K}^i}$ is such that

$$\mathcal{C}^i(\sigma) = \bigcup_{\tau \in \text{Faces}^{n-ik}(\sigma)} \Psi(\text{names}(\tau); [\tau : \sigma]) \tag{3}$$

for any $\sigma \in \mathcal{K}^{i-1}$, with $[\tau : \sigma]$ denoting the set of simplexes μ where $\tau \subseteq \mu \subseteq \sigma$.

▶ **Definition 10.** A *q-connected* carrier map $\Phi : \mathcal{K} \to 2^{\mathcal{L}}$ is a strict carrier map such that, for all $\sigma \in \mathcal{K}$, $\dim(\Phi(\sigma)) > q - \text{codim}_{\mathcal{K}}(\sigma)$ and $\Phi(\sigma)$ is $(q - \text{codim}_{\mathcal{K}}(\sigma))$-connected.

▶ **Definition 11.** A *q-shellable* carrier map $\Phi : \mathcal{K} \to 2^{\mathcal{L}}$ is a strict carrier map such that, for all $\sigma \in \mathcal{K}$, $\dim(\Phi(\sigma)) > q - \text{codim}_{\mathcal{K}}(\sigma)$ and $\Phi(\sigma)$ is shellable.

After r rounds, note that $\mathcal{K}^r$ only contains simplexes with dimension exactly $n - rk$. In [12, 11], the following lemmas are proved:

▶ **Lemma 12.** *For $1 \leq i \leq r$, the operator $\mathcal{C}^i : \mathcal{K}^{i-1} \to 2^{\mathcal{K}^i}$ is a $(k-1)$-shellable carrier map.*

▶ **Lemma 13.** *If $\mathcal{M}^1, \ldots, \mathcal{M}^x$ are all q-shellable carrier maps, and $\mathcal{M}^{x+1}$ is a q-connected carrier map, the composition $\mathcal{M}^1 \circ \ldots \mathcal{M}^x \circ \mathcal{M}^{x+1}$ is a q-connected carrier map, for any $x \geq 0$.*

5.2 Byzantine Systems: Equivocation and Interpretation

After the crash-failure rounds, if $m > 0$ the adversary picks one of the remaining processes to behave maliciously at round $r+1$. This process, say P_b, may send different views to different processes (which is technically called *equivocation*), but, informally speaking, all views are "plausible." For example, two non-faulty processes P_i and P_j could be indecisive after round r on whether the global state is σ_1 or σ_2 in $\mathcal{K}^r$, while P_b, a Byzantine process, sends a state corresponding to σ_1 to P_i, and a state corresponding to σ_2 to P_j. The faulty process P_b *does not reveal* its Byzantine nature, yet it *promotes ambiguity* in the state information diffusion.

At the final round, when a non-faulty process receives the states sent from the other processes, it must decide correctly even if one other process equivocates. If the non-faulty process can receive simplexes σ_1 and σ_2, representing global states that differ in only one process's contribution (that is, $\dim(\sigma_1 \cap \sigma_2) = n - rk - 1$), then the *interpretation* of a message containing one such state must be the same as a message containing the other. We capture this notion using the *equivocation* operator, called $\mathcal{E}$, describing the behavior of a Byzantine process, coupled with an *interpretation* operator, called Interp, describing the required behavior of non-faulty processes. Informally, $\text{Interp}(\sigma_1) = \text{Interp}(\sigma_2)$ for processes in $\text{names}(\tau)$, where $\tau = \sigma_1 \cap \sigma_2$ with $\dim(\tau) = n - rk - 1$. Formally:

▶ **Definition 14.** For arbitrary simplexes σ_1 and σ_2 in $\mathcal{K}$, with $\dim(\mathcal{K}) = n - rk$, let $(P_i, \text{Interp}(\sigma_1)) = (P_i, \text{Interp}(\sigma_2))$ if and only if $\sigma_1 = \sigma_2$; **or** $P_i \in \text{names}(\tau)$ where $\tau = \sigma_1 \cap \sigma_2$ and $\dim(\tau) = n - rk - 1$.

▶ **Definition 15.** For any pure simplicial complexes $\mathcal{K}$ and $\mathcal{L}$ with $\dim(\mathcal{K}) \leq n - rk$ and $\mathcal{K} \supseteq \mathcal{L}$, the $\mathcal{K}$-equivocation operator $\mathcal{E}_{\mathcal{K}}$ is

$$\mathcal{E}_{\mathcal{K}}(\mathcal{L}) = \bigcup_{\tau \in \text{Faces}^{n-rk-1}(\mathcal{L})} \Psi(\text{names}(\tau); \{\text{Interp}(\sigma^*) : \sigma^* \in \mathcal{K}, \sigma^* \supset \tau\}). \tag{4}$$

Note that $\mathcal{E}_{\mathcal{K}}(\mathcal{L}) = \emptyset$ whenever $\dim(\mathcal{L}) < n - rk - 1$ or $\dim(\mathcal{K}) < n - rk$, and also that

$$\mathcal{E}_{\mathcal{K}}(\sigma) = \bigcup_{\tau \in \text{Faces}^{n-rk-1}(\sigma)} \Psi(\text{names}(\tau); \text{Interp}(\sigma)) \tag{5}$$

for any $\sigma \in \mathcal{K}$ with $\dim(\sigma) = n - rk$. For convenience of notation, define $\mathcal{E}_{\mathcal{K}}(\mathcal{K}) = \mathcal{E}(\mathcal{K})$.

5.3 Connectivity under Equivocation

Next, we investigate some technical properties of these constructions that allow us to prove that the final complex is $(k-1)$-connected.

▶ **Lemma 16.** *For any pure, shellable simplicial complex with $\dim(\mathcal{K}) \leq n - rk$, the $\mathcal{K}$-equivocation operator $\mathcal{E}_{\mathcal{K}}$ is a carrier map.*

Proof. Let $\tau \subseteq \sigma \in \mathcal{K}$. We show that $\mathcal{E}_{\mathcal{K}}(\tau) \subseteq \mathcal{E}_{\mathcal{K}}(\sigma)$. If $\dim(\tau) < n - rk - 1$ then $\mathcal{E}_{\mathcal{K}}(\tau) = \emptyset$ and $\mathcal{E}_{\mathcal{K}}(\tau) \subseteq \mathcal{E}_{\mathcal{K}}(\sigma)$ for any $\sigma \supseteq \tau \in \mathcal{K}$. Otherwise, if $\dim(\tau) = \dim(\sigma)$ then $\tau = \sigma$ and $\mathcal{E}_{\mathcal{K}}(\tau) = \mathcal{E}_{\mathcal{K}}(\sigma)$, as we assumed that $\sigma \supseteq \tau \in \mathcal{K}$. The remaining case is when $\dim(\tau) = n - rk - 1$ and $\dim(\sigma) = n - rk$, which makes $\mathcal{E}_{\mathcal{K}}(\tau) \subseteq \mathcal{E}_{\mathcal{K}}(\sigma)$ in light of Definition 15. ◀

Let $(\mathcal{C}^r \circ \mathcal{E})$ be the composite map such that $(\mathcal{C}^r \circ \mathcal{E})(\sigma) = \mathcal{E}_{\mathcal{C}^r(\sigma)}(\mathcal{C}^r(\sigma))$. While, for an arbitrary complex $\mathcal{K}$, $\mathcal{E}_{\mathcal{K}}$ is not a strict carrier map *per se*, we show in the following lemmas that $(\mathcal{C}^r \circ \mathcal{E})$ is a strict $(k-1)$-connected carrier map. Lemma 17 shows that $(\mathcal{C}^r \circ \mathcal{E})$ is a strict carrier map, and Lemma 18 shows that for any $\sigma \in \mathcal{K}^{r-1}$, $(\mathcal{C}^r \circ \mathcal{E})(\sigma)$ is $((k-1) - \mathrm{codim}_{\mathcal{K}^{r-1}}(\sigma))$-connected.

▶ **Lemma 17.** *$(\mathcal{C}^r \circ \mathcal{E})$ is a strict carrier map.*

Proof. Consider $\sigma, \tau \in \mathcal{K}^{r-1}$, with $\mathcal{L} = \mathcal{C}^r(\sigma)$ and $\mathcal{M} = \mathcal{C}^r(\tau)$. Both $\mathcal{L}$ and $\mathcal{M}$ are pure, shellable simplicial complexes with dimension $n-rk$ (Definition 9 and Lemma 12). Therefore, both the $\mathcal{L}$-equivocation and $\mathcal{M}$-equivocation operators are well-defined. Also, $\mathcal{C}^r$ is a strict carrier map, hence $\mathcal{L} \cap \mathcal{M} = \mathcal{C}^r(\sigma) \cap \mathcal{C}^r(\tau) = \mathcal{C}^r(\sigma \cap \tau)$. Note that $\mathcal{L} \cap \mathcal{M} = \mathcal{C}^r(\sigma \cap \tau)$, if not empty, is a pure, shellable simplicial complex with dimension $n - rk$. Therefore, the $(\mathcal{L} \cap \mathcal{M})$-equivocation operator is well-defined.

First, we show that $\mathcal{E}(\mathcal{L}) \cap \mathcal{E}(\mathcal{M}) \subseteq \mathcal{E}(\mathcal{L} \cap \mathcal{M})$, which implies one direction of our equality:

$$\mathcal{E}(\mathcal{C}^r(\sigma)) \cap \mathcal{E}(\mathcal{C}^r(\tau)) \subseteq \mathcal{E}(\mathcal{C}^r(\sigma) \cap \mathcal{C}^r(\tau)) = \mathcal{E}(\mathcal{C}^r(\sigma \cap \tau)).$$

For clarity, let $F(\mathcal{K}) = \mathrm{Faces}^{n-rk-1}(\mathcal{K})$. Then,

$$\mathcal{E}(\mathcal{L}) \cap \mathcal{E}(\mathcal{M}) = \bigcup_{\mu \in F(\mathcal{L})} \mathcal{E}_{\mathcal{L}}(\mu) \quad \cap \quad \bigcup_{\nu \in F(\mathcal{M})} \mathcal{E}_{\mathcal{M}}(\nu) = \bigcup_{\substack{\mu \in F(\mathcal{L}) \\ \nu \in F(\mathcal{M})}} \mathcal{E}_{\mathcal{L}}(\mu) \cap \mathcal{E}_{\mathcal{M}}(\nu).$$

For arbitrary $\mu \in F(\mathcal{L})$ and $\nu \in F(\mathcal{M})$, if $\mathcal{E}_{\mathcal{L}}(\mu) \cap \mathcal{E}_{\mathcal{M}}(\nu) \neq \emptyset$, consider two cases:

1. μ and ν are proper faces of $\phi \in (\mathcal{L} \cap \mathcal{M})$. In this case,

   $$\mathcal{E}_{\mathcal{L}}(\mu) \cap \mathcal{E}_{\mathcal{M}}(\nu) = \Psi(\mathrm{names}(\mu) \cap \mathrm{names}(\nu); \mathrm{Interp}(\phi)),$$

   which is inside $\mathcal{E}_{\mathcal{L}\cap\mathcal{M}}(\phi) \subseteq \mathcal{E}_{\mathcal{L}\cap\mathcal{M}}(\mathcal{L} \cap \mathcal{M})$.
2. Otherwise, $\mu \subset \phi_1 \in \mathcal{L}$ or $\nu \subset \phi_2 \in \mathcal{M}$. In this case,

   $$\mathcal{E}_{\mathcal{L}}(\mu) \cap \mathcal{E}_{\mathcal{M}}(\nu) = \Psi(\mathrm{names}(\mu) \cap \mathrm{names}(\nu); \mathrm{Interp}(\phi_1) \cap \mathrm{Interp}(\phi_2)).$$

   By Definition 14, the above is non-empty only when $\mathrm{Interp}(\phi_1) = \mathrm{Interp}(\alpha)$ with $\alpha \in \mathcal{L}$, $\mathrm{Interp}(\phi_2) = \mathrm{Interp}(\beta)$ with $\beta \in \mathcal{M}$, and there exists a non-empty set $\mathbb{P}'$ such that $\mathbb{P}' \subseteq \mathrm{names}(\mu) \cap \mathrm{names}(\nu) \subseteq \mathrm{names}(\gamma)$, where $\gamma = \alpha \cap \beta$ with $\dim(\gamma) = n - rk - 1$. Let $\mathbb{P}''$ be a maximal $\mathbb{P}'$ satisfying such condition. Note that $\gamma \in (\mathcal{L} \cap \mathcal{M})$, so $(\mathcal{L} \cap \mathcal{M}) \neq \emptyset$. Since $(\mathcal{L} \cap \mathcal{M})$ is non-empty, it is pure, shellable with dimension $n - rk$, there must exist a simplex $\gamma' \supset \gamma$ with dimension $n - rk$. Moreover, $\mathrm{Interp}(\gamma') = \mathrm{Interp}(\alpha) = \mathrm{Interp}(\phi_1)$ and $\mathrm{Interp}(\gamma') = \mathrm{Interp}(\beta) = \mathrm{Interp}(\phi_2)$ for processes in $\mathrm{names}(\gamma)$, given the definition of Interp. In conclusion, we have $\mathcal{E}_{\mathcal{L}}(\mu) \cap \mathcal{E}_{\mathcal{M}}(\nu) = \Psi(\mathbb{P}''; \mathrm{Interp}(\gamma')) \subseteq \Psi(\mathrm{names}(\gamma); \mathrm{Interp}(\gamma'))$, which is inside $\mathcal{E}_{\mathcal{L}\cap\mathcal{M}}(\gamma') \subseteq \mathcal{E}_{\mathcal{L}\cap\mathcal{M}}(\mathcal{L} \cap \mathcal{M})$.

In the other direction, we have $\mathcal{E}(\mathcal{L}\cap\mathcal{M}) \overset{\text{def}}{=} \mathcal{E}_{\mathcal{L}\cap\mathcal{M}}(\mathcal{L}\cap\mathcal{M}) \subseteq \mathcal{E}_{\mathcal{L}}(\mathcal{L}\cap\mathcal{M}) \subseteq \mathcal{E}_{\mathcal{L}}(\mathcal{L}) \overset{\text{def}}{=} \mathcal{E}(\mathcal{L})$, since

(i) $\mathcal{E}_{\mathcal{L}\cap\mathcal{M}}(\mathcal{X}) \subseteq \mathcal{E}_{\mathcal{L}}(\mathcal{X})$ for any $\mathcal{X} \subseteq \mathcal{L} \cap \mathcal{M}$ (Definition 15); and
(ii) $\mathcal{E}_{\mathcal{L}}$ is a carrier map (Lemma 16).

The same argument proves that $\mathcal{E}(\mathcal{L} \cap \mathcal{M}) \subseteq \mathcal{E}(\mathcal{M})$, and therefore $\mathcal{E}(\mathcal{L} \cap \mathcal{M}) \subseteq \mathcal{E}(\mathcal{L}) \cap \mathcal{E}(\mathcal{M})$. ◀

▶ **Lemma 18.** *For any $\sigma \in \mathcal{K}^{r-1}$, $\mathcal{E}(\mathcal{C}^r(\sigma))$ is $((k-1) - \operatorname{codim}_{\mathcal{K}^{r-1}}(\sigma))$-connected.*

Proof. Consider $\sigma \in \mathcal{K}^{r-1}$ with $\operatorname{codim}_{\mathcal{K}^{r-1}}(\sigma) \leq k$. By Lemma 12, $\mathcal{M} = \mathcal{C}^r(\sigma)$ is a pure, shellable simplicial complex with $\dim(\mathcal{M}) = n - rk = d$. By Definition 15, $\mathcal{E}(\mathcal{M})$ is well-defined and $\dim(\mathcal{E}(\mathcal{M})) = n - rk - 1 = d'$. Note that $d' \geq n - t \geq 2t \geq 2k$, since $n + 1 > 3t$ and $t \geq k$.

First, we show that $\mathcal{E}(\mathcal{M})$ is "highly-connected" – that is, $(2k-1)$-connected. We proceed by induction on $\mu_0 \ldots \mu_\ell$, a shelling order of facets of $\mathcal{M}$.

Base. We show that $\mathcal{E}_{\mathcal{M}}(\mu_0)$ is $(2k-1)$-connected. Considering Definition 15, we have that $\mathcal{E}_{\mathcal{M}}(\mu_0) = \mathcal{E}_{\mathcal{M}}(\tau_0) \cup \ldots \cup \mathcal{E}_{\mathcal{M}}(\tau_d)$, with $\tau_0 \ldots \tau_d$ being all the proper faces of μ_0. Consider the cover $\{\mathcal{E}_{\mathcal{M}}(\tau_i) : 0 \leq i \leq d\}$ of $\mathcal{E}_{\mathcal{M}}(\mu_0)$, and its associated nerve $\mathcal{N}(\{\mathcal{E}_{\mathcal{M}}(\tau_i) : 0 \leq i \leq d\})$. For any index set $J \subseteq I = \{0 \ldots d\}$, let

$$\mathcal{K}_J = \bigcap_{j \in J} \mathcal{E}_{\mathcal{M}}(\tau_j) = \Psi(\bigcap_{j \in J} \operatorname{names}(\tau_j); \operatorname{Interp}(\mu_0))$$

For any J with $|J| \leq d$, we have $\cap_{j \in J} \operatorname{names}(\tau_j) \neq \emptyset$, making $\mathcal{K}_J$ a non-empty pseudosphere with dimension $d' - |J| + 1 \geq 2k - |J| + 1$. So, $\mathcal{K}_J$ is $((2k-1) - |J| + 1)$-connected by Lemmas 4 and 5. The nerve is hence the $(d-1)$-skeleton of I, which is $(d-2) = (d'-1) \geq (2k-1)$-connected. By the Nerve Theorem, $\mathcal{E}_{\mathcal{M}}(\mu_0)$ is also $(2k-1)$-connected.

IH. Assume that $\mathcal{Y} = \cup_{0 \leq y < x} \mathcal{E}_{\mathcal{M}}(\mu_y)$ is $(2k-1)$ connected, and let $\mathcal{X} = \mathcal{E}_{\mathcal{M}}(\mu_x)$. We must show that $\mathcal{Y} \cup \mathcal{X} = \cup_{0 \leq y \leq x} \mathcal{E}(\mu_y)$ is $(2k-1)$-connected. Note that $\mathcal{X}$ is $(2k-1)$-connected by an argument identical to the one above for the base case $\mathcal{E}_{\mathcal{M}}(\mu_0)$. Besides,

$$\mathcal{Y} \cap \mathcal{X} = \left(\bigcup_{0 \leq y < x} \mathcal{E}_{\mathcal{M}}(\mu_y) \right) \cap \mathcal{E}_{\mathcal{M}}(\mu_x) = \bigcup_{0 \leq y < x} (\mathcal{E}_{\mathcal{M}}(\mu_y) \cap \mathcal{E}_{\mathcal{M}}(\mu_x)) \stackrel{\star}{=} \bigcup_{i \in S} \mathcal{E}_{\mathcal{M}}(\tau_i),$$

where $i \in S$ is such that $(\cup_{0 \leq y < x} \mu_y) \cap \mu_x = \cup_{i \in S} \tau_i$. The set S is well-defined since $\mathcal{M}$ is shellable. The step $(\star)$ holds because:

1. $\mathcal{Y} \cap \mathcal{X}$ must include at least $\bigcup_{i \in S} \mathcal{E}_{\mathcal{M}}(\tau_i)$; and
2. $\mathcal{E}_{\mathcal{M}}(\mu_y) \cap \mathcal{E}_{\mathcal{M}}(\mu_x) \neq \emptyset$ only if $\psi = \Psi(\operatorname{names}(\mu_y \cap \mu_x); \operatorname{Interp}(\mu_x))$ exists, the latter inside $\psi' = \Psi(\operatorname{names}(\tau_j); \operatorname{Interp}(\mu_x))$ for some $j \in S$, or we contradict the fact that $\mathcal{M}$ is shellable.

Using an argument identical to the one for $\mathcal{E}_{\mathcal{M}}(\mu_0)$, yet considering the cover $\{\mathcal{E}_{\mathcal{M}}(\tau_i) : i \in S\}$, the nerve of $\mathcal{X} \cap \mathcal{Y}$ is either the $(d-1)$-skeleton of S (if $S = \{0 \ldots d\}$) or the whole simplex S (otherwise). By the Nerve Theorem, $\cup_{i \in S} \mathcal{E}_{\mathcal{M}}(\tau_i)$ is $(2k-1)$-connected. Once again, using the Nerve Theorem, since $\mathcal{Y}$ is $(2k-1)$-connected, $\mathcal{X}$ is $(2k-1)$-connected, and $\mathcal{Y} \cap \mathcal{X}$ is $(2k-1)$-connected, we have that $\mathcal{Y} \cup \mathcal{X}$ is $(2k-1)$-connected.

While the equivocation operator yields high connectivity $(2k-1)$ in the pseudosphere $\mathcal{C}^r(\sigma)$, the *composition* of $\mathcal{C}^r$ and $\mathcal{E}_{\mathcal{C}^r(\sigma)}(\mathcal{C}^r(\sigma))$ limits the connectivity to $(k-1)$, since the former map is only defined for simplexes with codimension $\leq k$. Formally, as $\mathcal{C}^r(\sigma) \neq \emptyset$ for any simplex $\sigma \in \mathcal{K}^{r-1}$ with $\operatorname{codim}_{\mathcal{K}^{r-1}}(\sigma) \leq k$, we have that $\mathcal{E}(\mathcal{C}^r(\sigma))$ is $((k-1) - \operatorname{codim}_{\mathcal{K}^{r-1}}(\sigma))$-connected. ◀

From Lemmas 17 and 18, we conclude the following.

▶ **Corollary 19.** *$(\mathcal{C}^r \circ \mathcal{E})$ is a $(k-1)$-connected carrier map.*

▶ **Theorem 20.** *An adversary can keep the protocol complex of a Byzantine synchronous system $(k-1)$-connected for $\lceil t/k \rceil$ rounds.*

Proof. If $m = 0$, $t \bmod k = 0$, and the adversary runs only the crash rounds failing k processes each time, for $r = \lfloor t/k \rfloor = \lceil t/k \rceil$ consecutive rounds. We have the following scenario:

$$(\mathcal{C}^1 \circ \ldots \circ \mathcal{C}^r)(\sigma).$$

Since $\mathcal{C}^i : \mathcal{K}^{i-1} \to 2^{\mathcal{K}^i}$ is a $(k-1)$-shellable carrier map for $1 \leq i \leq r$ (Lemma 12), the composition $(\mathcal{C}^1 \circ \ldots \circ \mathcal{C}^r)$ is a $(k-1)$-connected carrier map for any facet $\sigma \in \mathcal{I}$ (Lemma 13).

If $m > 0$, the adversary performs r crash rounds (failing k processes each time), followed by the extra equivocation round. We have the following scenario:

$$(\mathcal{C}^1 \circ \ldots \circ \mathcal{C}^{r-1} \circ (\mathcal{C}^r \circ \mathcal{E}))(\sigma). \tag{6}$$

Since $\mathcal{C}^i : \mathcal{K}^{i-1} \to \mathcal{K}^i$ is a $(k-1)$-shellable carrier map for $1 \leq i \leq r-1$ (Lemma 12), and $(\mathcal{C}^r \circ \mathcal{E})$ is a $(k-1)$-connected carrier map (Corollary 19), we have that the composition above $(\mathcal{C}^1 \circ \ldots \circ \mathcal{C}^{r-1} \circ (\mathcal{C}^r \circ \mathcal{E}))$ is a $(k-1)$-connected carrier map for any facet $\sigma \in \mathcal{I}$ (Lemma 13). ◀

6 k-Set Agreement and Lower Bound

The k-set agreement problem [7], is a fundamental task having important associations with protocol complex connectivity. In Byzantine systems, it can be difficult to characterize the input of a faulty process, since this process can ignore its "prescribed" input and behave as having a different one. This intrinsically leads to many alternative formulations for the problem in Byzantine systems [9]. In our algorithm, for each Byzantine process, we can commit to at most a single value transmitted as input. We define such value as the *apparent* input value of the Byzantine process. In our adopted formulation, each non-faulty process P_i starts with *any* value I_i from $V = \{v_0, \ldots, v_d\}$, with $d \geq k$ and $t \geq k \geq 1$, and finishes with a value O_i from V, respecting:

1. **Agreement.** At most k values are decided: $|\{O_i : P_i \in \mathbb{G}\}| \leq k$.
2. **Validity.** For any non-faulty process P_i, the output O_i is the input value of one of the participating processes.
3. **Termination.** The protocol finishes in a finite number of rounds.

The k-set agreement problem and connectivity are closely related. Lemma 21, proved in Appendix A, shows that no solution is possible for k-set agreement with a $(k-1)$-connected protocol complex, which, as seen in Section 5, can occur at least until round $\lceil t/k \rceil$.

▶ **Lemma 21.** *If, starting $\sigma \in \mathcal{I}$, the protocol complex $\mathcal{P}(\sigma)$ is $(k-1)$-connected, then no decision function δ solves the k-set agreement problem.*

Proof. Please refer to Appendix A. ◀

We now present a simple k-set agreement algorithm for Byzantine synchronous systems, running in $\lceil t/k \rceil + 1$ rounds. The procedure requires a relatively large number of processes compared to t: we assume $n + 1 \geq k(3t + 1)$. The procedure was designed with the purpose of tightening the connectivity lower bound, favoring simplicity over the optimality on the number of processes.

Algorithm 1 P_x.Agree(I)

if $k = 1$ **then**
  return `Decision`(Multiset(Cont(p) output by consensus algorithm))
Cont(w) $\leftarrow \perp$ for all $w \in T$
Cont(λ) $\leftarrow I$ ▷ Gossip
for $\ell : 1$ **to** $\lceil t/k \rceil + 1$ **do**
  send($S_x^{\ell-1} = \{(w, \text{Cont}(w)) : |w| = \ell - 1\}$)
  upon recv($S_y^{\ell-1} = \{(w, v) : |w| = \ell - 1, v \in V \cup \{\perp\}\}$) from P_y
    Cont(wP_y) $\leftarrow v$ for all $(w, v) \in S_y^{\ell-1}$
  end upon
$\mathbb{P}' \leftarrow \{P_i : P_i$ has a quorum$\}$ ▷ Validation
if $|\mathbb{P}'| = (n+1) - t$ **then**
  Apply completion rule for all wb where $b \in \mathbb{P} \setminus \mathbb{P}'$ and $|wb| = \lceil t/k \rceil$
$g \leftarrow$ any g such that $T(g)$ is pivotal ▷ Decision
for $\ell : \lceil t/k \rceil - 1$ **to** 1 **do**
  Apply consensus rule for all non-validated wb where $b \in \mathbb{P}(g)$ and $|wb| = \ell$
return `Decision`(Multiset(Cont(p) : $p \in T(g)$))

Non-faulty processes initially execute a *gossip phase* for $\lceil t/k \rceil + 1$ rounds, followed by a *validation phase*, and a *decision phase*, where the output is chosen. Define $R = \lceil t/k \rceil$, and consider the following tree, where nodes are labeled with words over the alphabet $\mathbb{P}$. The root node is labeled as λ, which represents an empty string. Each node w such that $0 \leq |w| \leq R$ has $n + 1$ child nodes labeled wp for all $p \in \mathbb{P}$. Any non-faulty process P_i maintains such tree, denoted T_i.

6.1 The Gossip Phase

For each of the trees maintained by the processes, as discussed above, all nodes w are associated with the value $\text{Cont}_p(w)$, called the *contents* of w. The meaning of those trees is well-known [1]: after the gossip phase, if node $w = p_1 \ldots p_x$ is such that $\text{Cont}_p(w) = v$, then p_x told that p_{x-1} told that $\ldots$ p_1 had input v to p. The special value $\perp$ represents an absent input. We omit the subscript p when the process is implied or arbitrary. We divide the processes into k disjoint groups: $\mathbb{P}(g) = \{P_x \in \mathbb{P} : x = g \bmod k\}$, for $0 \leq g < k$. For any tree T, we call $T(g)$ the subtree of T having only nodes $wp \in T$ such that $p \in \mathbb{P}(g)$.

6.2 The Validation Phase

In the validation phase, if we have a set $\mathbb{Q}$ containing $(n+1) - t$ processes that acknowledge all messages coming from process p (making sure that $p \in \mathbb{Q}$) in all rounds $1 \leq r \leq R$, we call such set the *quorum* of p, denoted Quorum(p). Formally, Quorum(p) $= \mathbb{Q} \subseteq \mathbb{P}$ such that $p \in \mathbb{Q}$, $|\mathbb{Q}| \geq (n+1) - t$, and $q \in \mathbb{Q}$ whenever Cont(wp) $= v$ implies Cont(wpq) $= v$, for any wp with $1 \leq |wp| \leq R$. It should be clear that every non-faulty process has a quorum containing at least all other non-faulty processes. If a process p has a quorum as seen by process $P_i \in \mathbb{G}$, we say that wp has been *validated* on P_i, for any wp with $1 \leq |wp| \leq R$. We also say that p has been validated on P_i in this case. Note that in our definition either all entries wp with $1 \leq |wp| \leq R$ are validated, or none is. Lemma 22 shows that validated entries are unique across non-faulty processes.

▶ **Lemma 22.** *If p has been validated on non-faulty processes P_i and P_j, then* $\mathrm{Cont}_i(wp) = \mathrm{Cont}_j(wp)$ *for any* $0 \leq |w| < R$.

Proof. If p has been validated on $P_i \in \mathbb{G}$, then $\mathrm{Cont}_i(wp) = v$ implies $\mathrm{Cont}_i(wpq) = v$ for $(n+1) - t$ different processes $q \in \mathbb{Q}_i$, and $\mathrm{Cont}_j(wp) = v'$ implies $\mathrm{Cont}_j(wpq) = v'$ for $(n+1) - t$ different processes $q \in \mathbb{Q}_j$, for any $0 \leq |w| < R$. As we have at most t non-faulty processes and $n+1 > 3t$, $|\mathbb{Q}_i \cap \mathbb{Q}_j| \geq (n+1) - 2t > t+1$, containing at least one non-faulty process that, in contradiction, would be broadcasting values consistently in its run. Hence, $v = \mathrm{Cont}_i(wp)$ and $v' = \mathrm{Cont}_j(wp)$ must be identical. ◀

6.3 The Decision Phase

In the decision phase, if we see t processes without a quorum, we have technically identified all non-faulty processes $\mathbb{B}$. In this case, we fill R-th round values of any $b \in \mathbb{B}$ using the *completion* rule: we make $\mathrm{Cont}(wb) = v$ if we have $(n+1) - 2t$ processes $\mathbb{G}' \subseteq \mathbb{G}$ where $\mathrm{Cont}(wbg) = v$ for any $g \in \mathbb{G}'$ and $|wb| = R$. If a process b has its R-round values completed as above in process $P_i \in \mathbb{G}$, we say that wb has been *completed* on P_i for any $|wb| = R$. Lemma 23 shows that completed entries are identical and consistent with validated entries across non-faulty processes. (Intuitively, the completion rule was done over identical values from correct processes.)

▶ **Lemma 23.** *If wp has been completed or validated on a non-faulty process P_i, and wp has been completed on a non-faulty process P_j, then* $\mathrm{Cont}_i(wp) = \mathrm{Cont}_j(wp)$.

Proof. Say wp has been validated on P_i and completed in P_j. Since wp has been validated on P_i, $\mathrm{Cont}_i(wp) = v$ implies $\mathrm{Cont}_i(wpq) = v$ for $(n+1)-t$ different processes $q \in \mathbb{Q}$. When P_j applies the completion rule on wp, we must have $\mathrm{Cont}_j(wpq) = v$ for $(n+1)-2t$ different processes $q \in \mathbb{G}$, as we have at most t faulty processes. Therefore, $\mathrm{Cont}_i(wp) = \mathrm{Cont}_j(wp)$.

If wp has been completed on all non-faulty processes, they all have identified t faulty processes, and the completion rule is performed over identical entries associated with non-faulty processes. Therefore, $\mathrm{Cont}_i(wp) = \mathrm{Cont}_j(wp)$ in this case as well. ◀

▶ **Definition 24.** We define a *pivotal subtree* as follows:

1. If there exists a subtree $T(g)$ with less than $\lceil t/k \rceil$ non-validated processes, define this subtree as pivotal;
2. Otherwise, we identified $k \cdot \lceil t/k \rceil \geq t$ Byzantine processes, so we apply the completion rule consistently to R-round values in $T(0)$, and define $T(0)$ as pivotal instead.

A pivotal subtree, therefore, must exist according to Definition 24. For that subtree, any sequence $p_1, (p_1p_2), \ldots, (p_1p_2 \ldots p_x)$, with $p_1 \neq \ldots \neq p_x$, has size $x < R = \lceil t/k \rceil$. As we see further ahead, this will allow us to suitably perform consensus over consistent values.

We first highlight that, essentially, our algorithm is separating the possible chains of unknown values across disjoint process groups, which either forces one of these chains to be smaller than $R = \lceil t/k \rceil$, or reveals all faulty processes, giving us the ability to perform the completion rule in a consistent way. This fundamental tradeoff underlies our algorithm, and ultimately explains *why* the $\lceil t/k \rceil$ connectivity bound is tight for relatively large numbers of n compared to t.

6.3.1 The Consensus Rule

Denote the set of processes in the word w as $\mathrm{SetProc}(w)$. For any non-validated wb with $b \in \mathbb{P}(g)$ in a pivotal subtree $T(g)$, where $1 \leq |wb| < R$, we establish consensus on $\mathrm{Cont}(wb)$. We apply the *consensus* rule: $\mathrm{Cont}(wb) = v$ if the majority of processes in $\mathbb{P}(g) \setminus \mathrm{SetProc}(wb)$

is such that $wbp = v$. This rule is applied first to entries labeled wb where $|wb| = R - 1$, and then moving upwards (please refer to Algorithm 1). Lemma 25 shows that the consensus rule indeed establishes consensus across non-faulty processes that identify $T(g)$ as the pivotal subtree.

▶ **Lemma 25.** *For any two-non-faulty processes P_i and P_j that applied the consensus rule on a pivotal subtree $T(g)$, with $0 \leq g < k$, we have that $\mathrm{Cont}_i(p) = \mathrm{Cont}_j(p)$ for any $p \in \mathbb{P}(g)$.*

Proof. Consider a non-faulty process P_i establishing the value of $\mathrm{Cont}_i(wp)$ with the consensus rule. Define $\mathrm{SetCons}(wp) = \mathbb{P}(g) \setminus \mathrm{SetProc}(wp)$ for any $wp \in T(g)$ with $|wp| < R$, noting that $|\mathrm{SetCons}(wp)| \geq 2t + 2$ as $|\mathbb{P}(g)| \geq 3t + 1$ and $|wp| < t$.

There are two possible cases:

1. If wp has been validated at a non-faulty process P_j with $\mathrm{Cont}_j(wp) = v$, at most t values from $S_i = \mathrm{Multiset}(\mathrm{Cont}_i(wpq) : q \in \mathrm{SetCons}(wp))$ will be different than v. Hence, there will always be a majority of values in S_i that will contain v, because $|S_i| \geq 2t + 2$.
2. Otherwise, if wp has not been validated at any non-faulty process, all $\mathrm{Cont}(wp)$ values are being calculated over consistent values, by Lemma 23, which makes all non-faulty processes establish $\mathrm{Cont}(wp)$ consistently with the consensus rule. ◀

▶ **Theorem 26.** *Algorithm 1 solves k-set agreement in $\lceil t/k \rceil + 1$ rounds for $n+1 > k(3t+1)$.*

Proof. Termination is trivial, as we execute exactly $R = \lceil t/k \rceil + 1$ rounds. By Lemma 25, each pivotal subtree yields a unique decision value. As we have at most k pivotal subtrees identified across non-faulty processes, up to k values are possibly decided across non-faulty processes. ◀

7 Conclusion

In Byzantine synchronous systems, the protocol complex can remain $(k - 1)$-connected for $\lceil t/k \rceil$ rounds, potentially *one* more round than in crash-failure systems. We conceive a combinatorial operator modeling the ability of Byzantine processes to *equivocate* without revealing their Byzantine nature, just after $\lfloor t/k \rfloor$ rounds of crash failures. We compose this operator with the regular crash-failure operators, extending $(k - 1)$-connectivity up to $\lceil t/k \rceil$ rounds. We tighten this bound, at least when n is relatively large compared to t, via a full-information protocol that solves a formulation of k-set agreement.

It may be surprising that Byzantine failures impose only *one* additional synchronous round over the crash-failure model, and *at most that* in our setting, where inputs are arbitrarily attributed to processes, and the number of processes is at least $k(3t + 1)$. In terms of solvability vs. number of rounds, the penalty for moving from crash to Byzantine failures can thus be *quite limited*. Previous work has hinted this possibility operationally, since

(i) in synchronous systems where n is large enough compared to t, we can simulate crash failures on Byzantine systems with a 1-round delay [2]; and
(ii) techniques similar to the reliable broadcast of [4, 25] deal with the problem of Byzantine equivocation, also with a 1-round delay.

This extra round is crucial – but enough – to limit the impact of Byzantine behavior in rather usual operational settings.

The algorithm that matches the connectivity bound was designed to separate chains of unresolved values, such that we suitably limit their size, or force the adversary to reveal the identity of all faulty processes. The prospect of an algorithm that applies similar ideas, however with better resilience, is a thought-provoking perspective for future work.

References

1. Hagit Attiya and Jennifer Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*. John Wiley Interscience, 2nd edition, March 2004.
2. Rida A. Bazzi and Gil Neiger. Simplifying fault-tolerance: Providing the abstraction of crash failures. *Journal of the ACM*, 48(3):499–554, May 2001. `doi:10.1145/382780.382784`.
3. A. Björner. Topological methods. In R. L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, volume 2, pages 1819–1872. MIT Press, Cambridge, MA, USA, December 1995. URL: `http://dl.acm.org/citation.cfm?id=233228.233246`.
4. G. Bracha. Asynchronous Byzantine agreement protocols. *Information and Computation*, 75(2):130–143, November 1987.
5. Christian Cachin, Rachid Guerraoui, and Luís Rodrigues. *Introduction to Reliable and Secure Distributed Programming*. Springer, 2 edition, February 2011.
6. Armando Castañeda, Yannai A. Gonczarowski, and Yoram Moses. Brief announcement: Pareto optimal solutions to consensus and set consensus. In *Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing*, PODC'13, pages 113–115, New York, NY, USA, 2013. ACM. `doi:10.1145/2484239.2484280`.
7. Soma Chaudhuri. More choices allow more faults: set consensus problems in totally asynchronous systems. *Information and Computation*, 105(1):132–158, July 1993.
8. Soma Chaudhuri, Maurice Herlihy, Nancy A. Lynch, and Mark R. Tuttle. Tight bounds for k-set agreement. *Journal of the ACM*, 47(5):912–943, September 2000. `doi:10.1145/355483.355489`.
9. Roberto de Prisco, Dahlia Malkhi, and Michael Reiter. On k-set consensus problems in asynchronous systems. *IEEE Transactions on Parallel and Distributed Systems*, 12(1):7–21, January 2001. `doi:10.1109/71.899936`.
10. Eli Gafni, Petr Kuznetsov, and Ciprian Manolescu. A generalized asynchronous computability theorem. In *Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing*, PODC'14, pages 222–231, New York, NY, USA, 2014. ACM. `doi:10.1145/2611462.2611477`.
11. Maurice Herlihy, Dmitry Kozlov, and Sergio Rajsbaum. *Distributed Computing Through Combinatorial Topology*. Morgan Kaufmann, December 2013.
12. Maurice Herlihy and Sergio Rajsbaum. Concurrent computing and shellable complexes. In Nancy Lynch and Alexander Shvartsman, editors, *Distributed Computing*, volume 6343 of *Lecture Notes in Computer Science*, pages 109–123. Springer Berlin / Heidelberg, 2010. `doi:10.1007/978-3-642-15763-9_10`.
13. Maurice Herlihy, Sergio Rajsbaum, and Mark Tuttle. An axiomatic approach to computing the connectivity of synchronous and asynchronous systems. *Electronic Notes in Theoretical Computer Science*, 230(0):79–102, March 2009. `doi:10.1016/j.entcs.2009.02.018`.
14. Maurice Herlihy, Sergio Rajsbaum, and Mark R. Tuttle. Unifying synchronous and asynchronous message-passing models. In *Proceedings of the Seventeenth Annual ACM Symposium on Principles of Distributed Computing*, PODC'98, pages 133–142, New York, NY, USA, 1998. ACM. `doi:10.1145/277697.277722`.
15. Maurice Herlihy and Nir Shavit. The asynchronous computability theorem for t-resilient tasks. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing*, STOC'93, pages 111–120, New York, NY, USA, 1993. ACM. `doi:10.1145/167088.167125`.
16. Maurice Herlihy and Nir Shavit. The topological structure of asynchronous computability. *Journal of the ACM*, 46(6):858–923, November 1999.
17. Dimitry N. Kozlov. *Combinatorial Algebraic Topology*, volume 21 of *Algorithms and Computation in Mathematics*. Springer, 1st edition, October 2007. `doi:10.1007/978-3-540-71962-5`.

18 Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine generals problem. *ACM Transaction on Programming Languages and Systems*, 4(3):382–401, July 1982. `doi:10.1145/357172.357176`.

19 Hammurabi Mendes, Christine Tasson, and Maurice Herlihy. Distributed computability in Byzantine asynchronous systems. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC'14, pages 704–713, New York, NY, USA, 2014. ACM. `doi:10.1145/2591796.2591853`.

20 James Munkres. *Elements of Algebraic Topology*. Prentice Hall, 2nd edition, January 1984. URL: `http://www.worldcat.org/isbn/0131816292`.

21 James Munkres. *Topology*. Pearson, 2nd edition, January 2000.

22 Gil Neiger. Distributed consensus revisited. *Information Processing Letters*, 49(4):195–201, February 1994. `doi:10.1016/0020-0190(94)90011-6`.

23 Gil Neiger and Sam Toueg. Automatically increasing the fault-tolerance of distributed algorithms. *Journal of Algorithms*, 11(3):374–419, 1990. `doi:10.1016/0196-6774(90)90019-B`.

24 Michael Saks and Fotios Zaharoglou. Wait-free k-set agreement is impossible: The topology of public knowledge. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing*, STOC'93, pages 101–110, New York, NY, USA, 1993. ACM. `doi:10.1145/167088.167122`.

25 T. Srikanth and S. Toueg. Simulating authenticated broadcasts to derive simple fault-tolerant algorithms. *Distributed Computing*, 2(2):80–94, June 1987.

A Appendix: Proofs for the Connectivity Arguments

Proof of Lemma 21. Consider a k-simplex $\alpha = \{u_0, \ldots, u_k\} \subseteq \{v_0, \ldots, v_d\}$ with $k+1$ different inputs. Let $\mathcal{I}_\beta = \Psi(\mathbb{P}, \beta)$ for any $\beta \subseteq \alpha$, and $\mathcal{I}_x = \bigcup_{\beta \in \text{skel}^x(\alpha)} \Psi(\mathbb{P}, \beta)$. We construct a sequence of continuous maps $g_x : |\,\text{skel}^x(\alpha)| \to |\mathcal{K}_x|$ where $\mathcal{K}^x$ is homeomorphic to $\text{skel}^x(\alpha)$ in $|\,\text{skel}^x(\mathcal{P}(\mathcal{I}_x))|$.

Base. Let g_0 map any vertex $v \in \alpha$ to a vertex in $\mathcal{K}_v = \mathcal{P}(\mathcal{I}_{\{v\}})$. We know that $\mathcal{K}_v$ is k-connected since $\dim(\mathcal{I}_{\{v\}}) = \dim(\mathcal{I})$ and $\mathcal{P}$ is a k-connected carrier map. We just constructed

$$g_0 : |\,\text{skel}^0(\alpha)| \to |\mathcal{K}_0|,$$

where $\mathcal{K}^0$ is isomorphic to a $\text{skel}^0(\alpha)$ in $|\,\text{skel}^0(\mathcal{P}(\mathcal{I}_0))|$.

Induction Hypothesis. Assume $g_{x-1} : |\,\text{skel}^{x-1}(\alpha)| \to |\mathcal{K}_{x-1}|$ for any $x \leq k$, where $\mathcal{K}_{x-1}$ is isomorphic to $\text{skel}^{x-1}(\alpha)$ in $|\,\text{skel}^{x-1}(\mathcal{P}(\mathcal{I}_{x-1}))|$. For any $\beta \in \text{skel}^x(\alpha)$, we have that $\text{skel}^x(\mathcal{P}(\mathcal{I}_\beta))$ is $(x-1)$-connected, hence the continuous image of the $(x-1)$-sphere in $\mathcal{P}(\mathcal{I}_\beta)$ can be extended to the continuous image of the x-disk in $\text{skel}^x(\mathcal{P}(\mathcal{I}_\beta))$. We just constructed

$$g_x : |\,\text{skel}^x(\alpha)| \to |\mathcal{K}_x|,$$

where $\mathcal{K}^x$ is isomorphic to $\text{skel}^x(\alpha)$ in $|\,\text{skel}^x(\mathcal{P}(\mathcal{I}_0))|$. In the end, we have $g_k : |\alpha| \to |\mathcal{K}_k|$ where $\mathcal{K}_k$ is isomorphic to α in $\text{skel}^k(\mathcal{P}(\mathcal{I}_k))$.

Now suppose, for the sake of contradiction, that k-set agreement is solvable, so there must be a simplicial map $\delta : \mathcal{P}(\mathcal{I}) \to \mathcal{O}$ carried by Δ. Then, induce the continuous map $\delta_c : |\mathcal{K}_k| \to |\alpha|$ from δ such that $\delta_c(v) \in |\,\text{views}(\delta(\mu))|$ if $v \in |\mu|$, for any $\mu \in \mathcal{K}_k$. Also, note that the composition of g_k with the continuous map δ_c induces another continuous map

$|\alpha| \to |\partial\,\alpha|$, since by assumption δ never maps a k-simplex of $\mathcal{K}_k$ to a simplex with $k+1$ different views (so δ_c never maps a point to $|\operatorname{Int}\alpha|$). We built a *continuous retraction* of α to its own border $\partial\,\alpha$, a contradiction (please refer to [20, 17]). Since our assumption was that there existed a simplicial map $\delta : \mathcal{P}(\mathcal{I}) \to \mathcal{O}$ carried by Δ, we conclude that k-set agreement is not solvable. ◀