Exploring the Computational Limits of Adaptive Networked Populations of Tiny Artefacts[*]

Ioannis Chatzigiannakis
Research Academic Computer Technology Institute
Patras, Greece
ichatz@cti.gr

Othon Michail
Computer Engineering and Informatics Department
University of Patras, Greece
michailo@ceid.upatras.gr

Paul G. Spirakis
Research Academic Computer Technology Institute
Patras, Greece
spirakis@cti.gr

1. INTRODUCTION

In the near future, it is reasonable to expect that new types of systems will appear, of massive scale, expansive and permeating their environment, of very heterogeneous nature, and operating in a constantly changing networked environment. We expect that most such systems will have the form of a very large society of unimpressive networked artefacts. Yet by cooperation, they will be organized in large societies to accomplish tasks that are difficult or beyond the capabilities of todays conventional centralized systems.

The Population Protocol model [1] introduced a novel approach towards the study of such systems by assuming that each artefact is an agent, so limited, that can be represented as a finite-state sensor of constant ($\mathcal{O}(1)$) total storage capacity. Such agents are passively mobile and communicate in pairs using a low-power wireless signal. It has been proven that, although such systems consist of extremely limited, cheap and bulk-produced hardware devices, they are still capable of carrying out very useful nontrivial computations. Based on this approach we investigate many new intriguing directions.

2. POPULATION PROTOCOL MODEL

The Population Protocol (PP) model [1] assumes a population V of $|V| = n$ agents. Each agent due to constraints in cost and size is severely limited. It consists of a *sensor* that translates a measurement to an input symbol from a finite *input alphabet* X, a *memory of constant size* (i.e. $\mathcal{O}(1)$ total storage capacity) that at any time contains a state from a finite set of *states* Q, a *control unit* that updates the agent states according to the interactions taking place, a *power supply* (battery) and is capable of interacting with other agents locally by using a low-power wireless signal.

After receiving a global start signal (e.g. from a base station), all agents sense their environment in order to produce an input symbol (e.g. a sensor indicates the presence of radioactivity by providing input 1 and 0 otherwise). Each input symbol is mapped to an initial state by the application of an input function $I : X \to Q$. Agents are assumed to interact in pairs under the commands of an *adversary scheduler* (e.g. modeling the *passive mobility* of the agents).

A *communication graph* $G = (V, E)$ is used to represent the permissible interactions. An $e = (u, v)$ in the set of arcs E (for a directed graph) means that the agents u and v are able to interact, with u playing the role of *initiator* and v the role of *responder* in the interaction.

In each step, the adversary selects an arc $(u, v) \in E$ and the corresponding agents interact with u being the initiator and v the responder. "Interact" simply means that they provide their current states $q_u, q_v \in Q$, respectively, to a global transition function $\delta : Q \times Q \to Q \times Q$ and update their states according to the output of δ. For example, if $\delta(q_u, q_v) = (q'_u, q'_v)$, then u replaces q_u with $\delta_1(q_u, q_v) = q'_u$ and v replaces q_v with $\delta_2(q_u, q_v) = q'_v$.

At any time during the computation, an *output function* $O : Q \to Y$, where Y is a finite *output alphabet*, determines the output value of each agent based on its current state.

For useful computations to take place in such a system, a strong global *fairness condition* is imposed on the adversary to ensure the protocol makes progress; otherwise the adversary could partition the agents into non-communicating clusters. It is worth noticing that the critical assumption that diversifies the PP model from traditional distributed systems is that the protocol specifications are independent of the population size, which is known as the *uniformity* property of PPs. Moreover, PPs are *anonymous* since there is no room in the agent states to store a unique identifier.

Generally, population protocols do not halt. Instead, halting is replaced by an interesting property called *stability*. Stability was defined in [1] to be a situation where execution reaches a *configuration* C (a snapshot of the population states), after which, no matter how the computation proceeds, no agent will be able to change its output value. Such a configuration C is called an *output-stable* configuration. A protocol $\mathcal{A}$ (stably) *computes* a function f that maps multisets of elements of X to Y if, for every such multiset x and every fair execution that starts from the initial configuration corresponding to x, the output value of every agent eventually stabilizes to $f(x)$.

3. OUR RESEARCH DIRECTIONS

3.1 Experimental Verification of PPs

In the area initiated by the proposal of the PP model, experimental evaluation through simulations, testbed development and testing in real sensor populations has been almost excluded. So, as the gap between theoretical results and their practical verification grows, we believe that now is the right time for approaches that deal with the latter. Moreover, any progress in the development of tools specialized on experimental evaluation of protocols could provide a powerful alternative option in cases where analysis fails or

[*]This work has been partially supported by the ICT Programme of the European Union under contract number ICT-2008-215270 (FRONTS).

turns out to be extremely hard.

Till now, only one (adversary) scheduler has been proposed in the relevant literature. We have experimented with five new schedulers and divided them into two categories, those that make interaction decisions without knowledge about the protocol executed and those that are allowed some knowledge, which we exploit to study performance. This new class of *protocol aware schedulers* has been partly motivated by the general question of how to construct fair but adversarial schedulers, i.e. that, despite the fairness condition, push the protocols to their worst-case performance, e.g. by exploiting the structure of the communication graph.

We have intensively tested some representative protocols appearing in the relevant literature on a wide range of possible real scenarios. Our main purpose is to verify stability and compare the experimental time to convergence with the known complexity bounds. To do so, we developed a specialized simulation tool, dedicated for the simulation of PPs. We have chosen to replace low-level effects with abstract and exchangeable models. This fact, together with the use of sophisticated data structures that support efficient lookup and updates in constant time, allows us to simulate for the first time extremely large populations (i.e. of up to 10^8 agents) in a reasonable time (see [2]).

3.2 Mediated Population Protocols

Mobile telecommunications are governed by the existence of service providers. Motivated by contemporary telecommunications reality we incorporated such a provider in the population protocol model (see [3]). We named it *Communication Facility* (CF) or *Mediator*. With its presence, all communication between population pairs is accomodated through a common medium. The new model turns out to be stronger in terms of computational power and the incorporation of the CF seems to resolve some open safety issues.

The CF is equipped with a memory of $\mathcal{O}(n^2) = \mathcal{O}(m)$ total capacity, where m denotes the number of arcs in the communication graph (i.e. $|E| = m$). The memory of the CF keeps the permissible interactions, i.e. communication links, that are organized in *communication classes* corresponding to link states. When a pair of agents (u, v) is about to interact, the agents transmit their ids to the CF (these ids cannot be used by the agents neither can be stored in their working memory) and await permission. The CF searches for the received pair (u, v) in its database and responds to the interacting pair with the state corresponding to the class in which (u, v) was located. If (u, v) was not found, the CF sends a message to the agents to abort the interaction.

We keep the CF quite passive, in the spirit of cheap and unimpressive hardware: it only keeps the permissible interacting pairs of agents in communication classes. The CF can be either a communication layer, or a provider, or even the Internet, or some other communication substrade. Thus, our goal is not to capture *direct* communication due to *physical proximity* (which we believe is quite primitive) but, instead, to allow for a communication *service*, *independent of the population functionality.* This service may also have many applications in security.

The resulting model is called the Mediated Population Protocol model (MPP). In order to study the computational power of the new model without the need to incorporate the CF explicitly in the model definition, we consider communication links each equipped with a buffer of $\mathcal{O}(1)$ size. So, the main modification of MPP in comparison to the PP model is that it allows the communication links to keep states, which the agents read before interacting and update according to the output of the transition function (the transition function is now of the form $\delta : Q \times Q \times S \rightarrow Q \times Q \times S$, where S is a finite set of *edge states*).

It turns out, that our mediated protocols are stronger than the classical population protocols in terms of computational power. For the *basic* population protocol model (that concerns only complete interaction graphs) there exists an exact characterization of the computable predicates: they are precisely the *semilinear predicates.* We have devised a mediated protocol that stably computes the product of two non-negative integer-valued variables, when G is the complete graph. This is not a semilinear predicate. To show this fact, we stated and proved a general Theorem about the Composition of two stably computing MPPs. We have also shown that all predicates stably computable in our model are (non-uniformly) in the class $NSPACE(m)$.

Finally, we have defined Randomized MPP (the scheduler selects equiprobably the next interacting pair) and showed that, any *Peano predicate* accepted by a MPP, can be verified in deterministic Polynomial Time.

3.3 Switching Probabilistic Protocols

We imagine here a continuum of agents (see [4]). By the law of large numbers, one can model the underlying aggregate stochastic process as a deterministic flow system. Our main proposal is, in this case, to exploit the powerful tools of continuous nonlinear dynamics in order to examine questions (such as stability) of such protocols.

We have extended the class of [1] by defining a general model of "Switching Population Protocols" (SPP). Our main point is that one can study stability and population dynamics of protocols, via nonlinear differential equations that describe quite accurately the (discrete) population protocol dynamics when the population is huge. Our approach provides a sufficient condition for stability of *probabilistic* PPs (the scheduler selects equiprobably) that can be checked in polynomial time. It also gives a more general way to specify population protocols, that reveals interesting classes.

4. REFERENCES

[1] D. Angluin, J. Aspnes, Z. Diamadi, M. J. Fischer, and R. Peralta. Computation in networks of passively mobile finite-state sensors. In *23rd Annual ACM Sympsium on Principles of Distributed Computing (PODC)*, pages 290–299, New York, NY, USA, 2004. ACM.

[2] I. Chatzigiannakis, O. Michail, and P. G. Spirakis. Experimental verification and performance study of extremely large sized population protocols. FRONTS Technical Report FRONTS-TR-2009-3, http://fronts.cti.gr/aigaion/?TR=61, January 2009.

[3] I. Chatzigiannakis, O. Michail, and P. G. Spirakis. Mediated population protocols. FRONTS Technical Report FRONTS-TR-2009-8, http://fronts.cti.gr/aigaion/?TR=65, February 2009.

[4] I. Chatzigiannakis and P. G. Spirakis. The dynamics of probabilistic population protocols. In *Distributed Computing, 22nd International Symposium, DISC*, volume 5218 of *Lecture Notes in Computer Science*, pages 498–499, 2008.