SEVENTH FRAMEWORK PROGRAMME
THEME 3
Information and Communication Technologies

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

Deliverable D1.2:

Models and Algorithmic Consequences for Dynamicity, Restricted Locally Computation, and Cooperation

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

Responsible Partner: UPC
Report Preparation Date: Friday, 12 February 2010

Contents

1 Introduction

In the second year of FRONTS, there have been 27 publications within WP1. In the present deliverable we are going to concentrate on a few of these publications We start in Section 2 by describing our progress in the development of the two models for the behavior of networks of sensors: population models and sensor field automata. In Section 3 we present a survey dealing with recent mobility issues. Finally, in Section 4 we present some related algorithmic work done in the second year of FRONTS.

2 Population Protocols and Sensor Field Automata models

In spite of the difficulties, there is no doubt on the importance of studying systems of tiny artefacts. Several proposals (taxonomies and surveys) which elucidate the distinguishing features of sensor networks, and which come from different scientific areas, have been published [ASSE02, ECPS02, Nag05, TAGH02, VRAC08, Xu02]. These proposals state clearly the need of formal models that capture the clue characteristics of sensor networks and that enable their study through the possibility of predictict their behavior, usefulness, effectiveness, efficiency and optimality in any kind of applications.

One of the aims of WP1 is to propose and analyze formal models that could play a fundamental role in the analysis and design of algorithms for such networks. During the first year or FRONTS we have isolated two general formal models for sensors, the Population Protocol model, introduced in [AAD$^+$06], and the Sensor Field model, that was defined within this project. During the second year we have carried out research in both of the models. The results are included in 9 FRONTS Technical Reports: 2009-{3, 8, 12, 16, 21, 29, 50, 51} and 2010-10.

2.1 Population Protocols: Extensions, Fairness, and Verification

The *Population Protocol* model [AAD$^+$04, AAD$^+$06] is a minimal theoretical framework for studying systems consisting of totally asynchronous agents, each equipped with $\mathcal{O}(1)$ bits of memory. The agents follow some completely unpredictable, but *fair*, mobility pattern and interact in pairs when they come sufficiently close to each other. In [AAD$^+$04, AAD$^+$06] it was proved that the class of solvable problems by this model is fairly small: *semilinearity* is a sufficient and necessary condition for solvability.

2.1.1 Mediated Population Protocol Model

In [CMSerb] we proposed the *Mediated Population Protocol* model (MPP), in order to extend the Population Protocol model. A communication graph (or digraph) denotes the permissible pairwise interactions. The main feature of our extended model is that it allows edges of the communication graph, G, to have *states* that belong to a constant size set. Moreover, edges have *readable only costs*, whose values also belong to a constant size set, and protocol rules for pairwise interactions can modify the corresponding edge state. Thus, our protocol specifications are still independent of the population size and do not use agent ids, i.e., they preserve *scalability*, *uniformity* and *anonymity*. Mediated Population Protocols (MPP) can stably compute interesting graph properties of the communication graph. We have shown this for the properties of maximal matchings (in undirected communication graphs), and also for finding the transitive closure of directed graphs and for finding all edges of small cost.

Most importantly, mediated protocols are stronger than the classical population protocols. In [CMSerb] we present a mediated protocol that stably computes the product of two positive integers, when G is the complete graph (which is not a semilinear predicate). In order to show the correctness of the protocol, we had to state and prove a general theorem about the composition of two stably computing mediated population protocols. Finally, we have obtained some non-uniform upper bound on computability: all predicates stably computable in this model are (non-uniformly) in the class $NSPACE(m)$, where m denotes the number of edges of the communication graph.

2.1.2 Graph Decision Mediated Model

The *Graph Decision Mediated Population Protocol* model (GDM) is a simplified version of MPP aiming to capture MPP's ability to *decide* (stably compute) *graph languages* (sets of communication graphs). Understanding properties of the communication graph is an important step in almost any distributed system. We proved in [CMSera] that any graph language is *undecidable* if we allow disconnected communication graphs. As a result, we focused on studying the computational limits of the GDM model in (at least) *weakly connected* communication graphs only, and gave several examples of decidable graph languages in this case. To simplify our task, first of all we proved that the class of decidable graph languages is *closed under complement*, *union* and *intersection* operations. Node and edge parity, bounded out-degree by a constant, existence of a node with more incoming than outgoing neighbors, and existence of some directed path of length at least $k = \mathcal{O}(1)$ are some examples of decidable properties by the GDM model (protocols for them are known, either directly by code or indirectly by exploiting the closure results).

An interesting finding of [CMSera] was the existence of symmetry in two specific communication (sub)graphs which may constitute the first step towards the proof of *impossibility results* in the GDM model. In particular, it has been proven that there exists no GDM, whose states eventually stabilize, to decide whether G contains some directed cycle of length 2 (2-cycle). Unfortunately, for an impossibility result to have important theoretical consequences, it should somehow avoid the *stabilizing states* restriction.

2.1.3 The PALOMA Model

We propose here another extension that we call the PALOMA model. In this model, the system consists of PAssively mobile LOgarithmic-space MAchines. The idea is to provide each agent with a memory whose size is logarithmic in the population size, which seems a very natural assumption: only 100 bits are required for 2^{100} agents (which is an astronomical population size)! Moreover, we can think of an agent as a small Turing Machine, which also seems natural: mobile phones, PDAs and many other common mobile devices are in fact sophisticated Turing Machines. It turns out that the PALOMA model is also very strong, since it can stably compute any symmetric predicate in $SPACE(n \log n)$.

2.1.4 Equivalence Between Schedulers

In [CDF$^+$er] we proposed a novel, generic definition of *probabilistic schedulers* for population protocols. Based on it, we identified the *consistent* probabilistic schedulers, and proved that any consistent scheduler that assigns a non-zero probability to any transition $i \to j$, where i and j are configurations satisfying $i \neq j$, is *fair* with probability 1. This new theoretical framework aims to simplify proving specific probabilistic schedulers fair. We additionally proposed two new probabilistic schedulers, the *State Scheduler* and the *Transition Function Scheduler*. Both possess the significant capability of being *protocol-aware*, i.e., they can assign transition probabilities based on information concerning the underlying protocol. Those schedulers, and also the *Random Scheduler* that was defined by Angluin et al. [AAD$^+$04] are fair with probability 1 (it can be proven with the help of our theoretical framework). In [CDF$^+$er] we also defined and studied *equivalence* between schedulers w.r.t. *performance* (*time equivalent* schedulers) and *correctness* (*computationally equivalent* schedulers) and reached the following conclusions:

1. The *protocol-oblivious* (or *agnostic*) Random Scheduler is not time equivalent to the State and Transition Function Schedulers, although all three are fair probabilistic schedulers (with probability 1). The statement can be proven by studying the performance of the *One-Way Epidemic Protocol* (*OR Protocol*) under these schedulers. To illustrate the unexpected performance variations of protocols under different fair probabilistic schedulers, one can slightly modify the State Scheduler to obtain a scheduler that may be adjusted to lead the One-Way Epidemic Protocol to *arbitrarily bad performance*.
2. The Random Scheduler is not computationally equivalent to the Transition Function Scheduler. To prove the statement we have studied the *Majority Protocol*

w.r.t. correctness under the Transition Function Scheduler. It turns out that the minority may win with constant probability under the same initial margin for which the majority w.h.p. wins under the Random Scheduler (as proven in [AAE07]).

2.1.5 Algorithmic Verification

We believe that, for applying our protocols to real, critical application scenarios, some form of computer-aided *verification* is necessary. Even if a protocol is followed by a formal proof of correctness, it would be safer to verify its code before loading it to the real sensor nodes.

Let $BP-VERIFICATION$ ('B' is from "Basic" and 'P' is from "Predicate") be the following problem: "Given a population protocol $\mathcal{A}$ for the basic model whose output alphabet $Y_{\mathcal{A}}$ is binary (i.e. $Y_{\mathcal{A}} = \{0, 1\}$), a first-order logical formula ϕ in Presburger arithmetic representing the specifications of $\mathcal{A}$, and an integer $k \geq 2$ (in binary), determine whether $\mathcal{A}$ conforms to its specifications for the complete digraph of k nodes-agents. (Here, "conforms to ϕ" means that for any legal input assignment x, which is a $|X_{\mathcal{A}}|$-vector with non-negative integer entries that sum up to k, and any computation beginning from the initial configuration corresponding to x on the complete communication graph of k agents, the population stabilizes to a configuration in which all agents output the value $\phi(x) \in \{0, 1\}$. On the other hand, "does not conform" means that there is at least one computation of $\mathcal{A}$ on the complete communication graph of k nodes which is unstable or the stable output does not agree with $\phi(x)$ - i.e. not all agents output the value $\phi(x)$.)".

The $BP-VERIFICATION$ is a surrogate of the more general $P-VERIFICATION$ problem (in which k is not provided and general correctness is required). Unfortunately, both problems and some of their special cases seem to be NP-hard.

Due to the hardness of protocol verification, we have focused on designing, implementing and extensively testing, under various combinations of protocols, specifications and communication graphs, a first exponential-time algorithm for $BP-VERIFICATION$ in order to observe its performance in practice and proceed with the required refinements.

In particular, the verifier expands the transition graph [AAD+06] corresponding to the pair $(\mathcal{A}, k)$ (only its reachable portion) and finds its strongly connected components by some known algorithm (e.g., [Tar72]). Based on the strongly connected components and the reachability between them, the algorithm constructs the corresponding dag and then checks whether there exists some directed path initiating from some *initial component* s (strongly connected component with some initial configuration) and ending to a *final component* s' (strongly connected component with no outgoing edges to other components) such that there exists some configuration in s' under which not all agents output the value (from $\{0, 1\}$) that all initial configurations of s require w.r.t. the specifications ϕ.

Finally, we would like to investigate the verification problem also for MPP and GDM protocols and possibly for the recently proposed *community protocols* [GR09] (in the latter model agents, are equipped with unique ids from the factory and are allowed to store a constant number of other agents' ids).

2.2 Sensor Field: Model Definition, Sensing Problems, Constant Memory Devices

The general sensing setting can be described by two elements: the observers or end users and the phenomenon, the entity of interest to the observers that is monitored and analyzed by a network with sensors. The corresponding information is discretized in two ways: first the environment is sampled on a discrete set of locations (sensor positions), and second the measures taken by the sensors are digitalized to the corresponding precision. To analyze the correctness and performance of the system, we are faced with a double task; on one side there is a computational problem to be solved by a particular network; on the other hand, it is necessary to assess whether a computed solution is a valid observation of the phenomenon. Both tasks require different analysis tools and we concentrate on the first. The distinctive peculiarities of the computational system define new parameters to be evaluated in order to measure the performance of the system. Metrics are needed to allow us to estimate the suitability of a specific or generic network topology or the possibility of emergent behavior with pre-specified requirements.

The computational system can be modeled by combining the notion of graph automata together with distributed data streams [GT01], a combination inspired by similar ideas developed in the context of concurrent programming [Hoa81]. Existing models coming from distributed systems [Pel00], hybrid systems and ad-hoc networks [Hen96, LSV01] capture some of such networks. Models coming from the area of population protocol models [AAD$^+$06, CMSerb] represent sensor networks, supposing that the corresponding sensing devices are extremely limited mobile agents (a finite state machine) that interact only in pairs by means of a communication graph and use a different approach.

2.2.1 The Sensor Field Model

In [ADGSer] we proposed a general model capturing some characteristic differences of sensor networks. A *Static Synchronous Sensor Field* (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 have 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, i.e., 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, etc.) and all what is needed to describe completely the instantaneous configuration of the local computation. Given a local state of a device, the communication 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 beginning 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 by sending data to their neighbors and to the environment.

The stream behavior of computation of a SSSF $\mathcal{F}$ with n devices is defined by a pair composed by $(\mathbf{u}, \mathbf{v})$, where $\mathbf{u}$ is an n-tuple of input data streams $\mathbf{u} = (u_k)_{1 \leq k \leq n}$, and $\mathbf{v}$ is a tuple of output data streams $\mathbf{v} = (v_k)_{1 \leq k \leq m}$, for some $m \leq n$. In this case, we considered that $\mathcal{F}$ outputs a tuple of output data streams $\mathbf{v}$ given a tuple of input data streams $\mathbf{u}$ and we defined the function $f_{\mathcal{F}}$ associated to the stream behavior of $\mathcal{F}$. We assumed that a function f defined on pairs of tuples of (infinite) data streams is computed by a sensor field $\mathcal{F}$ (with latency d) if f coincides with $f_{\mathcal{F}}$ (from the d-th data item of the data streams on).

The computational resources used by a sensor to compute a function of this kind 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 sensors, compute, send information and output data.
- *Space*. The space used by the device in each computation round.
- *Message Length*. The maximum number of data items of a message sent by the device in each computation round.
- *Number of messages*. The maximum number of messages sent by the device in each round.

We consider the following worst-case complexity measures taken over any device and computation round of a sensor field $\mathcal{F}$:

- $Size$: The number of nodes or devices of the communication graph G.
- $Time$ ($\mathcal{T}$): The maximum time used by any device in any of its rounds.
- $Space$ ($\mathcal{S}$): The maximum space used by any device of in any of its rounds.
- $MessageLength$ ($\mathcal{L}$): The maximum message length of any device in any of its rounds.
- $MessageNumber$ ($\mathcal{M}$): The maximum number of messages sent by any device in any of its rounds.

In general, we analyzed these complexity measures with respect to the $Size$ of the communication graph which usually coincides with the number n of data streams. We did not consider explicitly the energy consumption as a performance measure. For making an energy analysis we should have to incorporate a particular energy model to the sensor field. The performance measures defined above proportionate the basic ingredients for analyzing energy consumption where sending/receiving a message has the same cost for all the nodes, like for example the unit disk graphs. It is of interest (and a topic of future research) to consider energy models in which each link in the communication graph has different weights (or set of weights) representing the constants in the function that determines the cost of sending a message along the link.

We considered the possible memory restrictions of the *tiny* devices involved in a SSSF, since, in many applications, the devices have limited memory. The SSSF model can be adapted to take into account this fact, therefore we considered devices with constant or bounded memory capacity. To this end, we also assumed that each device has a buffer of bounded size to store the data received from its neighbors. We assumed that the communication graph might have any degree, but that a device cannot receive more packets in one round than those that can fit in the buffer. In the case that there are more incoming packets, then an arbitrary subset of them, filling the buffer, will be retrieved. In the opposite direction, we assumed that sending one data item to all the outgoing neighbors can be performed in constant time and space; here, the basic communication primitive is a broadcast to all neighbors, which is common in wireless networks. We denote by CMSF the variant of SSSF in which the devices have constant size memory.

It is of interest, and part of our future research, to analyze sensor fields (with or without constant memory) in which the reception of more than one message at the same time results in a collision.

2.2.2 Solving Sensing Problems

Computational problems that are susceptible of being solved by sensor fields can be stated in the following way:

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 $m \leq n$ so that output data streams and input data streams satisfy a given relation, i.e., the property defining the problem.

We considered that a sensing problem is solved by a sensor field $\mathcal{F}$ (with latency d) when the function $f_{\mathcal{F}}$ coincides with the relation or property that defines the problem.

We studied some particular sensing problems (the Average Monitoring and the Alerting problems) and, supposing that the sensor field has as many devices as input streams, we analyzed solutions for several topologies. We obtained upper and lower bounds on the solutions to the problem, and for some concrete topologies (rings, lines, trees) we obtained optimal algorithms.

Regarding the use of memory, as a function of the number of nodes, we showed that the Alerting problem requires logarithmic size memory per sensor, while the Alerting problem can be solved with constant memory per sensor.

In general, we proved that, by restricting the memory capacity of each device to be a constant w.r.t. the total number of devices, the class of problems solved by these SSSFs belongs to the class $DSPACE(O(n + m))$. Recall that in [CMSerb] it is shown that all predicates stably computed in the model of Mediated Population Protocols are in the class of $NSPACE(O(m))$. In this case, nondeterminism is required to verify that there exists a stable configuration reachable from the initial configuration, while in the Sensor Field case this is not required.

We also analyzed whether the use of additional nodes in the network that act as communicators, but that do not have access to input data streams, therefore they are not sensors, can improve the memory or the time required for solving a problem.

We showed that, for any given property $\mathcal{P}$ which is computable in polynomial time, there is a SSSF that solves the sensing problem associated to $\mathcal{P}$ in polynomial size and latency (with respect to n) with $\mathcal{S}(n) = \mathcal{T}(n) = \mathcal{L}(n) = \mathcal{M}(n) = O(1)$.

2.2.3 Simulating Population Protocols by Sensor Fields

We analyzed the computational power of sensor fields for the particular case in which the memory per device is constant, in relation to the Population Protocol model [AAD$^+$06] and one of its variations introduced in this project, the Mediated Population Protocol [CMSerb]. We showed that the constant memory Sensor Field (CMSF) model is more powerful that the population protocol (PP) by providing adequate simulation of the models. The results are given in [ASSer].

Recall that in a PP the interaction among components is described through an infinite sequence of pairwise interactions determined by a scheduler. This interaction scheduler models the interactions defined by the environment. In contrast, a Sensor Field interacts with the environment through input/output data streams in such a way that a data item can be read/written at each time step.

We show that any PP can be simulated by a CMSF. The simulation is done by defining a mapping of the input and the scheduler of a given PP, that models the input and the sequence of pairwise interactions, to an input data stream. Both systems act alike, except for a constant simulation overhead in time due to the required sequence of transmission and reception of messages. That is, the simulation of a transition of the PP requires a time step of constant duration on the local computation of the nodes in the CMSF. The simulation does not require any increase in the size of the network and it is state preserving. In Population Protocols it is required that the scheduler is fair in the sense that any possible pairwise interaction eventually occurs. Fair schedulers give rise to a particular kind of input data stream that verify this fairness condition.

Mediated Population Protocols keep not only the node state but also an arc state for any link in the communication network. For Mediated Population Protocols we provided two different simulations by SSSF. In the first one, the information about the link state is kept and managed by the environment; of course this, might require additional restriction on the set of valid data streams, but does not require additional devices. In the second simulation, we enlarge the sensor field with an additional node per link. In both cases, we showed that any Mediated Population Protocol can be simulated by a SSSF by an adequate transformation of the input and the scheduler and, when required, of the link states, into an input data stream. Again, the simulation of a transition on the PP requires a constant number of transitions on the CMSF.

It remains open to understand the exact conditions on the input data stream that guarantee that the sensor field obtained by the simulation correctly converges to the same final state (or final output), allowing for more freedom to the type of interaction than in the population protocol. In our simulation, the environment is used to keep only a pair of nodes alive and able to communicate at each time step, as required in the Population Protocol model.

2.2.4 Deciding Graph Properties

The Graph Decision Mediated Population Protocol (GDM) is a model in which the protocol does not have inputs, and it has been used within the project to assess decidability of properties of the communication graph [CMSera]. We adopted the same approach for the CMSF model. The results are given in [ASSer]. Now we assume that the sensor field has constant memory but no access to any input data stream. We analyzed the decidability of graph properties under the following variants of constant memory sensor fields:

- The simplistic CMSF: Here we assume that all the nodes have the same initial state and no input data stream is accessible.
- Node driven CMSF computation: We enlarge the computational power of the simplistic CMSF by adding an input data stream that at any time step selects a

node. This is a mechanism to control the execution of the protocol similar to the scheduler in Population Protocols, but now the environment is selecting a node rather than an edge. On a valid computation we assume, in addition, fairness for the input data stream in the sense that any existing node will always be eventually selected by the environment.

- Edge driven CMSF computation: We enlarge the computational power of the simplistic CMSF by adding an input data stream that at any time step selects a particular link and uses a link state, similar to the one used in the Mediated Population Protocol. Towards this end, we modify the communication graph with additional nodes that correspond to the edge state.

We showed that any non-trivial graph language in the graph class formed by all digraphs is undecidable in the three variations of the CMSF models. Any non-trivial graph property on the family of strongly connected digraphs ($\mathcal{S}$) is undecidable by simplistic BMSF with topology on $\mathcal{S}$. However, for the class of weakly connected digraphs ($\mathcal{W}$), any non-trivial property on $\mathcal{S}$ is undecidable by simplistic BMSF with topology on $\mathcal{W}$, but there are non trivial properties on $\mathcal{W}$ that are decidable.

For the node driven computation we showed that any non-trivial graph property in $\mathcal{S}$ is undecidable by node driven BMSF on $\mathcal{W}$. Any graph property decidable by simplistic CMSF on $\mathcal{W}$ is decidable by node driven CMSF on $\mathcal{W}$. However, when restricted to strongly connected graphs, some non-trivial graph languages are decidable:

- $kC = \{G \in \mathcal{S} \mid G$ contains a k-cycle$\}$, for any constant k.
- Directed simple path of constant length.

Observe that kC is a non-trivial graph property (even on strongly and weakly connected graphs), and therefore the language is not decidable neither in the simplistic CMSF nor in the node driven CMSF on topologies on $\mathcal{W}$.

Finally, we have showed that edge driven CMSF is more powerful than the other models. In particular, any graph property that is decidable by GDM is decidable by edge driven CMSF. Furthermore, it also has more computational power than the node driven GDM model, since any graph property that is decidable by node driven CMSF is decidable by edge driven CMSF. The exact computational limits of this model are the objective of our ongoing research.

2.2.5 Programming Sensor Fields

We conducted a validation of the capabilities of the definition of sensor field to analyze its viability, not only as a tool for analyzing different theoretical complexity or performance measures, but also as a tool to describe computation of sensor networks.

Based on the description of several protocols for the sensor field given in [ADGSer], we devised implementations of those protocols in the Nimo language. Nimo (Nets in Motion) is a programming language, a graphic-functional-data flow language, that allows a step by step visualization of the program executions under development at UPC [CZ]. The sensor field description provided the information required to code the corresponding programs in a reasonable amount of time. The results and the implementations are reported in [CDZer] and were presented at the 1st Workshop on New Challenges in Distributed Systems held in Valparaiso, Chile, April 6-9, 2009.

3 Dynamicity models

In [DMSer] we survey the main theoretical aspects of models for Mobile Ad Hoc Networks (MANETs). We present theoretical characterizations of mobile network structural properties and different dynamic graph models of MANETs, and we give detailed summaries of a few selected articles. In particular, we focus on articles dealing with connectivity of highly mobile networks, and on articles which show that mobility can be used to propagate information between nodes of the network, while at the same time maintaining small transmission distances and thus saving energy.

We devote a section to the different models of mobility that have been proposed by the scientific community. Recall that one of the main goals of these simplified models is to extract the topological properties of mobile networks, which might help both in improving simulation accuracy and in designing new protocols where mobility is used to reduce energy consumption and/or information propagation speed.

The most frequently used mobility models are the following two and their many variations: *The Random WayPoint model* (RWP) was first described in [JM96]. In this model, as usual, nodes are initially distributed uniformly at random on the region; then, each node chooses independently and uniformly at random a destination within the region, as well as a travel speed. The node then starts traveling towards the destination with the selected speed along a linear trajectory. When it reaches the destination (waypoint), it might optionally pause for a certain time, then chooses another waypoint in the region, and continues according to the same pattern.

A second type of model is *the Random Direction model* (RD), introduced in [Gue87], in which each node i in the region under consideration selects uniformly at random a direction $\theta_i \in [0, 2\pi)$, and chooses a speed that is kept constant during a certain amount of time. After a randomly chosen period of time, each node selects a new direction and speed, and continues moving. As the process evolves over time, some of the nodes might arrive at the boundary of the region, and a *border rule* has to be defined to determine how nodes behave when they hit the border. Then, the authors go into considering two of the important formal issues arising when using the RWP model of mobility, which is one of the most accepted models in the simulation community. The two structural properties are the *node spatial distribution* and the *instantaneous average nodal speed*.

We studied the impact of the above properties on the accuracy of RWP mobile network simulations, and theoretical characterizations of them that have been used to define a "perfect" simulation methodology, which completely removes the accuracy issues previously identified (following the line of work in [BRS03, YLN03]). Then, we discuss two of the main problems arising in the models: the *boundary effect* and the *speed decay*, explaining alternative solutions that have been proposed.

The remaining of the survey paper deals with the formal studies of connectivity on MANETs' models and their algorithmic consequences. We survey five theoretical models for MANETS. In the survey they are clustered by tasks, here they are mentioned by chronological order.

The first model is *the source-destination pairs-model.* [GT02], which is the first attempt to formally analyze a model of mobility, and suggests the important fact that high mobility can speed-up message passing using small connectivity range. The model is the following: there are n nodes ($n \to \infty$) all lying in the disk of unit area. The location of the i-th node at time t is given by the random variable $X_i(t)$. Each of the n nodes is a source node for one session and a destination node for another session, and each node i has an infinite stream of packets to send to its destination $d(i)$. The source-destination (S-D) association is established in the beginning and does not change over time. The nodes are mobile, but the mobility model described is non-constructive. It is a seminal great idea, but as discussed in more detail in the survey, the technical details are not clear. The second model, suggested by P. Santi [San05], studies the connectivity threshold for mobile networks: there are n vertices deployed uniformly at random in the unit square $[0,1]^2$. The nodes move randomly, but the mobility model is not fixed, it only must meet two conditions: it must be *bounded* (the support of the probability density function *pdf* of the long-term distribution of the nodes is contained in $[0,1]^2$) and *obstacle-free* (every subregion with non-zero measure has to have positive probability to contain at least one node at a given time). The paper proves that if $r_{\mathcal{M}}$ is the minimum value of the radius r, such that when taking a snapshot of the graph chosen from the long-term spatial distribution of the nodes the graph is connected, then a.a.s. $r_{\mathcal{M}} = c\sqrt{\frac{\log n}{\pi n}}$ with $c \geq 1$ (as in the static Random Geometric Graph's threshold). The author uses this result to analytically study the speed decay and the boundary effect for the RWP model, under different considerations of speed for each node.

During the first year of FRONTS ([DMPer]), we introduced *the DRGG model with radii r_c*, where r_c is the usual connectivity threshold for static Random Geometric Graphs. Based on the previous model, two new works have appeared in the second year of FRONTS. In [JMR09] we study a very general Random Direction type model with a radius below the threshold of the existence of a giant component. In particular, we use a a communication radius of $r_t = c/\sqrt{n}$, while the one in [DMPer] was $r_c = \sqrt{\frac{\ln n}{\pi n}}$. We give an upper bound on the speed at which information can be propagated between any pair of nodes. Recall that, in the static case, information between most pairs of nodes cannot be propagated since the largest connected component for the value of $\nu := n/\mathcal{A}$ to be considered has size $O(\log n)$. We show that mobility helps to propagate information. Technically the paper is very nice and uses innovative techniques to approaching this kind of problems.

Clementi et al. [CPS09] also consider the RGG model but with communication radius r between $\Theta(1)/\sqrt{n} \leq r \leq \sqrt{\frac{\ln n}{\pi n}}$. The authors provide tight asymptotic bounds for flooding the network with their choice of r. Again, they give a rigorous proof that, in their range of radii, flooding can be completed although at every time step the graph is disconnected. The proof uses a fine tessellation argument, where the RGG is approximated by a very fine grid on which the nodes are restricted to move. It is a discretized version (with respect to both time and space) of the models used in [JMR09, DMPer].

Considering the fact of using mobility to speed flooding, the papers of [JMR09] and [CPS09] are complementary: whereas the authors in [JMR09] study random geometric graphs with a radius below the thermodynamical threshold, [CPS09] considers the case of radii between the thermodynamical threshold and the threshold of connectivity. The work of [GT02] is orthogonal to these two since there is no absolute bound on the radius of transmission, but it also supports the hypothesis that mobility can help in propagating information.

4 Other Cooperation Models

In [CKSer] we study the performance of approximate Nash equilibria for linear congestion games. We consider how much the price of anarchy worsens and how much the price of stability improves as a function of the approximation factor ϵ. The paper gives (almost) tight upper and lower bounds for both the price of anarchy and the price of stability for atomic and non-atomic congestion games. These results not only encompass and generalize the existing results of exact equilibria to ϵ-Nash equilibria, but they also provide a unified approach which reveals the common threads of the atomic and non-atomic price of anarchy results. By expanding the spectrum, we also cast the existing results in a new light. For example, the Pigou network, which gives tight results for exact Nash equilibria of selfish routing, remains tight for the price of stability of ϵ-Nash equilibria.

In [DMRSer] we consider the possibility to achieve the “windfall of malice” even without the actual presence of malicious players. Recent results show that malicious players in a game may, counter-intuitively, improve social welfare. In [MSW06] it is shown that for a virus inoculation game, the existence of malicious players actually leads to better social welfare for the remaining players. This improvement in the social welfare with malicious players has been referred to as the "windfall of malice" [BKP09]. The existence of the windfall of malice for some games leads to an intriguing question: Can we achieve the windfall of malice even without the actual presence of malicious players? The paper provides positive and negative answers to the question. The authors introduce a general technique for designing mediators, which is inspired by a careful study of the "windfall of malice" effect. The mediator makes a random choice of one of two possible sets of proposed actions for each player, the first being optimal and the second being "fear inducing". The paper shows the applicability of the technique by designing a mediator for the virus inoculation game from [MSW06] that achieves a social welfare that is asymptotically optimal. On the other hand, the paper also shows the limits of the technique by proving an impossibility result that shows that for a large class of games, no mediator will improve the social welfare over the best Nash equilibrium. In particular, this impossibility result holds for the class congestion games, which are shown in [BKP09] to have a windfall of malice.

In [Toner] we consider the problem of communication scheduling in wireless networks with respect to the SINR (signal to interference plus noise ratio) constraint. The first result in this paper is a constant factor deterministic approximation algorithm for scheduling in wireless networks, in which all transmitter nodes have linear powers. To the best of our knowledge, this is the first constant factor approximation algorithm for this problem. In addition, we obtain the approximate value of the optimal schedule length with error at most a constant factor. This complements the results of [FKV09]. Other results in this paper include upper and lower bounds for the optimal schedule length for the case of uniform powers. The lower bound shows that the optimal schedule length for uniform power assignment is not shorter than the optimal schedule length for linear powers (for the same placement of the network nodes), if we admit an error of a constant factor.

A natural problem is to try to maintain the maximum value of the most recent items in a large sequence of values when there are restrictions on the memory of the algorithm. It is very useful to maintain some statistics over a sliding window, for example, the maximum value of the temperature observed by a set of sensor nodes over the last n seconds. Unlike previous approaches, we use the competitive analysis framework and compare the performance of the online streaming algorithm against an optimal adversary that knows the entire sequence in advance. We consider the problem of maximizing the *aggregate max*, i.e., the sum of the values of the largest items in the algorithm's memory over the entire sequence. For this problem, we prove an asymptotically tight competitive ratio, achieved by a simple heuristic, called partition-greedy, that performs stream updates efficiently and has almost optimal performance. In contrast, we prove that the problem of maximizing, for every time t, the value maintained by the online algorithm in memory, is considerably harder: in particular, we show a tight competitive ratio that depends on the maximum value of the stream. We further prove negative results for the closely related problem of maintaining the aggregate minimum and for the generalized version of the aggregate max problem in which every item comes with an individual window.

5 Ongoing and Future Work

The most important goal of WP1 of FRONTS is to formalize one or more models for heterogeneous MANETS. During the first two years of the project, individual and joint work of CTI and UPC has produced two models. During the third year, we plan to make a detailed taxonomy of network properties that are reflected by each model. We will also exploit the related research results conducted by other teams on sensor network mobility and communication scheduling, and incorporate them into the Population Protocol and Sensor Field models.

Another line of research is to use the ideas developed for dynamic networks in order to establish a theoretical model for studying message diffusion between agents in social networks. This research is a joint effort of UDRLS and UPC, and at the present it is at a very preliminary stage.

References

[AAD+04] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. In *23rd Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 290–299, New York, NY, USA, 2004. ACM.

[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*, 18(4):235–253, 2006.

[AAE07] Dana Angluin, James Aspnes, and David Eisenstat. A simple population protocol for fast robust approximate majority. In *21st International Symposium on Distributed Computing (DISC)*, volume 4731 of *Lecture Notes in Computer Science*, pages 20–32. Springer, 2007.

[ADGSer] Carme Àlvarez, Amalia Duch, Joaquim Gabarro, and Maria Serna. Sensor field: A computational model. In Shlomi Dolev, editor, *ALGOSENSORS 2009*, volume 5804 of *Lecture Notes in Computer Science*, pages
∗ 3–14. Springer-Verlag, 2009. FRONTS-TR-2009-50.

[ASSer] Carme Àlvarez, Maria Serna, and Paul Spirakis. On the computational
∗ power of constant memory sensor fields. 2010. FRONTS-TR-2010-10.

[ASSE02] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E.Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[BKP09] M. Babaioff, R. Kleinberg, and C.H. Papadimitriou. Congestion games with malicious players. *Games and Economic Behavior*, 67(1):22–35, 2009.

[BRS03] C. Bettstetter, G. Resta, and P. Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(3):257–269, 2003.

* [CDF$^+$er] Ioannis Chatzigiannakis, Shlomi Dolev, Sandor Fekete, Othon Michail, and Paul Spirakis. Not all fair probabilistic schedulers are equivalent. In *13th International Conference On Principles Of DIstributed Systems (OPODIS)*, volume 5923 of *LNCS*, pages 33–47. Springer-Verlag Berlin Heidelberg, December 2009. FRONTS-TR-2009-29.

* [CDZer] Silvia Clerici, Amalia Duch, and Cristina Zoltan. Implementing static synchronus sensor fields using nimo. Technical report, LSI UPC, 2009. FRONTS-TR-2009-51.

* [CKSer] George Christodoulou, Elias Koutsoupias, and Paul Spirakis. On the performance of approximate equilibria in congestion games. In *ESA*, pages 251–262, September 2009. FRONTS-TR-2009-60.

* [CMSera] Ioannis Chatzigiannakis, Othon Michail, and Paul Spirakis. Decidable graph languages by mediated population protocols. In *23rd International Symposium on Distributed Computing (DISC 2009)*, volume 5805 of *Lecture Notes in Computer Science*, pages 239–240. Springer-Verlag, September 2009. FRONTS-TR-2009-16.

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

[CPS09] A.E.F. Clementi, F. Pasquale, and R. Silvestri. MANETS: High mobility can make up for low transmission power. In *Proceedings of the 36th Internatilonal Collogquium on Automata, Languages and Programming: Part II*, pages 387–398. Springer, 2009.

[CZ] S. Clerici and C. Zoltan. A graphic functional-dataflow language. *Trends in Functional Programming*, 5:53–77.

∗ [DMPer] Josep Diaz, Dieter Mitsche, and Xavier Pérez. Connectivity for dynamic random geometric graphs. *IEEE Transactions in Mobile Computing*, 8:821–835, 2009. FRONTS-TR-2008-52.

∗ [DMRSer] Josep Diaz, Dieter Mitsche, Naveen Rustagi, and Jared Saia. On the power of mediators. In *5th Workshop on Internet and Network Economics (WINE 2009)*, volume 5929 of *Lecture Notes in Computer Science*, pages 455–463. Springer-Verlag Berlin Heidelberg, December 2009. FRONTS-TR-2009-44.

∗ [DMSer] Josep Diaz, Dieter Mitsche, and Paolo Santi. Theoretical aspects of graph models for manets. Technical report, UPC, 2009. FRONTS-TR-2009-52.

[ECPS02] D. Estrin, D. Culler, K. Pister, and G. Sukhatme. Connecting the physical world with pervasive networks. *IEEE pervasive computing*, pages 59–69, 2002.

[FKV09] Alexander Fanghänel, Thomas Kesselheim, and Berthold Vöcking. Improved algorithms for latency minimization in wireless networks. In *ICALP '09: Proceedings of the 36th International Colloquium on Automata, Languages and Programming*, pages 447–458, Berlin, Heidelberg, 2009. Springer-Verlag.

[GR09] Rachid Guerraoui and Eric Ruppert. Names trump malice: Tiny mobile agents can tolerate byzantine failures. In *36th Internatilonal Collogquium on Automata, Languages and Programming (ICALP 2009)*, volume 5556 of *Lecture Notes in Computer Science*, pages 484–495, 2009.

[GT01] P.B. Gibbons and S. Tirthapura. Estimating simple functions on the union of data streams. In *Proceedings of the thirteenth annual ACM symposium on Parallel algorithms and architectures*, pages 281–291. ACM New York, NY, USA, 2001.

[GT02] Matthias Grossglauser and David N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on Networking*, 10(4):477–486, 2002.

[Gue87] RA Guerin. Channel occupancy time distribution in a cellular radio system. *IEEE transactions on Vehicular Technology*, 36(3):89–99, 1987.

[Hen96] T. A. Henzinger. The theory of hybrid automata. In *LICS '96: Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science*, page 278, Washington, DC, USA, 1996. IEEE Computer Society.

[Hoa81] C. Hoare. Calculus of Total Correctness for Communicating Processes. *SCI. COMP. PROGRAM.*, 1(1/2):49–72, 1981.

[JM96] D.B. Johnson and D.A. Maltz. *Mobile Computing*, chapter Dynamic source routing in ad hoc wireless networks, pages 153–179. Kluwer Academic Publishers, 1996.

[JMR09] P. Jacquet, B. Mans, and G. Rodolakis. Information propagation speed in mobile and delay tolerant networks. In *29th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2009.

[LSV01] Nancy A. Lynch, Roberto Segala, and Frits W. Vaandrager. Hybrid i/o automata revisited. In *HSCC '01: Proceedings of the 4th International Workshop on Hybrid Systems*, pages 403–417, London, UK, 2001. Springer-Verlag.

[MSW06] T. Moscibroda, S. Schmid, and R. Wattenhofer. When selfish meets evil: Byzantine players in a virus inoculation game. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, page 44. ACM, 2006.

[Nag05] David J. Nagel. Wireless sensor systems and networks: Technologies, applications, implications and impacts, 2005.

[Pel00] D. Peleg. *Distributed computing: a locality-sensitive approach*. Society for Industrial Mathematics, 2000.

[San05] P. Santi. The critical transmitting range for connectivity in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 4(3):310–317, 2005.

[TAGH02] S. Tilak, N.B. Abu-Ghazaleh, and W. Heinzelman. A taxonomy of wireless micro-sensor network models. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(2):28–36, 2002.

[Tar72] Robert Endre Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1(2):146–160, 1972.

[Toner] Tigran Tonoyan. Scheduling in wireless networks with linear and uni-
∗ form power levels. FRONTS-TR-2010-2, 2010 .

[VRAC08] M. Vinyals, JA Rodriguez-Aguilar, and J. Cerquides. A survey on sensor networks from a multi-agent perspective. In *2th International Workshop on Agent Technology for Sensor Networks (ATSN-08)*, 2008.

[Xu02] N. Xu. A survey of sensor network applications. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[YLN03] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *21th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 2, pages 1312–1321. Citeseer, 2003.