SEVENTH FRAMEWORK PROGRAMME
THEME 3
Information and Communication Technologies

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

Deliverable D4.7:

Third Report on the Unification of Results

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

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

Contents

1 Introduction

The aim of the FRONTS project is to establish the foundations of adaptive networked societies of small or tiny heterogeneous artefacts. In a nutshell, we are working towards a science of adaptive organization of large networks of small or tiny artefacts.

Our approach to this goal is foundational and is based on algorithmic expertise. We intend to apply our models, methods and results to the scrutiny of large-scale simulations and experiments, from which we expect to obtain valuable feedback.

Our methodology follows basic principles of System Science. The networks of our study are (rather complex) "systems" of small and restricted parts, that may be organized via local communication. Dynamicity (and change) is a main characteristic, of both the network and its environment. Any computation is locally restricted, and no central control exists. Thus, the networks (systems) considered are *distributed* and cooperation among the tiny parts is not automatically guaranteed. Rather, it *has to be established* (in ad-hoc situations) and *maintained*, while adapting to dynamically changing external situations.

Still, if such systems are to be designed, they should be *"programmable"* and *trusted*. FRONTS aims to deliver at its end a set of well defined design rules for such systems.

We have decomposed our effort into four technical workpackages. As the contract indicates, they do no step in isolation, but they interact in a designed way. Here, we rephrase their goals:

- **WP1:** To state models, laws and complexity measures for our networks
- **WP2:** To answer how the network prepares internally to be ready to adapt
- **WP3:** To answer how the network reacts successfully to a dynamic external environment
- **WP4:** To verify our methods via experiments and simulations

For the 3rd year, we promised and delivered 4 technical deliverables for the 4 WPs, an issue in *Computer Science Review* and the unification of results:

- **D1.3:** A Set of Theories for Adaptively Organized Societies of Tiny Artefacts
- **D2.3:** A Set of Design Rules and Technical Recommendations for Adapting the Networks Infrastructure
- **D3.3:** A Set of Design Rules and Technical Recommendations for Adapting to the Dynamic Environment
- **D4.6:** A Complete Set of Real Testbed Experiments and Simulation Results
- Results Volume FRONTS: CSR Vol 5 Issue 1 Feb 2011

- 3^{rd} Unification Report

The effort of unifying our results, in the 3rd year of FRONTS, was seriously affected by the recommendations made by the review process at the end of the 2nd year. We summarize here the main points of the recommendations:

R1: Present a single and unified system providing a well-focused set of tasks.

R2: Evaluate the above on large-scale testbed deployments

R3: Focus on a selected set of FRONTS Schemes (from 2^{nd} year) to implement as components of the above.

R4: Report Conclusions on Theories Design Rules, Technical Recommendations.

The project took the recommendations into a very serious consideration. We provide below a high-level view of our answers and actions resulting from this process. We judge that the recommendations were of a great help in the direction of focusing our results and unifying our actions.

Highlight of Actions/Answers to the Recommendations:

R1: We delivered a single unified system, UNI, operating with a well-focused set of tasks. It has 2 layers.

R2: We jointly assessed the performance of our system based on experiments on real platforms, and on simulations (D4.6, Demo to the project, FET '11 Conference application)

R3: Considering UNI's layers we focused on the following components:

Layer 1: Neighborhood discovery, leader election/clustering, hierarchy construction and end-to-end communication

Layer 2: Data aggregation, group key establishment, motion planning, private tracking

We have implemented a set of FRONTS schemes as components based on: interfaces, realistic potential for integration, clarity of design and functionality, easiness of interaction with other components and innovative spirit.

R4: We concluded with 2 theories (formal models) in WP1 and a set of Design Rules and Technical Recommendations for adapting the infrastructure and adapting to the external environment (WP2 and WP3).

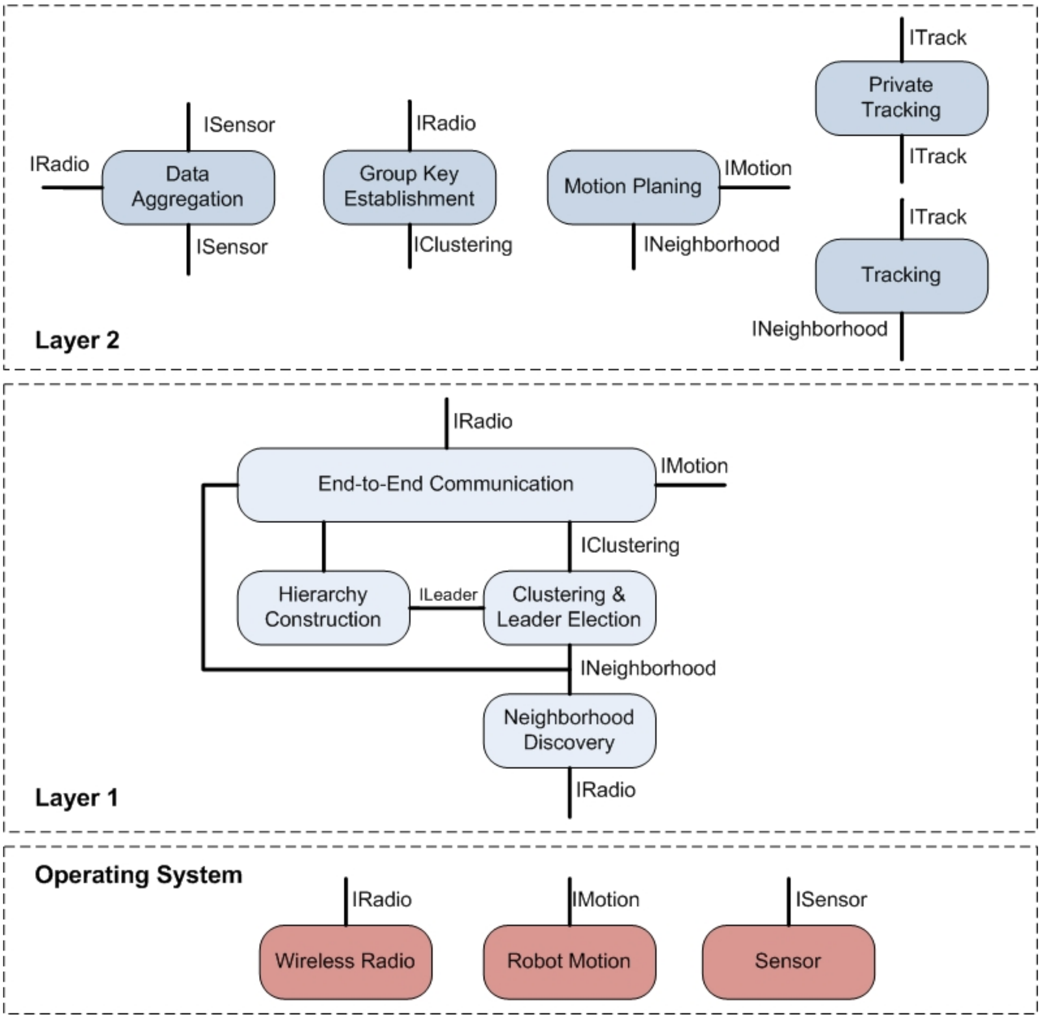

Figure 1: Relation of Components and Interfaces

2 Our Unified System

By the end of third year of FRONTS we delivered a single and unified system whose aim is to provide a self-stabilizing and self-organized society of tiny artefacts. The system comprises of two layers (from the bottom to the top):

Layer 1: Self-organizing and adaptive system reconfiguration

Layer 2: Reaction to external inputs

The *novelties* introduced by UNI concern: *Security* (keys, helpers, etc.), *Private Tracking* (additional security layer on top of Tracking module that prevents tracking by outsiders), *Delay-Tolerant communication* (E2E module, taking advantage of clustering, hierarchy and mobile agents to provide delay-tolerant and reliable connectivity), *Simple and Economical code per device* as well as *Clustering and Highways* (Virtual-representative nodes for each cluster) *construction*.

The interfaces for the inter-operation and integration of the components are depicted in Fig. 1. Figure 2 depicts the conceptual interaction of the components to

Component	Layer	Partner(s)
Neighborhood discovery	1	CTI
Leader election & Clustering	1	BGU, CTI
Hierarchy construction	1	UPC
End-to-end communication	1	UDRLS, UPB, TUBS
Data aggregation	2	CTI, UOA
Group key establishment	2	WROC
Motion Planning	2	UPB, TUBS
Tracking	2	UNIGE
Private tracking	2	UNISA

Table 1: The components of the FRONTS system, per layer and per responsible site

support applications (in the figure the pointed arrow $A \rightarrow B$ indicates a dependence of component A to component B).

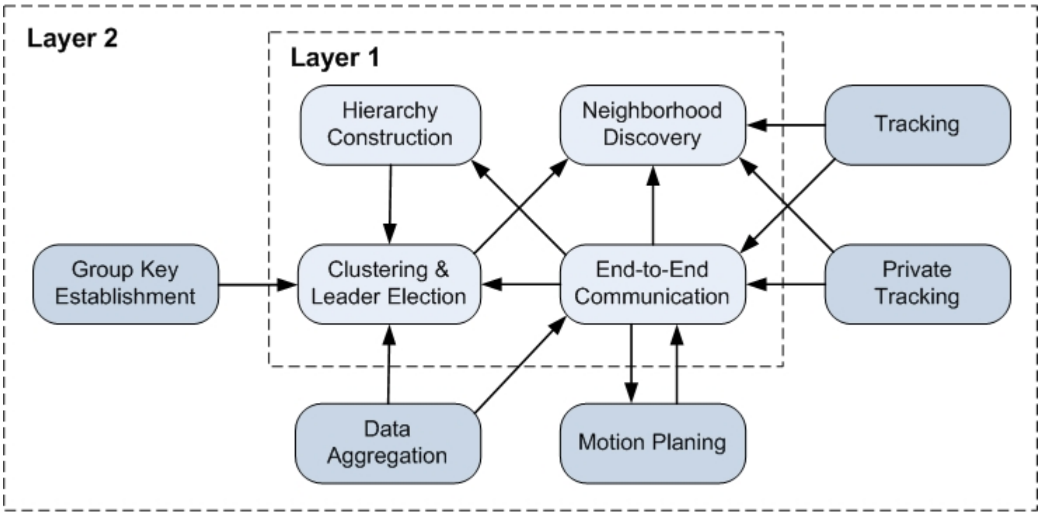

Figure 2: Component Interaction to Support Applications

The development of each component took into account existing code already in the code repository (from the previous years) which was further extended to support the new features previously described, and appropriately modified to make the components compatible with the FRONTS experimental testbed and to enable the inter-operation of the components. In Tab. 1 we list the 9 components for both layers and the partner sites that are responsible for the development.

Conclusions for UNI:

The various experiments and measurements conducted regarding the performance of UNI are presented in great detail in D4.6. In addition, a detailed list of notes, conclusions and open issues/further optimizations for each component is described in D4.6. These issues are summarized in Section 6. The development and experimentation phases involved the joint, hard work of all partners for many hours and numerous meetings.

3 Complementary Theories: Our Conclusions

The main goal of WP1 in FRONTS project was to define one or more theoretical models for networks of tiny artifacts, which could help in making simulations, designing new algorithms for communication, data retrieval and processing. During our research in the third year of FRONTS, we continued our efforts focusing on the models initially considered in the first year, the *Synchronous Sensor Field (SSF)* model and the *Population Protocols (PP)* model along with its various extensions we have proposed, *Mediated Population Protocols (MPP)*, *Passively mobile Machines (PM)*, *Mediated Graph Protocols (MGP)*, *Population Protocols with Absence Detection (PPAD)*.

General Properties and Characteristics:

The PP model and its enhancements (MPP, PM, MGP, PPAD) capture *stable convergence* of the system under *persistent input* (provided by the environment to the population). These models consider a system consisting of a population of agents that can interact with each other in pairs in unpredictable, uncontrollable but *fair* patterns. The agents' inability to control their interactions is called *passive mobility* and *fairness* is a characteristic of the interaction pattern which mainly ensures that all pairs of agents that are able to interact will eventually do so. The interactions are *local* and thus the systems described by such models are highly scalable. In addition, most of these models (in fact all except PM) consider *finite state agents* which renders the overall system *easily verifiable*. Each computational entity in the system is fairly restricted memory-wise but as it has been shown throughout our research, the computational power of the collective system is significantly greater (especially for the proposed extensions). The protocols considered within these models are *uniform* (each agent runs the same protocol) and *anonymous* (agents do not assume unique identifiers). Last but not least population protocols are now a well established theoretical model for networks of tiny artefacts.

The SSF model on the other hand captures continuous processing of *streams of input* (data streaming). The devices composing the system are considered *non-uniform* (some devices may be capable of environmental sensing while others may just process data streams from neighboring devices) and each device may have its *own transition function*. The systems is *synchronous*, that is, the computation is organized in rounds in which each device receives (either from the environment and/or from other agents), processes and broadcasts/outputs data values. The SSF is a new and promising model since it is a clear (from a mathematical point of view) and additionally takes into account the processing of streaming data which becomes increasingly popular especially in the context of wireless sensor networks.

Population Protocols Based Models:

Due to the basic PP model's weak computational power we introduced several extensions aiming at enhancing the protocols power and allow more useful computations. On the one hand, we raised the finite state constraint from the *agents* allowing them to *store information proportional to the size of the population in their local memory*. We call this model **PM** (referred as PALOMA in older bibliography) and we consider populations of interacting Turing Machines (TMs) instead of finite state automata. We showed that when allowing agents to use at least $\log n$, where n is the population size, the population can be organized into a distributed TM that uses the collective memory of the population to perform symmetric computations. We also showed that under a certain local memory threshold the protocols were computationally equivalent to the basic model.

On the other hand, we studied the possibility of *global memory distributed on states of the interactions*. In the model that we call **MPP** and that we introduced in the second year of FRONTS, we consider that the agents remain finite-state automata but their interactions could also be characterized by state drawn from a finite set. This addition proved to greatly strengthen the model in the case of a *complete interaction graph* (populations where any two agents are capable of interacting with each other). For this case we showed (during the third year of FRONTS) again that the population can exploit the full distributed memory and perform symmetric computations equivalent to those of a Nondeterministic TM (NTM) of $O(n^2)$ space, where again n denotes the size of the population.

Using the MPP model we explored the capability of *self-deciding* (becoming *self-aware*) *graph properties on the interaction graph* that the population runs on, by introducing two variations: The **GDMPP** model (initially referred as *GDM*) that was proposed in the second year of FRONTS, considers that *the input graph*/network, whose properties we are interested in, *is the whole interaction graph*. This variation was proven to be *fairly limited in the case of disconnected graphs*. The **MGP** model, introduced in the third year of FRONTS, deals with this case by considering a complete interaction graph where *the input graph is provided as some preprocessing on the network*. By exploiting the complete network infrastructure, MGP *can compute* non-trivial *graph properties on disconnected input graphs* (subgraphs of the complete interaction graph), even in the case where the input graph is *initially unstable* (changes for a finite period of time) and can simulate a nondeterministic Turing Machine of linear space that receives its input graph by its adjacency matrix representation.

Throughout the study of the models stated above, we have also proved that in the case of complete interaction graph the election of a leader/coordinator is easy. This discovery in combination with the PP *composition* property allows for clearer protocol descriptions. In addition, we studied the PP model from the perspective of protocol verification.

Finally, we explored the case where HW with new features is available in the population. More specifically, we very recently studied the case, that we call **PPAD**, where there is a special agent in the population capable of detecting the absence of a state of the protocol from the population. As we showed, such an addition substantially increases the computational power of the model.

The Sensor Field Computational Model:

Regarding the SSF model, we proposed in the previous years of FRONTS the Static Synchronous Sensor Field (SSSF) in which the links between the devices are fixed and cannot appear or disappear throughout the computation. The model allows us to express computation in terms of the stream behavior 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 static version of SSF model however fails to capture the inherent dynamic behavior that characterize such networks. During the third year of FRONTS we have extended the definition of the sensor field model to additionally capture some continuous sources of dynamicity and mobility, defining the *dynamic sensor field model*. We considered two potential sources of dynamicity and mobility: the *passive mobility of the targeted data* (like the one discussed above) 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.

Within this context we studied the *continuous average monitoring problem* and analyzed its message (number and length), time, space and distance complexity under three SSF scenarios. In a complementary analysis we have conducted different studies on *data streaming techniques* focusing more on *data stream aggregation*. The main limitation of the model is the assumption on synchronization. However, the experimental scenario of FRONTS considers synchronization at the lower layers of the communicating infrastructure.

4 Highlights & Focuses on Adapting the Internal Net Infrastructure

In the third year of FRONTS, we focused on a *few schemes per functionality* (as found in years 1 and 2) and developed them further in order to derive most interesting Design Rules and Recommendations.

Regarding the adaptation of the internal net infrastructure, we focused on two of the schemes presented in D2.2, namely the *fully decentralized recommendations in opportunistic networks* (SC1) and the *helper based security* (SC2) schemes and we discussed design rules and the technical recommendations derived by the performed experiments. The reasons behind these choices are that firstly, SC1 shows how a very limited amount of local information stored on nodes with limited computation and communication capabilities, can adapt due to opportunistic interactions to support the provisioning of global accurate services (e.g. recommendations) and secondly SC2 shows how the issues of constrained resources can be circumvented with the support of helpers, namely special devices woven into the environment allowing tiny artifacts to perform computations otherwise impossible with their limited resources (e.g. security functions).

Helpers:

In the helper based security scheme, *security helpers* are special devices, woven into the net, allowing all other devices to perform computations otherwise impossible. They are part of cryptographic protocols involving tiny artefacts with the specific task of helping these artefacts perform cryptographic operations. This set of special devices introduces heterogeneity in the network and as shown in the research conducted during the third year of FRONTS, it provides support to the rest of the devices to allow them quickly react to the environment in those cases where their computational and communication resources are not sufficient. Thus this addition advantages are traced beyond security.

With our research on helpers, we answered one crucial question: *"How much can we trust a security helper?"*. Ideally, one would like to be able to make minimal trust assumptions on the helpers but typically we believe it is reasonable to assume that the helper performs as previously described. In no case, however, the helper must receive secret information from the artefacts (like for example secret keys).

Efficient construction of untraceable tags:

In previous years of FRONTS, we discussed an efficient construction of untraceable tags in the model with randomizers, to be used for passive RFID tags *carrying private encrypted information*. The interaction with the *anonymizers* is intended to guarantee that tags are not traced, and this is achieved by re-randomizing the ciphertext (by producing a new ciphertext that carries the same message). The re-randomization procedure is carried on without having access to the private key used to generate the ciphertext. This extra important property *guarantees privacy of the information even with respect to the randomizers*. Our construction uses *pairing-based cryptography* that allows *high level of security with very limited storage*, making our construction applicable to RFIDs.

The results of the third year can be summarized in Table 2.

Our helpers and untraceable tags imply *a recommendation function* in a fully decentralized setting in which only *opportunistic communication* is available.

Table 2: Design Rules and Technical Recommendations for Adapting the Net Infrastructure

	Rule/Recommendation
1	Randomizers can help in preventing RFID tracing in a dynamic scenario.
2	The success probability of a tracer is directly influenced by its reading range.
3	The boundless mobility model appears to be the mobility model that is more resilient to tracing.
4	Tradeoffs between the number of tracers (adversaries) the number of tags and the number of helpers (randomizers) should be analysed.

5 Highlights & Focuses on Adapting to the Dynamic Environment

Within the context of adaptation to the dynamic environment, we focused, during the third year of FRONTS, on four schemes and provided some interesting Design Rules and Technical Recommendations. These schemes are summarized in Table 3.

Table 3: Schemes for Adapting to the Dynamic Environment

	Scheme
1	Building and maintaining formations of mobile sensors
2	Coping with Sybil attacks
3	Self-synchronized duty-cycling and minimum energy multicasting
4	Data Collection in mobile networks of small artefacts

Our focuses for each scheme throughout the third year are briefly presented below:

Building and maintaining formations of mobile sensors.

We focused on two formation problems: Firstly, we want the sensors to *form the shortest chain*, between two, given stationary nodes, in order to have a most efficient communication infrastructure. The second formation problem which we considered is the *gathering* problem. The goal is to move all robots to one common point, which is not predetermined. The robots have only a local view and should need as few capabilities as possible. In this context, we are mainly interested in energy efficient strategies Our findings showed that in order to cope with the changes in the environment, local algorithms are useful. The reason is that they have the property that local changes in the environment interfere only with the actions of nearby sensors, but still allow for energy efficient formation strategies.

Coping with Sybil attacks

Sybil attacks, i.e. cloning and forging identities is a common and major problem in ID-based systems. We addressed the problem of detection and counteraction against a Sybil type attack performed in order to increase the fraction of adversarial nodes in a network.

Self-synchronized duty-cycling and minimum energy multi-casting

Concerning *self-synchronized duty-cycling*, the main aim of our research was the development of a mechanism that allows sensor networks to perform duty-cycling in a self-organized way, without any central control. Interestingly enough, we found that the mechanisms of distributed cooperation exhibited by social insects, flocks of birds and fish schools, seem to be well-suited for the development of distributed, self-organized algorithms for the management and control of networks of tiny artefacts. Concerning the *minimum energy multi-casting* problem, the last year of the project was dedicated to a specific subproblem, namely the *minimum energy broadcast* (MEB) problem. The results on this work have shown that problems arising in networks of tiny artifacts should be modeled as closely as possible. As shown, it may happen that some algorithms suffer important performance losses when the problem is only slightly changed.

Data Collection in mobile networks of small artefacts

We considered wireless sensor networks where nodes have random and changeable mobility patterns. We study the problem where a particular node, called the base station, collects the data generated by the sensors/nodes. In order to limit the energy consumption due to the transmissions, data collection takes place in a multi hop way. Although the *static* version (where nodes are in fixed positions) of this energy-aware data gathering problem has been extensively studied, the mobile setting still requires a lot of work. Through our research we came to the conclusion that data gathering protocols in highly mobile wireless networks are able to achieve good performance *without having access to positions of nodes* (which is fairly expensive information).

6 Conclusions

Throughout our work in FRONTS during the third year we conducted various experiments as briefly listed in Table 4.

As our work on project FRONTS came to an end we have arrived to various Design Rules Technical Recommendations concerning our proposed Unified System.

- Concerning the hardware's behavior, the memory capabilities of the current WSN devices is very limiting. Extra memory would help increase buffer sizes, increase stack size and therefore the total number of scheduled tasks (that respond to event's triggered by external input), debugging information etc.. Regarding the robot motion, improved hardware abilities (more accurate sensors,

Table 4: Conducted Experiments

	Subject
1	Time to Stabilize
2	End-to-End Communication
3	Delay-Tolerant Communication
4	Failures
5	Establishment of Keys
6	Tracking
7	Private Tracking

compasses, localization of relative positions of sending nodes) would allow more advanced algorithms to be implemented for improved movement patterns.

- Concerning the severe communication restrictions of the nodes (e.g., compile time fixed payload restrictions on the radio messages, very low maximum payload per message, multiple modules controlling the radio), we recommend a very careful modular design to be used. Then each module should be first defined as an abstract datatype that will later on expose a generic handler (mediator) that is factored as an external sub-module. For each module it is important to make the module data public or providing/exposing methods to format the data to meet inter-module demands.
- It is important to establish from the early stages of the design and implementation cycle, an approach for checking the integrity of messages exchanged by the algorithms (e.g., checksums, error codes).
- An adequate method for preloading keyshares to each node must be determined. Our suggestions is that the keys should be loaded into flash memory so that they do not occupy RAM memory. Also closer examination (and, possibly, code patches) are required for different key-lengths, due to constraints imposed by AES implementation available in Wiselib.
- Careful selection of the module parameters, in particular those related to the adaptation to external events.
- In networks with dense topology it is preferable to select lower adaptation rates in order to avoid oscillations in message exchanges and therefore congestion in the wireless medium. The included micro delays on the spreading of information (e.g., in the cluster formation, traces in the tracking module etc.) are very slightly affected. On the other hand the overall stability of the system is greatly improved.

References

The union of references are included in the 4 WP reports (deliverables D1.3, D2.3, D3.3, D4.6). We do not repeat them here.