SEVENTH FRAMEWORK PROGRAMME
THEME 3
Information and Communication Technologies

Grant agreement for:
Collaborative project, Small and medium-scale focused research project (STREP)

Deliverable D1.3:

A Set of Theories for Adaptively Organized Societies of Tiny Artefacts

Project acronym: FRONTS
Project full title: Foundations of Adaptive Networked Societies of Tiny Artefacts
Grant agreement no.: 215270

Responsible Partner: UPC
Report Preparation Date: Monday, May 1, 2011

Contents

1 Introduction

The main goal of WP1 in FRONTS project was to define a set one or more theoretical models for networks of tiny artifacts, which could help in making simulations, designing new algorithm for communication, data retrieval and processing. In Section 2 and 3, we present the two main models that we have proposed, the Mediator Population Protocols and the Sensor Field. We sketch the results obtained during the first and second years and the new results obtained during the third year of project. The models presented in this section are general models, aiming to explain and deal with global properties of networks of artifacts. In Sections 2 and 3 of D-3-3, it is presented a more restricted, problem oriented models for dynamic networks. In Section 4 we present a new theoretical model of interest aware data transmission, which also could be used in non-synchronous networks of artifacts. In Section 5, we present some of the other research done in WP1 under FRONTS. Section 6 is mainly devoted to extract the conclusions on the two main models and further line of work to follow. FInally we present a list of the produced material disseminating the theoretical research done under FRONTS.

The results of this third year are included in 13 FRONTS Technical Reports and papers: [CMN$^+$10b, CMN$^+$10a, CMN$^+$11b, MCS11b, CMS10a, SMC11, CMS10b, ACD$^+$11, ASS10, FNP$^+$10, DMSM$^+$11, BCG11].

During our research in the context of WP1, we have compiled a book, a book chapter and a couple of surveys for the reader interested in models for networks of tiny artifacts. In [MCS11b, SMC11] we discuss Population Protocols and most of its well-known extensions emphasizing on the MPP, GDMPP, PALOMA and PM models. Moreover, in [ACD$^+$11, FNP$^+$10], we surveyed the previously discussed models along with the *static synchronous sensor field* (*SSSF*) model, discussed in Deliverable 2.1. Finally, MPP and PM models were also presented in the 5th Athens Colloquium on Algorithms and Complexity [MCN$^+$10] and the dynamic sensor field in the GRASTA conference, Dagstuhl.

2 Population Protocols

2.1 Introduction

Networks of tiny artifacts are about to be part of everyday life. Homes and workplaces capable of self-controlling and adapting air-conditioning for different temperature and humidity levels, sleepless forests ready to detect and react in case of a fire, vehicles able to avoid sudden obstacles or possibly able to self-organize routes to avoid congestion, and so on, will probably be commonplace in the very near future. *Mobility* plays a central role in such systems and so does *passive mobility*, that is, mobility of the network stemming from the environment itself. Theoretical models for such systems have received great attention and will play a fundamental role for their study as they constitute an abstract but yet formal and precise method for understanding the laws that govern their functionality and their inherent properties.

In [AAD$^+$06], Angluin *et al.* introduced the *Population Protocol* (*PP*) model, which captures the notion of computation by a population of extremely limited communicating agents. In this model, the system consists of a collection of agents, represented as finite-state machines. The agents exchange information via pairwise interactions, which they are unable to predict or control. Via these interactions, the system organizes its computation and provides complex behavior as a whole. In [AAD$^+$06, AAER07], the computational power of the model was studied and has been proved to be exactly the class of *semi-linear predicates*, consisting of all predicates definable by first-order logical formulas of Presburger arithmetic (see, e.g., [GS66]).

2.2 Passively Mobile Communicating Machines that Use Restricted Space

The PP is fairly minimalistic and as a result, computationally weak. The natural next step is to try and enhance the basic model with extra realistic and implementable assumptions, in order to gain more computational power and/or speed-up the time to convergence and/or improve fault-tolerance. In an effort to strengthen the model in [AAER07], we proposed a new theoretical model for passively mobile Wireless Sensor Networks, called PM, standing for *Passively mobile Machines* [CMN$^+$11a]. The main modification w.r.t. the PP model is that agents now, instead of being automata, are Turing Machines. In the second year of FRONTS, we introduced a special case of the model, called *PALOMA*, where all agents *use memory logarithmic in the population size n* (Deliverable 1.2 in [CMN$^+$10b]). In [CMN$^+$11a], however, we provided general definitions for unbounded memories and proved that our results hold for more general cases. We focused on *complete communication graphs* and defined the complexity classes $\mathbf{PMSPACE}(f(n))$ parametrically, consisting of all predicates that are stably computable by some PM protocol that uses $\mathcal{O}(f(n))$ memory in each agent. We provided a protocol that generates unique identifiers from scratch only by using $\mathcal{O}(\log n)$ memory, and used it to provide an exact characterization of the classes $\mathbf{PMSPACE}(f(n))$ when $f(n) = \Omega(\log n)$: *they are precisely the classes of all symmetric predicates in* $\mathbf{NSPACE}(nf(n))$. In this way, we also provided a space hierarchy of the PM model when the memory bounds are $\Omega(\log n)$. We then explored the computability of the PM model when the protocols use $o(\log \log n)$ space per machine and proved that $\mathbf{SEM} = \mathbf{PMSPACE}(f(n))$ when $f(n) = o(\log \log n)$, where $\mathbf{SEM}$ denotes the class of the semilinear predicates. Finally, when $f(n) = \mathcal{O}(\log \log n)$, we showed a threshold behavior of the $\log \log n$ bound, by showing that $\mathbf{SEM} \subsetneq \mathbf{PMSPACE}(f(n))$.

2.3 Mediated Population Protocols

2.3.1 The Computational Power of the Model

During the second year of FRONTS, we proposed another extension of the PP model, the *Mediated Population Protocol* (MPP) model, and showed that it is computationally stronger than the basic model [CMS09]. In the third year, we focused on the computational power of the MPP model on complete interaction graphs and initially identical edges (*SMPP*) [CMN$^+$10a]. In particular, we investigated the class MPS of all predicates that are stably computable by the SMPP model. From [CMS09] we knew that MPS is in the symmetric subclass of $NSPACE(n^2)$. In [CMN$^+$10a], we proved that this inclusion holds with equality, thus, providing the following exact characterization for MPS: *A predicate is in* MPS *iff it is symmetric and is in* $NSPACE(n^2)$. In [MCS11a], we have compiled thorough presentation of MPP's computability.

2.3.2 Decidability of Graph Properties and Self-Awareness

In order to capture MPP's ability to stably compute graph properties, we introduced in Deliverable 1.2 a simplified version of MPP called Graph Decision Mediated Population Protocol (GDM now referred as GDMPP). The findings of our study were published in [CMS10b].

The work [CMS10b] raised some interesting questions and provided promising directions for further research. Some of these issues were addressed in the research conducted in [CMNS11]. We explored the capability of a network of extremely limited computational entities (agents), communicating according to a *complete interaction graph*, to decide properties about any of its subnetworks (become self-aware of itself). We devise simple graph protocols that can decide properties of some *input subgraph* given by some preprocessing on the network. As in GDMPP, the agents are modeled as finite-state automata and run the same global graph protocol which is a fixed size grammar, that is, its description is independent of the size (number of agents) of the network. The main difference of this new variation is that the protocols run on complete interaction graphs, but can decide properties concerning any of its subgraphs. We call this variation *Mediated Graph Protocol* (*MGP*) model. In [CMNS11], we provided some interesting properties of the MGP model among which is the ability to decide properties on stabilizing (initially changing for a finite number of steps) input graphs and we showed that the MGP model has the ability to decide properties of disconnected input graphs. The latter property was absent in the GDMPP model (as shown in [CMS10b]). We additionally showed that the computational power within the connected components is fairly restricted. The computability within the connected components of the input subgraph is tightly coupled to the computability of the GDMPP on the family of all weakly connected graphs (an open issue of [CMS10b]). Finally, we gave an exact characterization of the class $\mathbf{GMGP}$, of graph languages decidable by the MGP model: it is equal to the class of graph languages decidable by a nondeterministic Turing Machine of linear space that receives its input graph by its adjacency matrix representation.

2.4 Algorithmic Verification of Population Protocols

In [CMS10a], we studied the Population Protocol model of Angluin et al. from the perspective of protocol verification. In particular, we focused on algorithmically solving the problem of determining whether a given population protocol conforms to its specifications given as a first-order logical formula of Presburger arithmetic. Since this was the first work on population protocols' verification, we redefined most notions of population protocols in order to make them suitable for an algorithmic solution. Moreover, we formally defined the general verification problem and some interesting special cases. All these problems were shown to be NP-hard. Then we proposed some first algorithmic solutions for the special case where the population size on which the protocol runs, is provided as part of the verifier's input. Finally, we conducted experiments and algorithmic engineering in order to improve our verifiers' running times.

3 The Sensor Field Computational Model

3.1 Introduction

In the FRONTS work [ADGS09] we proposed a general model capturing some characteristic differences of sensor networks. A *Static Synchronous Sensor Field* ($\mathcal{SSSF}$) consists of a *set of devices* and a *communication graph.* The communication graph specifies how the devices communicate one to the other. For the moment and without loss of generality, we assumed that all devices are sensing devices that can receive information from the environment and send information to the environment. We concentrated our study in the case in which the devices and the communication links do not appear and disappear, the model we considered is static. Moreover each device executes its own process, communicates with its neighbors (devices associated to adjacent nodes) and also with the environment. The local computation of each device is defined by a potentially infinite set of states and a transition function and it depends on the communication with its neighbors and with the environment. A state of a device codifies the values of some local set of variables (ordinary program variables, message buffers, program counters ...) and all what is needed to describe completely the instantaneous configuration of the local computation. Given a local state of a device and given the communications items received from its neighbors and the input data item received from the environment, the transition function specifies its new local state as well as the communication items to be sent to each of its neighbors and the data item to be output to the environment in one computation step.

The computation of a SSSF depends on the interaction between the devices and the environment. All the devices work in a synchronous way, at the begining of each round they receive data from their neighbors and from the environment, then they apply their own transition function changing in this way their actual configuration and finish the round sending data to their neighbors and to the environment. The model allow us to express computation in terms of the stream behaviour of a computation. This notion matches the generic definition of the *sensing problems* that we expect to be solved by a network of tiny artifacts.

The computational resources used by a device in a network to compute a function of data streams are the following. For each device and computation round we can measure the following parameters:

- *Time.* The number of operations performed in the given round of the device. This is a rough estimation of the “physical time” needed to input data, receive information from other sensor, compute, send information and output data.
- *Space.* The space used by the device in such computation round.
- *Message Length.* The maximum number of data items of a message sent by the device in such computation round.
- *Number of messages.* The maximum number of messages sent by the device in such round.
- *Distance.* The distance traversed by the device during the round.

From those measures we consider the $Size$, the number of nodes or devices of the communication graph G, and the following worst case complexity measures taken over any device and computation round of a sensor field. $Time$ $(\mathcal{T})$,, $Space$ $(\mathcal{S})$, $MessageLength$ $(\mathcal{L})$, $MessageNumber$ $(\mathcal{M})$, and $Distance$ $(\mathcal{D})$.

From those measure we can analyze also other derived performance measures of the network, like energy consumption.

During the third year of FRONTS we have extended the definition of the sensor field model to capture additionally different some sources of dynamicity and mobility, defining the *dynamic sensor field model*.

3.2 Dynamicity sources

We consider two potential sources of dynamicity and mobility: The *passive mobility of the targeted data* and the *active mobility of the network devices*. This approach will also allow us to incorporate other sources of dynamicity like failures or temporal disconnection.

The motivating scenario considers a set of sensors that are attached to mobile agents, those sensor have only the capability of transmit measurements. In terms of the general hypothesis for the sensor field this can be modeled by assuming that an input data streams is not originated in a fixed location during the computation. Furthermore, different devices (only those within hearing range) will be able to get the sensor measurements.

We assume, in this initial proposal, that the network computing devices are able to move in order to obtain readings from far away sensors or to carry out a particular computation and that the movement pattern is decided of the network computation. This model is different from the asynchronous ad-hoc mobility models surveyed in [DMS11].

This observation of dynamicity have and impact on the sensing problems that the network is willing to solve. Recall that computational problems that are susceptible of being solved by static sensor fields can be stated in the following way [ADGS09]: Sensing Problem Π: Given an n-tuple of data streams $\mathbf{u} = (u_k)_{1 \leq k \leq n}$ for some $n \geq 1$, compute an m-tuple of data streams $\mathbf{v} = (v_k)_{1 \leq k \leq m}$ for some $m \leq n$ such that $R_\Pi(\mathbf{u}[1,t], \mathbf{v}[1,t])$ is satisfied for every $t \geq 1$. R_Π is the relation that output data streams have to satisfy given the input data streams. A function f between data streams is *consistent* with a relation R when for every pair of data streams $\mathbf{u}$ and $\mathbf{v}$, and every $t \geq 1$, if $f(\mathbf{u}[1,t]) = \mathbf{v}[1,t]$ then $R(\mathbf{u}[1,t], \mathbf{v}[1,t])$. A particular case is the Average Monitoring: Given g data streams $(u_k)_{1 \leq k \leq g}$ for some $g \geq 1$, compute m data streams $(v_k)_{1 \leq k \leq m}$ such that $v_k[t] = (u_1[t] + \cdots + u_g[t])/g$.

In the above formulation a problem arises if the data is originated in mobile targets or when the devices that have to collect the data can move. In those situations, the precondition that the network has access to all the sensors at any time step might not be possible. Therefore, to monitor continuously a wide area where the g targets move, we relax this condition and require only to get a reading from any sensor inside a reporting period. This leads to the following formulation: Continuous Average Monitoring: Given a set of g mobile data streams $(u_\alpha)_{1 \leq \alpha \leq g}$ for some $g \geq 1$, compute m mobile data streams $(v_k)_{1 \leq k \leq m}$ such that, for some $p > 0$, any $1 \leq k \leq m$, and $t > 0$, $v_k[tp] = (u_1[t_1] + \cdots + u_g[t_g])/g$ for some $(t-1)p \leq t_1, \ldots, t_g < tp$. We refer to p as the *gathering period*.

Observe that the continuous monitoring problem is an extension of the average monitoring problem as the average monitoring requires $p = 1$. This leads us to generalize de definition of sensing problems: Generalized Sensing Problem Π: Given a set of g mobile data streams $\mathbf{u} = (u_k)_{1 \leq k \leq g}$ for some $g \geq 1$, compute m mobile data streams $\mathbf{v} = (v_k)_{1 \leq k \leq m}$ for some $m > 0$ such that, for some $p > 0$, any $1 \leq k \leq m$, $R_\Pi(u_1[t_1] + \cdots + u_g[t_g]), \mathbf{v}[1,t])$ is satisfied for every $t \geq 1$ and for some $(t-1)p \leq t_1, \ldots, t_g < tp$. R_Π is the relation that output data streams have to satisfy given the input data streams.

3.3 The Dynamic Sensor Field

The fundamental features of the *Dynamic Sensor Field* (DSSF) model [ADMS11] are the following: A device is able to receive readings from any sensor in its sensing range, instead of being attached to just one sensor. At the same time that the device performs its local computation, it may move.

In a Static Sensor Field the data items in a data stream were assumed to be produced as readings of some sensor placed in a fixed location and attached to a device in the same position. However, the data stream abstraction allows us to consider that any of the data items can be obtained at different locations at different time steps. Observe that, additional information, like location of the target at the moment of the reading, could be attached to the data items. In this paper we consider a set W of g data streams from the sensors together with a collection N of n devices that can move according to the network computation. The sensors either do not move at all or each of them moves following an independent random mobility pattern.

A *communication graph* is a directed graph $G = (N, E)$. Each $k \in N$ is associated to a device. Each edge $(i, j) \in E$ specifies that device i can send messages to device j. In a *Dynamic Sensor Field* the communication graph might change during the computation. Let $G_t = (N, E_t)$ denote the communication graph at time t. Due to mobility, the subset of input data items accessible by a device may change along time. Let $D_t \subseteq N \times W$ denote the data stream accessibility relation at time t. We denote by $(k, \alpha) \in D_t$ the event that sensor α can be detected by device k at time t.

We assume that all devices are able to receive information from the environment (input data stream) and send information to the environment (output data stream). Moreover each device executes its own process, communicates with their actual neighbors (devices associated to adjacent nodes) and, if required, changes location. All the devices work in a synchronous way, at each time step they receive data from their neighbors and from the environment, apply their own transition function changing in this way their actual configuration, possibly move, and send data to their neighbors and to the environment. We assume that the devices can change position while they are performing their local computation.

Apart from the worst case performance measures of interest for the SSSFs, as latency, message number, message length, etc. (see [ADGS09]) we consider two additional performance measures, the traveled distance per step and the gathering period.

3.4 Solving the continuous average monitoring problem

We consider the continuous monitoring problem and analyze different dynamic sensor fields to solve it in the following scenario. We assume that the data of interest is accessible in a predetermined square shaped area discretized as a grid. The de-

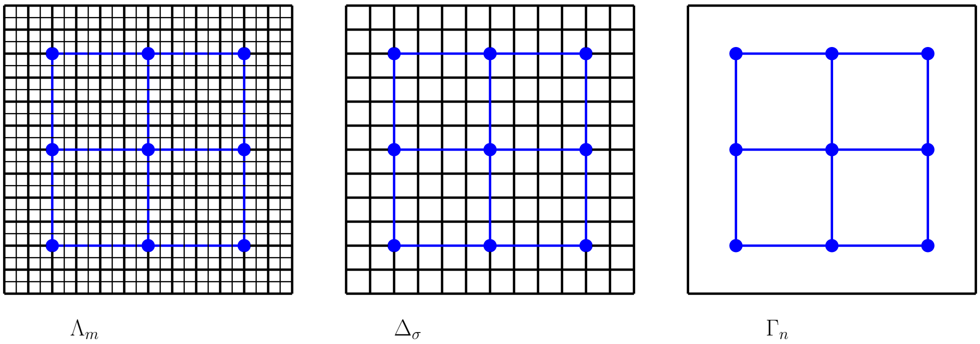

Figure 1: The three fundamental grids embedded in a terrain $\mathcal{T}$ with $n = 3$, $\sigma = 13$ and $m = 25$.

vices have two associated ranges, a *sensing range* s and a *communication range* r. We assume that the three squared grids (mobility, sensing and communicating) are embedded in the terrain (see Figure 1). The *mobility grid* Λ_m is formed by $m \times m$ nodes that serves as reference positions for the movement of the targets with attached sensors and the computing devices. We assume that sensors and devices stop at grid nodes labeled by coordinates (i, j), $1 \leq i, j \leq m$, and they move following paths on the grid. W.l.o.g assume the distance among to neighboring nodes in Λ_m is a unit length. As a subgrid of Λ_m, we have the *sensing grid* Δ_σ of size $\sigma \times \sigma$, where $2s$ is the distance between nodes in Δ_σ. As subgrid of Δ_σ we have the *communicating grid* Γ_n, with $n \times n$ nodes, where r is the distance among nodes in Γ_n. In the case that $r \leq 2s$ we have that Γ_n is also a sensing grid, in this case we take $\sigma = n$. Therefore, we can assume that $n \leq \sigma \leq m$. Let $\mathcal{T}(n, \sigma, m)$ denote this terrain.

In order to analyze the trade-off between size, latency and gathering period due to mobility, we propose a DSSF. The sensor fields are designed with two parts. The *gathering* part solves the problem of obtaining a reading from any sensors and will determine the gathering period. The *averaging* part computes the average of the measures taken during a gathering period. As we will see later on, the gathering part requires more steps than the averaging parts. Therefore the devices, just after the first gathering period finishes, run in parallel both algorithms. When both finalize the process is repeated with the new gathered data. The computing devices are arranged either as a line or as a grid of adequate dimensions.

In the **static data setting** we assume that input data streams are originated from some unknown positions in the grid Λ_m and that devices move on top of Λ_m. In this setting we have analysed three algorithms whose mobility patterns are the following:

- Line Sweeping: n devices are arranged as a line on the upper part of the terrain, the line sweeps the terrain from top to bottom and back.
- Surveilance grid: n^2 devices are arranged as a $n \times n$ mesh. Each device has associated a submesh of the mobility mesh. Devices perform a synchronized snake-like walk form upper left corner to lower most nodes and back again.

- Surveilance strip: n devices are arranged in a line. Each device has associated a strip of the mobility mesh. Devices perform a synchronized snake-like walk form upper left corner to bottom nodes and back again on the associated strip.

The following table presents the asymptotic bounds on the different resources needed for each one of the static data setting algorithms. The first algorithm only works for the case $n = \sigma$.

Algorithm	N	latency	gathering period	$\mathcal{T}(N)$	$\mathcal{L}(N)$	$\mathcal{S}(N)$	$\mathcal{M}(N)$	$\mathcal{D}(N)$
Line Sweeping	n	n	n	$O(g)$	$O(g)$	$O(g)$	2	r
Survei. grid	n^2	n	$2\max\{(r/2s)^2, n\}$	$O(g)$	$O(g)$	$O(g)$	2	$2s$
Survei. strip	n	n	$nr/2s$	$O(g)$	$O(g)$	$O(g)$	2	$2s$

For the case of **dynamic data** we assume a mobility pattern of the targets based on the *walkers* (see [DMS11] for references). We keep the term *walker* to refer to the moving targets with attached sensors. We consider a set of g walkers $\mathcal{W}$ moving on the mobility grid Λ_m, under the following random mobility model.

Initially g walkers are sprinkled uniformly at random on the m^2 vertices of the grid. At each step, every w_i, not on the boundary, chooses with probability $\frac{1}{4}$ one of the four possible directions and makes a step in the chosen direction. If w_i is in the corner, it chooses with probability $\frac{1}{2}$ any of the two possible directions, and if it is touching the boundary in one dimension only, w_i chooses with probability $\frac{1}{2}$ the only available direction in the dimension touching the boundary, and with probability $\frac{1}{4}$ the other two directions in the perpendicular dimension.

We propose three different sensor fields for solving the continuos monitoring problem when the input data streams follow the walker mobility model presented before. In the first two are static sensor fields (bounds on expected performance) and the third is a slower version of the Line sweeping algorithms. The analysis is the following:

Algorithm	N	latency	gathering period	$\mathcal{T}(N)$	$\mathcal{L}(N)$	$\mathcal{S}(N)$	$\mathcal{M}(N)$	$\mathcal{D}(N)$
SLine Sweeping	n	n	n	$O(g)$	$O(g)$	$O(g)$	2	$2r$
Grid	n^2	$2n$	$2g\left(\frac{m-1}{n}\right)^{20}$	$O(g)$	$O(g)$	$O(g)$	2	0
CentralLine	n	n	$gm^2/2$	$O(g)$	$O(g)$	$O(g)$	2	0

4 Models for interest-based routing

In [DMSM$^+$11], we address for the first time the problem of deriving theoretical, asymptotic characterization of social-aware forwarding protocols in opportunistic networks. We derive bounds on the expected message delivery time of two different routing protocols, representative of social-oblivious and social-aware forwarding. Message exchange in interest-based routing networks is governed by the *store*-*carry*-and-*forward* mechanism typical of delay-tolerant networks: a node (sender or relay nodes) stores the message in its buffer and carries it around, till a communication opportunity with another ”similar” node arises, upon which the message can be forwarded the node. Given this basic forwarding mechanism, a great deal of attention has been devoted in past years to optimize the forwarding policy of routing protocols. Significant performance improvement of social-aware approaches over social-oblivious approaches have been experimentally demonstrated [ED07, PH08]. Most existing social-aware forwarding approaches rely on the ability to store state information, storage is used to keep trace of history and to predict future meeting opportunities. Socially-oblivious routing protocols do not require storing additional information in the node buffers, which are then exclusively used to store the messages circulating in the network. Thus, comparing performance of social-aware vs. social-oblivious forwarding approaches would require modeling node buffers, which renders the resulting network model very complex. If storage capacity on the nodes is not accounted for in the analysis, unfair advantage would be given to social-aware approaches, which extensively use state information. This explains why the fundamental question of whether social-aware forwarding is superior to social-oblivious forwarding *per se* (and not due to storage of extensive status information) has remained unaddressed so far. Recently, a social-aware stateless approach has been introduced in [MMS11], where individuals with similar interests tend to meet relatively more often than individuals with diverse interests, this interest-based (IB) forwarding approach allows a fair comparison between social-aware and social-oblivious forwarding. In the present work, we provide a comparison of asymptotical performance of IB and BinarySW, the choice of that forwarding protocol is due to the fact that in [TS08], BinarySW is shown to be optimal within the class of Spray and Wait forwarding protocols. The contributions in this work are the following: 1.- A formal analysis of IB and BinarySW forwarding performance in two different mobility scenarios, the interest-based mobility, which models a situation where node mobility is highly correlated to similarity of individual interests, and social-oblivious mobility, which models the situation in which node mobility is independent of individual interests. The analysis reveals that, under certain conditions IB forwarding provides asymptotic performance benefits as compared to BinarySW. We also prove that IB forwarding always provide at least the same asymptotic performance as BinarySW, thus formally proving for the first time the superiority of social-aware vs. social-oblivious forwarding. We qualitatively confirm the previous analysis through simulations, which have been performed based both on a real-world data trace, and realistic human mobility model. 2.-We extend the analysis of 1 in the following ways: First, we consider a version of the forwarding algorithms in which up to q copies of the message can be created in the network, and travel along paths of length at most ℓ towards the destination. While in the previous analysis it was assumed $q = 2$ and $\ell = 2$, our results are shown to hold also for any constant values of q and ℓ. Second, we consider a version of the forwarding algorithm in which the sender does know the ID of the destination, but it does not know its *interest profile*. We show that expected message delivery time with IB forwarding remains bounded even in this more challenging networking scenario. Finally, we consider a model where source and destination do not necessarily have orthogonal interests, which can be interpreted as an average-case situation instead of the worst-case scenario considered in 1 and in all the generalizations considered so far. We show that, differently from what happens in worst-case conditions, in the average case IB forwarding and BinarySW provide the same asymptotic performance.

5 Data Stream Aggregation

During the last two years of the project we dealt with theoretical and practical aspects of efficiently and reliably extracting useful statistical summaries over all data streams that are observed continuously througout the entire network infrastructure, i.e. *aggregating* across our sensor network. While at the 2nd year we gave emphasis on a lower, sensor level in order to come up with novel algorithmic ideas regarding *time-window* aggregation [BK09] and also experimentally evaluate these ideas [Gia10], as well as other, existing, traditional aggregation techniques [RC09], at the final 3rd year, in the spirit of the Unifying Experiment, we focused on the higher, network level:

- We extensively surveyed the area of data stream aggregation in sensor networks, giving emphasis on *in-network* aggregation techniques, by contributing to the FRONTS State of the Art journal volume [BCG11]
- We performed, through the Unifying Experiment setting, real-life experiments as well as simulations, in order to see how the theoretical and experimental bounds and results of the 2nd year extend to the higher, sensor-network level. In particular, we used the *Partition Greedy* algorithm of [BK09] within each sensor in order to compute a sliding-window MAX aggregate and also implemented a traditional, duplicate-insensitive summary [CHL$^+$09, RC09] for the very important DCOUNT aggregate. Finally, after computing some interesting metrics (such as approximation precision and quantification pf resiliency to node failures) we try to see possible connections to the experimental results of [Gia10] and [RC09].

6 Conclusions - Research Achievements

As already stated, the main objective of WP1 in FRONTS, was to understand the computational process and behavior of the different types of artifact's networks with the aim of taking the maximum profit of the topology of the network. The computational system can be perceived and analyzed in two complementary ways. The first one has as goal to show the emergency of some designed behavior. In this scenario it is usual to assume that the system models some kind of interaction among the participating devices and that the final goal is to achieve a configuration with the pre-specified tasks. In the second approach, a task has to be carried out by the network, based on the subjacent communication model and the exchange of information among the participating devices. This information could be external data gathered by artifact, internal computation of the devices or a combination of both.

For the first approach, we have focused on models coming from the area of population protocols [AAD$^+$06, CMS09, CMN$^+$10b]. They represent mainly sensor networks, supposing that the corresponding sensing devices are extremely limited mobile agents, unable to control their own movement. The main limitation of the model is the assumption that interaction only occurs in pairs. Moreover, this is a key assumption, as it allow us to model the potential interactions by means of an interaction graph. Looking back to our initial research objectives, w.r.t. Population Protocols, described in Deliverable D1.1, we can observe substantial progress in most directions. Regarding the examination of the basic model's behavior under the assumption of *random transitions*, we defined and study various probabilistic schedulers [CDF$^+$09] for PPs as well as the *equivalence* between schedulers w.r.t. *performance* and *correctness*. In addition, we were able to devise protocols that solve *optimization* problems concerning the communication graph in the context of our MPP research [CMS09]. Finally, we proposed [CMS10a] algorithmic solutions for a special case of the *verification* problem for the basic PP model.

For the second approach, we have prosed a new model, the sensor field model, whose foundations rely in classic concepts of distributed in-network computation on a static or dynamic communication graph. The SSSF model can be seen as a non-uniform computational model in the sense that it is easy to introduce constraints for all or some of the devices of the sensor field and relate it to classic complexity classes. We also introduced the associated performance measures. The main limitation of the model is the assumption on synchronization. We have bounded the classes of fundamental problems to be solved by sensor fields, in the static and the dynamic settings. The input to those problems have been modeled as tuples of data stream. In a complementary analysis we have conducted different studies on data streaming techniques. Those techniques should be combined with our solution approach. In particular when the duration of the time step or the gathering period are large the overall performance will benefit by dealing with aggregate values instead of straight readings.

Moreover, we explored the capability of networks of tiny artifacts to achieve levels of self-awareness by deciding properties of its underlying infrastructure and obtaining different decidable properties depending on the model. In this way, it is our belief that we made an important step towards a better understanding of the inherent properties and limitations for systems using as platform networks of small devices, The fundamental question posted to WP1 at the beginning of the project was to evaluate the existence or one or more models adequate for networks of tiny artifacts. At the end of the project we have the conviction that the sensor field model captures most of the characteristics of those networks. It has allowed us to perform a theoretical analysis of constrains dealing with the subjacent communication topology that is the basis for the sensor networks considered in some of the experiments. The main restriction to the model at present is its synchronous nature. However, the experimental scenario of FRONTS considers synchronization at the lower layers of the communicating infrastructure.

References

[AAD^{+}06] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, pages 235–253, March 2006.

[AAER07] Dana Angluin, James Aspnes, David Eisenstat, and Eric Ruppert. The computational power of population protocols. *Distributed Computing*, 20(4):279–304, November 2007.

[ACD^{+}11] Carme Àlvarez, Ioannis Chatzigiannakis, Amalia Duch, Joaquim Gabarró, Othon Michail, Serna Maria, and Paul G. Spirakis. Computational models for networks of tiny artifacts: A survey. *Computer Science Review*, 5(1), January 2011. FRONTS-TR-2011-1. *

[ADGS09] Carme Àlvarez, Amalia Duch, Joaquim Gabarro, and Maria Serna. Sensor field: A computational model. In *5th Intl Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS)*, pages 3–14, Berlin, Heidelberg, 2009. Springer-Verlag. FRONTS-TR-2009-50. *

[ADMS11] Carme Àlvarez, Josep Díaz, Dieter Mitsche, and Maria Serna. Continuous monitoring in the dynamic sensor field model. Technical report, 2011. FRONTS-TR-2011-16. *

[ASS10] Carme Àlvarez, Maria Serna, and Paul G. Spirakis. On the computational power of constant memory sensor fields. Technical report, 2010. FRONTS-TR-2010-10. *

[BCG11] Luca Becchetti, Ioannis Chatzigiannakis, and Yiannis Giannakopoulos. Streaming techniques and data aggregation in networks of tiny artefacts. *Computer Science Review*, 5(1), 2011. FRONTS-TR-2011-7. *

[BK09] Luca Becchetti and Elias Koutsoupias. Competitive analysis of aggregate max in windowed streaming. In *ICALP '09: 36th International Colloquium on Automata, Languages and Programming*, pages 156–170. Springer-Verlag, July 2009. FRONTS-TR-2009-9. *

[CDF^{+}09] Ioannis Chatzigiannakis, Shlomi Dolev, Sándor P. Fekete, Othon Michail, and Paul G. Spirakis. Not all fair probabilistic schedulers are equivalent. In *13th International Conference on Principles of Distributed Systems (OPODIS)*, volume 5923 of *Lecture Notes in Computer Science*, pages 33–47, Berlin, Heidelberg, 2009. Springer-Verlag. *

[CHL^{+}09] Jeffrey Considine, Marios Hadjieleftheriou, Feifei Li, John Byers, and George Kollios. Robust approximate aggregation in sensor data management systems. *ACM Trans. Database Syst.*, 34(1):1–35, 2009.

[CMN^{+}10a] Ioannis Chatzigiannakis, Othon Michail, Stavros Nikolaou, Andreas Pavlogiannis, and Paul G. Spirakis. All symmetric predicates in $NSPACE(n^2)$ are stably computable by the mediated population protocol model. In *35th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 6281 of *Lecture Notes in Computer Science*, pages 270–281. Springer-Verlag, August 23–27 2010. FRONTS-TR-2010-17. *

[CMN^{+}10b] Ioannis Chatzigiannakis, Othon Michail, Stavros Nikolaou, Andreas Pavlogiannis, and Paul G. Spirakis. Passively mobile communicating logarithmic space machines. Technical report, RACTI, Patras, Greece, 2010. FRONTS-TR-2010-16. *

[CMN^{+}11a] Ioannis Chatzigiannakis, Othon Michail, Stavros Nikolaou, Andreas Pavlogiannis, and Paul Spirakis. Passively mobile communicating machines that use restricted space. *Theoretical Computer Science*, 2011. FRONTS-TR-2011-13. *

[CMN^{+}11b] Ioannis Chatzigiannakis, Othon Michail, Stavros Nikolaou, Andreas Pavlogiannis, and Paul G. Spirakis. Passively mobile communicating machines that use restricted space. Technical report, June 2011. FRONTS-TR-2011-28. *

[CMNS11] Ioannis Chatzigiannakis, Othon Michail, Stavros Nikolaou, and Paul Spirakis. The computational power of simple protocols for self-awareness on graphs. Technical report, Research Academic Computer Technology Institute (CTI), Patras, Greece and Computer Engineering and Informatics Department (CEID), University of Patras, Patras, Greece., 2011. FRONTS-TR-2011-14. *

[CMS09] Ioannis Chatzigiannakis, Othon Michail, and Paul G. Spirakis. Mediated population protocols. In *36th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 5556 of *Lecture Notes in Computer Science*, pages 363–374. Springer-Verlag, July 2009. FRONTS-TR-2009-8. *

[CMS10a] Ioannis Chatzigiannakis, Othon Michail, and Paul G. Spirakis. Algorithmic verification of population protocols. In *12th International Symposium on Stabilization, Safety, and Security of Distributed Sys-* *

tems (SSS), volume 6366 of *Lecture Notes in Computer Science*, pages 221–235. Springer-Verlag, September 2010. FRONTS-TR-2010-12.

* [CMS10b] Ioannis Chatzigiannakis, Othon Michail, and Paul G. Spirakis. Stably decidable graph languages by mediated population protocols. In *12th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*, volume 6366 of *Lecture Notes in Computer Science*, pages 252–266. Springer-Verlag, September 2010. FRONTS-TR-2010-20.

* [DMS11] Josep Díaz, Dieter Mitsche, and Paolo Santi. Theoretical aspects of graph models for MANETs. In S. Nikoetseas and J. Rolim, editors, *Theoretical Aspects of Distributed Computing in Sensor Networks*, Monographs in Theoretical Computer Science, pages 161–190. Springer, 2011. FRONTS-TR-2009-52.

* [DMSM$^+$11] Josep Diaz, Alberto Marchetti Spaccamela, Dieter Mitsche, Paolo Santi, and Julinda Stefa. Social-aware forwarding improves routing performance in pocket switched networks. Technical report, FRONTS-TR-2011-16, 2011. FRONTS-TR-2011-16.

[ED07] M. Haah E. Daly. Social network analysis for routing in disconnected delay-tolerant manets. In *Proceeding of ACM MobiHoc*, pages 32–40, 2007.

* [FNP$^+$10] Apostolos Filippas, Stavros Nikolaou, Andreas Pavlogiannis, Othon Michail, Ioannis Chatzigiannakis, and Paul G. Spirakis. Computational models for wireless sensor networks: A survey. In *1st International Conference for Undergraduate and Postgraduate Students in Computer Engineering, Informatics, related Technologies and Applications (EUREKA!)*, 2010. FRONTS-TR-2010-18.

* [Gia10] Yiannis Giannakopoulos. An experimental evaluation of the partition-greedy algorithm for the online aggregate max problem. Technical report, University of Athens, Department of Informatics, January 2010. FRONTS-TR-2010-13.

[GS66] Seymour Ginsburg and Edwin H. Spanier. Semigroups, presburger formulas, and languages. *Pacific Journal of Mathematics*, 16:285–296, 1966.

* [MCN$^+$10] Othon Michail, Ioannis Chatzigiannakis, Stavros Nikolaou, Andreas Pavlogiannis, and Paul Spirakis. New models for population protocols. 5th Athens Colloquium on Algorithms and Complexity, August 2010. FRONTS-TR-2010-23.

∗ [MCS11a] Othon Michail, Ioannis Chatzigiannakis, and Paul G. Spirakis. Mediated population protocols. *Theoretical Computer Science (TCS)*, 2011. FRONTS-TR-2011-12.

[MCS11b] Othon Michail, Ioannis Chatzigiannakis, and Paul G. Spirakis. *New Models for Population Protocols*. N. A. Lynch (Ed), Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool, 2011. FRONTS-TR-2011-10.

[MMS11] A. Mei, G. Morabito, and P. Santi. Social-aware stateless forwarding in pocket switched networks. In *Proceeding of ACM MobiHoc*, 2011.

[PH08] E. Yoneki P. Hui, J. Crowcroft. Bubble rap: Social-based forwarding in delay tolerant networks. In *Proceeding of ACM MobiHoc*, pages 241–250, 2008.

∗ [RC09] Michael Resvanis and Ioannis Chatzigiannakis. Experimental evaluation of duplicate insensitive counting algorithms. In *13th Panhellenic Conference on Informatics (PCI 2009)*. IEEE, September 2009. FRONTS-TR-2009-19.

∗ [SMC11] Paul Spirakis, Othon Michail, and Ioannis Chatzigiannakis. *Population Protocols and Related Models*, pages 109–160. Number 928 in Monographs in Theoretical Computer Science. Springer-Verlag, 1st edition, January 2011. FRONTS-TR-2011-11.

[TS08] C.S. Raghavendra T. Spyropoulos, K. Psounis. Efficient routing in intermittently connected mobile networks: The multi-copy case. *IEEE Transaction on Networking*, 16:77–90, 2008.