Parabyzantine Protocols

LIAM MONNINGER, Ramate LLC, USA

This write-up is for friends and colleagues to explain the beginnings of my study of a category of distributed computing protocols which I call Parabyzantine. These protocols act like Byzantine fault-tolerant protocols up to some well-understood point. Below, I begin with my motivation which lies in the need for fault-tolerant order and compute efficiency. I then provide a trivial combinatorial example of an efficient Parabyzantine protocol called Resample. I finally suggest more abstract constructions with which the study is properly concerned, including an example from a suggestive formal category $\mathfrak{P}$.

Most terms and notation loosely follow Herlihy, Kozlov, and Rajsbaum [1] (hereafter, HKR).

1 Motivation

My motivation for studying Parabyzantine protocols, as I have dubbed them, is derived primarily from two phenomena in the world of distributed computing:

- the need for fault-tolerant order;
- and, the compute inefficiencies of Byzantine fault-tolerant replica-based protocols.

1.1 The General Need for Order

In computing, we typically reason about partial functions. These are functions which deterministically map one value to another, but may not be defined for all inputs.

$$\hat{f} : S \to S \cup \{\bot\} \text{ is a lifted partial function.}$$
$$f : S \rightharpoonup S \text{ is a partial function in its native category.}$$

When composing these partial functions, we axiomatize that undefinedness propagates through the composition.

$$\begin{aligned} &f, g : S \rightharpoonup S \\ &f \circ g \text{ is defined if and only if} \\ &\quad g(x) \text{ is defined and} \\ &\quad f(g(x)) \text{ is defined} \end{aligned}$$

We find this to be a suitable model for programming modern machines, because we know how to design hardware which can logically represent functions with these properties and which is rarely corrupted by the surrounding environment.

However, for general computing tasks, we often have partial functions which are non-commutative.

$$f \circ g \neq g \circ f$$

A familiar example is subtraction on $\mathbb{N}$, which is naturally partial:

Author's Contact Information: Liam Monninger, liam@ramate.io, Ramate LLC, Durham, California, USA.

$$\text{sub} : \mathbb{N} \times \mathbb{N} \rightharpoonup \mathbb{N}$$
$$\text{sub}(a, b) = \begin{cases} a - b & \text{if } a \geq b, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

And is also non-commutative:

$$\text{sub}(5, 3) = 2 \neq \text{sub}(3, 5) \text{ (undefined).}$$

In general, for a family $f_1, \ldots, f_n$ of partial functions, the composite is determined by the ordered sequence $(f_1, \ldots, f_n)$ not the unordered set $\{f_1, \ldots, f_n\}$. In local hardware cases—such as a single hart—we can often form reasonable assumptions or conventions, such that something like a state machine reading through an assembly is equivalent to a sequence of partial functions. In the context of general distributed computing tasks, however, we tend to make these assumptions more minimal.

In distributed computing, we may instead refer to a set of processes P which are able to share information with one another via messages $m \in M$ to varying degrees of success. Then, within this context, we hope to achieve results up to some notion of correctness, typically up to some notion of equivalence with the local hardware case.

Put more provocatively, a distributed computing protocol often has to decide from a chaotic world of messages what a reasonable order of partial functions should be. Signatures can be placed on messages to ensure they are authentic. Prover systems can be built to attest to the correctness of the value of a partial function. But, these may mean little without a reasonable order.[1]

1.2 2f + 1

Often the success of a distributed computing protocol is framed in terms of its ability to induce agreement. And, in the abundant cases where order matters, this necessarily means agreement on said order.

If we take set of processes $|P| = 3f + 1$ and ask for the unique agreeing set of messages $M_i \subseteq M$ which indicate computing the partial function at index i, we may wonder what different requirements on this subset might mean.

For example, a simple majority $|M_i| > \frac{|P|}{2}$ would ensure by the Generalized Pigeonhole Principle that any two indexed subsets of messages M_i, M_j would intersect in at least one process. In a sense, at least one process can attest to each ordering step. By recurrence, this reads out a total order.[2]

More formally, let $Q : M \to 2^P$ be the function which maps a message to the set of processes which have sent it, representing a quorum. Then, we may define order $Order(M_i)$ as follows:

[1] This is not to disregard the value of modeling specific problem domains where the meaning of reasonable order may not require a total order. However, for simplicity, most of this write-up will be concerned with the total order case.

[2] In an applied setting, what constitutes a useful total order also invokes questions of authenticity, fairness, safety, and liveness.

$$
\begin{aligned}
&Q(M_i) = \bigsqcup_{m \in M_i} Q(m) \\
&O(M_i, M_j) := (Q(M_i) \cap Q(M_j) \neq \emptyset) \\
&Order(M_i) = Order(M_{i-1}) \wedge O(M_{i-1}, M_i) \\
&\quad = Order(M_0) \wedge O(M_0, M_1) \wedge \cdots \wedge O(M_{i-1}, M_i)
\end{aligned}
$$

Importantly, this simple majority model does not account for the possibility that a process may fault. As it turns out, the most aggressive assumption we can make and still retain the possibility of these intersecting sets amongst non-faulty processes—which give rise to our total order–is that at most f processes may fault and quorums must be formed by at least $2f + 1$ processes. This is the Byzantine fault model.

Let H be the set of honest processes and F be the set of faulty processes. Any two quora, under the Byzantine fault model, must then intersect in at least $f + 1$ processes:

$$
\begin{aligned}
|Q(M_1) \cap Q(M_2)| &= |Q(M_1)| + |Q(M_2)| - |Q(M_1) \cup Q(M_2)| \\
&\geq (2f + 1) + (2f + 1) - (3f + 1) = f + 1
\end{aligned}
$$

We can further deduce that any two quora must intersect in at least one honest process $h \in H$ in the presence of at most f faulty processes:

$$
\begin{aligned}
&|Q(M_1) \cap Q(M_2)| = |Q(M_1) \cap Q(M_2) \cap H| + |Q(M_1) \cap Q(M_2) \cap F|, \\
&|Q(M_1) \cap Q(M_2) \cap H| = |Q(M_1) \cap Q(M_2)| - |Q(M_1) \cap Q(M_2) \cap F|, \\
&f \geq |Q(M_1) \cap Q(M_2) \cap F| \\
&\quad \implies |Q(M_1) \cap Q(M_2) \cap H| \geq f + 1 - f = 1
\end{aligned}
$$

A straightforward interpretation of this fact renders the ubiquitous family of Byzantine fault-tolerant state machine replication protocols (BFT SMR). In these protocols, each process executes the same deterministic transition function to emit messages certifying the ordered outcome of state machine transitions, i.e., partial functions.

Without a prover system[3], BFT SMRs require at least $2f + 1$ of the honest processes to be engaged in the computing and broadcasting a partial function $f_i, f_{i+1}, \cdots, f_n$. With prover systems, this requirement holds up to the task of ordering.

If a Byzantine fault-tolerant SMR protocol tolerates f faulty processes, then the cost of computing one index worth of work is $\Theta(2f + 1)$.

[3]Prover systems are not common in practice owing to the computational expense of systems like ZKPs.

2 A Parabyzantine Protocol

If we want to better use available compute in a Byzantine fault-tolerant setting, an initial thought is to simply relax the requirement on order. Perhaps we can include fewer processes and allow our total order to fail with an acceptably low likelihood. Indeed, as I will show, even a fairly straightforward application of this concept yields a protocol which is computationally efficient and low loss.

Before we endeavor this protocol, let's first consider what computationally efficient and a low loss would mean.

2.0.1 Computationally Efficient. Computational efficiency is fairly straightforward. As referenced above, SMR requires at least $2f + 1$ processes to certify each partial function. Thus, each logical step costs $\Theta(2f + 1)$ replicated work. So, in abstract, if we use fewer than $2f + 1$ processes, we will have improved compute efficiency.

More interestingly, any constant-factor improvement over SMR yields a proportional increase in effective compute utilization across the replica set. In practical deployments this may have outsized effects on throughput, since the extra compute can be reinvested into tasks whose impact on subsequent rounds is non-linear, e.g., batching, compression, proof generation, or state access.

2.0.2 Low Loss. What is meant by low loss may be less obvious. In the Byzantine fault-tolerant setting, we have described a discrete mathematical limit that tells us we cannot have a property we called $Order(M_i)$ without involving at least $2f + 1$ processes. So, the notion that we can have protocol which is computationally more efficient seems to contradict this limit.

However, we may consider the following objective function which is minimized for M_i when $Order(M_i) = 1$:

$$Obj(M_i) = 0 \iff Order(M_i) = 1$$
$$Obj(M_i) = 1 \iff Order(M_i) = 0$$

In its current form, we cannot improve. But, if we constrain on the average case Loss $= E[Obj(M_i)] = Pr[Obj(M_i) = 1] \leq \epsilon$, we have much more flexibility.

In a sense, it is precisely this sort of assumption which gives rise to the name Parabyzantine. These are protocols which act like Byzantine fault-tolerant protocols up to some well-understood point. In the case of the Resample protocol, this point will be an average-case guarantee. While we are under the veil of average case, Resample will be Byzantine fault-tolerant. Once we remove the veil, Resample will not be.

2.1 The Resample Protocol

We define the following Byzantine fault-allowing protocol Resample:

(1) For a given index i on M, pick a random subcommittee $K_i \subseteq P$ such that $|K_i| = 3k + 1$.
(2) Accept M_i if and only if $|Q_{K_i}(M_i)| \geq 2k + 1$. Otherwise, pick another random subcommittee and repeat.

Let...

$$f' = |K_i \cap F|, h' = |K_i \cap H|, f' + h' = |K_i| = 3k + 1$$

Observe the following possible outcomes for selection of a random subcommittee:

- $|K_i \cap H| \geq 2k + 1 \iff$ Divine $= D$, representing a subcommittee which has an honest supermajority which computes M_i correctly.
- $|K_i \cap H| \leq k \iff$ Corrupt $= C$, representing a subcommittee which has a dishonest supermajority and may compute M_i incorrectly.
- $k < |K_i \cap H| < 2k + 1 \iff$ Undecided $= U$, representing all other cases wherein neither an honest nor dishonest supermajority exists and may either compute M_i correctly or disagree internally and not render a supermajority.

2.2 The Loss of the Resample Protocol

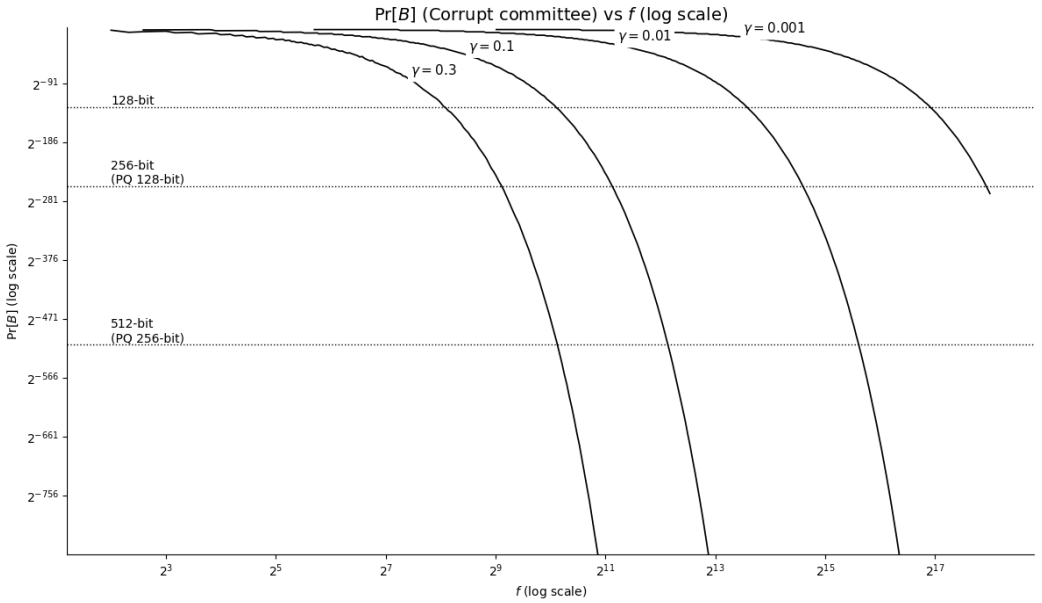

Fig. 1. $\Pr[B]$ representing the probability of selecting a corrupt subcommittee as a function of f for different sampling ratios $\gamma = \frac{k}{f}$.

Let $\mathcal{S}(f, k)$ represent the total number of ways to select the subcommittee without replacement:

$$\mathcal{S}(f, k) = \binom{3f + 1}{3k + 1}$$

The total number of ways to select a Corrupt subcommittee is:

$$\mathcal{S}_C(f,k) = \sum_{f'=2k+1}^{\min(3k+1,f)} \binom{f}{f'} \cdot \binom{2f+1}{3k+1-f'}$$

The total number of ways to select a Divine subcommittee is:

$$\mathcal{S}_D(f,k) = \sum_{h'=2k+1}^{\min(3k+1,2f+1)} \binom{2f+1}{h'} \cdot \binom{f}{3k+1-h'}$$

All other outcomes are by definition Undecided, so the total number of ways to select an Undecided subcommittee is:

$$\mathcal{S}_U(f,k) = \mathcal{S}(f,k) - \mathcal{S}_C(f,k) - \mathcal{S}_D(f,k)$$

Let S represent the transient state in which the RESAMPLE is sampling, i.e., has just started or produced an Undecided subcommittee. Let G represent the absorbing state in which RESAMPLE makes a "good" decision to select a Divine subcommittee. Let B represent the absorbing state in which RESAMPLE makes a "bad" decision to select a Corrupt subcommittee. The Markov Chain is then given by the state space Ω and transition matrix T.

$$\Omega = \{S, G, B\}$$
$$T = \begin{pmatrix} \Pr[U] & \Pr[D] & \Pr[C] \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The probability of the algorithm making a "bad" decision to select a Corrupt subcommittee $\Pr[B]$ is then:

Let $x := \Pr[$ eventually hitting B $\mid$ start in S $]$

$$x = \Pr[C] \cdot 1 + \Pr[D] \cdot 0 + \Pr[U] \cdot x$$

$$x - \Pr[U] \cdot x = \Pr[C] \implies x \cdot (1 - \Pr[U]) = \Pr[C] \implies x = \frac{\Pr[C]}{1 - \Pr[U]}$$

Observe that...

$$1 - \Pr[U] = \Pr[D] + \Pr[C] \implies \Pr[B] = \frac{\Pr[C]}{\Pr[D] + \Pr[C]}$$

Conversely, the probability of the algorithm making a "good" decision to select a Divine subcommittee $\Pr[G]$ is then:

$$\Pr[G] = 1 - \Pr[B] = 1 - \frac{\Pr[C]}{\Pr[D] + \Pr[C]} = \frac{\Pr[D]}{\Pr[D] + \Pr[C]}$$

The probability $\Pr[C]$ drops off quickly for a fixed sampling ratio $\gamma = \frac{k}{f}$ (with, say, $k = \lfloor \gamma f \rfloor$). First observe that the number of faulty replicas in a sampled subcommittee follows a hypergeometric distribution:

$$\begin{aligned} &X := |K_i \cap F| \sim \text{Hypergeometric}(N, K, n), \\ &C \equiv \{X \geq 2k + 1\}, \\ &\Pr[C](\gamma, f) = \frac{\mathcal{S}_C(f, \lfloor \gamma f \rfloor)}{\mathcal{S}(f, \lfloor \gamma f \rfloor)}. \end{aligned}$$

Standard large-deviation bounds for sampling without replacement give a KL-based tail bound:

$$\begin{aligned} &\Pr[X \geq qn] \leq \exp(-n \cdot D(q \,\|\, p)), \\ &D(q \,\|\, p) = q \log\left(\frac{q}{p}\right) + (1 - q) \log\left(\frac{1 - q}{1 - p}\right). \end{aligned}$$

In our case $p = \frac{f}{3f+1} \approx \frac{1}{3}$, $q = \frac{2k+1}{3k+1} \approx \frac{2}{3}$, and $n = 3k + 1$. Hence...

$$D\left(\frac{2}{3} \,\middle\|\, \frac{1}{3}\right) = \frac{2}{3} \log 2 + \frac{1}{3} \log \frac{1}{2} = \frac{1}{3} \log 2.$$

Therefore...

$$\begin{aligned} &\Pr[C] \lesssim \exp\left(-(3k + 1) \cdot \frac{1}{3} \log 2\right) = 2^{-(3k+1)/3}, \\ &\Pr[C](\gamma, f) \lesssim 2^{-(3\gamma f+1)/3} = 2^{-\gamma f - 1/3} = 2^{-\Omega(f)}. \end{aligned}$$

In particular, for any fixed $\gamma \in (0, 1)$...

$$\lim_{f \to \infty} \Pr[C](\gamma, f) = 0 \forall \gamma \in (0, 1)$$

That is, Resample is Parabyzantine protocol with exponentially decreasing loss as the fault parameter f increases for a fixed sampling ratio γ.

2.3 The Cost of the Resample Protocol

The expected cost of Resample, measured in the total number of processes engaged across all sampling attempts, remains on the order $\Theta(k)$, but admits a constant-factor improvement over the $2f + 1$ quorum required by standard SMR.

Fix an index i and write $\Pr[U] := \Pr[\mathcal{U}]$ for the probability that a uniformly sampled subcommittee of size $3k + 1$ is *Undecided*. Each sampling attempt terminates (i.e., yields a decision) precisely when the outcome is not $\mathcal{U}$. Thus, if T denotes the number of sampling attempts until termination, then T is geometric with success probability $1 - \Pr[U]$:

$$\Pr[T = t] = \Pr[U]^{t-1} \cdot (1 - \Pr[U]) \qquad \text{for } t \in \mathbb{N},$$
$$\mathbb{E}[T] = \sum_{t=1}^{\infty} t \cdot \Pr[U]^{t-1}(1 - \Pr[U]) = \frac{1}{1 - \Pr[U]}.$$

Each attempt engages at most $3k + 1$ processes. Therefore the expected number of process-participations consumed per decision is

$$\mathbb{E}[\text{Processes Used}] = (3k + 1) \cdot \mathbb{E}[T] = \frac{3k + 1}{1 - \Pr[U]}.$$

In regimes of interest (e.g., for a fixed sampling ratio $\gamma = \frac{k}{f}$ with large f), $\Pr[U]$ is typically bounded away from 1; in particular, if $\Pr[U] \approx \frac{1}{2}$, then

$$\mathbb{E}[T] \approx \frac{1}{1 - \frac{1}{2}} = 2,$$
$$\mathbb{E}[\text{Processes Used}] \approx 2(3k + 1) = \Theta(k).$$

Thus, RESAMPLE is Parabyzantine in the sense described above: it replaces a worst-case quorum requirement with an average-case sampling cost, remaining computationally efficient while permitting a tunably low probability of failure.

3 An Expanded View of Parabyzantine Protocols

Now that we have found one Parabyzantine protocol, we might wonder if we can find others and under what conditions these others act like Byzantine fault-tolerant protocols. This is, however, a big universe, and it helps to explore it in a structured manner.

To this end, I have chosen to align my views along two vantages:

(1) the rich perspective of topological frameworks for distributed computing;
(2) and, the category theoretic consideration of relationships between Parabyzantine protocols.

My exploration is further guided by the search for a non-trivial bridge between topological descriptions of protocols and non-topological descriptions of protocols—one that can be expressed in the morphisms of a category of Parabyzantine protocols.

3.1 Topological Protocols

The fields of topology and distributed computing have a well-established and fruitful intersection. We know, for example, that we can formulate Byzantine agreement as a problem of barycentric subdivision on a simplicial complex via the toplogical framework of Herlihy and Shavit [2] such as in HKR's BaryAgree [1].[4] Further, we know that BaryAgree achieves $Obj(M_i) = 0$ in the Byzantine fault-tolerant setting.

Much of this intersection is, in a word, unavoidable: it is hard to model agreement problems without something that looks like the sort of complex geometric object with which topology is concerned. And, indeed I have spent most of my own study digesting the various topological tools and constructions which are already invoked in the catalog of Byzantine agreement problems.

Rather than spending time reciting various twists and turns which I have not yet synthesized into a coherent contribution, I would instead like to point to two reasons why the topological formulation of Byzantine agreement is acutely interesting to my study of Parabyzantine protocols:

(1) its relevance to connected structure;
(2) and, its rich supply of non-standard equivalence relations.

3.1.1 Connected Structure. Byzantine agreement is often concerned with some kind of connected structure. Consider our recursive definition of $Order(M_i)$, which required consecutive quorums to intersect. Intuitively, these intersections form a connected structure in the process set P such that we can move from one quorum to the next.

To make this intuition precise, consider the following nerve construction. Let P be the set of processes, and for each index i let $Q_i \subseteq P$ denote the quorum of processes whose messages certify the ith step.[5]

Consider the family of subsets $\mathcal{U} = \{Q_i\}_{i \in I}$. Viewing $\mathcal{U}$ as a cover of

$$X := \bigcup_{i \in I} Q_i \subseteq P,$$

[4] Barycentric agreement is a form of the more general lattice agreement.

[5] More generally, one may take $\mathcal{Q}$ to be a family of quorums and regard $\{Q\}_{Q \in \mathcal{Q}}$ as a cover of the participating process set.

we may form the nerve simplicial complex $N(\mathcal{U})$ defined by:

$$\mathrm{Vert}(N(\mathcal{U})) = \mathcal{U},$$

$$\{Q_{i_0}, \ldots, Q_{i_d}\} \text{ spans a } d\text{-simplex} \iff \bigcap_{j=0}^{d} Q_{i_j} \neq \emptyset.$$

In this language, our intersection predicate $O(M_{i-1}, M_i)$ asserts precisely that consecutive vertices in the nerve share an edge:

$$O(M_{i-1}, M_i) \iff Q_{i-1} \cap Q_i \neq \emptyset \iff \{Q_{i-1}, Q_i\} \in N(\mathcal{U})^{(1)}.$$

Thus, a run satisfying $Order(M_i)$ determines a path in the 1-skeleton of the nerve. Or, in other words, it is the connectedness of a space with a topology describing the intersection of quorums with which the fundamental Byzantine assumption is concerned.

3.1.2 Non-Standard Equivalence Relations. The second reason for my interest in the topological formulation of Byzantine agreement is that topology is a rich discipline for studying non-standard equivalence relations.

Recall that non-commutativity was a key consideration in arriving at our Byzantine fault-tolerant state machine replica protocol.

$$f \circ g \neq g \circ f$$

When studying topologically, however, we may be able to show, for example, that $f \circ g$ and $g \circ f$ are equivalent up to homotopy:

$$f \circ g \simeq g \circ f$$

In topology generally, we can access many of these non-standard equivalence relations. Homotopy, indistinguishability under a quotient space, and various more specific deformation relations, give us a lot of ways to relate two objects which are not equivalent in the standard sense. Similar to how we used the average case to find the Parabyzantine RESAMPLE protocol, we may be able to infer other protocols which act like Byzantine fault-tolerant protocols up to some well-understood topological relationship and hence are Parabyzantine.

Thus, in addition to its extensive corpus, the topological formulation of Byzantine agreement is both a fitting tool for describing the sorts of properties we would like to preserve and a source of instruments for relaxing requirements into Parabyzantine protocols.

3.2 A Category of Parabyzantine Protocols

My category-theoretic interest in Parabyzantine protocols is motivated by two questions:

(1) Can we define meaningful morphisms between objects representing Parabyzantine protocols?
(2) Can composition of these morphisms reveal useful non-trivial protocols?

A category of distributed protocols is a large undertaking. For the purposes of this write-up, I will instead describe a small provisional construction which captures one dimension of protocol variation: the tradeoff between quorum cost and the strength of the order guarantee.

Let $\mathfrak{P}$ be a provisional category whose objects are Parabyzantine protocols together with their parameters. In particular, let $\mathcal{R} \hookrightarrow \mathfrak{P}$ be the full subcategory whose objects are resampling instances

$$X_{k,f} := \textsc{Resample}(k, f)$$

where f is the global fault parameter ($|P| = 3f + 1$) and k is the resampling parameter ($|K| = 3k + 1$).

We will regard two protocol instances as equivalent when they induce the same quorum structure and the same induced order semantics. In particular, when $k = f$ the resampling committee is the full population, so $\textsc{Resample}(f, f)$ is equivalent to the "full quorum" protocol $\mathcal{P}$.

3.2.1 A Thin Category of Refinements. Rather than attempt to define all morphisms in $\mathfrak{P}$, we focus on a simple and compositional family of maps inside $\mathcal{R}$. Define a morphism

$$\lceil_{k,f} : X_{k,f} \to X_{k+1,f}$$

which increments the sampling parameter by one. Intuitively, $\lceil_{k,f}$ is a refinement map: it increases committee size and thus moves the protocol closer to the full-quorum behavior.

Because these refinements compose, we obtain for any $k \leq \ell \leq f$ a canonical composite

$$\lceil^{\ell-k} : X_{k,f} \to X_{\ell,f},$$
$$\lceil^{\ell-k} \triangleq \lceil_{\ell-1,f} \circ \cdots \circ \lceil_{k,f}.$$

In particular, applying $\lceil$ $f - k$ times yields a morphism from any resampling instance to the full-quorum instance:

$$X_{k,f} \xrightarrow{\lceil^{f-k}} X_{f,f} \simeq \mathcal{P}.$$

This expresses, in categorical language, a basic but useful fact: within the resampling family, there is a directed path of refinements from cheaper, smaller committees to more classical, full-quorum behavior.

One may view $\mathcal{P}$ as an absorbing point of this refinement chain: once k reaches f, further increments no longer change the induced quorum semantics. In this sense, $\mathcal{P}$ behaves like a fixed point of the refinement operation.

$$X_{k,f} \xrightarrow{\lceil} \cdots \xrightarrow{\lceil} X_{f,f} \simeq \mathcal{P} \circlearrowright \lceil$$

3.2.2 A Suggestive Bridge to Topology. Now suppose we have a class of protocols $\mathcal{T} \hookrightarrow \mathfrak{P}$ which admit a topological semantics, e.g., via simplicial complexes and subdivision. It is tempting to imagine a functorial assignment

$$\Phi : \mathcal{R} \to \mathcal{T}$$

which sends a resampling instance $X_{k,f}$ to a topological object $Y_{k,f} := \Phi(X_{k,f})$ whose semantics approximate those of the full-quorum protocol $\mathcal{P}$, but only up to a weaker equivalence relation (e.g. homotopy or a quotient identifying indistinguishable executions).

In the most optimistic case, one might have

$$\Phi(X_{f,f}) \simeq \Phi(\mathcal{P})$$
and
$$\Phi(X_{k,f}) \simeq \Phi(\mathcal{P}) \text{ up to a controlled deformation.}$$

Even in this trivial resampling example, such a construction would amount to a bridge between a combinatorial protocol family and a topological description.

The hope, as the category $\mathfrak{P}$ is explored beyond $\mathcal{R}$, is that more meaningful morphisms and their compositions reveal non-trivial Parabyzantine protocols—not merely refinements in committee size, but systematic transformations which preserve some notion of correctness while relaxing others in a controlled way.

To analogize, $\mathfrak{P}$ becomes a map of regions in Byzantine fault-tolerant computing. Building bridges between these regions may reveal higher-order patterns which lead us to non-trivial constructions and better protocols.

4 Afterword

As alluded to previously, my study of Parabyzantine protocols has naturally encouraged me to review much of the existing literature on topological frameworks for distributed computing. I expect to be in this review process for some time. Hence, for those readers coming from my work on the OAC project, the Gwrdfa API will likely contain only protocols related to the RESAMPLE protocol for some time—though adapted for an applied setting.

References

[1] Maurice Herlihy, Dmitry Kozlov, and Sergio Rajsbaum. 2013. *Distributed Computing Through Combinatorial Topology.* Morgan Kaufmann, San Francisco, CA, USA.

[2] Maurice P. Herlihy and Nir Shavit. 1999. The Topological Structure of Asynchronous Computability. *J. ACM* 46, 6 (1999), 858–923. https://doi.org/10.1145/331524.331563