Universal Protocols for Information Dissemination using Emergent Signals*

Bartłomiej Dudek
Institute of Computer Science, University of Wrocław
Wrocław, Poland
bartlomiej.dudek@cs.uni.wroc.pl

Adrian Kosowski
IRIF and Inria
Paris, France
adrian.kosowski@inria.fr

ABSTRACT

We consider a population of n agents which communicate with each other in a decentralized manner, through random pairwise interactions. One or more agents in the population may act as authoritative sources of information, and the objective of the remaining agents is to obtain information from or about these source agents. We study two basic tasks: *broadcasting*, in which the agents are to learn the bit-state of an authoritative source which is present in the population, and *source detection*, in which the agents are required to decide if at least one source agent is present in the population or not.

We focus on designing protocols which meet two natural conditions: (1) universality, i.e., independence of population size, and (2) rapid convergence to a correct global state after a reconfiguration, such as a change in the state of a source agent. Our main positive result is to show that both of these constraints can be met. For both the broadcasting problem and the source detection problem, we obtain solutions with an expected convergence time of $O(\log n)$, from any starting configuration. The solution to broadcasting is exact, which means that all agents reach the state broadcast by the source, while the solution to source detection admits one-sided error on a ε-fraction of the population (which is unavoidable for this problem). Both protocols are easy to implement in practice and are self-stabilizing, in the sense that the stated bounds on convergence time hold starting from any possible initial configuration of the system.

Our protocols exploit the properties of self-organizing oscillatory dynamics. On the hardness side, our main structural insight is to prove that *any* protocol which meets the constraints of universality and of rapid convergence after reconfiguration must display a form of non-stationary behavior (of which oscillatory dynamics are an example). We also observe that the periodicity of the oscillatory behavior of the protocol, when present, must necessarily depend on the number $^{\#}X$ of source agents present in the population. For instance, our protocols inherently rely on the emergence of a signal passing through the population, whose period is $\Theta(\log \frac{n}{^{\#}X})$ rounds for most starting configurations. The design of phase clocks with tunable frequency may be of independent interest, notably in modeling biological networks.

CCS CONCEPTS

• **Mathematics of computing → Stochastic processes**; • **Theory of computation → Self-organization**;

KEYWORDS

epidemic processes, oscillatory dynamics, population protocols, broadcasting, self-stabilization

ACM Reference Format:
Bartłomiej Dudek and Adrian Kosowski. 2018. Universal Protocols for Information Dissemination using Emergent Signals. In *Proceedings of 50th Annual ACM SIGACT Symposium on the Theory of Computing (STOC'18).* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3188745.3188818

1 INTRODUCTION

Information-spreading protocols, and more broadly epidemic processes, appear in nature, social interactions between humans, as well as in man-made technology, such as computer networks. For some protocols we have a reasonable understanding of the extent to which the information has already spread, i.e., we can identify where the information is located at a given step of the process: we can intuitively say which nodes (or agents) are "informed" and which nodes are "uninformed". This is the case for usual protocols in which uninformed agents become informed upon meeting a previously informed agent, cf. e.g. mechanisms of rumor spreading and opinion spreading models studied in the theory community [25, 27]. Arguably, most man-made networking protocols for information dissemination also belong to this category.

By contrast, there exists a broad category of complex systems for which it is impossible to locate which agents have acquired some knowledge, and which are as yet devoid of it. In fact, the question of "where the information learned by the system is located" becomes somewhat fuzzy, as in the case of both biological and synthetic neural networks. In such a perspective, information (or knowledge) becomes a global property of the entire system, whereas the state of an individual agent represents in principle its *activation*, rather than whether it is informed or not. The convergence from an uninformed population to an informed population over time is far from monotonous. Even so, once some form of "signal" representing global knowledge has emerged, agents may try to read and copy this signal into their local state, thus each of them eventually also becomes informed. At a very informal conceptual level, we refer

*The full version of this paper is available online at: https://arxiv.org/abs/1705.09798. This work was supported by ANR project DESCARTES and by NCN grant 2015/17/B/ST6/01897.

STOC'18, June 25–29, 2018, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5559-9/18/06…$15.00
https://doi.org/10.1145/3188745.3188818

to this category of information-dissemination protocols as protocols with *emergent behavior.* At a more technical level, emergent protocols essentially need to rely on non-linear dynamical effects, which typically include oscillatory behavior, chaotic effects, or a combination of both. (This can be contrasted with simple epidemic protocols for information-spreading, in which nodes do not become deactivated.)

This work exhibits a simple yet fundamental scenario of self-stabilizing information-spreading which can only be addressed efficiently using emergent protocols. The source or sources of information act only as a catalyst in information spreading, boosting an ongoing oscillatory dynamical process in the population to prevent it from stopping. Both the efficient operation of the designed protocols, and the need for non-stationary dynamical effects in any efficient protocol for the considered problems, can be formalized through rigorous theoretical analysis.

1.1 Problems and Model

We consider a population of n identical agents, each of which may be in a *constant* number of possible states. Interactions between agents are pairwise and random. A fair scheduler picks a pair of interacting agents independently and uniformly at random in each step. The protocol definition is provided through a finite sequence of state transition rules, following the precise conventions of the randomized Population Protocol model [5, 8] or (equivalently) of fixed-volume Chemical Reaction Networks [20].[1]

The *input* to the problem is given by fixing the state of some subset of agents, to some state of the protocol, which is not available to any of the other agents. Intuitively, the agents whose state has been fixed are to be interpreted as authoritative sources of information, which is to be detected and disseminated through the network (i.e., as the rumor source node, broadcasting station, etc.). For example, the problem of spreading a bit of information through the system is formally defined below.

Problem BitBroadcast
Input States: X_1, X_2.
Promise: The population contains a non-zero number of agents in exactly one of the two input states $\{X_1, X_2\}$.
Question: Decide if the input state present in the population is X_1 or X_2.

We can, e.g., consider that the transmitting station (or stations) choose whether to be in state X_1 or X_2 in a way external to the protocol, and thus transmit the "bit" value 1 or 2, respectively, through the network. Broadcasting a bit is one of the most fundamental networking primitives.

The definition of the population protocol includes a partition of the set of states of the protocol into those corresponding to the possible answers to the problem. When the protocol is executed on the population, the *output* of each agent may be read at every step by checking, for each agent, whether its state belongs to the subset of an output state with a given answer (in this case, the answer of the agent will be the "bit" it has learned, i.e., 1 or 2). We will call a protocol *exact* if it eventually converges to a configuration, such that starting from this configuration all agents always provide the correct answer. We will say it operates *with ε-error*, for a given constant $\varepsilon > 0$, if starting from some step, at any given step of the protocol, at most an ε-fraction of the population holds the incorrect answer, with probability $1 - O(1/n)$.

Time is measured in *steps* of the scheduler, with n time steps called a *round*, with the expected number of activations of each agent per round being a constant. Our objective is to design protocols which converge to the desired outcome rapidly. Specifically, a protocol is expected to converge in $O(\text{poly} \log n)$ rounds (i.e., in $O(n \text{poly} \log n)$ steps), with probability $1 - O(1/n)$, starting from *any possible starting configuration* of states in the population, conforming to the promise of the problem.[2]

Motivated by both applications and also a need for a better understanding of the broadcasting problem, we also consider a variant of the broadcasting problem in which no promise on the presence of the source is given. This problem, called Detection, is formally defined below.

Problem Detection
Input State: X.
Question: Decide if at least one agent in state X is present in the population.

Detection of the presence of a source is a task which is not easier than broadcasting a bit. Indeed, any detection protocol is readily converted into a broadcasting protocol for states $\{X_1, X_2\}$ by identifying $X = X_1$ and treating X_2 as a dummy state which does not enter into any interactions (i.e., is effectively not visible in the network). Intuitively, the detection task in the considered setting is much harder: a source X may disappear from the network at any time, forcing other agents to spontaneously "unlearn" the outdated information about the presence of the source. This property is inherently linked to the application of the Detection problem in suppressing false rumors or outdated information in social interactions. Specifically, it may happen that a certain part of a population find themselves in an informed state before the original rumor source is identified as a source of false information, a false rumor may be propagated accidentally because of an agent which previously changed state from "uninformed" to "informed" due to a fault or miscommunication, or the rumor may contain information which is no longer true. Similar challenges with outdated information and/or false-positive activations are faced in Chemical Reaction Networks, e.g. in DNA strand displacement models [4]. In that context, the detection problem has the intuitive interpretation of detecting if a given type of chemical or biological agent (e.g., a contaminant, cancer cell, or hormonal signal) is present in the population, and spreading this information among all agents.

1.2 Our Results

In Section 3, we show that both the BitBroadcast, and the Detection problem can be solved with protocols which converge in an expected $O(\log n)$ rounds to a correct outcome, starting from

[1] The activation model is thus asynchronous. The same protocols may be deployed in a synchronous setting, with scheduler activations following, e.g., the independent random matching model (with only minor changes to the analysis) or the PULL model [27] (at the cost of significantly complicating details of the protocol formulation).

[2] We adhere to this strong requirement for self-stabilizing behavior from *any* initial configuration in the design of our protocols. The presented impossibility results still hold under significantly weaker assumptions.

any possible starting configuration of the system (i.e., in a self-stabilizing manner). The solution to BitBroadcast guarantees a correct output. The solution to Detection admits one-sided ε-error: in the absence of a source, all agents correctly identify it as absent, whereas when the source is present, at any moment of time after convergence the probability that at least $(1-\varepsilon)n$ agents correctly identify the source as present is at least $1 - O(1/n)$.[3] Here, $\varepsilon > 0$ is a constant influencing the protocol design, which can be made arbitrarily small.

They rely on the same basic building block, namely, a protocol realizing oscillatory dynamics at a rate controlled by the number of present source states in the population. Thus, these protocols display non-stationary behavior. In Section 4, we show that such behavior is a necessary property in the following sense. We prove that in any protocol which solves Detection in sub-polynomial time in n and which uses a constant number of states, the number of agents occupying some state has to undergo large changes: by a polynomially large *factor* in n during a time window of length proportional to the convergence time of the protocol. For the BitBroadcast problem, we show that similar volatile behavior must appear in a synthetic setting in which a unique source is transmitting its bit as random noise (i.e., selecting its input state $\{X_1, X_2\}$ uniformly at random in subsequent activations).

We note that, informally speaking, our protocols rely on the emergence of a "signal" passing through the population, whose period is $\Theta(\log \frac{n}{^{\#}X})$ rounds when the number of source agents in state X is $^{\#}X$. In Section 5, we then discuss how the behavior of *any* oscillatory-type protocol controlled by the existence of $^{\#}X$ has to depend on both n and $^{\#}X$. We prove that for any such protocol with rapid convergence, the cases of subpolynomial $^{\#}X$ and $^{\#}X = \Theta(n)$ can be separated by looking at the portion of the configuration space regularly visited by the protocol. This, in particular, suggests the nature of the dependence of the oscillation period on the precise value of $^{\#}X$, and that the protocols we design with period $\Theta(\log \frac{n}{^{\#}X})$ are among the most natural solutions to the considered problems.

The proofs of all theorems are deferred to the closing sections of the paper.

1.3 Comparison to the State-of-the-Art

Our work fits into lines of research on rumor spreading, opinion spreading, population protocols and other interaction models, and emergent systems. We provide a more comprehensive literature overview of some of these topics in Subsection 1.4.

Other work on the problems. The BitBroadcast problem has been previously considered by Boczkowski, Korman, and Natale in [12], in a self-stabilizing but synchronous (round-based) setting. A protocol solving their problem was presented, giving a stabilized solution in $\tilde{O}(\log n)$ time, using a number of states of agents which *depends on population size*, but exchanging messages of bit size $O(1)$ (this assumption can be modeled in the population protocol framework as a restriction on the permitted rule set). In this sense, our result can be seen as providing improved results with respect to their approach, since it is applicable in an asynchronous setting and reduces the number of states to constant (the latter question was not resolved in [12] and posed as open [11]). We remark that their protocol has a more general application to the problem of deciding which of the sources X_1 or X_2 is represented by a larger number of agents, provided these two numbers are separated by a multiplicative constant. (Our approach could also be used in such a setting, but the required separation of agent numbers to ensure a correct output would have to be much larger: we can compare values if their *logarithms* are separated by a multiplicative constant.) The protocol of [12] involves a routine which allows the population to create a synchronized modulo clock, working in a synchronous setting. The period of this clock is independent of the input states of the protocol, which should be contrasted with the oscillators we work on in this paper, which encode the input into a signal (with a period depending on the number of agents in a given input state).

The Detection problem was introduced in a work complementary to this paper [4]. Therein we look at applications of the confirmed rumor spreading problem in DNA computing, focusing on performance on protocols based on a time-to-live principle and on issues of fault tolerance in a real-world model with leaks. The protocols designed there require $O(\log n)$ states, and while self-stabilizing, do not display emergent behavior (in particular, agents can be categorized as "informed" and "uninformed", the number of correctly informed agents tends to increase over time, while the corresponding continuous dynamical system stabilizes to a fixed point attractor).

Originality of methods. The oscillatory dynamics we apply rely on an *input-parameter-controlled* oscillator. The *uncontrolled* version of the oscillator which we consider here is the length-3 cyclic oscillator of the cyclic type, known in population dynamics under the name of rock-paper-scissors (or RPS). This has been studied intensively in the physics and evolutionary dynamics literature (cf.e.g. [44] for a survey), while algorithmic studies are relatively scarce [17]. We remark that the uncontrolled cyclic oscillator with a longer (but $O(1)$ length cycle) has been applied for clock/phase synchronization in self-stabilizing settings, in variants of the population protocol model which involve a predetermined leader [6], in extensions of the population protocol model which allow a very large number of states [15], and very recently also in the standard population protocol setting when resolving the leader election problem [24]. (In the latter work, the connection to oscillatory dynamics is not made explicit, and the longer cycle provides for a neater analysis, although it does not seem to be applicable to our parameter-controllable setting.) Whereas we are not aware of any studies of parameter-controlled oscillators in a protocol design setting (nor for that matter, of rigorous studies in other fields), we should note that such oscillators have frequently appeared in models of biological systems, most notably in biological networks and neuroendocrinology ([26] for a survey). Indeed, some hormone release and control mechanisms (e.g., for controlling GnRH surges in vertebrates) appear to be following a similar pattern. To the best of our knowledge, no *computational* (i.e., interaction-protocol-based) explanation for these mechanisms has yet been proposed, and we

[3] The existence of some form of one-sided error is inherent to the Detection problem in the asynchronous setting: indeed, if no agent of the population has not made any communication with the source over an extended period of time, it is impossible to tell for sure if the source has completely disappeared from the network, or if it is not being selected for interaction by the random scheduler.

Table 1: Summary of results and comparison with the state-of-the-art.

Problem:	BitBroadcast	Detection
Non-stationarity property: (Applies to all fast $O(1)$-state protocols)	no fixed points while source transmits random bits	no fixed points while source is present
New protocols with emergent signal:	universal, 74 states	universal, 55 states
Expected convergence time:		
– No error (exact output)	$O(\log n)$	impossible
– One-sided ε-error		$O(\log n)$
Other self-stabilizing protocols with $\omega(1)$-states:	Clock-Sync (in synchronized round model) using $O(\log n)$ states [12]	Time-to-Live using $O(\log n)$ states [4]

hope that our work may provide, specifically on the Detection problem, may provide some insights in this direction.

In terms of lower bounds, we rely on rather tedious coupling techniques for protocols allowing randomization, and many of the details are significantly different from lower-bound techniques found in the population protocols literature. We remark that a recent line of work in this area [2, 20] provides a powerful set of tools for proving lower bounds on the number of states (typically $\Omega(\log\log n)$ states) for fast (typically polylogarithmic) population protocols for different problems, especially for the case of deterministic protocols. We were unable to leverage these results to prove our lower bound for the randomized scenario studied here, and believe our coupling analysis is complementary to their results.

1.4 Other Related Work

Our work fits into the line of research on rumor spreading, population protocols, and related interaction models. Our work also touches on the issue of how distributed systems may spontaneously achieve some form of coordination with minimum agent capabilities. The basic work in this direction, starting with the seminal paper [31], focuses on synchronizing timers through asynchronous interprocess communication to allow processes to construct a total ordering of events. A separate interesting question concerns local clocks which, on their own, have some drift, and which need to synchronize in a network environment (cf. e.g. [33, 35], or [32] for a survey of open problems).

Rumor spreading. Rumor spreading protocols are frequently studied in a synchronous setting. In a synchronous protocol, in each parallel round, each vertex independently at random activates a local rule, which allows it either to spread the rumor (if it is already informed), or possibly also to receive it (if it has not yet been informed, as is the case in the push-pull model). The standard push rumor spreading model assumes that each informed neighbor calls exactly one uninformed neighbor. In the basic scenario, corresponding to the complete interaction network, the number of parallel rounds for a single rumor source to inform all other nodes is given as $\log_2 n + \ln n + o(\log n)$, with high probability [23, 40]. More general graph scenarios have been studied in [21] in the context of applications in broadcasting information in a network. Graph classes studied for the graph model include hypercubes [21], expanders [43], and other models of random graphs [22]. The push-pull model of rumor spreading is an important variation: whereas for complete networks the speedup due to the pull process is in the order of a multiplicative constant [27], the speed up turns out to be asymptotic, e.g., on preferential attachment graphs, where the rumor spreading time is reduced from $\Theta(\log n)$ rounds in the push model to $\Theta(\log n/\log\log n)$ rounds in the push-pull model [19], as well as on other graphs with a non-uniform degree distribution. The push-pull model often also proves more amenable to theoretical analysis. We note that asynchronous rumor spreading on graphs, in models closer to our random scheduler, has also been considered in recent work [25, 38], with [25] pointing out the tight connections between the synchronous (particularly push-pull) and asynchronous models in general networks.

Population protocols. Population protocols are a model which captures the way in which the complex behavior of systems (biological, sensor nets, etc.) emerges from the underlying local interactions of agents. The original model of Angluin et al. [5, 7] was motivated by applications in sensor mobility. Despite the limited computational capabilities of individual sensors, such protocols permit at least (depending on available extensions to the model) the computation of two important classes of functions: threshold predicates, which decide if the weighted average of types appearing in the population exceeds a certain value, and modulo remainders of such weighted averages. The majority function, which belongs to the class of threshold functions, was shown to be stably computable for the complete interaction graph [5]; further results in the area of majority computation can be found in [7, 9, 10, 36]. A survey of applications and models of population protocols is provided in [9, 37]. An interesting line of research is related to studies of the algorithmic properties of dynamics of chemical reaction networks [20]. These are as powerful as population protocols, though some extensions of the chemical reaction model also allow the population size to change in time. Two very recent results in the population protocol model are worthy of special attention. Alistarh, Aspnes, and Gelashvili [3] have resolved the question of the number of states required to solve the Majority problem on a complete network in polylogarithmic time as $\Theta(\log n)$. For the equally notable task of Leader Election, the papers of Gasieniec and Stachowiak [24] (for the upper bound) together with the work of Alistarh, Aspnes, Eisenstat, Gelashvili, and Rivest [2] (for the lower bound) put the number of states required to resolve this question in polylogarithmic time as $\Theta(\log\log n)$. Both of these results rely on a notion of a self-organizing phase clock.[4]

Nonlinearity in interaction protocols. Linear dynamical systems, as well as many nonlinear protocols subjected to rigorous analytical study, have a relatively simple structure of point attractors and repellers in the phase space. The underlying continuous dynamics (in the limit of $n \to +\infty$) of many interaction protocols defined for complete graphs would fit into this category: basic models of randomized rumor spreading [40]; models of opinion propagation (e.g. [1, 14]); population protocols for problems such as majority and thresholds [5, 7]; all reducible Markov chain processes, such as random walks and randomized iterative load balancing schemes.

Nonlinear dynamics with non-trivial limit orbits are fundamental to many areas of systems science, including the study of physical, chemical and biological systems, and to applications in control science. In general, population dynamics with interactions between pairs of agents are non-linear (representable as a set of quadratic difference equations) and have potentially complicated structure if the number of states is 3 or more. For example, the simple continuous Lotka-Volterra dynamics [34] gives rise to a number of discrete models, for example one representing interactions of the form $A + B \to A + A$, over some pairs A, B of states in a population (cf. [44] for further generalizations of the framework or [17] for a rigorous analysis in the random scheduler model). The model describes transient stability in a setting in which several species are in a cyclic predator-prey relation. Cyclic protocols of the type have been consequently identified as a potential mechanism for describing and maintaining biodiversity, e.g., in bacterial colonies [28, 29]. Cycles of length 3, in which type A_2 attacks type A_1, type A_3 attacks type A_2, and type A_1 attacks type A_3, form the basis of the basic oscillator, also used as the starting point for protocols in this work, which is referred to as the RPS (rock-paper-scissors) oscillator or simply the 3-cycle oscillator. This protocol has been given a lot of attention in the statistical physics literature. The original analytical estimation method applied to RPS was based on approximation with the Fokker-Planck equation [42]. A subsequent analysis of cyclic 3- and 4-species models using Khasminskii stochastic averaging can be found in [18], and a mean field approximation-based analysis of RPS is performed in [39]. In [17], we have performed a study of some algorithmic implications of RPS, showing that the protocol may be used to perform randomized choice in a population, promoting minority opinions, in $\tilde{O}(n^2)$ steps. All of these results provide a good qualitative understanding of the behavior of the basic cyclic protocols. We remark that the protocol used in this paper is directly inspired by the properties of RPS, as we discuss further on, but has a more complicated interaction structure (see Fig. 1).

For protocols with convergence to a single point in the configuration space in the limit of large population size, a discussion of the limit behavior is provided in [13], who provide examples of protocols converging to limit points at coordinates corresponding to any algebraic numbers.

We also remark that local interaction dynamics on arbitrary graphs (as opposed to the complete interaction graph) exhibit a much more complex structure of their limit behavior, even if the graph has periodic structure, e.g., that of a grid. Oscillatory behavior may be overlaid with spatial effects [44], or the system may have an attractor at a critical point, leading to simple dynamic processes displaying self-organized criticality (SOC, [41]).

2 PRELIMINARIES: BUILDING BLOCKS FOR POPULATION PROTOCOLS

2.1 Protocol Definition

A randomized *population protocol* for a population of n agents is defined as a pair $P = (K_n, R_n)$, where K_n is the set of *states* and R_n is the set of *interaction rules*. The interaction graph is complete. We will simply write $P = (K, R)$, when considering a protocol which is *universal* (i.e., defined in the same way for each value of n) or if the value of n is clear from the context. All the protocols we design are universal; our lower bounds also apply to some non-universal protocols. The set of rules $R \subseteq K^4 \times [0, 1]$ is given so that each rule $j \in R$ is of the form $j = (i_1(j), i_2(j), o_1(j), o_2(j), q_j)$, describing an interaction read as: "$(i_1(j), i_2(j)) \to (o_1(j), o_2(j))$with probability q_j". For all $i_1, i_2 \in K$, we define $R_{i_1,i_2} = \{j \in R : (i_1(j), i_2(j)) = (i_1, i_2)\}$ as the set of rules acting on the pair of states i_1, i_2, and impose that $\sum_{j\in R_{i_1,i_2}} q_j \le 1$.

For a state $A \in K$, we denote the number of agents in state A as $^{\#}A$, and the *concentration* of state A as $a = {}^{\#}A/n$, and likewise for a set of states $\mathcal{A}$, we write $^{\#}\mathcal{A} = \sum_{A\in\mathcal{A}} {}^{\#}A$.

In any configuration of the system, each of the n agents from the population is in one of the states from K_n. The protocol is executed by an asynchronous scheduler, which runs in *steps*. In every step the scheduler uniformly at random chooses from the population a pair of distinct agents to interact: the *initiator* and the *receiver*. If the initiator and receiver are in states i_1 and i_2, respectively, then the protocol executes at most one rule from set protocol R_{i_1,i_2}, selecting rule $j \in R_{i_1,i_2}$ with probability q_j. If rule j is executed, the initiator then changes its state to $o_1(j)$ and the receiver to $o_2(j)$. The source has a special state, denoted X in the Detection problem, or one of two special states, denoted $\{X_1, X_2\}$ in the BitBroadcast problem, which is never modified by any rule.

All protocols are presented in the randomized framework, in which transition probabilities are some rational absolute constants (independent of population size). The universal protocols considered here are amenable to a form of conversion into deterministic rules discussed in [2], which simulates randomness of rules by exploiting the inherent randomness of the scheduler in choosing interacting node pairs to distribute weakly dependent random bits around the system.

All protocols designed in this work are *one-way* (cf. [8]), which means that for any rule $j \in R$, we have $o_1(j) = i_1(j)$ (i.e., have all rules of the form $A + B \to A + C$, also more compactly written as $A\colon\ B \to C$), which makes them relevant in a larger number of application. As an illustrative example, we remark that the basic rumor spreading (epidemic) model is one-way and given simply as $1\colon\ 0 \to 1$. All protocols can also obviously be rewritten to act

[4] For the sake of precision, we prefer to use the term *self-organizing*, rather than *self-stabilizing*, when referring to properties of a system which need to be observed over time and cannot be inferred by observing any specific configuration (snapshot) of the population.

on unordered pairs of agents picked by the scheduler, rather than ordered pairs.

2.2 Protocol Composition Technique

Our protocols will be built from simpler elements. Our basic building block is the input-controlled oscillatory protocol P_o (see Fig. 1). We then use protocol P_o as a component in the construction of other, more complex protocols, without disrupting the operation of the original protocol.

Formally, we consider a protocol P_B using state set $B = \{B_i : 1 \le i \le k_b\}$ and rule set R_B, and a *protocol extension* P_{BC} using a state set $B \times C = B \times \{C_i : 1 \le i \le k_c\}$, where C is disjoint from B, and *rule extension* set R_{BC}. Each rule extension defines for each pair of states from $B \times C$ (i.e., for each element of $(B \times C) \times (B \times C)$) a probability distribution over elements of $C \times C$.

The *composed protocol* $P_B \circ P_{BC}$ is a population protocol with set of states $B \times C$. Its rules are defined so that, for a selected pair of agents in states (B_i, C_j) and $(B_{i'}, C_{j'})$, we obtain a pair of agents in states (B_{i^*}, C_{j^*}) and $(B_{i'^*}, C_{j'^*})$ according to a probability distribution defined so that:

- Each pair $B_{i^*}, B_{i'^*}$ appears in the output states of the two agents with the same probability as it would in an execution of protocol P_B on a pair of agents in states B_i and $B_{i'}$.
- Each pair $C_{j^*}, C_{j'^*}$ appears in the output states of the two agents with the probability given by the definition of P_{BC}.

In the above, the pairs of agents activated by P_B and P_{BC} are not independent of each other. This is a crucial property in the composition of protocol P_o when composing it with further blocks to solve the Detection problem.

We denote by $\mathbf{1}_B$ the identity protocol which preserves agent states on set of states B. For a protocol P, we denote by $P/2$ a lazy version of a protocol P in which the rule activation of P occurs with probability $1/2$, and with probability $1/2$ the corresponding rule of the identity protocol is activated. Note that all asymptotic bounds on expected and w.h.p. convergence time obtained for any protocol P also apply to protocol $P/2$, in the regime of at least a logarithmic number of rounds. We also sometimes treat a protocol extension P_{BC} as a protocol in itself, applied to the identity protocol $\mathbf{1}_B$.

The *independently composed protocol* $P_B + P_{BC}$ is defined as an implementation of the composed protocol $(P_B/2) \circ (P_{BC}/2)$, realized with the additional constraint that in each step, either the rule of P_B is performed with an identity rule extension, or the rule extension of P_{BC} is performed on top of the identity protocol $\mathbf{1}_B$. Such a definition is readily verified to be correct by a simple coupling argument (akin to arguments used when analysing coupled random walks), and allows us to analyze protocols P_B and P_{BC}, observing that the pairs (identities) of agents activated by the scheduler in the respective protocols are independent.

All the composed protocols (and protocol extensions) we design are also one-way, i.e., $C_{j^*} = C_j$ and $B_{i^*} = B_i$, with probability 1. In notation, rules omitted from the description of protocol extensions are implicit, occurring with probability 0 (where $C_{j^*} \neq C_j$) or with the probability necessary for the normalization of the distribution to 1, where the state is preserved (where $C_{j^*} = C_j$).

As a matter of naming convention, we name the states in the separate state sets of the composed protocols with distinct letters of the alphabet, together with their designated subscripts and superscripts. The rumor source X is treated specially and uses a separate letter (and may be seen as a one state protocol without any rules, on top of which all other protocols are composed; in particular, its state is never modified). The six remaining states of protocol P_o are named with the letters $A_?^?$, as usual in its definition. Subsequent protocols will use different letters, e.g., $M_?$ and $L_?$.

3 OVERVIEW OF PROTOCOL DESIGNS

3.1 Main Routine: Input-Controlled Oscillator Protocol P_o

We first describe the main routine which allows us to convert local input parameters (the existence of the source) into a form of global periodic signal on the population. This main building block is the construction of a 7-state protocol P_o following oscillatory dynamics, whose design we believe to be of independent interest.

The complete design of protocol P_o is shown in Fig. 1. The source state is denoted by X. Additionally, there are six states, called A_i^+ and A_i^{++}, for $i \in \{1, 2, 3\}$. The naming of states in the protocol is intended to maintain a direct connection with the dynamics of the RPS oscillator, which is defined by the simple rule "$A_i : \quad A_{i-1} \mapsto A_i$, for $i = 1, 2, 3$". In fact, we will retain the convention $A_i = \{A_i^+, A_i^{++}\}$ and $a_i = a_i^+ + a_i^{++}$, and consider the two states A_i^+ and A_i^{++} to be different flavors of the same species A_i, referring to the respective superscripts as either lazy ($^+$) or aggressive ($^{++}$).

The protocol has the property that in the absence of X, it stops in a corner state of the phase space, in which only one of three possible states appears in the population, and otherwise regularly (every $O(\log n)$ rounds) moves sufficiently far away from all corner states. An intuitive formalization of the basic properties of the protocol is given by the theorem below.

Theorem 3.1. *There exists a universal protocol P_o with $|K| = 7$ states, including a distinguished source state X, which has the following properties.*

1. *For any starting configuration, in the absence of the source ($^\#X = 0$), the protocol always reaches a configuration such that:*
   - *all agents are in the same state: either A_1^{++}, or A_2^{++}, or A_3^{++};*
   - *no further state transitions occur after this time.*

   Such a configuration is reached in $O(\log n)$ rounds, with constant probability (thus: in $O(\log n)$ rounds in expectation, and in $O(\log^2 n)$ rounds with probability $1 - O(1/n)$).
2. *For any starting configuration, in the presence of the source ($^\#X \ge 1$), we have with probability $1 - O(1/n)$:*
   - *for each state $i \in K \setminus \{X\}$, there exists a time step in the next $O(\log \frac{n}{^\#X})$ rounds when at least a constant fraction of all agents are in state i;*
   - *during the next $O(\log \frac{n}{^\#X})$ rounds, at least a constant fraction of all agents change their state at least once.*

All omitted proofs can be found in the full version of the paper.

The RPS dynamics provides the basic oscillator mechanism which is still largely retained in our scenario. Most of the difficulty lies in controlling its operation as a function of the presence or absence of the rumor source. We do this by applying two separate mechanisms.

(1) Interaction with an initiator from the same species makes receiver aggressive:

$$A_i^?: \quad A_i^? \quad \mapsto \quad A_i^{++}$$

(2) Interaction with an initiator from a different species makes receiver lazy (case of no attack):

$$A_i^?: \quad A_{i+1}^? \mapsto \quad A_{i+1}^{+}$$

(3) A lazy initiator has probability p of performing a successful attack on its prey:

$$A_i^+: \quad A_{i-1}^? \mapsto \begin{cases} A_i^+, & \text{with probability } p, \\ A_{i-1}^+, & \text{otherwise.} \end{cases}$$

(4) An aggressive initiator has probability $2p$ of performing a successful attack on its prey:

$$A_i^{++}: \quad A_{i-1}^? \mapsto \begin{cases} A_i^+, & \text{with probability } 2p, \\ A_{i-1}^+, & \text{otherwise.} \end{cases}$$

(5) The source converts any receiver into a lazy state of a uniformly random species:

$$X: \quad A_?^? \quad \mapsto \begin{cases} A_1^+, & \text{with probability } 1/3, \\ A_2^+, & \text{with probability } 1/3, \\ A_3^+, & \text{with probability } 1/3. \end{cases}$$

Figure 1: Rules of the basic oscillator protocol P_o. The adopted notation for one-way rules is of the form $A\colon\ B \mapsto C$, corresponding to transitions written as $A + B \to A + C$ in the notation of chemical reaction networks or $(A, B) \to (A, C)$ in the notation of population dynamics. All rules apply to $i \in \{1, 2, 3\}$, whereas a question mark $?$ in a superscript or subscript denotes a wildcard, matching any permitted combination of characters, which may be set independently for each agent. Probability $p > 0$ is any (sufficiently small) absolute constant.

The presence of rumor source X shifts the oscillator towards an orbit closer to the central orbit $(A_1, A_2, A_3) = (1/3, 1/3, 1/3)$ through rule (5), which increases the value of potential $\phi := \ln(a_1 a_2 a_3)$, where $a_i = {}^{\#}A_i/n$. Conversely, independent of the existence of rumor source X, a second mechanism is intended to reduce the value of potential ϕ. This mechanism exploits the difference between the aggressive and lazy flavors of the species. Following rule (1), an agent belonging to a species becomes more aggressive if it meets another from the same species, and subsequently attacks agents from its prey species with doubled probability following rule (4). This behavior somehow favors larger species, since they are expected to have (proportionally) more aggressive agents than the smaller species (in which pairwise interactions between agents of the same species are less frequent) — the fraction of agents in A_i which are aggressive would, in an idealized static scenario, be proportional to a_i. (This is, in fact, often far from true due to the interactions between the different aspects of the dynamics). As a very loose intuition, the destabilizing behavior of the considered rule on the oscillator is resemblant of the effect an eccentrically fitted weight has on a rotating wheel, pulling the oscillator towards more external orbits (with smaller values of ϕ).

The intuition for which the proposed dynamics works, and which we will formalize and prove rigorously in the full version of the paper, can now be stated as follows: in the presence of rumor source X, the dynamics will converge to a form of orbit on which the two effects, the stabilizing and destabilizing one, eventually compensate each other (in a time-averaged sense). The period of a single rotation of the oscillator around such an orbit is between $O(1)$ and $O(\log n)$, depending on the concentration of X. In the absence of X, the destabilizing rule will prevail, and the oscillator will quickly crash into a side of the triangle.

For small values of ${}^{\#}X > 0$, the protocol can be very roughly (and non-rigorously) viewed as cyclic composition of three dominant rumor spreading processes over three sets of states A_1, A_2, A_3, one converting states A_1 to A_3, the next from A_3 to A_2, and the last from A_2 to A_1, which spontaneously take over at moments of time separated by $O(\log n)$ parallel rounds. For other starting configurations, and especially for the case of ${}^{\#}X = 0$, the dynamics of the protocol, which has 5 free dimensions, is more involved to describe and analyze. In the full version of the paper, we provide some further insights into the operation of the protocol, notably formalizing the notion that an intuitively understood oscillation (going from a small number of agents in some state A_i, to a large number of agents in state A_i, and back again to a small number of agents in state A_i) takes $\Theta(\log \frac{n}{{}^{\#}X})$ rounds, with probability $1 - O(1/n)$. As such, protocol P_o can be seen as converting *local input* ${}^{\#}X$ into a *global periodic signal* with period $\Theta(\log \frac{n}{{}^{\#}X})$. What remains is allowing nodes to extract information from this periodic signal.

Simulation timelines shown in Fig. 6 illustrate the idea of operation of protocol P_o and its composition with other protocols.

We remark that in the case of ${}^{\#}X = 0$, the time for the protocol to stabilize is by the oscillator subroutine is given as $O(\log n)$ rounds only with constant probability. Due to the fact that we exploit the variance of the random choices made by the scheduler in the analysis, this does not readily extend to a comparable w.h.p. analysis. Using the fact that this result holds for any starting configuration of the process, we have for any positive integer a that the process stabilizes in $O(a \log n)$ rounds with probability at least $1 - 2^{-\Omega(a)}$. Consequently, the expected convergence time of the process is $O(\log n)$, and the best w.h.p. bound we know (which appears tight) is $O(\log^2 n)$ rounds with probability $1 - O(1/n)$, as stated in the theorem. The analysis we perform is largely insensitive to the details of the scheduler and may be performed similarly for other fair random scheduler models (including parallel schedulers).

3.2 Protocols for BitBroadcast

A solution to BitBroadcast is obtained starting with an independent composition of two copies of oscillator P_o, called $P_{o[1]}$ and $P_{o[2]}$, with states in one protocol denoted by subscript [1] and in the other by subscript [2]. The respective sources are thus written as $X_{[1]}$ and $X_{[2]}$. In view of Theorem 3.1, in this composition $P_{o[1]} + P_{o[2]}$, under the promise of the BitBroadcast problem, one of the oscillators will be running and the other will stop in a corner of its state space. Which of the oscillators is running can be identified by the presence of states $A_i^+[z]$, which will only appear for $z \in \{1, 2\}$ corresponding to the operating oscillator. Moreover, by the same Theorem, every $O(\log n)$ rounds a constant fraction of agents of this oscillator will be in such a state $A_i^+[z]$, for any choice of $i \in \{1, 2, 3\}$. We can thus design the protocol extension P_b to detect this. This is given by the pair of additional output states $\{Y_1, Y_2\}$ and the rule extension consisting of the two rules shown in Fig. 2.

Theorem 3.2 (Protocol for BitBroadcast). *Protocol $(P_{o[1]} + P_{o[2]}) + P_b$, having $|K| = 74$ states, including distinguished source states $X_{[1]}$, $X_{[2]}$ converges to an exact solution of BitBroadcast. This occurs in $O(\log n)$ parallel rounds in expectation. In the output encoding, agent states of the form $(\cdot, \cdot, Y_z)$ represent answer "z", for $z \in \{1, 2\}$.*

The protocol $(P_{o[1]} + P_{o[2]}) + P_b$ is not "silent", i.e., it undergoes perpetual transitions of state, even once the output has been decided. As a side remark, we note that for the single-source broadcasting problem, or more generally for the case when the number of sources is small, $\max\{{}^{\#}X_{[1]}, {}^{\#}X_{[2]}\} = O(1)$, we can propose the following simpler silent protocol. We define protocol P'_o, by modifying protocol P_o as follows. We remove from it Rule (5), and replace it by to the four rules shown in Fig. 3. The analysis of the modified protocol follows from the same arguments as those used to prove Theorem 3.1(1). In the regime of $\max\{{}^{\#}X_{[1]}, {}^{\#}X_{[2]}\} = O(1)$, the effect of the source does not influence the convergence of the process and each of the three possible corner configurations, with exclusively species $\{A_1, A_2, A_3\}$, is reached in $O(\log n)$ rounds with constant probability. However, rules $(5a)$ – $(5d)$ enforce that the only stable configuration which will persist is the one in a corner corresponding to the identity of the source, i.e., A_1 for source $X_{[1]}$ and A_2 for source $X_{[2]}$; the source will restart the oscillator in all other cases. We thus obtain the following side result, for which we leave out the details of the proof.

Observation 1. *Protocol P'_o, having $|K| = 6 + 2 = 8$ states, including distinguished source states $X_{[1]}$, $X_{[2]}$ converges to an exact solution of BitBroadcast, eventually stopping with all agents in state A_1^{++} if source $X_{[1]}$ is present and stopping with all agents in state A_2^{++} if source $X_{[2]}$ is present, with no subsequent state transition. The stabilization occurs within $O(\log n)$ parallel rounds in expectation if $\max\{{}^{\#}X_{[1]}, {}^{\#}X_{[2]}\} = O(1)$, i.e., the broadcast originates from a constant number of sources.*

3.3 Protocol for Detection

The solution to problem Detection is more involved. It relies on two auxiliary extensions added on top of a single oscillator P_o. The first, P_m, runs an instance of the 3-state majority protocol of Angluin et al. [7] *within* each species A_i of the oscillator. For this reason, the composition between P_o and P_m has to be of the form $P_o \circ P_m$ (i.e., it cannot be independent). The operation of this extension is shown in Fig. 4 and analyzed in the full version of the paper. It relies crucially on an interplay of two parameters: the time $\Theta(\log \frac{n}{{}^{\#}X})$ taken by the oscillator to perform an orbit, and the time $\Omega(\log \frac{n}{{}^{\#}X})$ it takes for the majority protocol (which is reset by the oscillator in its every oscillation) to converge to a solution. When parameters are tuned so that the second time length is larger a constant number of times than the first, a constant proportion of the agents of the population are involved in the majority computation, i.e., both of the clashing states in the fight for dominance still include $\Omega(n)$ agents. In the absence of a source, shortly after the oscillator stops, one of these states takes over, and the other disappears.

The above-described difference can be detected by the second, much simpler, extension P_l, designed in Fig. 5 and analyzed in the full version of the paper. The number of "lights" switched on during the operation of the protocol will almost always be more than $(1 - \varepsilon)n$, where $\varepsilon > 0$ is a parameter controlled by the probability of lights spontaneously disengaging, and may be set to an arbitrarily small constant.

Theorem 3.3 (Protocol for Detection). *For any $\varepsilon > 0$, protocol $(P_o \circ P_m) + P_l$, having $|K| = 6 \cdot 3 \cdot 3 + 1 = 55$ states, including a distinguished source state X, which solves the problem of spreading confirmed rumors as follows:*

1. *For any starting configuration, in the presence of the source (${}^{\#}X \geq 1$), after an initialization period of $O(\log n)$ rounds, at an arbitrary time step the number of agents in an output state corresponding to a "yes" answer is $(1 - \varepsilon)n$, with probability $1 - O(1/n)$.*
2. *For any starting configuration, in the absence of the source (${}^{\#}X = 0$), the system always reaches a configuration such that all agents are in output states corresponding to a "no" answer for all subsequent time steps. Such a configuration is reached in $O(\log n)$ rounds, in expectation.*

4 IMPOSSIBILITY RESULTS FOR PROTOCOLS WITHOUT NON-STATIONARY EFFECTS

For convenience of notation, we identify a configuration of the population with a vector $z = (z^{(1)}, \ldots, z^{(k)}) \in \{0, 1, \ldots, n\}^k = Z$, where $z^{(i)}$, for $1 \leq i \leq k$, denotes the number of agents in the population having state i, and $\|z\|_1 = n$. Our main lower bound may now be stated as follows.

(6a) Interaction with agent state $A^{+}_{?[1]}$ sets agent output to Y_1:

$$(A^{+}_{?[1]}, A^{++}_{?[2]}, Y_?): \quad (A^{?}_{?[1]}, A^{?}_{?[2]}, Y_?) \mapsto Y_1$$

(6a) Interaction with agent state $A^{+}_{?[2]}$ sets agent output to Y_2:

$$(A^{++}_{?[1]}, A^{+}_{?[2]}, Y_?): \quad (A^{?}_{?[1]}, A^{?}_{?[2]}, Y_?) \mapsto Y_2$$

Figure 2: Protocol extension P_b of protocol $(P_{o[1]} + P_{o[2]})$, with additional states $\{Y_1, Y_2\}$.

(5a-5b) Interaction with source $X_{[1]}$ alters corner configurations different from A_1:

$$X_{[1]}: \quad A^{?}_{3} \mapsto A^{+}_{1}$$
$$X_{[1]}: \quad A^{?}_{2} \mapsto A^{+}_{3}$$

(5c-5d) Interaction with source $X_{[2]}$ alters corner configurations different from A_2:

$$X_{[2]}: \quad A^{?}_{1} \mapsto A^{+}_{2}$$
$$X_{[2]}: \quad A^{?}_{3} \mapsto A^{+}_{1}$$

Figure 3: Protocol P'_o is obtained from the basic oscillator protocol P_o by removing Rule (5) and replacing it by the two rules shown above.

(6) Meeting an agent of a different type resets majority setting:

$$((X \text{ or } A^{?}_{j}), M_?): \quad (A^{?}_{i}, M_?) \mapsto \begin{cases} M_{+1}, & \text{with probability } 1/2 \\ M_{-1}, & \text{with probability } 1/2 \end{cases}, \quad \text{for } j \neq i.$$

(7-10) Three-state majority protocol among agents of the same type:

$$\begin{aligned}
(7):& \quad (A^{?}_{i}, M_{-1}): \quad (A^{?}_{i}, M_{+1}) \mapsto M_0, \quad \text{with probability } r.\\
(8):& \quad (A^{?}_{i}, M_{+1}): \quad (A^{?}_{i}, M_{-1}) \mapsto M_0, \quad \text{with probability } r.\\
(9):& \quad (A^{?}_{i}, M_{+1}): \quad (A^{?}_{i}, M_0) \mapsto M_{+1}, \quad \text{with probability } r.\\
(10):& \quad (A^{?}_{i}, M_{-1}): \quad (A^{?}_{i}, M_0) \mapsto M_{-1}, \quad \text{with probability } r.
\end{aligned}$$

Figure 4: Protocol extension P_m of protocol P_o, with additional states $\{M_{-1}, M_0, M_{+1}\}$. This protocol extension is applied through the composition $(P_o \circ P_m)$. Probability $r > 0$ is given by an absolute constant, depending explicitly on s and p, whose value is sufficiently small.

Theorem 4.1 (Fixed points preclude fast stabilization). *Let $\varepsilon_1 > 0$ be arbitrarily chosen, let P be any k-state protocol, and let z_0 be a configuration of the system with at most n^{ε_0} agents in state X, where $\varepsilon_0 \in (0, \varepsilon_1]$ is a constant depending only on k and ε_1. Let B be a subset of the state space around z_0 such that the population of each state within B is within a factor of at most n^{ε_0} from that in z_0 (for any $z \in B$, for all states $i \in \{1, \dots, k\}$, we have $z_0^{(i)}/n^{\varepsilon_0} < z^{(i)} \leq n^{\varepsilon_0} \max\{1, z_0^{(i)}\}$).*

Suppose that in an execution of P starting from configuration z_0, with probability $1 - o(1)$, the configurations of the system in the next $n^{2\varepsilon_1}$ parallel rounds are confined to B.

Then, an execution of P for $n^{2\varepsilon_0}$ parallel rounds, starting from a configuration in which state X has been removed from z_0, reaches a configuration in a $O(n^{6\varepsilon_1})$-neighborhood of B, with probability $1 - o(1)$.

In the statement of the Theorem, for the sake of maintaining the size of the population, we interpret "removing state X from z_0" as replacing the state of all agents in state X by some other state, chosen adversarially (in fact, this may be any state which has sufficiently many representatives in configuration z_0). The $O(n^{6\varepsilon_1})$-neighborhood of B is understood in the sense of the 1-norm or, asymptotically equivalently, the total variation distance, reflecting

(11) Light switch progresses from L_{-1} to L_{+1} in the presence of initiator M_{-1}:

$$(A^?_?, M_{-1}, L_?): \quad (A^?_?, M_?, L_{-1}) \mapsto L_{+1}$$

(12) Light switch progresses from L_{+1} to L_{on} in the presence of initiator M_{+1}:

$$(A^?_?, M_{+1}, L_?): \quad (A^?_?, M_?, L_{+1}) \mapsto L_{on}$$

(13) Light spontaneously turns off:

$$(A^?_?, M_?, L_?): \quad (A^?_?, M_?, L_{on}) \mapsto L_{-1}, \quad \text{with probability } q(\varepsilon).$$

Figure 5: Protocol extension P_l of protocol $(P_o \circ P_m)$, with additional states $\{L_{-1}, L_{+1}, L_{on}\}$. This protocol extension is applied through the composition $(P_o \circ P_m) + P_l$. Probability $q(\varepsilon) > 0$ is given by an absolute constant, depending explicitly on s, p, r, and ε, whose value is sufficiently small.

configurations which can be converted into a configuration from B by flipping the states of $O(n^{6\varepsilon_1})$ agents.

The proof of Theorem 4.1 is provided in the full version of the paper. It proceeds by a coupling argument between a process starting from z_0 and a perturbed process in which state X has been removed. The analysis treats differently rules and states which are seldom encountered during the execution of the protocol from those that are encountered with polynomially higher probability (such a clear separation is only possible when $k = O(\text{poly} \log \log n)$). Eventually, the probability of success of the coupling reduces to a two-dimensional biased random walk scenario, in which the coordinates represent differences between the number of times particular rules have been executed in the two coupled processes.

We have the following direct corollaries for the problems we are considering. For Detection, if B represents the set of configurations of the considered protocol, which are understood as the protocol giving the answer "$^\#X > 0$", then our theorem says that, with probability $1 - o(1)$, the vast majority of agents will not "notice" that $^\#X$ had been set to 0, even a polynomial number of rounds after this has occurred, and thus cannot yield a correct solution. An essential element of the analysis is that it works only when state X is removed in the perturbed process. Thus, there is nothing to prevent the dynamics from stabilizing even to a single point in the case of $X = 0$, which is indeed the case for our protocol P_r. The argument for BitBroadcast only applies to situations where the source agent is sending out white noise (independently random bits in successive interactions). Such a source can be interpreted as a pair of sources in states X_1 and X_2 in the population, each disclosing itself with probability $1/2$ upon activation and staying silent otherwise. In the cases covered by the lower bound, the scenario in which the source X_1 is completely suppressed cannot be distinguished from the scenario in which both X_1 and X_2 appear; likewise, the scenario in which the source X_2 is completely suppressed cannot be distinguished from the scenario in which both X_1 and X_2 appear. By coupling all three processes, this would imply the indistinguishability of the all these configurations, including those with only source X_1 and only source X_2, which would imply incorrect operation of the protocol.

Whereas we use the language of discrete dynamics for precise statements, we informally remark that the protocols covered by the lower bound of Theorem 4.1 include those whose dynamics z_t/n, described in the continuous limit ($n \to +\infty$), has only point attractors, repellers, and fixed points. In this sense, the use of oscillatory dynamics in our protocol seems inevitable.

The impossibility result is stated in reference to protocols with a constant number of states, however, it may be extended to protocols with a non-constant number of states k, showing that such protocols require $n^{\exp[-O(\text{poly}(k))]}$ time to reach a desirable output. (This time is larger than polylogarithmic up to some threshold value $k = O(\text{poly} \log \log n)$.) The lower bound covers randomized protocols, including those in which rule probabilities depend on n (i.e., non-universal ones).

5 INPUT-CONTROLLED BEHAVIOR OF PROTOCOLS FOR DETECTION

In this Section, we consider the periodicity of protocols for self-organizing oscillatory dynamics, in order to understand how the period of a phase clock must depend on the input parameters. We focus on the setting of the Detection problem, considering the value $^\#X$ of the input parameter. In Section 3.1, we noted informally that the designed oscillatory protocol performs a complete rotation around the triangle in $\Theta(\log n/^\#X)$ rounds. Here, we provide partial evidence that the periodicity of any oscillatory protocol depends both on the value of $^\#X$ and n. We do this by bounding the portions of the configuration space in which a protocol solving Detection finds itself in most time steps, separating the cases of sub-polynomial $^\#X$ (i.e., $^\#X < n^{\varepsilon_0}$, where $\varepsilon_0 > 0$ is a constant dependent on the specific protocol), and the case of $^\#X = \Theta(n)$.

Any protocol on k states (not necessarily of oscillatory nature) can be viewed as a Markov chain in its k-dimensional configuration space $[0, n]^k$, and as in Section 4 we identify a configuration with a vector $z \in \{0, 1, \ldots, n\}^k = Z$. The configuration at time step t is denoted $z(t)$. In what follows, we will look at the equivalent space of log-configurations, given by the bijection:

$$\begin{aligned} Z \ni z = &(z^{(1)}, \ldots, z^{(k)}) \mapsto \\ &(\ln^\circ z^{(1)}, \ldots, \ln^\circ z^{(k)} \equiv \ln^\circ z \in \{\ln^\circ 0, \ln^\circ 1, \ldots, \ln^\circ n\}^k), \end{aligned}$$

where $\ln^\circ a = \ln a$ for $a > 0$ and $\ln^\circ a = -1$ for $a = 0$.

For $z_0 \in Z$, we will refer to the *d-log-neighborhood* of z_0 as the set of points $\{z \in Z : |\ln^\circ z - \ln^\circ z_0| < d\}$.

Notice first that the notion of a box in the statement of Theorem 4.1 is closely related to the set of points in the $(\varepsilon_0 \ln n)$-log-neighborhood of configuration z_0. It follows from the Theorem that any protocol for solving Detection within a polylogarithmic number of rounds T with probability $1 - o(1)$ must, in the case of $0 < {}^{\#}X < n^{\varepsilon_0}$, starting from z_0 at some time t_0, leave the $(\varepsilon_0 \ln n)$-log-neighborhood of z_0 within T rounds with probability $1 - o(1)$. We obtain the following corollary.

Proposition 5.1. *Fix a universal protocol P which solves the Detection problem with ε-error in $T = O(\text{poly} \log n)$ rounds with probability $1 - o(1)$. Set $0 < {}^{\#}X < n^{\varepsilon_0}$, where $\varepsilon_0 > 0$ is a constant which depends only on the definition of protocol ${}^{\#}X$. Let t_0 be an arbitrarily chosen moment of time after at least T rounds from the initialization of the protocol in any initial state. Then, within T rounds after time t_0, there is a moment of time t such that $z(t)$ is not in the $(\varepsilon_0 \ln n)$-log-neighborhood of $z(t_0)$, with probability $1 - o(1)$.* □

The above Proposition suggests that oscillatory or quasi-oscillatory behavior at low concentrations of state X must be of length $\Omega(\log n)$. By contrast, the following Proposition shows that in the case ${}^{\#}X = \Theta(n)$, the protocol may remain tied to a constant-size log-neighborhood of its configuration space.

Proposition 5.2. *Fix a universal protocol P with set of states K which solves the Detection problem with ε-error in $T = O(\text{poly} \log n)$ rounds with probability $1 - o(1)$. Then, there exists a constant $\delta_0 > 0$, depending only on the design of protocol P, with the following property. Fix ${}^{\#}X \in [cn, n/2]$, where $0 < c < 1/2$ is an arbitrarily chosen constant. Let t be an arbitrarily chosen moment of time, after at least T rounds from the initialization of the protocol at an adversarially chosen initial configuration $z(0)$, such that each coordinate $z^{(i)}(0)$ satisfies $z^{(i)}(0) = 0$ or $z^{(i)}(0) > 1/(2|K|)$, for all $i \in \{1, \ldots, |K|\}$. Then, with probability $1 - e^{-n^{\Omega(1)}}$, $z(t)$ is in the δ_0-log-neighborhood of $z(0)$.*

Note that, in the regime of a constant-size log-neighborhood of configuration $z(0)$, the discrete dynamics of the protocol adheres closely to the continuous-time version of its dynamics in the limit $n \to +\infty$. Since the latter is independent of n, any oscillatory behavior "inherited" from the continuous dynamic would have a period of $O(1)$ rounds. We leave as open the question whether some form of behavior of a protocol with polylogarithmic (i.e., or more broadly, non-constant and subpolynomial) periodicity for Detection can be designed in the regime of ${}^{\#}X = \Theta(n)$ despite this obstacle. In particular, the authors believe that the existence of an input-controlled phase clock with a period of $\Theta(\log n)$ for any ${}^{\#}X > 0$, and the absence of operation for ${}^{\#}X = 0$, is unlikely in the class of discrete dynamical systems given by the rules of population protocols.

Acknowledgment. We sincerely thank Dan Alistarh and Przemek Uznański for inspiring discussions, and Lucas Boczkowski for many detailed comments which helped to improve this manuscript.

REFERENCES

[1] Mohammed Amin Abdullah and Moez Draief. 2012. Majority Consensus on Random Graphs of a Given Degree Sequence. *CoRR* abs/1209.5025 (2012). http://arxiv.org/abs/1209.5025
[2] Dan Alistarh, James Aspnes, David Eisenstat, Rati Gelashvili, and Ronald L. Rivest. 2017. Time-Space Trade-offs in Population Protocols, See [30], 2560–2579. https://doi.org/10.1137/1.9781611974782.169
[3] Dan Alistarh, James Aspnes, and Rati Gelashvili. 2018. Space-Optimal Majority in Population Protocols, See [16], 2221–2239. https://doi.org/10.1137/1.9781611975031.144
[4] Dan Alistarh, Bartlomiej Dudek, Adrian Kosowski, David Soloveichik, and Przemyslaw Uznanski. 2017. Robust Detection in Leak-Prone Population Protocols. In *DNA (Lecture Notes in Computer Science)*, Vol. 10467. Springer, 155–171.
[5] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. 2006. Computation in networks of passively mobile finite-state sensors. *Distributed Computing* 18, 4 (2006), 235–253. https://doi.org/10.1007/s00446-005-0138-3
[6] Dana Angluin, James Aspnes, and David Eisenstat. 2008. Fast computation by population protocols with a leader. *Distributed Computing* 21, 3 (2008), 183–199. https://doi.org/10.1007/s00446-008-0067-z
[7] Dana Angluin, James Aspnes, and David Eisenstat. 2008. A simple population protocol for fast robust approximate majority. *Distributed Computing* 21, 2 (2008), 87–102. https://doi.org/10.1007/s00446-008-0059-z
[8] Dana Angluin, James Aspnes, David Eisenstat, and Eric Ruppert. 2007. The computational power of population protocols. *Distributed Computing* 20, 4 (2007), 279–304. https://doi.org/10.1007/s00446-007-0040-2
[9] James Aspnes and Eric Ruppert. 2007. An Introduction to Population Protocols. *Bulletin of the EATCS* 93 (2007), 98–117.
[10] Luca Becchetti, Andrea E. F. Clementi, Emanuele Natale, Francesco Pasquale, Riccardo Silvestri, and Luca Trevisan. 2013. Simple Dynamics for Majority Consensus. *CoRR* abs/1310.2858 (2013). http://arxiv.org/abs/1310.2858
[11] Lucas Boczkowski. 2017. Personal communication.
[12] Lucas Boczkowski, Amos Korman, and Emanuele Natale. 2017. Minimizing Message Size in Stochastic Communication Patterns: Fast Self-Stabilizing Protocols with 3 bits, See [30], 2540–2559. https://doi.org/10.1137/1.9781611974782.168
[13] Olivier Bournez, Pierre Fraigniaud, and Xavier Koegler. 2012. Computing with Large Populations Using Interactions. In *MFCS (Lecture Notes in Computer Science)*, Vol. 7464. Springer, 234–246.
[14] Colin Cooper, Robert Elsässer, and Tomasz Radzik. 2014. The Power of Two Choices in Distributed Voting. In *Proc. 41st International Colloquium on Automata, Languages, and Programming, ICALP 2014, Copenhagen, Denmark, Part II (Lecture Notes in Computer Science)*, Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias (Eds.), Vol. 8573. Springer, 435–446. https://doi.org/10.1007/978-3-662-43951-7_37
[15] Colin Cooper, Anissa Lamani, Giovanni Viglietta, Masafumi Yamashita, and Yukiko Yamauchi. 2017. Constructing self-stabilizing oscillators in population protocols. *Inf. Comput.* 255 (2017), 336–351. https://doi.org/10.1016/j.ic.2016.12.002
[16] Artur Czumaj (Ed.). 2018. *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018.* SIAM. https://doi.org/10.1137/1.9781611975031
[17] Jurek Czyzowicz, Leszek Gasieniec, Adrian Kosowski, Evangelos Kranakis, Paul G. Spirakis, and Przemyslaw Uznanski. 2015. On Convergence and Threshold Properties of Discrete Lotka-Volterra Population Protocols. In *Proc. 42nd International Colloquium on Automata, Languages, and Programming, ICALP 2015, Kyoto, Japan, Part I (Lecture Notes in Computer Science)*, Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann (Eds.), Vol. 9134. Springer, 393–405. https://doi.org/10.1007/978-3-662-47672-7_32
[18] Alexander Dobrinevski and Erwin Frey. 2012. Extinction in neutrally stable stochastic Lotka-Volterra models. *Phys. Rev. E* 85 (May 2012), 051903. Issue 5. https://doi.org/10.1103/PhysRevE.85.051903
[19] Benjamin Doerr, Mahmoud Fouz, and Tobias Friedrich. 2011. Social Networks Spread Rumors in Sublogarithmic Time. *Electronic Notes in Discrete Mathematics* 38 (2011), 303–308. https://doi.org/10.1016/j.endm.2011.09.050
[20] David Doty. 2014. Timing in chemical reaction networks. In *Proc. Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, Chandra Chekuri (Ed.). SIAM, 772–784. https://doi.org/10.1137/1.9781611973402.57
[21] Uriel Feige, David Peleg, Prabhakar Raghavan, and Eli Upfal. 1990. Randomized Broadcast in Networks. In *Algorithms, International Symposium SIGAL '90, Tokyo, Japan, 1990 (Lecture Notes in Computer Science)*, Tetsuo Asano, Toshihide Ibaraki, Hiroshi Imai, and Takao Nishizeki (Eds.), Vol. 450. Springer, 128–137. https://doi.org/10.1007/3-540-52921-7_62
[22] Nikolaos Fountoulakis and Konstantinos Panagiotou. 2013. Rumor spreading on random regular graphs and expanders. *Random Struct. Algorithms* 43, 2 (2013), 201–220. https://doi.org/10.1002/rsa.20432

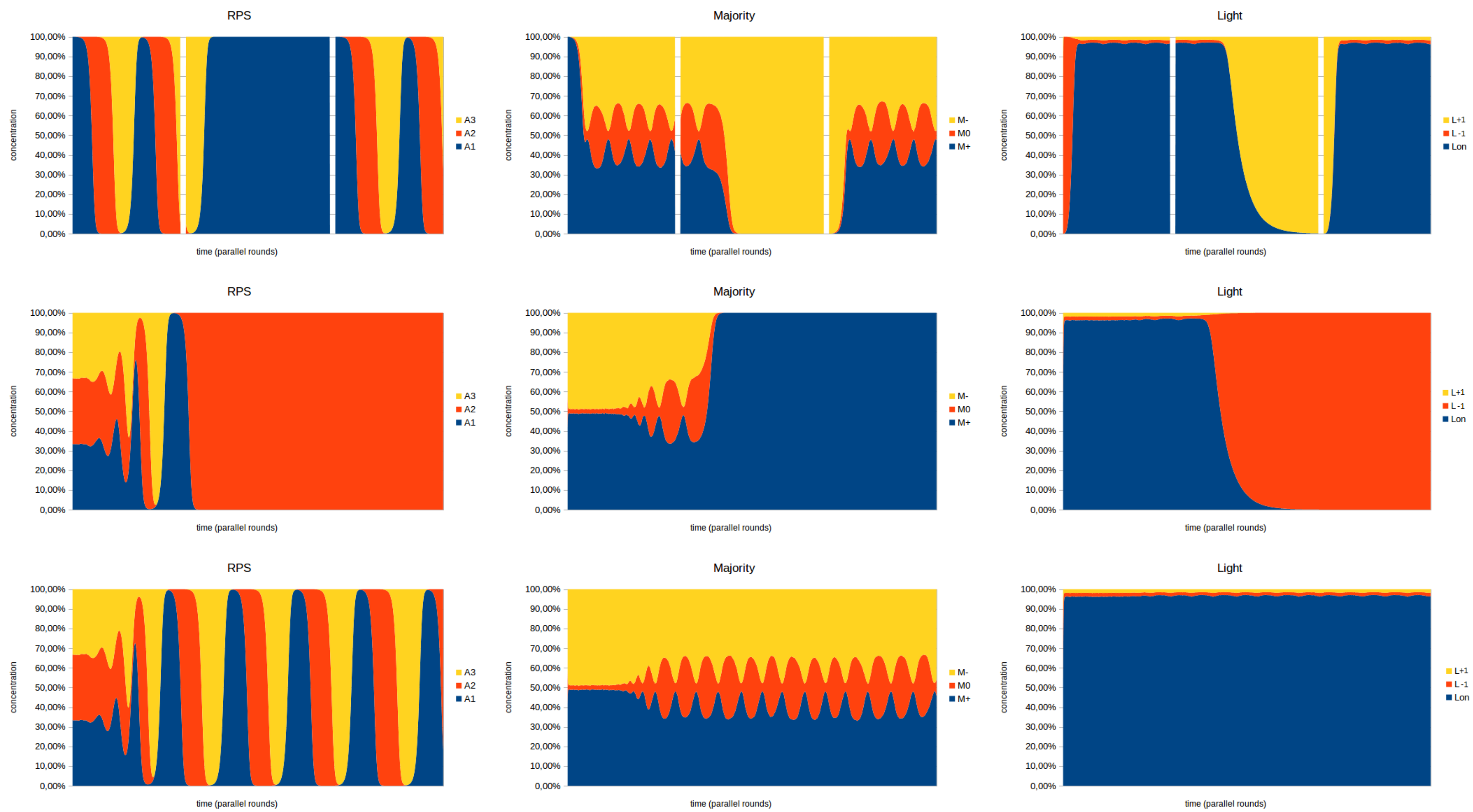

Figure 6: Illustration of concentration of various features as a function of time steps for a simulation of the protocol P_o and solution to Detection for $n = 10^6, p = 6 \cdot 10^{-2}, q = 10^{-2}, r = 10^{-1}, s = 1$ in various scenarios. The left column shows concentration of species A_1, A_2, A_3, the middle one the majority layer, and the right one the light. First row: initialization from a corner configuration ($^{\#}X = 1$, left panel), further dynamics of the protocol after rumor source is removed ($^{\#}X = 0$, middle panel), further dynamics of the protocol after rumor source is reinserted ($^{\#}X = 1$, right panel). Middle row: initialization from a configuration with $A_1 = A_2 = A_3 = 1/3$ and $^{\#}X = 0$. Bottom row: as above, with $^{\#}X = 1$. The range of the horizontal time scale corresponds to 2500 parallel rounds of the protocol.

[23] Alan M. Frieze and Geoffrey R. Grimmett. 1985. The shortest-path problem for graphs with random arc-lengths. *Discrete Applied Mathematics* 10, 1 (1985), 57–77. https://doi.org/10.1016/0166-218X(85)90059-9

[24] Leszek Gasieniec and Grzegorz Stachowiak. 2018. Fast Space Optimal Leader Election in Population Protocols, See [16], 2653–2667. https://doi.org/10.1137/1.9781611975031.169

[25] George Giakkoupis, Yasamin Nazari, and Philipp Woelfel. 2016. How Asynchrony Affects Rumor Spreading Time. In *Proc. 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016, Chicago, IL, USA*, George Giakkoupis (Ed.). ACM, 185–194. https://doi.org/10.1145/2933057.2933117

[26] Albert Goldbeter. 2002. Computational approaches to cellular rhythms. *Nature* 420, 6912 (Nov. 2002), 238–245. http://dx.doi.org/10.1038/nature01259

[27] Richard M. Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vöcking. 2000. Randomized Rumor Spreading. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*. IEEE Computer Society, 565–574. https://doi.org/10.1109/SFCS.2000.892324

[28] Benjamin Kerr, Margaret A. Riley, Marcus W. Feldman, and Brendan J. M. Bohannan. 2002. Local dispersal promotes biodiversity in a real-life game of rock-paper-scissors. *Nature* 418, 6894 (11 Jul 2002), 171–174. https://doi.org/10.1038/nature00823

[29] Benjamin C. Kirkup and Margaret A. Riley. 2004. Antibiotic-mediated antagonism leads to a bacterial game of rock-paper-scissors in vivo. *Nature* 428, 6981 (25 Mar 2004), 412–414. https://doi.org/10.1038/nature02429

[30] Philip N. Klein (Ed.). 2017. *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*. SIAM. https://doi.org/10.1137/1.9781611974782

[31] Leslie Lamport. 1978. Time, Clocks, and the Ordering of Events in a Distributed System. *Commun. ACM* 21, 7 (1978), 558–565. https://doi.org/10.1145/359545.359563

[32] Christoph Lenzen, Thomas Locher, Philipp Sommer, and Roger Wattenhofer. 2010. Clock Synchronization: Open Problems in Theory and Practice. In *Proc. 36th Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM 2010, Spindleruv Mlýn, Czech Republic (Lecture Notes in Computer Science)*, Jan van Leeuwen, Anca Muscholl, David Peleg, Jaroslav Pokorný, and Bernhard Rumpe (Eds.), Vol. 5901. Springer, 61–70. https://doi.org/10.1007/978-3-642-11266-9_5

[33] Christoph Lenzen, Thomas Locher, and Roger Wattenhofer. 2008. Clock Synchronization with Bounded Global and Local Skew. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*. IEEE Computer Society, 509–518. https://doi.org/10.1109/FOCS.2008.10

[34] Alfred J. Lotka. 1909. Contribution to the Theory of Periodic Reactions. *The Journal of Physical Chemistry* 14, 3 (1909), 271–274. https://doi.org/10.1021/j150111a004 arXiv:http://dx.doi.org/10.1021/j150111a004

[35] Jennifer Lundelius and Nancy A. Lynch. 1984. A New Fault-Tolerant Algorithm for Clock Synchronization. In *Proc. Third Annual ACM Symposium on Principles of Distributed Computing, Vancouver, B. C., Canada*, Tiko Kameda, Jayadev Misra, Joseph G. Peters, and Nicola Santoro (Eds.). ACM, 75–88. https://doi.org/10.1145/800222.806738

[36] George B. Mertzios, Sotiris E. Nikoletseas, Christoforos L. Raptopoulos, and Paul G. Spirakis. 2017. Determining majority in networks with local interactions and very small local memory. *Distributed Computing* 30, 1 (2017), 1–16. https://doi.org/10.1007/s00446-016-0277-8

[37] Othon Michail, Ioannis Chatzigiannakis, and Paul G. Spirakis. 2011. *New Models for Population Protocols.* Morgan & Claypool Publishers. https://doi.org/10.2200/S00328ED1V01Y201101DCT006

[38] Konstantinos Panagiotou and Leo Speidel. 2016. Asynchronous Rumor Spreading on Random Graphs. *CoRR* abs/1608.01766 (2016). http://arxiv.org/abs/1608.01766

[39] Matthew Parker and Alex Kamenev. 2009. Extinction in the Lotka-Volterra model. *Phys. Rev. E* 80 (2009), 021129. Issue 2. https://doi.org/10.1103/PhysRevE.80.021129

[40] Boris Pittel. 1987. On Spreading a Rumor. *SIAM J. Appl. Math.* 47, 1 (March 1987), 213–223. https://doi.org/10.1137/0147013

[41] V. B. Priezzhev, Deepak Dhar, Abhishek Dhar, and Supriya Krishnamurthy. 1996. Eulerian Walkers as a Model of Self-Organized Criticality. *Phys. Rev. Lett.* 77 (Dec 1996), 5079–5082. Issue 25. https://doi.org/10.1103/PhysRevLett.77.5079

[42] Tobias Reichenbach, Mauro Mobilia, and Erwin Frey. 2006. Coexistence versus extinction in the stochastic cyclic Lotka-Volterra model. *Phys. Rev. E* 74 (Nov 2006), 051907. Issue 5. https://doi.org/10.1103/PhysRevE.74.051907

[43] Thomas Sauerwald. 2010. On Mixing and Edge Expansion Properties in Randomized Broadcasting. *Algorithmica* 56, 1 (2010), 51–88. https://doi.org/10.1007/s00453-008-9245-4

[44] Attila Szolnoki, Mauro Mobilia, Luo-Luo Jiang, Bartosz Szczesny, Alastair M. Rucklidge, and Matjaz Perc. 2014. Cyclic dominance in evolutionary games: A review. *CoRR* abs/1408.6828 (2014). http://arxiv.org/abs/1408.6828