Unifying Synchronous and Asynchronous Message-Passing Models

Maurice Herlihy
Computer Science Department
Brown University
Providence, RI 02912
herlihy@cs.brown.edu

Sergio Rajsbaum
Instituto de Matemáticas
U.N.A.M., D.F. 04510, México
rajsbaum@servidor.unam.mx

Mark R. Tuttle
Digital Equipment Corporation
Cambridge Research Lab
One Kendall Square, Building 700
Cambridge, MA 02139
tuttle@crl.dec.com

Abstract

We take a significant step toward unifying the synchronous, semi-synchronous, and asynchronous message-passing models of distributed computation. The key idea is the concept of a *pseudosphere*, a new combinatorial structure in which each process from a set of processes is independently assigned a value from a set of values. Pseudospheres have a number of nice combinatorial properties, but their principal interest lies in the observation that the behavior of protocols in the three models can be characterized as simple unions of pseudospheres, where the exact structure of these unions is determined by the timing properties of the model. We use this pseudosphere construction to derive new and remarkably succinct proofs of bounds on consensus and k-set agreement in the asynchronous and synchronous models, as well as the first lower bound on wait-free k-set agreement in the semi-synchronous model.

1 Introduction

The field of distributed computing embraces a bewildering variety of models [LL90, Gaf98]. A fundamental dimension along which these models differ is the degree to which process activity is synchronized. At one end of the spectrum is the *synchronous model* in which computation proceeds in a sequence of rounds. In each round, a process sends messages to the other processes, receives the messages sent to it by the other processes in that round, and changes state. All processes take steps at exactly the same rate, and all messages are delivered with exactly the same message delivery time. At the other end is the *asynchronous model* in which there is no bound on the amount of time that can elapse between process steps, and there is no bound on the time it can take for a message to be delivered. Between these extremes is the *semi-synchronous model* in which process step times and message delivery times can vary, but are bounded between constant upper and lower bounds. Proving a lower bound in any of these models requires a deep understanding of the global states that can arise in the course of a protocol's execution, and of how these global states are related.

The notion of *indistinguishability* or *similarity* [FLP85, HM90] has played a fundamental role in nearly every lower bound in distributed computation. Two global states are considered indistinguishable if one process has the same local state in both, and therefore cannot distinguish between them. The graph-theoretic representation of similarity, in which two global states are joined by an edge labeled with a process P if the global states are indistinguishable to P, has proven to be immensely powerful.

While the classical notion of similarity captures the notion of two states being indistinguishable to a single process, higher degrees of similarity have proved essential for understanding problems such as k-set agreement [Cha91] and renaming [ABND$^+$87, ABND$^+$90]. For example, it may matter that a pair of global states are indistinguishable to two processes, to three processes, and so on. To capture these higher degrees of similarity it is convenient to represent the global state of a system of $n+1$ processes with the n-dimensional analog of a triangle, called a *simplex*, where each vertex of the simplex representing a global state is labeled with the local states of processes in this global state. The set of all global states at the end of a protocol represented in this way forms a *simplicial complex*, sometimes called the *protocol complex*. The degree of similarity between two global states is

PODC 98 Puerto Vallarta Mexico

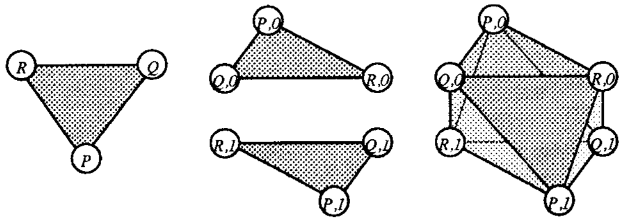

Figure 1: Construction of a three-process binary pseudosphere.

represented geometrically by the number of vertices the corresponding simplexes have in common: two global states similar to one process share one vertex (namely that vertex labeled with the local state of this process having the same local state in both global states), states similar to two processes share two vertices, states similar to three processes share three vertices, and so on.

In this paper, we take a significant step toward unifying the synchronous, semi-synchronous, and asynchronous models of computation. The key, unifying idea in this paper is the concept of a ***pseudosphere***, a simplicial complex in which each process from a set of processes is independently assigned a value from a set of values. Pseudospheres have a number of nice combinatorial properties, but their principal interest lies in the observation that protocol complexes in the three models can be characterized as simple unions of pseudospheres, where the exact structure of these unions is determined by the timing properties of the model. Because of the simple combinatorial properties of pseudospheres, reasoning about these unions can be accomplished by straightforward combinatorial arguments. We use this pseudosphere construction to derive new and remarkably succinct proofs of bounds on consensus [PSL80, Fis83] and k-set agreement [Cha91] in the asynchronous and synchronous models, as well as the first lower bound on wait-free k-set agreement in the semi-synchronous model.

A pseudosphere can be defined very simply. Start with an n-dimensional simplex where each vertex is labeled with a process id, and choose a finite set of values taken from an arbitrary domain. The pseudosphere is the complex constructed by taking multiple copies of this simplex and independently labeling each vertex with a value from the domain. For example, Figure 1 shows how to construct a pseudosphere by independently assigning binary values to a set of three processes. The left-hand figure shows a triangle labeled with process ids P, Q, and R. The central figure shows an intermediate stage where two copies of the triangle are each labeled with zeros and ones. The right-hand figure shows the complete construction, where copies of the triangle are labeled with all combinations of zeros and ones. We can just as easily assign values from a set larger than $\{0, 1\}$, although the result is harder to illustrate. We call this construct a pseudosphere because it is easily shown that the result of assigning binary values to $n + 1$ processes is topologically equivalent to an n-dimensional sphere.

The collection of initial global states for consensus or k-set agreement clearly forms a pseudosphere whose vertices are labeled with input values. For example, the right-hand figure in Figure 1 is the input complex for three-process consensus. The basic insight underlying the work presented in this paper is that protocol complexes in the models of interest have natural representations as unions of pseudospheres, except that the vertices are labeled with timing and failure information instead of input values. Reasoning about these protocol complexes reduces to the purely combinatorial problem of reasoning about unions of pseudospheres, and indeed the formal manipulations needed to derive our results are remarkably similar in all three models. In each model, we define a "round" structure appropriate for that model, and express the one-round executions as the union of pseudospheres. An r-round execution is constructed by inductively replacing each simplex in the single-round execution with the union of pseudospheres produced by the $(r - 1)$-round protocol. The protocol complex produced by this iterative construction represents only a subset of the global states reachable in the model, but this set is large enough to prove the desired results for consensus, k-set agreement, and so on.

2 Related Work

Of course, we are not the first to propose iterative constructions representing the global states at the end of a protocol. As one example, Borowsky and Gafni [BG93] state a construction for the asynchronous model, and while the intuition behind the construction is compelling, it is not easy to write down a formal description of that construction. In a later paper [BG97], they define an iterated immediate snapshot model that has a recursive structure and has proven to be useful [HS97]. As another example, Chaudhuri, Herlihy, Lynch, and Tuttle [CHLT93] give an inductive construction for the synchronous model, and while the resulting "Bermuda Triangle" is visually appealing and an elegant combination of proof techniques from the literature, there is a fair amount of machinery needed in the formal description of the construction. In this sense, the formal presentation of our construction is substantially more succinct than these constructions.

We are also not the first to attempt to unify synchronous and asynchronous models of computation. All such attempts, including ours, restrict attention to a subset of well-behaved, round-based executions. There are a number of ways one could consider unifying models. One approach is to translate algorithms written for a synchronous model into an asynchronous model. This was the approach used by Awerbuch [Awe85] in the message-passing model when he constructed his *synchronizer* and showed how (in the absence of faults) synchronous protocols can be run in asynchronous systems in the presence of a synchronizer. This was also the approach used by Gafni [Gaf98] in the shared-memory model when he described a *round-by-round failure detector* and how these detectors can be used to run in an asynchronous model an algorithm written for a synchronous model. Another approach, which is basically our approach, is to identify a set of concepts that can be used to describe or reason about multiple models. This was the direction followed by Moses and Rajsbaum [MR98] when they showed how the concepts of communication layering and mobile faults can be used to reason in a uniform way about the synchronous and asynchronous models, and news of their results motivated us to work out our own direction. The translation approach assumes that the synchronous model is an easier model in which to work. This assumption seems justified for algorithm design, but lower bounds are typically easier to derive in the asynchronous model. Our pseudosphere construction illustrates this distinction in a striking way, since the asynchronous construction has a much simpler combinatorial structure.

We are the first, however, to unify the synchronous, semi-synchronous, and asynchronous models of message-passing computation with a single concept, namely the pseudosphere. Gafni [Gaf98] does his most formal work in a synchronous and asynchronous shared-memory model, and while he sketches how his ideas might be extended to a semi-synchronous message-passing model, this extension requires changing the nature of the failure detector. Moses and Rajsbaum [MR98] focus on synchronous and asynchronous models. Their stated results apply to consensus whereas our results apply to both consensus and k-set agreement, although they consider other issues as well.

One indication that our construction is fundamental is that the pseudosphere constructions originally developed to unify the synchronous and asynchronous models extended cleanly to the semi-synchronous model. We consider this significant. Although variants of the semi-synchronous model have been around for a long time, we are aware of only one substantial lower bound in this model: the consensus bound of Attiya, Dwork, Lynch, and Stockmeyer [ADLS94]. The absence of other results suggests that it is very difficult to prove significant lower bounds in this model, and that results and proof techniques from other models do not translate into the semi-synchronous model as easily as one might hope. With our pseudosphere construction, however, we can prove the first lower bound for wait-free k-set agreement in this model. Another indication is that our construction can be used to simplify the proof of known results. For example, our protocol complex construction is significantly more succinct than the construction used by Herlihy and Shavit [HS93] in their asynchronous computability theorem, and our construction could be used to simplify the proof of one direction of that theorem. And, as mentioned, the formal analysis underlying our construction can be presented considerably more succinctly than the constructions used by Borowsky and Gafni [BG93] and by Chaudhuri, Herlihy, Lynch, and Tuttle [CHLT93].

Our constructions are guided by concepts and theorems taken from elementary combinatorial topology. As described above, we believe our results are interesting in their own right, even to readers unfamiliar with or uninterested in topological techniques. For readers interested in applications of topology to distributed computing, however, our constructions should be even more interesting. Our approach here replaces the existential arguments used by Herlihy and Shavit [HS93] to analyze protocol complexes in the asynchronous model with a constructive, inductive analysis. Although the existential analysis encompasses the entire complex, and ours is restricted to a well-structured subcomplex, we feel that the concise and constructive nature of our treatment makes a contribution, both in terms of simplicity and brevity, and in terms of intuitive appeal. Like us, Chaudhuri, Herlihy, Lynch, and Tuttle [CHLT93] explicitly construct a subset of the protocol complex in the synchronous model. It is not clear, however, how to translate that construction into the asynchronous model.

3 Basic Topology

A *vertex* $\vec{v}$ is a point in a high-dimensional Euclidean space. Vertexes $\vec{v}_0, \ldots, \vec{v}_n$ are *affinely independent* if $\vec{v}_1 - \vec{v}_0, \ldots, \vec{v}_n - \vec{v}_0$ are linearly independent. An n-

dimensional simplex (or *n-simplex*) $S^n = (\vec{s}_0, \ldots, \vec{s}_n)$ is the convex hull of a set of $n+1$ affinely-independent vertexes. For example, a 0-simplex is a vertex, a 1-simplex a line segment, a 2-simplex a solid triangle, and a 3-simplex a solid tetrahedron. Where convenient, we use superscripts to indicate dimensions of simplexes. We say that the $\vec{s}_0, \ldots, \vec{s}_n$ *span* S^n. By convention, a simplex of dimension $d < 0$ is an empty simplex. Simplex S^m is a (proper) *face* of T^n if the vertexes of S^m are a (proper) subset of the vertexes of T.

A *simplicial complex* (or complex) is a set of simplexes closed under containment and intersection. The *dimension* of a complex is the highest dimension of any of its simplexes. $\mathcal{L}$ is a *subcomplex* of $\mathcal{K}$ if every simplex of $\mathcal{L}$ is a simplex of $\mathcal{K}$. A map $\mu : \mathcal{K} \to \mathcal{L}$ carrying vertexes to vertexes is *simplicial* if it also induces a map of simplexes to simplexes. Two complexes $\mathcal{K}$ and $\mathcal{L}$ are *isomorphic*, written $\mathcal{K} \cong \mathcal{L}$, if there is a surjective and one-to-one simplicial map $\iota : \mathcal{K} \to \mathcal{L}$.

Informally, a complex is k-connected if it has no holes in dimensions k or less. More precisely,

Definition 1: A complex $\mathcal{K}$ is *k-connected* if every continuous map of the k-sphere to $\mathcal{K}$ can be extended to a continuous map of the $(k+1)$-disk [Spa66, p. 51]. By convention, a complex is (-1)*-connected* iff it is nonempty, and every complex is *k-connected* for $k < -1$.

This definition says that a complex is 0-connected if it is connected in the graph-theoretic sense. The definition of k-connectivity may appear difficult to use, but fortunately we can do all our reasoning in a combinatorial way, using the following elementary consequence of the Mayer-Vietoris sequence [Mun84, p. 142].

Theorem 2: If $\mathcal{K}$ and $\mathcal{L}$ are complexes such that $\mathcal{K}$ and $\mathcal{L}$ are k-connected, and $\mathcal{K} \cap \mathcal{L}$ is nonempty and $(k-1)$-connected, then $\mathcal{K} \cup \mathcal{L}$ is k-connected.

This theorem allows us to reason about a complex's connectivity in terms of the connectivity of its components.

4 Model

A set of $n+1$ sequential *processes* communicate by sending messages to one another. At any point, a process may *crash*: it stops and sends no more messages. There is a bound f on the number of processes that can fail. In this paper, we consider three distinct message-passing models. In the *asynchronous* model, there is no bound on process speed or message delivery time. In the *synchronous* model, processes take steps at the same rate, and messages take the same amount of time to be delivered. In the *semi-synchronous* model, the time between two consecutive steps of a process is at least c_1 and at most c_2, and the time to deliver a message is at most d, where c_1, c_2, and d are known constants. (The synchronous and asynchronous models are limiting cases of the semi-synchronous model.) In all three models, message delivery is reliable and FIFO.

Each process starts with an *input value* taken from a set V, and then executes a deterministic *protocol* in which it repeatedly receives one or more messages, changes its local state, and sends one or more messages. After a finite number of steps, each process chooses a *decision value* and halts. At any instant, a process's local state is given by the input value and the the sequence of messages received so far. A protocol is uniquely determined by its *message function* and its *decision function*. The message function determines which messages a process should send in a given state, and the decision function determines which output value a process should choose in a given state (if any). A protocol is a *full-information protocol* [Had83, FL82, PSL80] if the message function causes each process to send its entire local state when it sends a message. We can assume without loss of generality that all protocols $\mathcal{P}$ we consider are *full-information* protocols [Had83, FL82, PSL80, DM90].

In the k-set agreement task [Cha91], processes are required to (1) choose a decision value after a finite number of steps, (2) choose as their decision values some process's input value, and (3) collectively choose no more than k distinct decision values. When $k = 1$, this problem is usually called *consensus* [PSL80, Fis83].

We now show how to apply concepts from combinatorial topology to this model. An initial local state of process P is modeled as a vertex $\vec{v} = \langle P, v \rangle$ labeled with P's process id and initial value v. An initial global state is modeled as an n-simplex $S^n = (\langle P_0, v_0 \rangle, \ldots, \langle P_n, v_n \rangle)$, where the P_i are distinct. We use *ids*(S^n) to denote the set of process ids associated with S^n, and *vals*(S^n) the set of values. The set of all possible initial global states forms a complex, called the *input complex*.

Any protocol has an associated *protocol complex* $\mathcal{P}$, defined as follows. Each vertex is labeled with a process id and a possible local state for that process. A set of vertexes $\langle P_0, v_0 \rangle, \ldots, \langle P_d, v_d \rangle$ spans a simplex of $\mathcal{P}$ if and only if there is some protocol execution in which $P_0, \ldots, P_d$ finish the protocol with respective local states $v_0, \ldots, v_d$. Each simplex thus corresponds

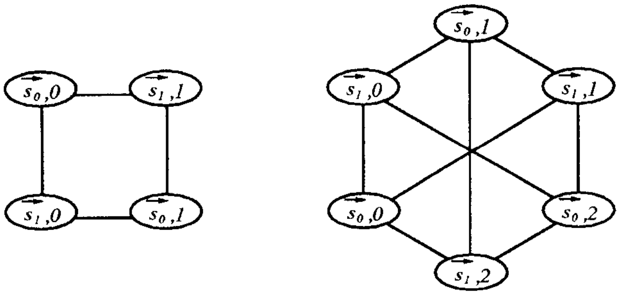

Figure 2: Pseudospheres $\psi(S^1;\{0,1\})$ and $\psi(S^1;\{0,1,2\})$.

to an equivalence class of executions that "look the same" to the processes at its vertexes. The protocol complex $\mathcal{P}$ depends both on the protocol and on the timing and failure characteristics of the model.

We use $\mathcal{P}(S^m)$ to denote the subcomplex of $\mathcal{P}$ corresponding to executions in which only the processes in *ids*(S^m) participate (the rest fail before sending any messages). If $m < n - f$, then there are no such executions, and $\mathcal{P}(S^m)$ is empty. More generally, if $\mathcal{I}$ is a subcomplex of the input complex, then we define $\mathcal{P}(\mathcal{I})$ to be the union of $\mathcal{P}(S^m)$ for all S^m in $\mathcal{I}$. A protocol *solves* k-set agreement if the protocol's decision map δ carries vertexes of $\mathcal{P}$ to values in V such that if $\vec{p} \in \mathcal{P}(S^n)$ then $\delta(\vec{p}) \in$ *vals*(S^n), and such that δ maps the vertices of any given simplex in $\mathcal{P}(S^n)$ to at most k distinct values.

5 Pseudospheres

Informally, a pseudosphere is a combinatorial structure in which each process from a set of processes is independently assigned a value from a set of values.

Definition 3: Let $S^m = (\vec{s}_0,\ldots,\vec{s}_m)$ be a simplex and $U_0,\ldots,U_m$ be a sequence of finite sets. The *pseudosphere* $\psi(S^m;U_0,\ldots,U_m)$ is the following complex. Each vertex is a pair $\langle\vec{s}_i,u_i\rangle$, where $\vec{s}_i$ is a vertex of S^m and $u_i \in U_i$. Vertexes $\langle\vec{s}_{i_0},u_{i_0}\rangle,\ldots,\langle\vec{s}_{i_\ell},u_{i_\ell}\rangle$ span a simplex of $\psi(S^m;U_0,\ldots,U_m)$ if and only if the $\vec{s}_i$ are distinct. A pseudosphere in which all U_i equal to U is simply written $\psi(S^m;U)$.

We call this construct a pseudosphere because if S^n is an n-dimensional simplex, then $\psi(S^n;\{0,1\})$ is homeomorphic to an n-dimensional sphere. Pseudospheres are important because every complex considered here is either a pseudosphere or the union of pseudospheres. Because any process can start with any input from V, the input complex to k-set agreement is the pseudosphere $\psi(P^n;V)$, where P^n is a simplex whose vertexes are labeled with the $n+1$ distinct process ids.

Lemma 4: Pseudospheres satisfy the following simple combinatorial properties.

1. If U is a singleton set, then $\psi(S^m,U) \cong S^m$.
2. Let $S^m = (\vec{s}_0,\ldots,\vec{s}_m)$, and let $S^{m-1} = (\vec{s}_0,\ldots,\widehat{\vec{s}_i},\ldots\vec{s}_m)$, where circumflex denotes omission. If $U_i = \emptyset$, then
$$\psi(S^m;U_0,\ldots,U_m) \cong \psi(S^{m-1};U_0,\ldots,\widehat{U_i},\ldots,U_m).$$
3. If $S_0 \cap S_1 = (\vec{s}_0,\ldots,\vec{s}_\ell)$, then
$$\psi(S_0;U_0,\ldots,U_m) \cap \psi(S_1;V_0,\ldots,V_m) \cong \psi(S_0 \cap S_1;U_0 \cap V_0,\ldots,U_\ell \cap V_\ell).$$

The next theorem shows how to exploit the nice combinatorial properties of pseudospheres. It states that if applying a protocol to a single simplex preserves connectivity below some dimension, then applying that protocol to any input pseudosphere also preserves that degree of connectivity. If we view a protocol complex as a map from simplexes to complexes, then it is actually a theorem in topology, and so it applies to any model of computation.

Theorem 5: Let $\mathcal{P}$ be a protocol, S^m be a simplex, and $c \geq 0$ be a constant. If $\mathcal{P}(S^\ell)$ is $(\ell - c - 1)$-connected for every face S^ℓ of S^m, then $\mathcal{P}(\psi(S^m;U_0,\ldots,U_m))$ is $(m-c-1)$-connected for every sequence $U_0,\ldots,U_m$ of nonempty sets.

A consequence of this theorem is that any n-dimensional pseudosphere is $(n-1)$-connected (just let $\mathcal{P}$ be the identity map, a trivial protocol in which each process halts immediately):

Corollary 6: If $U_0,\ldots,U_m$ are all nonempty, then $\psi(S^m;U_0,\ldots,U_m)$ is $(m-1)$-connected.

Naively, one might think that S^m is always m-connected, but note that although the empty simplex has dimension -1, it is not (-1)-connected. We can generalize Theorem 5 to multiple pseudospheres.

Theorem 7: Let $\mathcal{P}$ be a protocol satisfying the precondition of Theorem 5, and let $A_0,\ldots,A_\ell$ be a sequence of finite sets. If $\cap_{i=0}^{\ell} A_i \neq \emptyset$ then
$$\mathcal{P}\left(\bigcup_{i=0}^{\ell}\psi(S^m;A_i)\right) \text{ is } (m-c-1)\text{-connected.}$$

Letting $\mathcal{P}$ be the identity map again, the trivial protocol in which each process decides its input, we have:

Corollary 8: If $A_0, \ldots, A_\ell$ is a sequence of finite sets such that $\cap_{i=0}^{\ell} A_i \neq \emptyset$ then

$$\bigcup_{i=0}^{\ell} \psi(S^m; A_i) \text{ is } (m-1)\text{-connected.}$$

Our results about k-set agreement are based on the corollary to the following theorem, proved using Sperner's Lemma [Lef49, Lemma 5.5].

Theorem 9: Let $V = \{v_0, \ldots, v_k\}$ be a set of $k+1$ possible input values, let P^n be a simplex with process ids $P_0, \ldots, P_n$, and $\mathcal{P}$ be a protocol with input complex $\psi(P^n; V)$. If $\mathcal{P}$ has the property that for every n-dimensional pseudosphere $\psi(P^n; U)$, where U is a nonempty subset of V, $\mathcal{P}(\psi(P^n; U))$ is $(k-1)$-connected, then $\mathcal{P}$ cannot solve k-set agreement.

Corollary 10: If $\mathcal{P}(S^m)$ is $(m-(n-k)-1)$-connected for all m with $n-f \leq m \leq n$, then $\mathcal{P}$ cannot solve k-set agreement in the presence of f failures.

6 Asynchronous Computation

In this section, we define the r-round asynchronous protocol complex $\mathcal{A}^r(S^m)$. We restrict attention to global states that arise during a set of well-behaved, round-based executions of the full-information protocol defined as follows. In each round, each process sends its state to every other process, receives the messages delivered to it during that round, and undergoes a state transition. Because the model is asynchronous, a message m sent from P to Q in round r may not be delivered in that round. When m is delivered, however, all previously undelivered messages sent from P to Q in rounds 1 through r are delivered at the same time. This means that messages are delivered in FIFO order. In each round, each process receives at least $n-f+1$ messages sent during that round, including its message to itself. This is the greatest number of messages a process can count on receiving when up to f processes can fail. This set of executions looks something like a message-passing analog of the executions arising in the iterated immediate snapshot model defined by Borowsky and Gafni [BG97] for shared memory.

Let $\mathcal{A}^1(S^n)$ be the protocol complex of all one-round, f-faulty, $(n+1)$-process protocol executions with input simplex S^n. Let P be the set of all $n+1$ processes. For any set U, let 2^U denote the power set of U, and let 2^U_k denote the subset of 2^U consisting of sets of size at least k. Our first result says that the one-round protocol complex is a single pseudosphere:

Lemma 11: $\mathcal{A}^1(S^n) \cong \psi(S^n; 2^{P-\{P_0\}}_{n-f}, \ldots, 2^{P-\{P_n\}}_{n-f})$.

Proof: Each vertex in $\mathcal{A}^1(S^n)$ has the form $\langle P_i, M \rangle$, where $P_i \in P$ and M is the set of messages delivered to P_i during the round. Every process receives a message from itself, and also from at least $n-f$ other processes (since at most f processes can fail). Since each process can hear from an independently assigned set of at least $n-f$ other processes, these sets induce a pseudosphere on S^n. Define the vertex map

$$\iota : \mathcal{A}^1(S^n) \to \psi(S^n; 2^{P-\{P_0\}}_{n-f}, \ldots, 2^{P-\{P_n\}}_{n-f})$$

by $\iota(P_i, M) = \langle \vec{s}_i, \mathit{ids}(M) - \{P_i\} \rangle$. It is easy to see that ι is simplicial, one-to-one, and onto. □

Let $\mathcal{A}^r(S^m)$ be the r-round protocol complex defined by induction to be the result of taking the union of $\mathcal{A}^{r-1}(T)$ for every simplex T in $\mathcal{A}^1(S^m)$. (Recall that for $S^m \subset S^n$, $\mathcal{A}^1(S^m)$ is the subcomplex of $\mathcal{A}^1(S^n)$ of executions in which only the processes in $\mathit{ids}(S^m)$ participate, and the remaining processes fail before sending any messages). We can prove that the one-round protocol complex $\mathcal{A}^1(S^m)$ is $(m-(n-f)-1)$-connected for all $n \geq m \geq 0$, and we can iterate this argument to prove that the r-round complex is also highly connected:

Lemma 12: $\mathcal{A}^r(S^m)$ is $(m-(n-f)-1)$-connected.

It follows that the asynchronous protocol complex is $(f-1)$-connected (when $m = n$), and thus we can prove the impossibility of asynchronous k-set agreement [BG93, HS93, SZ93]:

Corollary 13: There is no asynchronous f-resilient k-set agreement protocol for $k \leq f$.

7 Synchronous Computation

We now define the r-round synchronous protocol complex $\mathcal{S}^r(S^m)$. Here too, we consider only a subset of all possible executions: executions in which no more than k processes fail in any round. Informally, we will show that the one-round protocol complex is the union of pseudospheres, where each pseudosphere corresponds to the set of executions in which a fixed set of processes fail. For example, Figure 3 illustrates the possible executions of a one-round protocol for

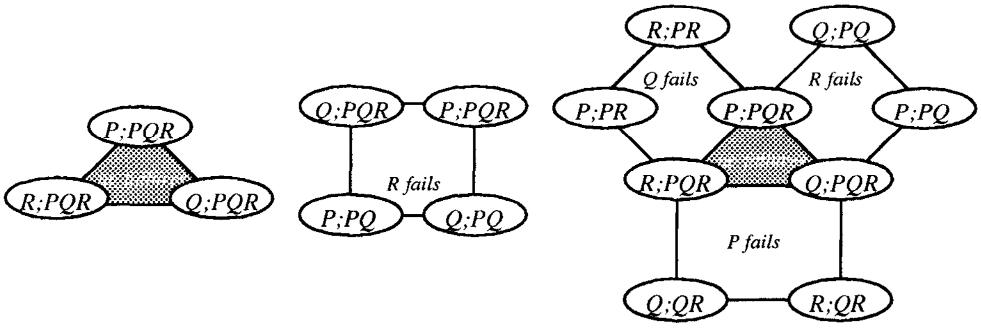

Figure 3: Construction of a one-round three-process protocol complex.

three processes, P, Q, and R, starting from a fixed input simplex, in which no more than one process fails. Here, each vertex is labeled with a process, followed by the processes from which it has received messages. The figure on the left represents the execution in which in which no processes fail: this is a (degenerate) pseudosphere in which each process receives the same set of messages. The figure in the middle represents the executions in which R alone fails. This complex is a pseudosphere: P and Q independently do or do not receive a message from R. The figure on the right represents the entire one-faulty protocol complex. It is the union of the failure-free pseudosphere with the three single-failure pseudospheres.

Let $\mathcal{S}^1(S^n)$ be the complex of one-round executions of an $(n+1)$-process protocol with input simplex S^n in which at most k processes fail. It is the union of complexes $\mathcal{S}^1_K(S^n)$ of one-round executions starting from S^n in which *exactly* the processes in K fail. Given a set K of process ids, let $S^n\backslash K$ be the face of S^n labeled with the process ids *not* in K. Our next result says that $\mathcal{S}^1_K(S^n)$ is a pseudosphere, which means that $\mathcal{S}^1(S^n)$ is a union of pseudospheres:

Lemma 14: $\mathcal{S}^1_K(S^n) \cong \psi(S^n\backslash K; 2^K)$.

Proof: Each vertex in $\mathcal{S}^1_K(S^n)$ has the form $\langle P_i, M\rangle$, where $P_i \in \mathit{ids}(S^n\backslash K)$ and M is a set of messages received from other processes in the round. Every nonfaulty process receives a message from every other nonfaulty process, and also from some subset of faulty processes. Define the vertex map

$$\iota : \mathcal{S}^1_K(S^n) \to \psi(S^n\backslash K; 2^K)$$

by $\iota(P_i, M) = \langle \vec{s}_i, K - \mathit{ids}(M)\rangle$. The vertex map ι is simplicial, one-to-one, and onto. □

Note the similarity between the two models: the one-round complex is a single pseudosphere in the asynchronous model (Lemma 11) and a union of pseudospheres in the synchronous model (Lemma 14). To compute the connectivity of this union using Theorem 2, we need to understand the intersections. The next lemma shows that these intersections have a simple structure: they are themselves the union of pseudospheres. Order the process sets lexicographically: the empty set first, followed by singleton sets, followed by two-element sets, and so on. Let $K_0, \ldots, K_\ell$ be the sequence of sets of process ids less than or equal to K_ℓ, listed in lexicographic order.

Lemma 15:

$$\bigcup_{i=0}^{\ell-1} \mathcal{S}^1_{K_i}(S^n) \cap \mathcal{S}^1_{K_\ell}(S^n) \cong \bigcup_{P\in K_\ell} \psi(S^n\backslash K_\ell; 2^{K_\ell - \{P\}}).$$

Lemma 16: $\mathcal{S}^1(S^m)$ is $(m-(n-k)-1)$-connected if $n \geq 2k$.

Let $\mathcal{S}^r(S^n)$ be the protocol complex of r-round synchronous executions of an $(n+1)$-process protocol with input simplex S^n where no more than than k processes fail in each round. We can decompose this complex as follows. Let $K_0, \ldots, K_\ell$ be the sequence of sets of k or fewer process ids in lexicographic order. Recall that $\mathcal{S}^1_{K_i}(S^n) = \psi(S^n\backslash K_i; 2^{K_i})$ is the complex of one-round executions in which exactly the processes in K_i fail. The set of r-round executions in which exactly the processes in K_i fail in the first round can be written as $\mathcal{S}^{r-1}_i(\mathcal{S}^1_{K_i}(S^n))$, where $\mathcal{S}^{r-1}_i$ is the complex for an $(r-1)$-round, $(f-|K_i|)$-faulty, $(n-|K_i|+1)$-process full-information protocol. The $\mathcal{S}^{r-1}_i$ are considered distinct protocols because the $\mathcal{S}^1_{K_i}(S^n)$ have varying dimensions. Taking the union over all the K_i, we have

$$\mathcal{S}^r(S^n) = \bigcup_{i=0}^{\ell} \mathcal{S}^{r-1}_i(\mathcal{S}^1_{K_i}(S^n)).$$

We can prove an r-round analog of Lemma 15, which yields the following r-round analog of Lemma 16.

Lemma 17: $S^r(S^m)$ is $(m-(n-k)-1)$-connected if $n \geq rk+k$.

The connectivity of this protocol complex implies the lower bound for synchronous k-set agreement [CHLT93]:

Theorem 18: If $n \geq f+k$, then any synchronous f-resilient k-set agreement protocol requires $\lfloor f/k \rfloor + 1$ rounds. If $n < f+k$, then any synchronous f-resilient k-set agreement protocol requires $\lfloor f/k \rfloor$ rounds.

8 Semi-Synchronous Computation

Finally, we define the r-round semi-synchronous protocol complex $\mathcal{M}^r(S^m)$. In this model, the time between two consecutive steps of a process is at least c_1 and at most c_2, and the time to deliver a message is at most d. The values c_1, c_2, and d are known constants, and we define $C = c_2/c_1$.

Once again, we restrict attention to *round-structured* executions defined as follows. Each *round* takes exactly time d. All messages sent during a round are delivered at the very end of that round (at multiples of time d). All processes take steps in lockstep as quickly as possible (at multiples of time c_1). The interval between process steps is called a *microround*, and there are $\mu = \lfloor d/c_1 \rfloor$ microrounds per round. Each message is labeled with the microround in which it was sent. A process P_i'*s view* of a failure pattern at the end of a round is a sequence $\langle \mu_0, \ldots, \mu_n \rangle$, where μ_j is the microround of the last message received from P_j (or 0 if no message was received from P_j).

Consider a set K of processes, and consider all single-round executions in which K is the set of failing processes. A *failure pattern* for K is a function F mapping each process $P_j \in K$ to the microround μ_j in which it fails. At the end of the round, there are a number of views consistent with F, since the last message received by P_i from a process P_j failing in microround μ_j will be sent either in microround μ_j or $\mu_j - 1$. We define $[F]$ to be the set of possible views $\langle \mu_0, \ldots, \mu_n \rangle$ produced by F:

$$\mu_i = \begin{cases} F(P_i) - 1 \text{ or } F(P_i) & \text{if } P_i \in K \\ \mu & \text{otherwise} \end{cases}$$

We define $[F \uparrow j]$ to be the subset of $[F]$ in which P_j's last message is delivered in microround μ_j (and not $\mu_j - 1$). If F is a failure pattern for K, and $j \in K$, then $[F \uparrow j]$ is defined to be the set of views $\langle \mu_0, \ldots, \mu_n \rangle$, where

$$\mu_i = \begin{cases} F(P_i) - 1 \text{ or } F(P_i) & \text{if } P_i \in K - \{j\} \\ F(P_i) & \text{if } i = j \\ \mu & \text{otherwise.} \end{cases}$$

We order the failure patterns for K in reverse lexicographical order: the first pattern fails all processes in K at microround μ, and the last at 0.

Let $\mathcal{M}^1(S^n)$ be the complex of one-round executions of an $(n+1)$-process protocol with input simplex S^n in which at most k processes fail. It is the union of complexes $\mathcal{M}^1_{K,F}(S^n)$ denoting protocol complex of one-round executions starting from S^n in which the processes in K fail with pattern F. The next result says that $\mathcal{M}^1_{K,F}(S^n)$ is a pseudosphere, so the one-round protocol complex is a union of pseudospheres:

Lemma 19: $\mathcal{M}^1_{K,F}(S^n) \cong \psi(S^n \backslash K; [F])$.

Proof: Each vertex in $\mathcal{M}^1_{K,F}(S^n)$ has the form $\langle P_i, M \rangle$, where $P_i \in ids(S^n \backslash K)$ and M is a set of messages received from other processes in the round. This set contains a message from every nonfaulty process in every microround, and a message from every faulty process in some (possibly empty) prefix of microrounds. Each message is labeled with the microround in which it was sent. Let ι map each vertex v of $\mathcal{M}^1_{K,F}(S^n)$ to the vertex v' of $\psi(S^n \backslash K; [F])$ where the j-th component of the view labeling v' is the microround of the last message from P_j in the set of messages labeling v. This is an isomorphism. □

Once again, notice the similarity among the models (Lemmas 11, 14, and 19). In each case, the one-round complex is the union of pseudospheres, where the structure of the union reflects the timing and failure properties of the model. The pseudospheres $\psi(S^n \backslash K; [F])$ forming the one-round complex $\mathcal{M}^1(S^n)$ are lexicographically ordered by the ordering on process sets K and the ordering on failure patterns F, ordering first by K and then by F. Let $\psi(S^n \backslash K_0; [F_0]), \ldots, \psi(S^n \backslash K_\ell; [F_\ell])$ be the sequence of pseudospheres less than or equal to $\psi(S^n \backslash K_\ell; [F_\ell])$, listed in this order. Let $\mathcal{K} = \bigcup_{i=0}^{\ell-1} \psi(S^n \backslash K_i, [F_i])$ and $\mathcal{L} = \psi(S^n \backslash K_\ell; [F_\ell])$.

Lemma 20: $\mathcal{K} \cap \mathcal{L} = \bigcup_{j \in K_\ell} \psi(S^n \backslash K_\ell; [F_\ell \uparrow j])$.

This result says that the intersections of the pseudospheres making up $\mathcal{M}^1(S^n)$ are just the unions of other pseudospheres. This makes is possible to use Theorem 2 to compute the connectivity of their union $\mathcal{M}^1(S^n)$. For the case of one-round protocols, we can prove that one particular union $\mathcal{M}^1(S^m)$

of pseudospheres is $(m-(n-k)-1)$-connected for all $m \leq n$ when $n \geq 2k$. Iterating this construction r times to define the r-round protocol complex $\mathcal{M}^r(S^m)$, using techniques analogous to those used in the synchronous model, we can prove that $\mathcal{M}^r(S^m)$ is also highly-connected:

Lemma 21: $\mathcal{M}^r(S^m)$ is $(m-(n-k)-1)$-connected if $n \geq (r+1)k$.

So far, we have shown that if $n \geq (r+1)k$, the r-round protocol executions in which at at most k processes fail comprise a $(k-1)$-connected complex, which cannot solve k-set agreement. These executions take time rd, which is short. We now show how to "stretch" the final round of this protocol.

In the wait-free case, we have $n = (r+1)k+1$ processes, and we are allowed a "failure budget" of $f = (r+1)k$. We have established that k-set agreement has no decision map on $\mathcal{M}^r(\mathcal{I})$.

Let $\mathcal{M}_\epsilon^{r+1}(\mathcal{I})$ denote the protocol complex at time $(r+1)d-\epsilon$ for $d > \epsilon > 0$, which is just ϵ time before the start of round $r+1$. We claim that k-set agreement has no decision map on $\mathcal{M}_\epsilon^{r+1}(\mathcal{I})$. No process has received a message since time rd, so any decision it could make after waiting $d-\epsilon$ without a message could have been made at time rd, implying that $\mathcal{M}^r(\mathcal{I})$ would a decision map, which it does not.

Let $\mathcal{M}_\infty^{r+1}(\mathcal{I})$ denote the protocol complex corresponding to the following executions: for each vertex $\vec{v}$ in $\mathcal{M}^r(\mathcal{I})$, where $P = id(\vec{v})$, fail all processes but P, and run P as slowly as possible, (taking steps at multiples of time c_2). At time $rd+Cd$, P will time out, but at time $rd+Cd-\epsilon$, this execution is indistinguishable to P from the corresponding execution in $\mathcal{M}_\epsilon^{r+1}(\mathcal{I})$, and therefore $\mathcal{M}_\infty^{r+1}(\mathcal{I})$ has no decision map. We have just shown a worst-case lower bound of time

$$\left(\left\lfloor \frac{f}{k} \right\rfloor - 1\right) d + Cd$$

to solve k-set agreement wait-free with $f = n$ failures.

Corollary 22: Any wait-free protocol for k-set agreement and $n+1$ processes in the semi-synchronous model requires time $\lfloor \frac{n}{k} \rfloor d + Cd$.

As noted above, this corollary is the first substantial new lower bound for the semi-synchronous model since the Attiya, Dwork, Lynch, and Stockmeyer consensus bound of 1993 [ADLS94]. We believe that this result can be extended to the f-resilient case, but this will require further work.

References

[ABND+87] Hagit Attiya, Amotz Bar-Noy, Danny Dolev, Daphne Koller, David Peleg, and Rudiger Reischuk. Achievable cases in an asynchronous environment. In *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science*, pages 337–346, October 1987.

[ABND+90] Hagit Attiya, Amotz Bar-Noy, Danny Dolev, David Peleg, and Rudiger Reischuk. Renaming in an asynchronous environment. *Journal of the ACM*, July 1990.

[ADLS94] Hagit Attiya, Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. Bounds on the time to reach agreement in the presence of timing uncertainty. *Journal of the ACM*, 41(1):122–152, January 1994.

[Awe85] Baruch Awerbuch. Complexity of network synchronization. *Journal of the ACM*, 32:801–823, October 1985.

[BG93] Elizabeth Borowsky and Eli Gafni. Generalized FLP impossibility result for t-resilient asynchronous computations. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, May 1993.

[BG97] Elizabeth Borowsky and Eli Gafni. A simple algorithmically reasoned characterization of wait-free computations. In *Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing*, pages 189–198, August 1997.

[Cha91] Soma Chaudhuri. Towards a complexity hierarchy of wait-free concurrent objects. In *Proceedings of the 3rd IEEE Symposium on Parallel and Distributed Processing.* IEEE, December 1991. Also appeared as Technical Report No. 91-024, Iowa State University, 1991.

[CHLT93] Soma Chaudhuri, Maurice Herlihy, Nancy Lynch, and Mark R. Tuttle. A tight lower bound for k-set agreement. In *Proceedings of the 34th IEEE Symposium on Foundations of Computer Science*, pages 206–215. IEEE, October 1993.

[DM90] Cynthia Dwork and Yoram Moses. Knowledge and common knowledge in a Byzantine environment: Crash failures. *Information and Computation*, 88(2):156–186, October 1990.

[Fis83] Michael J. Fischer. The consensus problem in unreliable distributed systems (a brief survey). In Marek Karpinsky, editor, *Proceedings of the 10th International Colloquium on Automata, Languages, and Programming*, pages 127–140. Springer-Verlag, 1983. A preliminary version appeared as Yale Technical Report YALEU/DCS/RR-273.

[FL82] Michael J. Fischer and Nancy A. Lynch. A lower bound for the time to assure interactive consistency. *Information Processing Letters*, 14(4):183–186, June 1982.

[FLP85] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faulty processor. *Journal of the ACM*, 32(2):374–382, April 1985.

[Gaf98] Eli Gafni. A round-by-round failure detector — unifying synchrony and asynchrony. In *Proceedings of the 17th Annual ACM Symposium on Principles of Distributed Computing*. ACM, June 1998.

[Had83] Vassos Hadzilacos. A lower bound for Byzantine agreement with fail-stop processors. Technical Report TR–21–83, Harvard University, 1983.

[HM90] Joseph Y. Halpern and Yoram Moses. Knowledge and common knowledge in a distributed environment. *Journal of the ACM*, 37(3):549–587, July 1990.

[HS93] Maurice P. Herlihy and Nir Shavit. The asynchronous computability theorem for t-resilient tasks. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, pages 111–120. ACM, May 1993.

[HS97] Gunnar Hoest and Nir Shavit. Towards a topological characterization of asynchronous complexity. In *Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing*, pages 199–208, August 1997.

[Lef49] S. Lefschetz. *Introduction to Topology*. Princeton University Press, Princeton, New Jersey, 1949.

[LL90] Leslie Lamport and Nancy Lynch. Distributed computing: Models and methods. In J. van Leeuwen, editor, *Formal Models and Semantics*, volume B of *Handbook of Theoretical Computer Science*, chapter 19, pages 1157–1199. MITpress, 1990.

[MR98] Yoram Moses and Sergio Rajsbaum. The unified structure of consensus: a layered analysis approach. In *Proceedings of the 17th Annual ACM Symposium on Principles of Distributed Computing*. ACM, June 1998.

[Mun84] J. R. Munkres. *Elements Of Algebraic Topology*. Addison Wesley, Reading MA, 1984.

[PSL80] Marshall Pease, Robert Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228–234, 1980.

[Spa66] Edwin H. Spanier. *Algebraic Topology*. Springer-Verlag, New York, 1966.

[SZ93] Michael Saks and Fotis Zaharoglou. Wait-free k-set agreement is impossible: The topology of public knowledge. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, May 1993.