Coordination Mechanism and Customizable Hardware Platform to Provide Heterogeneous Wireless Sensor Networks Support

Edison P. de Freitas¹,², Rodrigo S. Allgayer², Tales Heimfarth³, Flávio R. Wagner⁴, Tony Larsson¹, Carlos E. Pereira², Armando M. Ferreira⁵

¹School of Information Science, Computer and Electrical Engineering
Halmstad University, Halmstad – Sweden

²Electrical Engineering Department – Federal University of Rio Grande do Sul
Porto Alegre – Brazil

³Computer Science Department – Federal University of Lavras
Lavras – Brazil

⁴Institute of Informatics – Federal University of Rio Grande do Sul
Porto Alegre – Brazil

⁵Defense Engineering Graduate Program – Military Institute of Engineering
Rio de Janeiro – Brazil

edison.pignaton@hh.se, allgayer@ece.ufrgs.br, tales@dcc.ufla.br,
flavio@inf.ufrgs.br, tony.larsson@hh.se, cpereira@ece.ufrgs.br, armando@ime.eb.br

Abstract. This paper presents an effort to support emerging Wireless Sensor Networks applications composed by different types of sensor nodes. The work is composed by two parts, in which the first is dedicated to provide cooperation abilities to sensor nodes, while the second is a customizable hardware platform intended to provide different types of sensor nodes, from those more resource constrained up to the resource-rich ones. A description of a testbed demonstrator of the proposed system is provided and comparisons with previous published simulation results denote the feasibility of the proposal.

1. Introduction

A number of emerging applications are being developed having as basis Wireless Sensor Networks as driving technologies. An important feature of the new systems that implement these applications is the usage of different types of sensors working in a unique network, cooperating in order to accomplish with users’ expectations. An example of such applications is area surveillance systems, which use sensor nodes with different sensing, computing and mobility capabilities to gather data of an area of interest.

The main issues in developing heterogeneous sensor networks are: (i) support for cooperation among heterogeneous nodes; and (ii) customization of sensor nodes [Erman et al. 2008]. The former is related to concerns such as message exchange synchronization, QoS requirements management, task (re-) allocation, network adaptation,
among others. The later is related to the diversity of node platforms, which may be built upon very distinct hardware components controlled by very different pieces of software.

Considering (i), the use of a middleware services represent a suitable approach to address the mentioned concerns, since they can integrate the technologies used in different nodes by means of common communication interfaces and cooperation mechanisms. Regarding (ii), customizable architectures can be very useful to build platforms for different sensor network nodes, from the very simple to the more sophisticated ones. This kind of architecture can provide a common base capability for all nodes. However, for nodes that need more advanced capabilities, the required resources can be incorporated. Hence, even though all nodes have the same base capability, some of them could be equipped with additional resources, thus making the sensor network more powerful due to this allowed heterogeneity.

This paper presents testbed results of a flexible and adaptable platform infrastructure intended to support heterogeneous sensor network applications composed by static sensor nodes on the ground and mobile sensors carried by Unmanned Aerial Vehicles (UAVs). It is based on the proposal of (i) a flexible coordination mechanism [Freitas et al. 2009], and (ii) on a customizable hardware architecture aimed for sensor nodes, called FemtoNode [Allgayer et al. 2009]. The key idea is to use this customizable platform to deploy different kinds of sensor nodes, from very tiny and resource constrained up to more sophisticated ones. Both types of nodes run a common coordination software, which provides the desired interoperability that will allow the cooperative work among different sensor nodes. In the demonstrator presented, nodes may be built upon the FemtoNode architecture and alternatively upon nodes with another hardware platform, namely SunSPOT [Microsystems 2010].

The remaining of this paper is organized as follows: In Section 2 the application scenario is highlighted, characterizing the network heterogeneity. Section 3 presents a pheromone-based coordination mechanism used to promote the collaborative work among static and mobile sensors. In Section 4, the FemtoNode customizable hardware architecture is described. Section 5 presents a description of a case study and highlights previous obtained simulation results, while Section 6 presents demonstrator and the practical results achieved with it, as well as a comparison with the previous simulations. Section 7 discusses related work in the area. Finally, Section 8 draws concluding remarks and gives directions for future work.

2. Motivation: Application Scenario and Network Heterogeneity

In the following, heterogeneity means that nodes in the network may have different sensing capabilities, computation power, and communication abilities. Additionally, it means that they may run on different hardware and operating systems. Therefore, such sensor networks are made up of low- and high-end nodes. Moreover, sensor nodes may have fixed positions or be able to move, being carried by UAV platforms, which can also vary from very small, as in [Walter et al. 2005], up to huge aircraft platforms, like GlobalHawk [Leonard and Dreznner 2002].

Low-end sensor nodes are those with constrained capabilities, for instance piezoelectric resistive tilt sensors, with limited processing support and communication resource capabilities. High-end sensor nodes include powerful devices like radar, high definition
visible light cameras, or infrared sensors, which are supported by moderate to rich computing and communication resources.

Mobility, as mentioned, is another important characteristic related to the heterogeneity addressed in this work and requires special attention. Sensor nodes can be statically placed on the ground or can move on the ground or fly at some altitude over the target area in which the observed phenomenon is occurring. Figure 1 graphically represents the idea of the three heterogeneity dimensions considered in this work, in which each axis represents one of the considered characteristics.

![Figure 1. Heterogeneity Dimensions.](image)

The reason for heterogeneity in the sensor nodes is to support a large range of applications that deal with very dynamic and challenging scenarios, which require different types of sensor capabilities in order to gather a wide diversity of data. Moreover, these different scenarios may require adaptations in the network, in terms of choosing suitable sensors for the tasks at hand as well as feasible QoS parameters, among others. The decisions related to these issues need additional data to be supported [Erman et al. 2008].

In order to illustrate the above idea, suppose that a network has the mission of providing a certain kind of information during a given period of time over an area of interest. The network must be able to choose a better alternative, among the set of all available options, in order to accomplish the mission. For example, an area surveillance system may receive the mission to observe if certain types of vehicles that are not allowed to pass through the surveyed area make any such violation and report if that is the case. To perform this in an efficient way ground sensors are set to alarm in the presence of unauthorized vehicles. If these sensors are not capable to confirm the violation, the alarms have to be delivered to more sophisticated sensors, carried by UAVs, in order to these last ones check and confirm the possible threat. Moreover, as the UAVs may be equipped with different types of sensor devices, they also should be able to decide which among them is the most suitable to respond to a given alarm. For example, UAVs equipped with visible-light cameras may provide poor results if employed in areas where the weather conditions are bad, e.g. in foggy or cloudy areas.

### 3. Pheromone-based Coordination Approach

The coordination strategy used in this work to make mobile sensor nodes cooperate with static sensor nodes is based on pheromone traces handed over by the mobile sensors to the static ones. Artificial pheromones are usually applied to distributed coor-
ordination by means of stigmergy, the indirect communication using environment cues [Bonabeau et al. 1999]. A pheromone trail is deposited in the environment when the entities are moving.

The pheromone provides information to other entities when they pass over it. Artificial pheromone also looses its strength along the time, modeling the evaporation of the real pheromones. In the UAV research field, pheromones are used to guide the movement of UAV swarms, for instance in surveillance and patrolling applications [P. Gaudino et al. 2003, Sauter et al. 2005].

Differently from other existing approaches, in the present work pheromones are used to guide the selection and assignment of a suitable UAV to handle an alarm issued by a ground sensor node. When an alarm is issued by the detection of a target, the network is responsible for selecting an appropriate UAV to respond to the alarm. This is performed by routing a given alarm to the UAV that has the strongest pheromone trace over the area. Having this information, the UAVs will base their movement decisions in a way to respond to the received alarms. This strategy is called here heuristic-P. The main difference from the mentioned approaches is that they rely on global UAV-network connectivity to spread information about the pheromone map, while the present approach explores a local connectivity among UAVs and ground sensor nodes.

Following the above outlined principles, the UAVs that are not engaged in the handling of any target will leave pheromone traces over the area which they cross. This pheromone trace is represented by a piece of information that is taken by the ground sensor nodes that are deployed in the area through which the UAVs have passed. When a target is detected by a ground sensor node, an alarm is issued. The decision about which UAV that will handle the potential target indicated by the issued alarm will be taken by the ground sensor nodes, by routing the alarm in the direction that points to the UAV which has the strongest pheromone trace over that area of the network. This process just considers the pheromone trace handed over by the UAVs to ground sensor nodes. This means that the only parameter taken into account is the time interval since a UAV passed by that specific location. Heuristic-P is inspired in [Heimfarth and Janacik 2008], which

Figure 2. (a) Illustrative scenario for the pheromone strategy (b) Choice of a UAV based on the pheromone strategy.
presents a pheromone-based strategy to migrate services in a sensor network, in which the pheromone concentration determines the places where the services are required.

In heuristic-P, instead of services, alarms are moved through the network following the pheromone concentration. Figure 2(a) presents a scenario that illustrates the strategy. A ground sensor node in the left border of the area detects a target. Then it issues an alarm, which is received by its neighbors. However, only those which have pheromone information about a UAV stronger than that of the alarm issuer will forward the alarm. This way, the alarm will follow a path to the closest UAV, which is represented in the figure by the shaded sensors, until the alarm delivery.

Figure 2(b) illustrates the choice of the strongest pheromone trace to be followed by an issued alarm. It is possible to observe that the alarm follows the strongest trace, which corresponds to UAV-A, until its delivery to this UAV. The arrows illustrated besides each sensor node represent how strong the pheromone of each UAV is. As it is possible to see, the pheromone level of UAV-A is increasing to the left, while the pheromone level of UAV-B is increasing to the right.

When an alarm reaches the UAV indicated by the strongest pheromone trace, if this UAV is not engaged in the handling of another alarm it sends a confirmation message to the node that had delivered the alarm. If the suggested UAV is already engaged in another alarm, the current alarm follows the second strongest pheromone trace to find another UAV to engage. When an idle UAV detects a new target, it takes the responsibility for handling it. In case that the UAV is already busy with another alarm response mission, it relays the incoming alarm that will be routed to another UAV, according to the pheromone-based heuristic-P strategy explained above.

In order to increase the robustness of the proposal, in case an alarm is issued by a node that has no pheromone trace, a direction is randomly chosen and the alarm is sent in that direction until it finds a pheromone trace. When the trace is found, it follows the trace as explained above. This situation is more likely to occur in the initialization of the system, especially in cases in which the number of UAVs deployed in the system is very low with regard to the area under surveillance.

Figure 3 presents an example of how an alarm issued by a sensor node (Figure 3-A) is routed through the network, following the pheromone traces (Figure 3 from A to D), until it is delivered to a UAV (Figure 3-E). The pheromone traces in the nodes are represented by the numbers in the center of the circles representing the ground sensor nodes in the figure. The smaller the number is, the stronger the pheromone. This translates the idea of the time past since a ground sensor node received the last pheromone beacon from a UAV. When a ground sensor node receives this pheromone beacon, it sends this information to its neighbors with a pheromone one point weaker (a number one unit greater than the one representing the node’s pheromone information). This is an indirect beacon that helps the other nodes find the traces to route the alarms. The nodes that receive the indirect beacons do not forward it. The symbol \( \infty \) means that the node has no pheromone trace, i.e. the last beacon (directly from a UAV or indirectly from another ground node) was received a long time before, above a tunable threshold. The number representing the pheromone is periodically incremented, indicating that the pheromone trace becomes weaker when time elapses, until disappearing (become \( \infty \)).

The architecture of a sensor node aims at efficiently supporting specific application needs. It requires a dedicated processing module, including a wireless communication interface, which meets both energy and performance requirements, as well as respects footprint constraints. The fact that application requirements as well as environment and other operational conditions may change during system run time imposes a major challenge [Hinkelmann et al. 2007]. In this context, the use of reconfigurable hardware [Garcia et al. 2006] appears as an interesting alternative.

Therefore, a customizable sensor node called FemtoNode is proposed. It contains a customizable ASIC and a wireless communication interface, which are configured according to application requirements.

The nodes use the RT-FemtoJava processor [Ito et al. 2001], a stack-based microcontroller that natively executes Java byte-codes. It implements an execution engine for Java in hardware, through a stack machine that is compatible with the specification of Java Virtual Machine. The customized application code is generated by the Sashimi design environment [UFRGS 2006]. The code also includes a VHDL description of the processor core and ROM (programs) and RAM (variables) memories. The Sashimi environment has been extended to incorporate an API that supports concurrent tasks, implementing the RTSJ standard [Wehrmeister et al. 2006].

As RT-FemtoJava is customizable, its code can be optimized according to the application requirements, reducing the occupied hardware area and also the energy consumption. The customizable hardware architecture of the FemtoNode allows the use of the sensor node as either a low- or high-end node. If the application requires higher performance resources to handle more complex data, such as image processing, additional resources can be included in the FemtoNode implementation. However, if the application is aimed at processing simple data, such as those from presence sensors, a reduced set of resources is used in the processor. This feature is important for the sensor node, because energy consumption is a great concern in wireless sensor networks, due to the nodes’ limited energy resource. Besides, reducing the unused resources during its synthesis the sensor node architecture allows its implementation in reconfigurable circuits with fewer available logical units, which is a feature that provides a larger application portability between different reconfigurable architectures with fewer available resources.
In the current implementation, the FemtoNode includes a wireless transceiver of Texas Instruments CC2420, which utilizes the IEEE802.15.4 standard communication protocol targeted to wireless sensor network applications with a low data rate. A module adapter described in VHDL implements the interface with the wireless transceiver. The module uses data and address buses to communicate with the processor, performing the exchange of data and allowing the transceiver parameters configuration.

As the data transfer rate from the wireless transceiver is low, compared to the processor frequency, the wireless communication module implements a buffer to store data, preventing delays while providing the necessary data to the processor. The module uses an interrupt system to inform the processor when a reception occurred.

To facilitate the use of the wireless communication module by the application developers, a communication API has been developed. The Wireless-API abstracts details of the communication media between the sensor nodes, offering a simplified form for the configuration of the data transfer module.

5. Case Study Description and Simulation Results

In order to illustrate the use of the proposed platform infrastructure, including the customizable FemtoNode and the cooperation mechanism, an area surveillance application is studied. In this application, low-end sensors nodes are scattered on the ground along a borderline. In case an unauthorized vehicle crosses the borderline limit, the sensors on the ground issue an alarm which will trigger the use of UAVs, which are equipped with more sophisticated sensors, such as radars or visible light cameras, in order to perform the recognition of the vehicle, and confirm a possible threat. Figure 4 presents this scenario.

![Figure 4. Area Surveillance Application Scenario.](image)

The setup of this system is done in such way that a predefined threshold triggers the activation of the mechanism that issues the alarms. This is done when a static sensor is not able to assess the detected vehicle is or is not a possible threat. So, the usage of a more sophisticated sensor is required and so, the cooperation mechanism has to be employed in order to drive one of the UAV-carried sensor to the area where the alarm was issued.

According to the coordination strategy based on pheromones presented in Section 3, the sensor nodes on the ground route the alarms according to the pheromone trace left by the UAVs, choosing the strongest trace to follow. When the alarm achieves a node close to the UAV, the alarm is delivered. This mechanism addresses several problems related to the communication and interoperation between nodes, such as message routing, controlled delay, delivery assurance, and appropriate node selection to cooperate with.
For the demonstrator implementation presented in the following section, sensor nodes with two different architectures compose the described surveillance system, one based on the FemtoNode and another one on the SunSpot. Each of them includes all necessary resources to meet the requirements of their utilization. Thus, based on the application specifications, a customization of the FemtoNode architecture was implemented. The UAV’s architecture is a FemtoNode with a large set of resources, capable of processing a large amount of data. Further details will be presented in Section 6.

5.1. Simulation Results for the Presented Scenario

As already mentioned, the presented scenario was already target of simulation experiments, which provided the results that will be presented in the following. These results will then be compared with the ones achieved with the demonstrator and discussed in Section 6.

Experiments using ShoX simulator [Lessmann et al. 2008] were performed and reported in [Freitas et al. 2009]. The accessed metrics presented in [Freitas et al. 2009] were: 1) the mean response time to the alarms generated in the system; 2) the number of alarms lost, due to communication failures; and 3) the utility in employing a given UAV to handle a given alarm.

For the purposes of this paper, the metrics 1 and 2 are taken into account, as the issue related to the evaluation of the utility in employing a UAV to handle a given target was not implemented so far in the demonstrator.

The simulation setup was the following: The surveillance area has dimensions 10 Km x 10 Km, in which 20,000 ground sensor nodes are randomly deployed with independent uniform probability, 500 meters communication range. Six UAVs of three different types, equally distributed, patrol the area, having a communication range of 1.5 Km and flying at the altitude of 250 meters and with speeds from 100 Km/h up to 120 Km/h. Three different runs were simulated, with one, three, and five targets respectively. The targets can further be of five different types, randomly chosen, with speeds from 50 Km/h up to 80 Km/h.

Figure 5 presents the simulation results in terms of the mean time required to respond to the alarms. Both raw data from each run (total of 20 runs for each number of targets) and the average value (lines with squared dots) are plotted in the figure. It is possible to observe that, in the worst case, the mean time to find a UAV that is idle to engage in the handling of an alarm is around 4 seconds, in the scenario with the maximum number of targets. On the other hand, in the best case, when there is just one target, the time needed to find a UAV is in average less than 1 second. An explanation for this behavior is that it is more probable to find an idle UAV when the number of targets is smaller. This may happen because, when there are more targets, an alarm message may follow a pheromone trace of a UAV that has just engaged in handling a target announced by another alarm, so the alarm must be retransmitted to the network and follow another trace. However, the solution does scale, as the increase in the mean time to find an idle UAV is linear with the increase in the number of targets, as can be concluded by taking the average values for all runs for each number of targets.

The second considered metric evaluates the system efficiency in terms of detecting a target and correctly routing the alarm message to an idle UAV. For all simulation runs,
no alarm was lost, which means that the system had 100% efficiency for the simulated scenario and correctly found an idle UAV at all occasions when an alarm was issued.

6. Demonstrator Presentation and Results

The simulations reported in [Freitas et al. 2009] showed that the proposed approach works well in the described scenario. However, wireless communications are very sensible to interferences and unpredictable variations. This means that simulation data, such as communication reachability and delays, are not always confirmed in real deployments. This fact motivated the deployment of a demonstrator to assess if the properties of the proposed approach are also observable in a physical implementation.

The deployed demonstrator is composed as a network consisting of sixteen static ground sensor nodes (nine SunSpots and the others FemtoNodes) and one mobile node (FemtoNode). The ground sensor nodes are equally distributed in a grid in an area of 225 square meters. The mobile FemtoNode, moved manually, represents a UAV that “flies” over this area leaving pheromones over the ground sensor nodes via a periodic beacon message sent to the network. Upon the occurrence of an alarm, the nodes route it in the direction of the nodes with stronger pheromone traces, until it arrives at a node which has communication with the UAV, as explained in Section 3. Figure 6 presents the demonstrator setup. The radio in the nodes was adjusted to provide a communication range of 5 meters, such that the nodes are capable of communicating only with their immediate vertical and horizontal neighbors, which are 5 meters apart, but not with their
diagonal neighbors or any other node in the grid. The mobile node, representing the UAV, has the same communication range configuration as the static nodes.

Twenty runs were performed. In each of them, an alarm was generated by one of the static nodes, randomly chosen, which had to be routed to the UAV according to the pheromone mechanism described in Section 3 and implemented as described above.

In order to stress the network and test these mechanisms, random messages were generated by the static nodes, which competed with the beacon and alarm messages for the utilization of the communication resources.

The evaluated metric with the described testbed was the time to respond to the alarms generated in the system. By obtaining this metric, the delay of one hop communication was calculated and compared with the one achieved in the simulation results described before. Figure 7 presents the time taken by the system to deliver the alarm to the UAV.

![Figure 7. Alarm Response Time Achieved by the Demonstrator.](image)

The average number of hops to deliver the alarm was 5 hops for the 20 runs of the testbed. Taking the average of the time to deliver an alarm, 538.85 ms, and the average number of hops, the average delay calculated is of 107.77 ms in each hop.

Considering the simulation results, taking the worst case scenario, the one with 5 targets, in average the number of hops for an alarm to be delivered was 13.78. Taking the average of worst case scenario, 1,821.65 ms to deliver an alarm, 132.14 ms is the delay for an alarm to be forwarded among the static nodes in each hop.

Comparing the delays obtained from the simulation runs and from the demonstrator, it is possible to observe that they are very close to each other. The delays obtained with the demonstrator are even better than the ones achieved by simulation, which is an evidence of the applicability of the proposed approach.

### 7. Related Work

AWARE [Erman et al. 2008] is a middleware whose goal is to provide integration of the information gathered by different types of sensors, including low-end sensor nodes in a wireless sensor network and mobile robots equipped with more sophisticated sensors. Our proposal not only addresses heterogeneous sensors and their coordination, but also concerns like QoS, e.g. message delay, which is missing in [Erman et al. 2008].
In [Walter et al. 2005], an approach using digital pheromones to control a swarm of UAVs is presented. The method proposed by the authors uses digital pheromones to bias the movements of individual units within a swarm toward particular areas of interest that are attractive, from the point of view of the mission that the swarm is performing, and away from areas that are dangerous or just unattractive. In the large sense, the pheromone-based strategy used in our work has a similar goal, driving the UAVs to areas of interest. However, differently from their approach, we use the pheromone traces to localize the UAVs when an alarm is issued by a ground sensor node informing an event of interest and then drive the UAVs to the location where the event happened.

In [Caldas et al. 2005] a sensor node was presented incorporating re-configurable hardware resources to improve and to expand the set of features executed by conventional sensor nodes. These features allow the processing of complex events that requires high computing efficiency and accuracy. Our proposal enhances type of flexibility by optimizing the microcontroller architecture by synthesize only the resources that the applications need.

8. Conclusions and Future Work

This paper presented a system solution to provide interoperability and coordination support for heterogeneous sensor networks composed by ground static sensor nodes and mobile sensors carried by UAVs, and a customizable hardware to implement the different types of sensors needed in such networks. The first part of the work is represented by a bio-inspired pheromone-based approach, while the second part is represented by the FemtoNode platform, which provides a support to the development of different types of nodes, customized according to specific requirements.

The assessment of the viability in using of the proposed approach in real networks was done by means of a comparison between results obtained by simulation experiments, representing a large scale scenario, with a small scale testbed demonstrator, which uses the pheromone mechanism and the FemtoNode. The evaluation provided evidences that the proposed solution indeed sounds.

As future works, a large demonstrator is being planned, in which we aim also to evaluate the selection of the utility in employing a given UAV to handle a given target, and like this, be able to compare the additional simulation results reported in [Freitas et al. 2009] with results from a demonstrator.

References


